

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Інформаційних управляючих систем
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів і моделей рекрутингу персоналу
в корпоративній інформаційній системі

(тема)

Виконав:

здобувач 2 року навчання,
групи ІУСТМ-24-1

Дмитро АФЕНЧЕНКО

(власне ім'я, прізвище)

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


Тип програми освітньо- професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Інформаційні управляючі системи та технології
(повна назва освітньої програми)

Керівник: доц. Тетяна БОРИСЕНКО
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри ІУС


(підпис)

Костянтин ПЕТРОВ
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Інформаційних управляючих систем _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)Освітня програма _____ Інформаційні управляючі системи та технології _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____ 
(підпис)

“ 24 ” листопада 20 25 р.

ЗАВДАННЯ**НА КВАЛІФІКАЦІЙНУ РОБОТУ**здобувачеві _____ Афенченку Дмитру Романовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів і моделей рекрутингу персоналу в корпоративній інформаційній системі _____

затверджена наказом по університету від “ 24 ” листопада 2025 р. № 1055Ст _____

2. Термін подання здобувачем роботи до екзаменаційної комісії “ 18 ” грудня 2025 р. _____

3. Вихідні дані до роботи _____ матеріали передатестаційної практики, науково-технічні публікації та інтернет-джерела з тематики кваліфікаційної роботи _____

4. Перелік питань, що потрібно опрацювати у роботі 1) аналіз предметної галузі та постановка задачі; 2) дослідження методів інтелектуального аналізу та розробка моделей підтримки прийняття рішень в рекрутингу; 3) розробка інформаційної технології рекрутингу персоналу та її програмна реалізація; 4) експериментальна перевірка гібридного підходу до інтелектуального ранжування кандидатів на основі LLM та нечіткої логіки. _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз процесів рекрутингу та змістовна постановка задачі дослідження	24.11.2025 - 25.11.2025	Виконано
2	Огляд існуючих ATS-систем та методів автоматизованого оцінювання кандидатів	26.11.2025 - 27.11.2025	Виконано
3	Формулювання вимог до підсистеми інтелектуального відбору персоналу	28.11.2025 - 29.11.2025	Виконано
4	Дослідження теоретичних основ Large Language Model та апарату нечіткої логіки	30.11.2025 - 01.12.2025	Виконано
5	Розробка гібридного методу оцінювання та формування бази нечітких правил	02.12.2025 - 03.12.2025	Виконано
6	Розробка інформаційної технології рекрутингу та архітектури підсистеми рекрутингу	04.12.2025 - 06.12.2025	Виконано
7	Розробка промпт-інжинірингу для Large Language Model та алгоритмів парсингу резюме	07.12.2025 - 08.11.2025	Виконано
8	Програмна реалізація підсистеми оцінювання та ранжування кандидатів	09.12.2025 - 11.12.2025	Виконано
9	Експериментальна перевірка ІТ на наборі тестових вакансій та резюме	12.12.2025 - 13.12.2025	Виконано
10	Порівняльний аналіз методів, оцінка ефективності та оформлення результатів	14.12.2025 - 15.12.2025	Виконано
11	Оформлення пояснювальної записки до кваліфікаційної роботи	16.12.2025	Виконано
12	Перевірка кваліфікаційної роботи на плагіат	17.12.2025	Виконано
13	Попередній захист кваліфікаційної роботи	18.12.2025	Виконано


Дата видачі завдання 24 листопада 2025 р.

Здобувач



(підпис)

Керівник роботи



(підпис)

доц. Тетяна БОРИСЕНКО

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 127 с., 16 рис., 6 табл., 3 дод., 16 джерел.

ВЕЛИКА МОВНА МОДЕЛЬ, НЕЧІТКА ЛОГІКА, РЕКРУТИНГ, СЕМАНТИЧНИЙ АНАЛІЗ; ATS, FUZZY LOGIC, GPT, LLM, NLP, PYTHON.

Об'єктом дослідження є процес інтелектуального відбору та оцінювання кандидатів на вакансії в корпоративних інформаційних системах.

Метою роботи є підвищення ефективності рекрутингу за рахунок розробки інформаційної технології на основі гібридного підходу до інтелектуального ранжування кандидатів (Large Language Model + Нечітка логіка).

Під час дослідження застосовувалися такі методи: системний аналіз, методи обробки природної мови (NLP), теорія нечітких множин, порівняльний експеримент.

Наукова новизна роботи полягає у запропонованому гібридному підході до ранжування кандидатів, який поєднує семантичний аналіз (LLM) та нечітку логіку та розробленій інформаційній технології інтелектуального ранжування кандидатів в корпоративній інформаційній системі.

Отримані результати мають практичне значення для HR-підрозділів компаній та рекрутингових агентств і можуть бути використані для автоматизації первинного скринінгу резюме та зменшення суб'єктивізму при наймі.

Кваліфікаційну роботу виконано згідно методичних вказівок щодо розробки та оформлення кваліфікаційної роботи [1], ДСТУ 3008:2015 [2] та ДСТУ 8302:2015 [3].

ABSTRACT

Master's thesis: 127 pages, 16 figures, 6 tables, 3 appendices, 16 sources.

FUZZY LOGIC, LARGE LANGUAGE MODEL, RECRUITMENT, SEMANTIC ANALYSIS; ATS, FUZZY LOGIC, GPT, LLM, NLP, PYTHON.

The object of the study is the process of intelligent selection and evaluation of candidates for vacancies in corporate information systems.

The aim of the thesis is to improve recruitment efficiency by developing an information technology based on a hybrid approach to intelligent candidate ranking that combines large language models with fuzzy logic.

The research employs the following methods: system analysis, natural language processing (NLP) techniques, fuzzy set theory, and a comparative experimental study.

The scientific novelty of the work consists in the proposed hybrid candidate ranking approach that integrates semantic analysis using large language models with fuzzy logic, as well as in the development of an information technology for intelligent candidate ranking within a corporate information system.

The obtained results have practical significance for HR departments and recruitment agencies and can be applied to automate the initial resume screening process and reduce subjectivity in hiring decisions.

The qualification thesis was completed in accordance with the methodological guidelines for the preparation and formatting of qualification works [1], DSTU 3008:2015 [2], and DSTU 8302:2015 [3].

ЗМІСТ

	С.
Скорочення та умовні позначки	9
Вступ.....	10
1 Аналіз предметної галузі та постановка задачі.....	12
1.1 Еволюція підходів до управління персоналом та роль інформаційних технологій.....	12
1.2 Системний аналіз бізнес-процесу рекрутингу	14
1.3 Класифікація та аналіз існуючих засобів автоматизації рекрутингу.....	15
1.4 Огляд методів та моделей підтримки прийняття рішень при відборі персоналу.....	18
1.4.1 Метод булевого пошуку (Boolean Search).....	18
1.4.2 Векторні моделі та Vector Space Model	19
1.4.3 Метод Fuzzy Logic	20
1.4.4 Метод Machine Learning	21
1.4.5 Підходи на основі великих мовних моделей (LLM)	21
1.5 Обґрунтування вибору методу дослідження.....	22
1.6 Постановка задач дослідження	23
2 Дослідження методів інтелектуального аналізу та розробка моделей підтримки прийняття рішень під час рекрутингу	25
2.1 Обґрунтування підходу до вирішення задачі дослідження	25
2.2 Моделі представлення та екстракції даних	27
2.3 Математична модель векторизації даних	29
2.4 Методи розрахунку відповідності та ранжування кандидатів	30
2.4.1 Keyword Matching як базовий метод.....	30
2.4.2 Метод «Семантична подібність».....	31
2.4.3 Суть гібридного підходу	31

2.5	Теоретичні основи моделі оцінювання на основі нечіткої логіки	32
2.5.1	Фазифікація.....	32
2.5.2	База правил та нечіткого висновку	33
2.5.3	Етап дефазифікації.....	34
3	Інформаційна технологія рекрутингу персоналу та програмна реалізація	36
3.1	Опис інформаційної технології інтелектуального оцінювання кандидатів.....	36
3.2	Програмна архітектура та механізми реалізації	39
3.3	Інтеграція з API мовної моделі	41
4	Експериментальна перевірка гібридного підходу рекрутингу на основі LLM та нечіткої логіки	45
4.1	Опис вхідних даних та первинної обробки даних	45
4.2	Експериментальна перевірка гібридного підходу	49
4.2.1	Результати LLM-аналізу.....	49
4.2.2	Фазифікація проміжних оцінок	51
4.2.3	Активація правил нечіткої логіки та формування вихідного балу кандидата.....	52
4.2.4	Результати застосування гібридного підходу	54
4.3	Результати експерименту з базовими методами.....	56
4.3.1	Результати методу Keyword Matching	57
4.3.2	Результати методу Vector Semantic Analysis.....	58
4.4	Порівняльний аналіз трьох методів та інтерпретація результатів	60
	Висновки	62
	Перелік джерел посилання	64
	Додаток А Програмний код реалізації гібридного підходу до інтелектуального ранжування кандидатів	67
	Додаток Б Тексти файлів опису вакансії та резюме для проведення експерименту	85
Б.1	Опис вакансії Senior Python Developer	85

Б.2 Резюме першого кандидата.....	87
Б.3 Резюме другого кандидата.....	92
Б.4 Резюме третього кандидата	97
Б.5 Резюме четвертого кандидата.....	101
Б.6 Резюме п'ятого кандидата.....	106
Додаток В Графічний матеріал кваліфікаційної роботи	113

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

IT – інформаційна технологія
KIC – корпоративні інформаційні системи
API – Application Programming Interface
ATS – Applicant Tracking Systems
CV – Curriculum Vitae
ERP – Enterprise Resource Planning
GDPR – General Data Protection Regulation
HCM – Human Capital Management
HR – Human Resources
HRM – Human Resource Management
JSON – JavaScript Object Notation
LLM – Large Language Model
MCDM – Multi-Criteria Decision Making
ML – Machine Learning
NLP – Natural Language Processing

ВСТУП

Сучасний світ переживає фундаментальну трансформацію економічних моделей, де на зміну індустріальній епосі приходить економіка знань. У цій новій реальності людський капітал перестає бути просто одним із ресурсів виробництва, а стає головним стратегічним активом, що визначає ринкову вартість та конкурентоспроможність будь-якого підприємства. Здатність організації залучати талановитих фахівців, ефективно оцінювати їхній потенціал та інтегрувати в робочі процеси стає критичним фактором виживання бізнесу. Водночас, ціна помилки при наймі персоналу неухильно зростає: за даними провідних консалтингових агентств, втрати від найму невідповідного співробітника можуть сягати річного розміру його заробітної плати, враховуючи витрати на пошук, навчання та втрачену вигоду.

Ринок праці сьогодні характеризується безпрецедентною динамікою та величезними обсягами інформації. Популярні вакансії у провідних технологічних компаніях та великих корпораціях можуть отримувати сотні, а іноді й тисячі відгуків щодня. В умовах такого інформаційного перевантаження традиційні методи рекрутингу, які століттями покладалися на інтуїцію HumanResources-менеджерів (HR) та ручний перегляд паперових резюме, виявляють свою повну неспроможність. Людський фактор стає «вузьким місцем» процесу найму: рекрутери фізично не встигають якісно опрацювати потік вхідних даних, що призводить до поверхового оцінювання, суб'єктивізму та пропуску дійсно цінних кандидатів.

Ситуація ускладнюється тим, що корпоративні інформаційні системи, які використовуються на більшості підприємств, часто мають обмежений функціонал у частині інтелектуальної підтримки процесів підбору персоналу. Більшість існуючих рішень зводяться до автоматизації документообігу та зберігання анкет, тоді як алгоритми пошуку базуються на примітивному співпадінні ключових слів. Такий підхід ігнорує семантичну складність

природної мови, не враховує синонімію професійних термінів та не здатен працювати з нечіткими критеріями, такими як рівень компетентності чи особисті якості. Це створює суттєвий технологічний розрив між реальними потребами бізнесу в якісному відборі та можливостями існуючих інструментів.

Вирішення цієї проблеми лежить у площині цифрової трансформації HR-процесів та впровадження методів інтелектуального аналізу даних. Поєднання сучасних математичних моделей для обробки природної мови та алгоритмів прийняття рішень в умовах невизначеності відкриває нові перспективи для автоматизації рекрутингу. Створення систем, здатних імітувати логіку експерта-рекрутера, розуміти контекст резюме та об'єктивно ранжувати кандидатів за комплексом критеріїв, є актуальним завданням сучасної інформатики. Такі системи дозволяють не лише автоматизувати рутинні операції, але й значно підвищити якість кадрових рішень, забезпечуючи бізнесу конкурентну перевагу у боротьбі за таланти.

Саме тому дослідження методів та моделей, що дозволяють підвищити ефективність рекрутингу в корпоративних інформаційних системах, є своєчасним та необхідним кроком для розвитку сучасних технологій управління персоналом. Розробка гібридних підходів, які поєднують точність векторного аналізу з гнучкістю нечіткої логіки, дозволяє подолати обмеження традиційних систем та створити інструмент, що відповідає викликам сучасного ринку праці [4].

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Еволюція підходів до управління персоналом та роль інформаційних технологій

Сучасний етап розвитку світової економіки характеризується стрімкою інтелектуалізацією бізнес-процесів, що докорінно змінює підходи до управління організаційними структурами. В умовах глобалізації та переходу до інформаційного суспільства традиційні фактори виробництва (земля, сировина, фінансовий капітал) поступово втрачають домінуючу роль, поступаючись місцем нематеріальним активам [5]. Серед них ключове значення набуває людський капітал – сукупність знань, навичок, креативних здібностей і досвіду співробітників, які забезпечують інноваційний розвиток підприємства. У цьому контексті процес рекрутингу набуває особливого значення, оскільки він є вхідною точкою для формування кадрового потенціалу організації.

Ця трансформація зумовила еволюцію концепції управління персоналом: від Personnel Administration до Human Resource Management (HRM) і, зрештою, до Human Capital Management (HCM). У межах парадигми HCM співробітник розглядається не як витратна стаття бюджету, яку необхідно мінімізувати, а як стратегічний ресурс, від якого залежить конкурентоспроможність та ринкова стійкість компанії. Якість реалізації цього етапу має довгостроковий вплив на діяльність підприємства, оскільки помилки при наймі призводять до кумулятивного негативного ефекту.

У цьому контексті процес рекрутингу набуває особливого значення, оскільки саме він виступає «вхідною точкою» формування кадрового потенціалу організації. Рекрутинг перестає бути простою функцією заповнення вакантних місць і перетворюється на складний процес залучення талантів, що вимагає:

- аналізу ринку праці та каналів залучення кандидатів;

- стандартизації вимог до ролі та компетенцій;
- технологічної підтримки оцінювання великої кількості резюме у стислий час.

Якість реалізації цього етапу має довгостроковий вплив на діяльність підприємства, оскільки помилки при наймі формують кумулятивний негативний ефект. До основних ризиків неефективного рекрутингу доцільно віднести:

- прямі фінансові збитки: витрати на рекламу вакансії, оплату праці рекрутерів, супутні податки, витрати на адаптацію, компенсації при звільненні невідповідного працівника;
- непрямі втрати: зниження продуктивності команди, погіршення морального клімату, репутаційні втрати, ризик втрати клієнтів через неякісний сервіс;
- втрата часу: повторний запуск циклу пошуку, повторне навчання та інтеграція нового співробітника.

Корпоративні інформаційні системи (КІС) класу Enterprise Resource Planning (ERP) історично включали модулі управління персоналом. Проте тривалий час їх функціональність була зосереджена переважно на кадровому діловодстві та розрахунку заробітної плати. Натомість процеси пошуку та первинного відбору кандидатів часто реалізовувалися за допомогою розрізнених інструментів (електронна пошта, таблиці, окремі бази резюме). В умовах цифрової економіки такий підхід стає гальмом розвитку, оскільки не дозволяє ефективно працювати з великими обсягами документів і даних ринку праці.

Цифрова трансформація HR-сфери (HR Digital) диктує нові вимоги до інформаційних систем. Сучасна корпоративна інформаційна система повинна не просто фіксувати дані про співробітників, а й виступати інструментом підтримки прийняття рішень на етапі відбору [6]. Необхідність впровадження інтелектуальних технологій зумовлена такими факторами:

- експоненціальне зростання обсягів інформації: кількість резюме на

популярні вакансії може вимірюватися сотнями;

– вимога скорочення часу закриття вакансій: простий робочого місця призводить до втрати прибутку;

– підвищення вимог до об'єктивності: необхідність мінімізації людського фактору та упередженості при відборі.

1.2 Системний аналіз бізнес-процесу рекрутингу

Для побудови ефективної інформаційної технології необхідно провести декомпозицію та системний аналіз предметної області. Рекрутинг доцільно розглядати як багатоетапний бізнес-процес з високою динамікою та інтенсивним обміном інформацією.

Типовий життєвий цикл закриття вакансії можна розділити на декілька ключових етапів (рисунок 1.1).

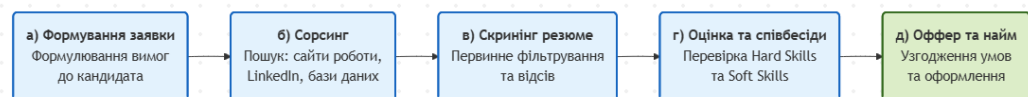


Рисунок 1.1 – Схема життєвого циклу закриття вакансії

Кожен з етапів має свої інформаційні особливості та проблеми:

а) на кроці формування заявки на підбір ініціатор пошуку формулює вимоги до майбутнього співробітника. Основна проблема тут полягає у формалізації знань: замовники часто оперують нечіткими поняттями, наприклад, «потрібен ініціативний працівник» або «гарне знання технологій», які важко інтерпретувати однозначно. Відсутність чіткого профілю посади на вході створює «шум» у каналі комунікації;

б) етап сорсингу – це процес активного пошуку та залучення кандидатів.

Інформація надходить з безлічі гетерогенних джерел;

- 1) сайтів пошуку роботи;
- 2) професійних соціальних мереж, наприклад LinkedIn;
- 3) корпоративних порталів та баз даних кадрових агентств. Це

створює проблему агрегації даних, оскільки вхідні документи мають різну структуру та формати файлів, що вимагає уніфікації.

в) скринінг резюме – це етап первинного фільтрування та відсіву нерелевантних кандидатів, який є найбільш критичним для автоматизації. В умовах високої конкуренції на одну вакансію може надходити сотні відгуків. Фізична обробка такого масиву інформації є надзвичайно трудомісткою. Дослідження показують, що рекрутер витрачає в середньому 6-10 секунд на перегляд одного резюме, що призводить до;

- 1) когнітивних викривлень та втоми;
- 2) помилок першого роду (відхилення гідного кандидата);
- 3) помилок другого роду (пропуск невідповідного кандидата);

г) на етапі «Оцінка та співбесіди» перевіряються професійні навички (Hard Skills) та особистісні якості (Soft Skills). Головною проблемою тут залишається суб'єктивізм оцінювання, коли різні інтерв'юери по-різному трактують відповіді кандидата;

д) етап «Оффер та найм» – це фінальний етап, що включає узгодження умов, оформлення документів та інтеграцію нового співробітника в облікові системи компанії.

1.3 Класифікація та аналіз існуючих засобів автоматизації рекрутингу

Ринок програмного забезпечення для HCM пропонує широкий спектр рішень для автоматизації процесів найму. Основним класом систем, що забезпечують цей функціонал, є Applicant Tracking Systems (ATS) – системи

управління кандидатами. Аналіз сучасної IT-інфраструктури дозволяє класифікувати ці системи за декількома критеріями: еволюційним розвитком, архітектурою розгортання та ступенем інтелектуальності.

З точки зору еволюційного розвитку, можна виділити три покоління систем:

– Applicant Tracking System (ATS) 1.0 - бази даних: системи, орієнтовані на просте зберігання та оцифрування паперових резюме. Їх основна функція - створення електронного архіву;

– ATS 2.0 - Workflow-системи: системи, що автоматизують бізнес-процес рекрутингу. Вони дозволяють відстежувати статус кандидата, планувати співбесіди та інтегруватися з job-сайтами. Це найбільш поширений тип систем сьогодні;

– ATS 3.0 - Інтелектуальні системи: системи нового покоління, що використовують алгоритми машинного навчання для предиктивної аналітики та рекомендацій, проте вони займають незначну частку ринку через високу вартість [7].

За архітектурним підходом доцільно виділити наступні категорії:

а) інтегровані модулі рекрутингу у складі ERP/HCM-систем, такі як SAP SuccessFactors, Oracle Taleo, Workday:

1) концепція: «All-in-One». Рекрутинг є лише частиною глобальної системи управління підприємством;

2) переваги: повна наскрізна інтеграція даних. Кандидат, який прийняв оффер, автоматично стає співробітником у системі, йому створюється обліковий запис, нараховується зарплата тощо. Високий рівень безпеки даних;

3) недоліки: громіздкість, складність інтерфейсу, висока вартість ліцензій та впровадження. Часто функціонал рекрутингу реалізований за залишковим принципом і поступається спеціалізованим рішенням у гнучкості;

б) спеціалізовані хмарні ATS, такі як Greenhouse, Lever, Breezy HR,

SmartRecruiters, які мають такі характеристики;

1) концепція: вузька спеціалізація на процесах Talent Acquisition, цей підхід відрізняється від традиційного найму тим, що розглядає рекрутинг не як реактивну процедуру закриття вакансій, а як довгострокову стратегію формування бренду роботодавця.;

2) переваги: сучасний інтерфейс, швидкість впровадження, потужні інструменти Customer Relationship Management (CRM) для кандидатів, аналітика ефективності каналів пошуку;

3) недоліки: проблеми інтеграції з основною ERP-системою підприємства, зберігання персональних даних на серверах провайдера, але є потреба відповідності General Data Protection Regulation (GDPR).

в) кастомні розробки, які можна охарактеризувати таким чином:

1) концепція: створення унікального програмного забезпечення (ПЗ) силами внутрішнього IT-департаменту;

2) переваги: повна адаптація під унікальні бізнес-процеси, можливість впровадження специфічних, наукоємних алгоритмів оцінки, які не доступні у «коробкових» версіях;

3) недоліки: високі витрати на розробку та підтримку.

Аналізуючи функціональні обмеження, можна сказати, що, незважаючи на розвиненість інтерфейсів, більшість комерційних ATS використовують застарілі методи пошуку. Ключовою проблемою є семантичний розрив: система шукає точне входження слів, вказаних рекрутером, і ігнорує контекст. Наприклад, запит «Java Developer» може відсіяти резюме з заголовком «Software Engineer (Backend)», навіть якщо навички ідентичні. Це підтверджує необхідність розробки більш досконалих методів інтелектуального аналізу для інтеграції їх у корпоративні системи.

1.4 Огляд методів та моделей підтримки прийняття рішень при відборі персоналу

Наукова проблематика автоматизованого відбору персоналу лежить на перетині інформаційного пошуку, Natural Language Processing (NLP) та теорії прийняття рішень. Розглянемо основні методологічні підходи, їх концептуальні основи та застосовність.

1.4.1 Метод булевого пошуку (Boolean Search)

Boolean Search – це детермінований підхід, що базується на класичній теорії множин та булевій алгебрі. Він має такі особливості:

а) концептуальна сутність: пошуковий запит формується як чіткий набір умов, з'єднаних логічними операторами «І», «АБО», «НІ». Резюме розглядається як множина слів. Якщо ця множина задовольняє заданій логічній формулі, резюме вважається релевантним. В іншому випадку – воно повністю відкидається;

б) переваги підходу;

1) прозорість: користувач повністю контролює результати пошуку і розуміє, чому те чи інше резюме потрапило у вибірку;

2) швидкість: метод не вимагає складних обчислень і працює миттєво навіть на великих базах даних;

в) недоліки підходу;

1) проблема синонімії: жорсткі умови відсіюють релевантних кандидатів, які використовують інші назви для тих самих навичок, наприклад, «керівник» замість «начальник»;

2) проблема полісемії: багатозначність слів призводить до появи

«інформаційного шуму»;

3) відсутність ранжування: метод бінарний – він ділить кандидатів лише на «підходить» та «не підходить». Всі знайдені кандидати вважаються рівнозначними, що ускладнює вибір найкращого.

1.4.2 Векторні моделі та Vector Space Model

Цей підхід представляє собою більш гнучку альтернативу булевому пошуку, базуючись на статистичному аналізі тексту. Він має такі особливості:

а) концептуальна сутність: у рамках цього методу будь-який текстовий документ (резюме або опис вакансії) представляється як набір числових характеристик (вектор) у багатовимірному просторі. Кожне слово тексту стає окремим виміром цього простору. Для визначення важливості слів використовується статистичний підхід;

1) враховується, наскільки часто слово зустрічається в конкретному документі, що частіше – то важливіше;

2) враховується, наскільки рідкісним є це слово у всій базі резюме, що рідкісніше – то більшу вагу воно має, оскільки дозволяє краще вирізнити унікальні навички. Ступінь відповідності визначається шляхом геометричного вимірювання «кута» між вектором резюме та вектором вакансії. Чим менший кут – тим більше схожі тексти за змістом;

в) переваги підходу;

1) автоматичне ранжування: система самостійно сортує кандидатів від найбільш до найменш відповідних;

2) врахування контексту: метод дозволяє знаходити документи, де ключові слова зустрічаються частіше, що може свідчити про більший досвід;

г) обмеження: метод «мішка слів» ігнорує порядок слів у реченні та синтаксичні зв'язки, що може призводити до помилкового тлумачення фраз із запереченнями;

1.4.3 Метод Fuzzy Logic

Оскільки людські характеристики, такі як компетенції, досвід, комунікабельність часто мають якісний характер і не мають чітких меж, класична «жорстка» логіка є недостатньою. Теорія нечітких множин дозволяє моделювати невизначеність, притаманну людському мисленню.

Концептуальна сутність теорії нечіткої множин: замість того, щоб оперувати точними числами, система використовує лінгвістичні змінні – слова природної мови, наприклад, «досвід роботи» може набувати значень «Малий», «Середній», «Значний». Ключовим елементом є поняття «функції належності», яка визначає ступінь впевненості у тому, що кандидат відповідає певному критерію. Наприклад, 5 років досвіду можуть на 100% належати до поняття «Досвідчений», а 3 роки – лише на 60%. Це дозволяє уникнути ситуації, коли кандидат відхиляється через нестачу одного місяця стажу. Прийняття рішень базується на базі правил типу «ЯКЩО... ТО... », які імітують міркування експерта-рекрутера.

Переваги теорії нечітких множин:

- гнучкість: дозволяє працювати з якісними критеріями та суб'єктивними оцінками;
- інтерпретованість: логіку прийняття рішення можна легко пояснити замовнику, показавши правила, за якими діє система;
- «м'який» відсів: кандидати, які трохи не дотягують до ідеалу, не відкидаються, а отримують трохи нижчий рейтинг.

1.4.4 Метод Machine Learning

Machine Learning (ML) – це найбільш сучасний підхід, що передбачає «навчання» системи на історичних даних компанії.

Концептуальна сутність ML: алгоритми, такі як класифікатори, нейронні мережі аналізують великі масиви даних про попередні найми: які резюме були відібрані, кого запросили на співбесіду, хто успішно пройшов випробувальний термін. На основі цього аналізу система будує власну модель ідеального кандидата і намагається передбачити успішність нових претендентів;

Проблеми застосування ML:

- проблема «холодного старту»: для роботи методу потрібна величезна накопичена база історичних даних, якої часто немає у нових компаній;

- упередженість: якщо в історії компанії були приховані дискримінаційні тенденції (наприклад, несвідома перевага чоловікам), алгоритм виявить цю закономірність і буде автоматично відтворювати її [8], [9];

- ефект «чорної скриньки»: складні нейронні мережі часто не можуть пояснити причину свого рішення, що є критичним недоліком для HR-сфери, де важлива прозорість.

1.4.5 Підходи на основі великих мовних моделей (LLM)

Окремо доцільно виділити сучасний клас підходів, що ґрунтуються на великих мовних моделях, здатних аналізувати текст у контексті та формувати структуровані висновки.

Загальна характеристика великих мовних моделей:

- сутність: LLM виконує семантичний аналіз резюме та може

формувати оцінки за різними критеріями (наприклад, відповідність обов'язковим вимогам, бонусний потенціал, особисті якості), а також надавати пояснення;

– переваги LLM;

- 1) урахування контексту та складних формулювань;
- 2) можливість структурування неформалізованих даних (різні формати резюме);
- 3) підтримка багатокритеріальної оцінки;
- 4) необхідність забезпечення прозорості фінального рішення на рівні правил/логіки;
- 5) доцільність використання як компонента у складі більш інтерпретованої системи прийняття рішень.

Саме ці міркування підводять до гібридного поєднання LLM-аналізу та нечіткої логіки.

1.5 Обґрунтування вибору методу дослідження

Проведений аналіз показав, що ізолюване застосування окремих підходів має суттєві обмеження. Метод ключових слів, хоч і залишається поширеним через простоту, є морально застарілим у контексті роботи з природною мовою, оскільки не розв'язує проблему семантичного розриву. Векторні моделі дають семантичну близькість, однак не завжди адекватно відображають саме професійну відповідність, а також є складними для пояснення рекрутеру. Методи машинного навчання потребують розмічених історичних даних і мають ризики упередженості, а також часто залишаються «чорною скринькою».

Враховуючи специфіку задачі – необхідність аналізу неструктурованого тексту резюме та одночасного прийняття рішення на основі

багатокритеріальних, часто нечітких вимог бізнесу – у даній дипломній роботі пропонується гібридний підхід до інтелектуального рекрутингу, який поєднує сильні сторони двох комплементарних підходів:

- LLM-аналіз як механізм глибокого розуміння тексту та формування проміжних оцінок кандидата за декількома критеріями;

- нечітку логіку як інтерпретований механізм узгодження критеріїв та формування фінального рейтингу через експертні правила.

Обґрунтування цієї комбінації базується на таких перевагах:

- комплементарність: LLM забезпечує контекстний аналіз вакансії та резюме, здатний виділяти як явні відповідності, так і додаткові компетенції й особисті якості, а нечітка логіка перетворює ці оцінки у кероване рішення, відтворюючи експертні міркування;

- інтерпретованість і керованість: на відміну від «чорної скриньки» ML-моделей, нечітка система оперує правилами, які можна пояснити і за потреби адаптувати під конкретну вакансію (через зміну лінгвістичних категорій та правил) [10];

- відповідність практичним вимогам рекрутингу: рішення формуються з урахуванням кількох компонентів, а не лише «схожості текстів», що було експериментально підтверджено у розділі 4 шляхом порівняння з baseline-методами.

1.6 Постановка задач дослідження

Актуальність теми дослідження зумовлена наявним протиріччям між постійно зростаючими обсягами інформаційних потоків на ринку праці, які вимагають оперативної обробки, та обмеженими можливостями традиційних інструментів рекрутингу.

Метою роботи є підвищення ефективності процесу рекрутингу

персоналу в корпоративній інформаційній системі за рахунок розробки та реалізації методу інтелектуального аналізу даних про кандидатів та їх ранжування [11]. Досягнення цієї мети дозволить забезпечити:

- мінімізацію помилок першого та другого роду: зниження ймовірності відхилення релевантних кандидатів та пропуску невідповідних;
- скорочення часових витрат: автоматизація первинного скринінгу дозволить рекрутерам фокусуватися на співбесідах, а не на читанні анкет;
- підвищення об'єктивності оцінювання: усунення когнітивних викривлень та суб'єктивізму, притаманних ручній обробці даних;
- зниження трудомісткості: звільнення HR-персоналу від рутинних операцій з аналізу великих масивів текстової інформації.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести системний аналіз процесу рекрутингу як об'єкта автоматизації, визначити структуру вхідних та вихідних інформаційних потоків;
- розробити уніфіковану формальну модель представлення вимог вакансії та компетентнісного профілю кандидата для забезпечення їх співставності;
- розробити підхід до багатокритеріального оцінювання та ранжування кандидатів з використанням апарату нечіткої логіки, яка враховуватиме як кількісні, так і якісні показники [12];
- розробити інформаційну технологію (ІТ) рекрутингу персоналу в корпоративній ІС;
- виконати програмну реалізацію гібридного підходу до інтелектуального ранжування кандидатів;
- провести експериментальну перевірку отриманих наукових результатів.

2 ДОСЛІДЖЕННЯ МЕТОДІВ ІНТЕЛЛЕКТУАЛЬНОГО АНАЛІЗУ ТА РОЗРОБКА МОДЕЛЕЙ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ПІД ЧАС РЕКРУТИНГУ

У першому розділі було визначено, що існуючі системи рекрутингу страждають від «семантичного розриву» між описом вакансії та резюме, а також від нездатності коректно працювати з нечіткими критеріями (наприклад, «достатній потенціал», «високі комунікативні навички», «сильний кандидат»). У цьому розділі виконується формалізація предметної області, описуються математичні моделі представлення текстових даних та алгоритми їх обробки, які лежать в основі розробленого у розділі 4 програмного модуля підсистеми рекрутингу.

Ключовою особливістю запропонованого рішення є те, що оцінювання кандидата виконується не як «просте порівняння двох текстів», а як багатокритеріальна процедура: LLM формує структуровані проміжні оцінки за різними аспектами придатності кандидата, а нечітка логіка (алгоритм Мамдані) агрегує ці оцінки у фінальний рейтинг, моделюючи експертні правила прийняття рішення.

2.1 Обґрунтування підходу до вирішення задачі дослідження

Задача рекрутингу з математичної точки зору є задачею багатокритеріального прийняття рішень (MCDM, Multi-Criteria Decision Making) в умовах невизначеності. Складність задачі зумовлена асиметрією та неоднорідністю вхідних даних:

– вакансія (V) – це набір обмежень та бажаних характеристик. Вона часто формулюється нечітко;

– кандидат (C) – це набір фактів та компетенцій. Резюме може містити інформацію, яка не запитувалась у вакансії (надлишковість), що не повинно знижувати рейтинг кандидата.

З огляду на це підхід прямого порівняння текстів за принципом «якщо тексти схожі – кандидат підходить» є методологічно недостатнім. Причина полягає в тому, що семантична близькість (зокрема отримана векторними моделями) не гарантує відповідності критичним вимогам і не відображає таких аспектів, як додатковий потенціал або особисті якості. Тому у роботі застосовується підхід, у якому:

– дані вакансії та резюме нормалізуються та подаються у вигляді, зручному для аналізу;

– формуються проміжні критерії оцінювання кандидата, які відповідають практиці рекрутингу;

– виконується агрегування критеріїв у фінальний рейтинг з використанням нечіткої логіки.

Процес обробки даних у модулі інтелектуального ранжування кандидатів складається з наступних логічних етапів:

– попередня обробка та уніфікація текстів: завантаження документів у форматі DOCX, очищення тексту, збереження структури за секціями (де це можливо). Це створює стабільну основу для подальших методів аналізу та мінімізує вплив «шуму» форматування;

– базові методи порівняння:

1) Keyword Matching як baseline, що визначає частку збігів ключових слів вакансії у тексті резюме;

2) Vector Semantic Analysis як baseline, що перетворює тексти у векторні подання та оцінює семантичну близькість через косинусну подібність;

– інтелектуальне багатокритеріальне оцінювання (LLM): велика мовна модель формує структурований результат аналізу, який включає щонайменше три такі компоненти;

1) явна відповідність (Explicit Match) – наскільки кандидат задовольняє вимоги вакансії;

2) бонусний потенціал (Bonus Potential) – наявність додаткових компетенцій, що не є прямими вимогами, але підсилюють цінність кандидата;

3) особисті якості (Soft Skills) – індикатори лідерства, комунікації, командної взаємодії тощо.

– агрегування результатів за допомогою нечіткої логіки: використовується алгоритм Мамдані для перетворення трьох числових оцінок (0–100) у фінальний рейтинг (0–100) на основі бази правил «ЯКЩО... ТО...», яка відтворює експертне міркування рекрутера [13].

Наявність етапів використання baseline-методів є важливою для подальшого експериментального порівняння з гібридним підходом.

Таким чином, запропонований підхід дозволяє оцінювати не лише «схожість текстів», а придатність кандидата як комплексний результат, де різні аспекти можуть підсилювати або компенсувати один одного відповідно до експертної логіки.

2.2 Моделі представлення та екстракції даних

Для коректного порівняння вакансії та кандидатів необхідно привести документи до уніфікованого внутрішнього представлення. У межах розробленого методу це представлення поєднує:

- текстову основу (очищений текст вакансії та резюме);
- структурні компоненти (секції документів, якщо їх можна виділити);
- оцінювальні компоненти (результат LLM-аналізу у вигляді трьох критеріїв).

Нехай вакансія подана об'єктом V , а кожне резюме кандидата – об'єктом

C_k . Для кожного документа формується кортеж:

$$D = \langle T_{raw}, T_{clean}, S, meta \rangle$$

де T_{raw} – вихідний текст, отриманий з DOCX;

T_{clean} – очищений текст після нормалізації;

S – множина секцій, що відображає внутрішню структуру документа;

$meta$ – метадані (ідентифікатор кандидата, кількість слів тощо).

Таке представлення відповідає практичній реалізації: система зберігає і «сирий» текст, і нормалізований, що дає змогу, за потреби, контролювати якість очищення або використовувати первинну структуру.

Ключовою відмінністю розробленого підходу є те, що формальна «оцінювальна модель» кандидата будується не шляхом жорстко заданих словників чи шаблонів, а через структурований висновок LLM, який повертається у стандартизованому форматі.

Для кожного резюме кандидата C_k – формується трійка оцінок:

$$R_k = \langle E_k, B_k, S_k \rangle$$

де E_k – оцінка явної відповідності від 0 до 100;

B_k – оцінка бонусного потенціалу від 0 до 100;

S_k – оцінка особистих якостей від 0 до 100.

Ці оцінки мають дві принципові властивості:

- структурованість: модель повертає значення у заданому форматі (JSON), що робить результат машинно-читаним та придатним для подальшої агрегації [14];

- контекстність: оцінювання виконується на основі одночасного аналізу тексту вакансії та резюме, а не за окремими локальними збігами термінів.

У межах методології важливо підкреслити: проміжні оцінки не є «остаточним рішенням», а слугують вхідними параметрами для формального блоку прийняття рішення – нечіткої системи.

2.3 Математична модель векторизації даних

Окремим компонентом дослідження є семантичний *baseline*, який дозволяє оцінити, наскільки кандидати «в цілому» близькі до тексту вакансії. На відміну від підходів на основі Word2Vec/TF-IDF, у розробленому рішенні застосовуються *sentence embeddings*, що формуються трансформерною моделлю по формулі 2.1.

$$Enc: T \rightarrow R^d, \quad (2.1)$$

де T – множина текстів;

d – розмірність простору вкладень (у реалізації використовується багатомовна модель, що забезпечує роботу з українською та англійською лексикою).

Для вакансії та резюме кандидата отримуємо формулу 2.2.

$$\vec{V} = Enc(T_V), \vec{C}_k = Enc(T_{C_k}), \quad (2.2)$$

Цей *baseline* є важливим, оскільки дозволяє:

- порівнювати тексти незалежно від точного збігу слів;
- зменшити вплив синонімії та різних формулювань;
- отримати ранжування кандидатів за «семантичною близькістю» як орієнтир для подальшого аналізу.

Разом з тим, семантичний *baseline* не розділяє вимоги на «критичні/важливі/бажані», а також не виділяє бонусний потенціал та особисті якості як окремі категорії. Саме тому він використовується для порівняння, але не є фінальним методом у роботі.

2.4 Методи розрахунку відповідності та ранжування кандидатів

У межах роботи використовуються три логічно різні методи оцінювання: Keyword Matching, Vector Semantic Analysis та Hybrid (LLM + Fuzzy). Формально кожен метод будує власну функцію оцінки кандидатів і породжує ранжування.

2.4.1 Keyword Matching як базовий метод

Нехай з тексту вакансії виділено множину ключових слів K_v (наприклад, топ-N найбільш частотних змістовних слів після фільтрації). Для кандидата формується множина слів K_{C_k} , що присутні у резюме. Тоді базовий показник збігу можна подати як частку перетину по формулі 2.3.

$$Score_{kw}(C_k, V) = \frac{|K_v \cap K_{C_k}|}{|K_v|} * 100, \quad (2.3)$$

Перевагою методу є простота та прозорість: легко пояснити, які саме терміни збіглися. Недоліком є те, що підхід:

- не враховує синонімію та контекст;
- може занижувати оцінку релевантним кандидатам через відмінні формулювання;
- погано працює зі складними вимогами, що не виражаються конкретним словом.

Тому Keyword Matching використовується як baseline і орієнтир для експериментального порівняння.

2.4.2 Метод «Семантична подібність»

Для оцінювання семантичної близькості між вакансією і резюме використовується косинусна подібність по формулі 2.4.

$$Sim(\vec{V}, \vec{C}_k) = \frac{\vec{V} * \vec{C}_k}{\|\vec{V}\| * \|\vec{C}_k\|}, \quad (2.4)$$

Оскільки у практичній інтерпретації зручно працювати зі шкалою 0–100, значення може бути приведене до інтервалу [0,100] шляхом лінійної нормалізації.

Переваги семантичного методу:

- автоматичне ранжування кандидатів;
- краща робота з контекстом і різними формулюваннями порівняно з keyword-пошуком.

Однак цей метод залишається «загальним»: він вимірює «схожість текстів», але не гарантує коректної багатокритеріальної оцінки з позиції рекрутингу. Зокрема, він не відокремлює:

- явну відповідність вимогам;
- додаткові компетенції (які інколи важливі як потенціал);
- особисті якості та поведінкові індикатори.

2.4.3 Суть гібридного підходу

Основним у роботі є підхід, який формує фінальний рейтинг кандидата у два кроки:

- 1) LLM отримує пару (T_v, T_{C_k}) і повертає структуровані проміжні оцінки

E_k, B_k, S_k ;

2) нечітка система Мамдані агрегує ці оцінки у фінальний рейтинг F_k .

У результаті для кожного кандидата будується фінальна оцінка за формулою 2.5.

$$F_k = Fuzzy(E_k, B_k, S_k), \quad (2.5)$$

Ранжування кандидатів визначається сортуванням F_k за спаданням.

Перевагою цього підходу є те, що він поєднує:

- інтелектуальний аналіз змісту – для виділення складних характеристик кандидата;
- формалізоване прийняття рішення – для стабільної, інтерпретованої агрегації критеріїв.

2.5 Теоретичні основи моделі оцінювання на основі нечіткої логіки

Семантичні та лексичні методи дають певні оцінки схожості, але не відображають логічні нюанси рекрутингу (компенсації, «м'які» межі, різні сценарії прийняття рішень). Тому у роботі використовується нечітка модель, яка приймає на вході три проміжні критерії LLM і повертає фінальний рейтинг кандидата.

2.5.1 Фазифікація

Фазифікація – це перетворення чітких числових значень E_k, B_k, S_k у ступені належності до лінгвістичних термів. Замість твердження «explicit =

76» система оперує поняттями типу «явна відповідність – середня/висока» зі ступенем належності $\mu \in [0, 1]$.

У роботі застосовується універсальна шкала 0–100 для всіх трьох входів та виходу (фінального рейтингу). Для кожної вхідної змінної визначаються лінгвістичні терми:

- x_1 : Явна відповідність – {низька, середня, висока};
- x_2 : Бонусний потенціал – {низький, середній, високий};
- x_3 : Особисті якості – {погані, середні, відмінні}.

Для вихідної змінної у Final Rating задаються терми {дуже низький, низький, середній, високий, дуже високий}.

З огляду на практичну природу задачі доцільно використовувати трапецієподібні та трикутні функції належності, оскільки:

- трапецієподібні дозволяють моделювати «плато впевненості» (діапазон значень, які експерт трактує однаково);
- трикутні зручні для проміжних категорій, де важливо описати «перехід» між крайніми термами.

2.5.2 База правил та нечіткого висновку

База правил визначає логіку прийняття рішення. У межах розробленого методу правила мають вигляд продукцій ЯКЩО (умови по x_1, x_2, x_3) ТО (рівень y).

Нижче наведено повний набір запропонованих правил:

- 1) ЯКЩО $x_1 =$ висока І $x_2 =$ високий І $x_3 =$ відмінні ТО $y =$ дуже високий;
- 2) ЯКЩО $x_1 =$ висока І $x_2 =$ високий ТО $y =$ дуже високий;
- 3) ЯКЩО $x_1 =$ висока І $x_3 =$ відмінні ТО $y =$ високий;
- 4) ЯКЩО $x_1 =$ висока І $x_2 =$ середній І $x_3 =$ відмінні ТО $y =$ високий;
- 5) ЯКЩО $x_1 =$ середня І $x_2 =$ високий І $x_3 =$ відмінні ТО $y =$ високий;

- 6) ЯКЩО $x_1 = \text{середня}$ І $x_2 = \text{високий}$ ТО $y = \text{середній}$;
- 7) ЯКЩО $x_1 = \text{висока}$ І $x_2 = \text{середній}$ ТО $y = \text{високий}$;
- 8) ЯКЩО $x_1 = \text{висока}$ І $x_2 = \text{низький}$ І $x_3 = \text{відмінні}$ ТО $y = \text{середній}$;
- 9) ЯКЩО $x_1 = \text{середня}$ І $x_2 = \text{середній}$ І $x_3 = \text{середні}$ ТО $y = \text{середній}$;
- 10) ЯКЩО $x_1 = \text{середня}$ І $x_2 = \text{середній}$ ТО $y = \text{середній}$;
- 11) ЯКЩО $x_1 = \text{висока}$ І $x_2 = \text{низький}$ І $x_3 = \text{середні}$ ТО $y = \text{середній}$;
- 12) ЯКЩО $x_1 = \text{низька}$ І $x_2 = \text{низький}$ ТО $y = \text{низький}$;
- 13) ЯКЩО $x_1 = \text{низька}$ І $x_2 = \text{середній}$ ТО $y = \text{низький}$;
- 14) ЯКЩО $x_1 = \text{середня}$ І $x_2 = \text{низький}$ І $x_3 = \text{погані}$ ТО $y = \text{низький}$;
- 15) ЯКЩО $x_1 = \text{низька}$ І $x_2 = \text{низький}$ І $x_3 = \text{погані}$ ТО $y = \text{дуже низький}$.

У підсумку наведена база правил реалізує практичну логіку рекрутингу: домінування явної відповідності, можливість компенсації середньої відповідності сильними додатковими характеристиками, а також жорстке зниження рейтингу у випадку низької відповідності без компенсуючих факторів.

2.5.3 Етап дефазифікації

Результатом роботи нечіткої системи є агрегована функція належності $\mu_{agg}(y)$. Для отримання чіткого числового рейтингу використовується метод центру тяжіння, формула 2.6.

$$Rank = \frac{\int y * \mu_{agg}(y) dy}{\int \mu_{agg}(y) dy}, \quad (2.6)$$

Цей підхід забезпечує збалансовану оцінку, оскільки враховує внесок усіх активованих правил і дозволяє формувати фінальне ранжування кандидатів за числовою шкалою.

Логіка виконання процесу рекрутингу представлена на рисунку 2.1. Схема відображає послідовність проходження даних через етапи обробки.

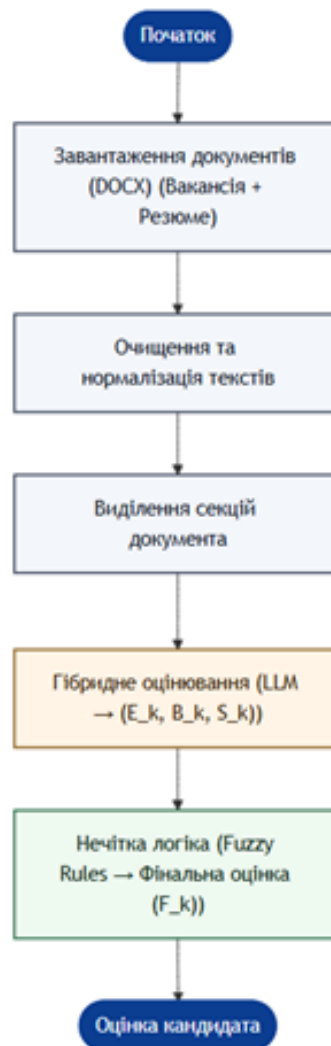


Рисунок 2.1 – Логіка виконання процесу рекрутингу

3 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РЕКРУТИНГУ ПЕРСОНАЛУ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Опис інформаційної технології інтелектуального оцінювання кандидатів

Для реалізації процесу автоматизованого відбору персоналу розроблено інформаційну технологію (ІТ) інтелектуального аналізу резюме, що базується на гібридному поєднанні великих мовних моделей (LLM) та нечіткої логіки. Опис технології виконано з використанням методології функціонального моделювання IDEF0.

На рисунку 3.1 наведена діаграма IDEF0, яка відображає розроблену ІТ та демонструє повний цикл обробки даних: від завантаження файлів вакансії та резюме до формування ранжованого списку кандидатів.

У межах діаграми показано послідовну трансформацію вхідних неструктурованих даних у формалізоване рішення про найм. Функціонування кожного етапу підтримується відповідними механізмами, зокрема програмними модулями підсистеми рекрутингу, прикладний програмний інтерфейс (API) OpenAI та обчислювальними бібліотеками Python.

Етап 1 «Завантаження вакансії та резюме» відповідає за введення даних у систему. Вхідними об'єктами є файли у форматі DOCX (опис вакансії та множина резюме кандидатів). Керуючим елементом виступають конфігураційні правила, що визначають шляхи до директорій та параметри зчитування. Механізмом реалізації є клас DataLoader. Вихідними даними є «сирий» текст документів.

Етап 2 «Очищення та нормалізація текстів»: на цьому кроці відбувається попередня обробка отриманого тексту. Керування здійснюється правилами очищення (Regular Expressions), які видаляють зайві пробіли, спецсимволи та нормалізують переноси рядків. Це критично важливо для зменшення «шуму» перед подачею даних в LLM. Результатом є нормалізований текст.

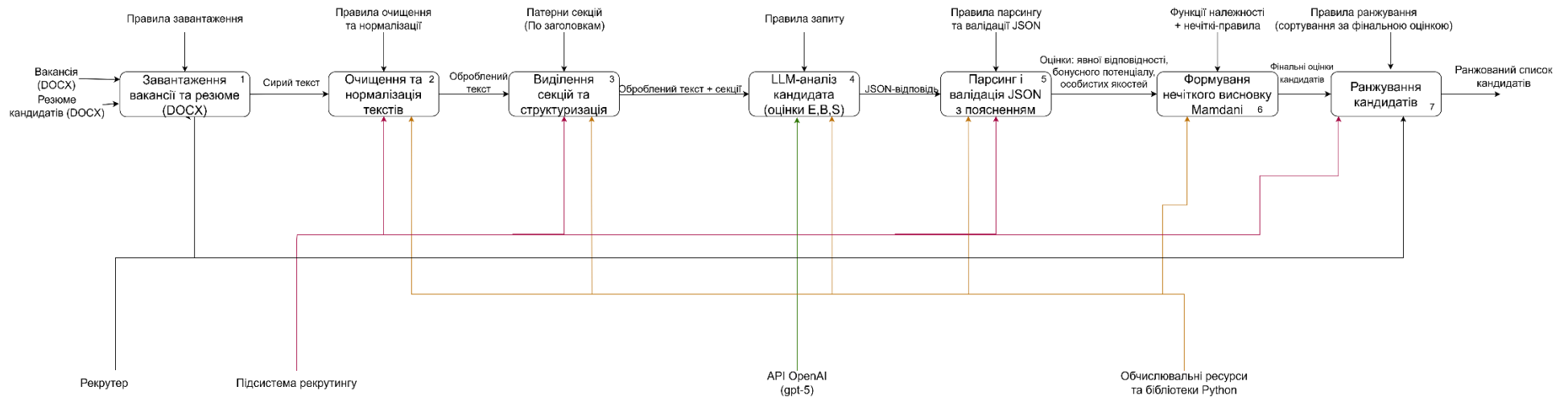


Рисунок 3.1 – IDEF0-діаграма ІТ рекрутингу персоналу

Етап 3 «Виділення секцій та структуризація» виконує сегментацію тексту резюме на логічні блоки (навички, досвід, освіта). Керуючим елементом є патерни заголовків (Markdown, Bold, двокрапки), реалізовані у класі DocumentProcessor. Це дозволяє подавати моделі не суцільний текст, а структурований документ, що підвищує точність аналізу.

Етап 4 «LLM-аналіз кандидата» – це центральний етап обробки, де відбувається семантичний аналіз тексту. Вхідними даними є оброблений текст вакансії та резюме. Керування здійснюється через складний інженерний промпт (Prompt), що містить інструкції щодо оцінювання та вимоги до JSON-формату. Механізмом виступає API OpenAI (модель gpt-5-mini або аналог). Результатом є JSON-відповідь із розрахованими метриками чіткої відповідності, бонусного потенціалу та особистих якостей.

Етап 5 «Парсинг і валідація JSON» забезпечує технічну стійкість системи. Він відповідає за перетворення текстової відповіді моделі у словник Python (dict), видалення маркерів форматування та перевірку наявності обов'язкових ключів. У разі помилки валідації система може ініціювати повторний запит або повернути помилку.

Етап 6 «Формування нечіткого висновку Mamdani»: на цьому етапі чіткі числові оцінки від LLM перетворюються на лінгвістичні змінні. Керуючим елементом є база правил нечіткої логіки («ЯКЩО... ТО...») та функції належності. Механізмом виступає бібліотека skfuzzy (клас FuzzyController). Вихідними даними є агрегована оцінка кандидата.

Етап 7 «Ранжування кандидатів» – це фінальний етап, який виконує сортування списку кандидатів за спаданням фінального рейтингу. Керування здійснюється правилами ранжування. Результатом роботи ІТ є ранжований список кандидатів, готовий для прийняття кадрового рішення рекрутером.

3.2 Програмна архітектура та механізми реалізації

Програмна реалізація системи виконана мовою Python з використанням об'єктно-орієнтованого підходу. Архітектура застосунку побудована модульно, що забезпечує гнучкість налаштування та можливість розширення.

Основні класи та їх призначення наведено нижче:

- клас `Config` виступає єдиною точкою входу для налаштувань системи.

У ньому інкапсульовано таке;

- 1) API-ключі та параметри моделі (`OPENAI_MODEL`); Шляхи до файлової системи;

- 2) параметри нечіткої логіки (універси для змінних `Explicit`, `Bonus`, `Soft` та `Rating` у діапазоні 0–100);

- 3) налаштування візуалізації. Така централізація дозволяє змінювати поведінку системи (наприклад, вагові коефіцієнти або модель ембедінгів) без втручання в основний код логіки;

- клас `DocumentProcessor` реалізує механізми роботи з текстовими даними. Особливістю реалізації є метод `extract_sections`, який використовує регулярні вирази для ідентифікації структури документа;

- клас `LLMAnalyzer` відповідає за взаємодію з генеративним штучним інтелектом. У цьому класі реалізовано механізм кешування (`analysis_cache`), що дозволяє уникнути повторних запитів до API для одних і тих самих кандидатів, заощаджуючи час та ресурси. Метод `parse_llm_response` містить логіку очищення відповіді від Markdown-тегів (```json`), що є поширеною проблемою при роботі з LLM;

- клас `FuzzyController` є програмною реалізацією математичної моделі, описаної у розділі 2. Він використовує бібліотеку `scikit-fuzzy` для побудови системи контролю [15]. У методі `_build_fuzzy_system` програмно задаються функції належності (трапецієподібні `trapmf` та трикутні `trimf`) для вхідних та вихідних змінних. База знань сформульована у вигляді списку об'єктів

ctrl.Rule, які пов'язують антецеденти з консеквентами. Цей механізм дозволяє реалізувати нелінійну залежність фінальної оцінки від вхідних параметрів. Фрагмент програмної реалізації бази правил, яка визначає логіку прийняття рішення системою, наведено у лістингу 1;

- клас HybridRecruitmentSystem виконує роль оркестратора (Facade pattern). Він поєднує роботу LLMAnalyzer та FuzzyController, передаючи результати аналізу тексту на вхід системи нечіткого висновку. Також клас відповідає за обробку виключень та формування підсумкового pandas.DataFrame.

Лістинг 1 – Програмна реалізація правил нечіткої логіки

```
rules = [
    # Ідеальні кандидати
    ctrl.Rule(explicit['висока'] & bonus['високий'] &
soft['відмінні'], rating['дуже_високий']),

    # Дуже хороші кандидати
    ctrl.Rule(explicit['висока'] & bonus['високий'],
rating['дуже_високий']),
    ctrl.Rule(explicit['висока'] & soft['відмінні'],
rating['високий']),
    ctrl.Rule(explicit['висока'] & bonus['середній'] &
soft['відмінні'], rating['високий']),

    # Хороші кандидати з бонусами компенсують недостатню
явну відповідність
    ctrl.Rule(explicit['середня'] & bonus['високий'] &
soft['відмінні'], rating['високий']),

    # Слабкі кандидати
    ctrl.Rule(explicit['низька'] & bonus['низький'],
rating['низький']),
    ctrl.Rule(explicit['низька'] & bonus['середній'],
rating['низький']),
```

```

        ctrl.Rule(explicit['низька'] & bonus['низький'] &
soft['погані'], rating['дуже_низький']),
    ]

```

3.3 Інтеграція з API мовної моделі

Ключовим компонентом розробленої технології є інтеграція з API OpenAI, яка забезпечує семантичний аналіз неструктурованого тексту. Взаємодія реалізована через клієнтську бібліотеку `openai` у класі `LLMAnalyzer` [16].

Процес запиту до API налаштовано з урахуванням специфіки задачі рекрутингу:

- параметр `model`: використовується модель класу `gpt-5-mini` (або `gpt-4o-mini` в залежності від конфігурації), яка забезпечує оптимальний баланс між вартістю та якістю розуміння контексту.

- параметр `temperature`: встановлено значення 1, що дозволяє моделі проявляти певну гнучкість при інтерпретації неявних навичок, але у поєднанні з жорстким системним промптом зберігає структуру.

- параметр `response_format`: встановлено значення `{"type": "json_object"}`. Це критично важливе налаштування, яке примушує модель генерувати валідний JSON, що усуває необхідність складного парсингу тексту та використання регулярних виразів для вилучення оцінок.

Метод `call_llm` також включає обробку помилок (`try-except block`), що забезпечує стабільність роботи системи при мережевих збоях або проблемах з доступом до API. Реалізація методу `call_llm` наведена у лістингу 2.

Лістинг 2 – Реалізація методу запиту до LLM API

```

def call_llm(self, prompt: str, max_tokens: int = 10000) ->

```

```

str:
    """
    Виконує запит до OpenAI API.
    Args:
        prompt: Промпт для моделі
        max_tokens: Максимальна довжина відповіді
    Returns:
        Відповідь моделі (текст)
    """
    try:
        response = self.client.chat.completions.create(
            model=self.model,
            messages=[
                {"role": "system", "content": "Ти експерт-рекрутер у IT."},
                {"role": "user", "content": prompt}
            ],
            max_completion_tokens=max_tokens,
            temperature=1
        )

        return response.choices[0].message.content.strip()

    except Exception as e:
        print(f"⚠ Помилка при запиті до LLM: {e}")
        raise

```

Запит (Prompt) у розробленій системі виступає не просто запитом, а керує когнітивним процесом моделі. Реалізований у методі `create_analysis_prompt` промпт має чітку модульну структуру, спрямовану на мінімізацію галюцинацій та забезпечення об'єктивності оцінювання.

Структура промпту включає наступні блоки:

- визначення формату виводу. На самому початку промпту задається жорстка схема JSON, яка наведена в лістингу 3. Це реалізує принцип "Code

First", змушуючи модель "думати" в термінах структурованих даних;

– блоки "вакансія" та "резюме кандидата" чітко розмежовані розділювачами, що допомагає моделі розрізнити джерело вимог та об'єкт аналізу;

– алгоритм оцінювання. Промпт містить покрокову інструкцію для розрахунку кожної метрики, що імітує логіку експерта за наступними атрибутами;

1) для явної відповідності введено класифікацію вимог: "Критичні" (вага 50), "Важливі" (30), "Бажані" (20). Визначено шкалу близькості технологій: 100% (точний збіг), 80% (пряма альтернатива), 50% (суміжна екосистема). Це дозволяє системі оцінювати не лише ключові слова, а й семантичну відповідність, вирішуючи проблему синонімії;

2) для бонусного потенціалу встановлено правило "Рахуй ТІЛЬКИ те, чого НЕМАЄ у вимогах". Це запобігає подвійному врахуванню одних і тих самих навичок. Введено категорії бонусів: суміжні технології, сертифікати, участь у Open-source, знання іноземних мов;

3) для особистих якостей замість абстрактної оцінки, запит вимагає шукати конкретні індикатори в тексті: "Очолював команду" (Лідерство), "Оптимізував на X%" (Результативність), "Менторив колег" (Комунікація);

– обмежувальні правила. Блок "КРИТИЧНІ ПРАВИЛА" містить директиви, що забороняють моделі вигадувати факти ("НІКОЛИ не здогадуйся") та вимагають наводити математичні розрахунки у полі explanation. Це забезпечує прозорість прийнятого рішення.

Лістинг 3 – Структура JSON файлу

```

{{
  "explicit_match": {{
    "score": [число 0-100],
    "explanation": "Формула: X+Y+Z=[результат]"
  }}
}}

```

```
}},  
"bonus_potential": {{  
  "score": [число 0-100],  
  "explanation": "Всього: X+Y+Z+W+V=[результат]"  
}},  
"soft_skills": {{  
  "score": [число 0-100],  
  "explanation": "Всього: X+Y+Z+W+V=[результат]"  
}}}
```

Така деталізація промпту дозволяє перетворити стохастичну природу генеративної моделі на детермінований інструмент оцінювання, де кожен бал має обґрунтування, засноване на вхідному тексті.

4 ЕКСПЕРЕМЕНТАЛЬНА ПЕРЕВІРКА ГІБРИДНОГО ПІДХОДУ РЕКРУТИНГУ НА ОСНОВІ LLM ТА НЕЧІТКОЇ ЛОГІКИ

Для проведення експерименту був використаний програмний засіб який є реалізацією модуля рекрутингу. Для експерименту було обрано задачу первинного відбору кандидатів на IT-вакансії.

На відміну від базових методів, які оцінюють відповідність резюме вакансії лише через збіг ключових слів або загальну семантичну подібність текстів, запропонований гібридний підходу реалізує більш повний цикл прийняття рішення: аналіз резюме мовною моделлю, формування проміжних оцінок за трьома незалежними критеріями, фазифікація, застосування набору експертних правил та обчислення фінального рейтингу.

4.1 Опис вхідних даних та первинної обробки даних

Для практичної апробації методу було використано комплект документів у форматі DOCX, що складається з однієї вакансії та п'яти резюме кандидатів. Така структура експерименту дозволяє перевірити роботу системи у сценарії «одна вакансія → множина кандидатів», що є типовим випадком для рекрутингу.

Вакансія сформована для позиції Senior Python Developer. Текст вакансії містить вимоги до технологічного стеку, досвіду роботи, умов взаємодії в команді та очікуваних нефункціональних компетенцій. Резюме кандидатів відрізняються як за обсягом, так і за стилем подачі інформації (таблиця 4.1). Це важливо, оскільки в реальних умовах резюме майже ніколи не мають єдиного стандарту.

Таблиця 4.1 – Характеристики вхідних документів

Документ	Позначення	Кількість слів
Вакансія	vacancy	314
Резюме 1 (ідеальний кандидат)	resume1_ideal	820
Резюме 2 (хороший кандидат)	resume2_good	740
Резюме 3 (середній кандидат)	resume3_middle	352
Резюме 4 (суміжний профіль)	resume4_adjacent	650
Резюме 5 (junior-кандидат)	resume5_junior	901

З наведених даних видно, що обсяг резюме коливається у широких межах (від 352 до 901 слова), тобто система має коректно працювати як із короткими документами, так і з великими текстами, що містять багато зайвої інформації.

Додатково важливим є не лише обсяг тексту, а й його структурованість, оскільки гібридний підхід використовує виділення секцій документа (наприклад, досвід, навички, проекти, освіта).

Саме тому виконується підготовчий етап, оскільки якість подальшого порівняння текстів прямо залежить від того, наскільки коректно виконано очищення та нормалізацію даних. На цьому етапі було виконано такі дії:

- зчитування DOCX-документів: текст витягується з усіх непорожніх параграфів документа. Для збереження логічної структури об'єднання виконується з переносами рядків, після чого формується суцільний текст;

- очищення та нормалізація: виконується уніфікація пробілів, усунення зайвих символів та нормалізація переносів рядків. Така обробка зменшує шум у тексті та підвищує стабільність роботи як базових методів, так і LLM

(рисунок 4.1 та 4.3);

– виділення секцій: застосовано універсальний механізм, що шукає заголовки різних типів (markdown-заголовки, заголовки з двокрапкою, виділені жирним тощо). Результатом є словник секцій, який дозволяє надалі подати текст LLM більш структуровано (рисунок 4.2).

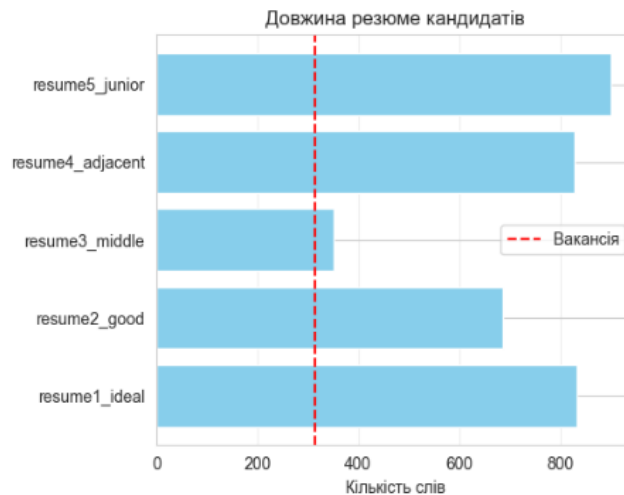


Рисунок 4.1 – Діаграма довжини резюме кандидатів

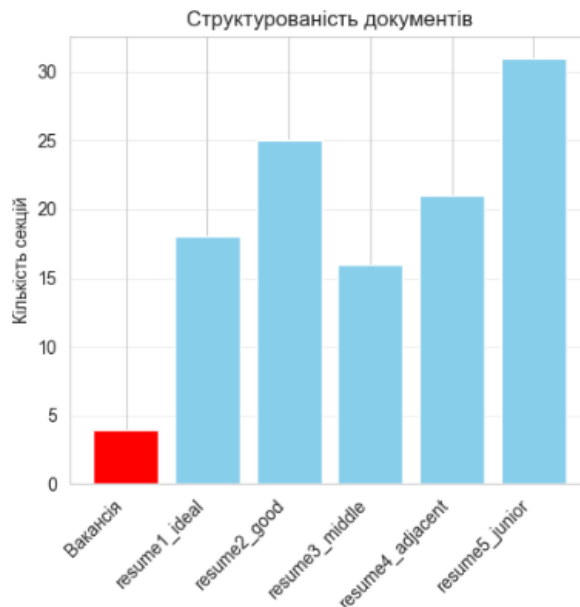


Рисунок 4.2 – Діаграма структурованості документів

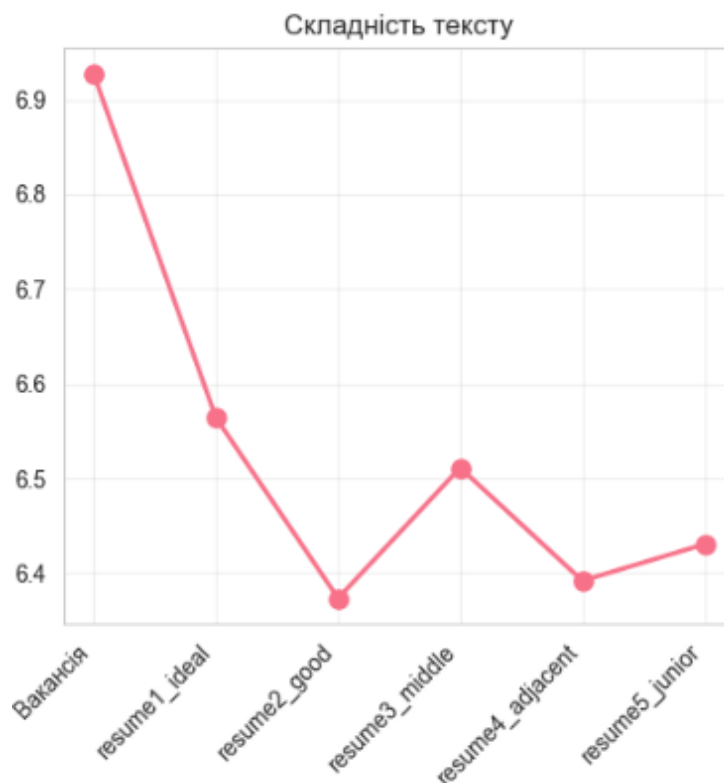


Рисунок 4.3 – Діаграма складності тексту резюме

На рисунку 4.1 показано, що найоб’ємнішим є резюме `resume5_junior`, яке суттєво перевищує вакансію за кількістю слів. Рисунок 4.2 демонструє, що вакансія має невелику кількість секцій, тоді як резюме є більш «розбитими» на частини, що характерно для сучасних резюме (CV) шаблонів. Рисунок 4.3 відображає середню довжину слова як простий індикатор «текстової складності»: вакансія є найбільш насиченою термінами (найвища середня довжина слова), а резюме мають близькі між собою значення. Це важливо, оскільки вакансії часто містять більше технічних термінів та аббревіатур, ніж резюме.

Для відтворюваності експерименту налаштування (шляхи до файлів, модель векторних представлень, параметри нечіткої логіки та параметри візуалізації) винесені у конфігураційний клас. Результати експерименту та графіки зберігаються у окремій директорії `results`, що забезпечує контроль збережених проміжних та фінальних результатів.

4.2 Експериментальна перевірка гібридного підходу

Гібридний підхід є основним результатом розробки, оскільки поєднує два взаємодоповнюючих механізми:

1) LLM-компонент, який виконує роль «інтелектуального аналітика» та формує проміжні оцінки кандидата за трьома критеріями: явна відповідність, бонусний потенціал, особисті якості;

2) контролер нечіткої логіки, який реалізує експертні правила прийняття рішення, працює з нечіткими поняттями («високий», «середній» тощо) і формує фінальний рейтинг.

Саме така архітектура дозволяє зробити рішення більш близьким до реального процесу рекрутингу: на практиці рекрутер ніколи не приймає рішення лише за одним критерієм.

4.2.1 Результати LLM-аналізу

На першому кроці гібридного підходу для кожного кандидата формується структурований результат LLM-аналізу. У підході закладено принцип «контрольованої оцінки»: LLM повертає не довільний текст, а строго задану JSON-структуру з числовими оцінками та короткими поясненнями.

Результати які було отримано від LLM наведено на рисунку 4.4 та в таблиці 4.2.

```
[5/5] Обробка resume5_junior...
Аналіз кандидата resume5_junior...
AAAAAAAAAAAA
{
  "explicit_match": {
    "score": 75,
    "explanation": "Формула: КРИТИЧНІ = (100*0.7 + 100*1.0 + 100*1.0 + 100*1.0) / 4 * 50 = (70 + 100 + 100 + 100) / 4 * 50 = 92.5 * 50/100 = 46.25; ВАЖЛИВІ = (100*1.0 + 100*1.0 + 50*1.0 + 80*1.0 + 80*1.0) / 5 * 30 = (100 + 100 + 50 + 80 + 80) / 5 * 30 = 82 * 30/100 = 24.6; БАЖАНІ = (80*1.0 + 30*1.0 + 50*1.0 + 0*1.0 + 0*1.0 + 30*1.0 + 0 + 0 + 0) / 9 * 20 = (80 + 30 + 50 + 0 + 0 + 30 + 0 + 0 + 0) / 9 * 20 = 21.111111... * 20/100 = 4.222222...; EXPLICIT_MATCH = 46.25 + 24.6 + 4.222222... = 75.072222... ≈ 75"
  },
  "bonus_potential": {
    "score": 20,
    "explanation": "Всього: А(дод. технології same ecosystem)=0 + Б(суміжні технології)=12 + В(освіта/сертифікати)=5 + Г(open-source/проекти)=3 + Д(дод. мови)=0 => 0+12+5+3+0=20"
  },
  "soft_skills": {
    "score": 39,
    "explanation": "Всього: Лідерство та вплив = менторинг(5) + code review(4) = 9; Досягнення та результати = запуск/модуль(8) + оптимізація(6) = 14; Команда робота = Agile/Scrum(4) + pair programming/code review(5) = 9; Комунікація = 0; Самостійність = remote readiness(4) + self-taught(3) = 7; SOFT_SKILLS = 9+14+9+0+7 = 39"
  }
}
✓ Рейтинг: 55.00
```

Рисунок 4.4 – Приклад результатів LLM-аналізу для resume5_junior

Таблиця 4.2 – Результати роботи LLM

Кандидат	Оцінка явної відповідності	Оцінка бонусного потенціалу	Оцінка особистих якостей
resume1_ideal	100	65	69
resume2_good	90	36	93
resume3_middle	76	37	51
resume4_adjacent	51	56	14
resume5_junior	75	20	39

Отримані значення відображають логіку експертної оцінки:

- resume1_ideal має максимальну явну відповідність, що відповідає очікуванню для «ідеального» кандидата;
- resume2_good має дуже високий рівень особистих якостей, що сигналізує про управлінські/комунікаційні переваги;
- resume4_adjacent має високий бонусний потенціал (суміжні компетенції), але суттєво слабкі особисті якості, що у реальному рекрутингу часто є «ризиком»;
- resume5_junior демонструє середні значення та низький бонусний потенціал, що узгоджується з типовим профілем junior-рівня.

Таким чином, після LLM-аналізу система вже має інтерпретовану картину кандидатів, однак остаточне рішення ще не формується – воно

визначається наступним кроком.

4.2.2 Фазифікація проміжних оцінок

На етапі фазифікації числові значення переводяться у нечіткі лінгвістичні категорії. Важливість цього кроку полягає у тому, що кандидат не завжди належить лише до однієї категорії. Наприклад, значення 65 може одночасно бути частково «середнім» і частково «високим». Це дозволяє моделювати людське мислення, де оцінювання є плавним, а не дискретним.

Результати фазифікації для всіх кандидатів наведені у таблиці 4.3, де значення показують ступінь належності до категорій.

Таблиця 4.3 – Результати фазифікації вхідних метрик

Кандидат	Явна відповідність (низька / середня / висока)	Бонусний потенціал (низький / середній / високий)	Особисті якості (погані / середні / відмінні)
resume1_ideal	0.00 / 0.00 / 1.00	0.00 / 0.25 / 0.25	0.00 / 0.05 / 0.45
resume2_good	0.00 / 0.00 / 1.00	0.20 / 0.30 / 0.00	0.00 / 0.00 / 1.00
resume3_middle	0.00 / 0.00 / 0.80	0.15 / 0.35 / 0.00	0.00 / 0.95 / 0.00
resume4_adjacent	0.00 / 0.95 / 0.00	0.00 / 0.70 / 0.00	1.00 / 0.00 / 0.00
resume5_junior	0.00 / 0.00 / 0.75	1.00 / 0.00 / 0.00	0.05 / 0.45 / 0.00

Таблиця 4.3 наочно демонструє таке:

- високі значення явної відповідності для перших двох кандидатів однозначно відносяться до категорії «висока»;
- бонусний потенціал для resume1_ideal знаходиться на межі між «середній» і «високий», що відкриває можливість для підсилення рейтингу;
- resume4_adjacent має середню явну відповідність і середній бонусний

потенціал, але особисті якості однозначно належать до «поганих», що суттєво впливатиме на рішення.

4.2.3 Активація правил нечіткої логіки та формування вихідного балу кандидата

Після фазифікації система активує набір експертних правил типу «ЯКЩО–ТО». У реалізації використано 15 правил, які відображають типові рекрутингові сценарії. Наприклад:

- ідеальні кандидати формуються при поєднанні високої явної відповідності, високого бонусного потенціалу і відмінних особистих якостей;
- хороші кандидати можуть мати високий явну відповідність і компенсувати нижчий бонусний потенціал за рахунок сильних особистих якостей;
- кандидати із середньою явною відповідністю можуть підвищуватися за рахунок високого бонусного потенціалу (суміжних компетенцій);
- слабкі особисті якості можуть знижувати рейтинг навіть при наявності бонусних навичок.

Фактично на цьому кроці система переходить від простого «порівняння текстів» до логічного висновку, де важливі не лише значення метрик, а їх комбінації.

У коді робота нечіткої логіки пояснюється через функції належності, які відображено на рисунку 4.5 та 4.6.

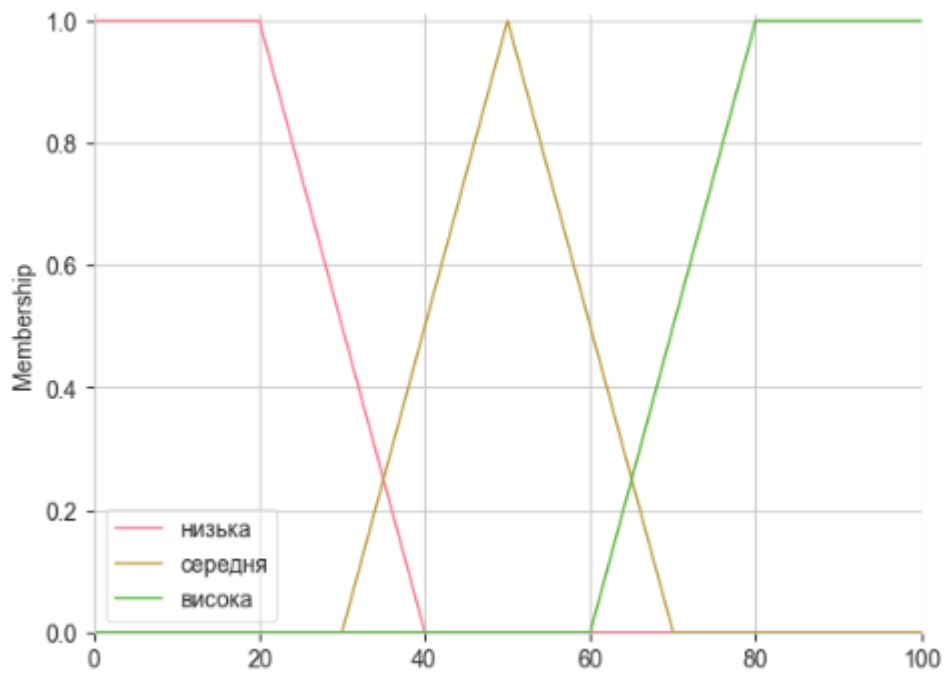


Рисунок 4.5 – Функція належності для атрибутів

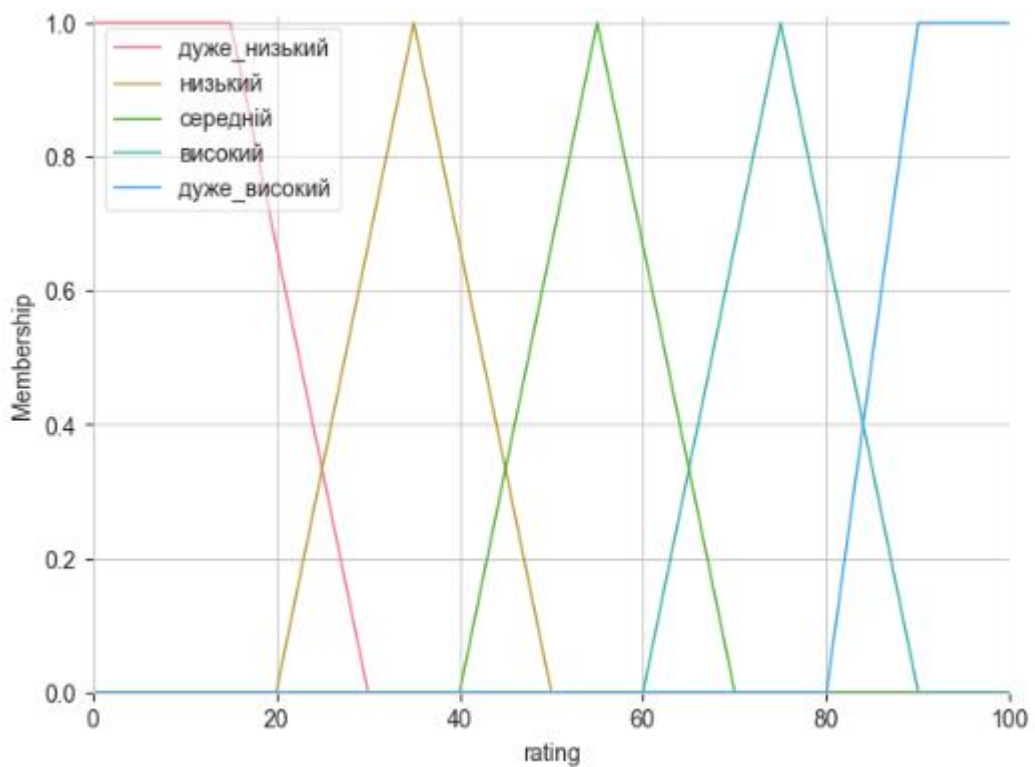


Рисунок 4.6 – Функція належності для Final Rating

На рисунках 4.5 та 4.6 видно, що для кожної змінної визначені три категорії для входів та п'ять категорій для виходу (від «дуже низького» до «дуже високого» рейтингу). Це забезпечує достатню гнучкість, щоб система могла відрізнити «середнього кандидата» від «хорошого» та «дуже хорошого», навіть якщо їхні числові показники близькі.

4.2.4 Результати застосування гібридного підходу

Після активації правил і агрегації нечітких висновків виконується дефазифікація та отримується фінальний рейтинг кожного кандидата. Результати гібридного методу наведено в таблиці 4.4.

Таблиця 4.4 – Фінальний рейтинг кандидатів

Ранг	Кандидат	Фінальна оцінка	Оцінка явної відповідності	Оцінка бонусного потенціалу	Оцінка особистих якостей
1	resume1_ideal	79.21	100	65	69
2	resume2_good	70.05	90	36	93
3	resume3_middle	68.86	76	37	51
4	resume4_adjacent	55.00	51	56	14
5	resume5_junior	55.00	75	20	39

Результати таблиці 4.4 є показовими з точки зору «людської» логіки відбору:

- resume1_ideal отримує найвищий рейтинг завдяки максимальній явній відповідності та достатніх бонусному потенціалу та особистим якостям для підтвердження зрілості профілю;

- resume2_good поступається першому кандидату через нижчий

бонусний потенціал, але отримує високу оцінку завдяки дуже сильним особистим якостям;

- resume3_middle має адекватний середній рейтинг. Явна відповідність достатня, але бонусний потенціал і особисті якості не дають підстав для більш високого результату;

- resume4_adjacent і resume5_junior отримали однаковий фінальний рейтинг, проте з різних причин: перший має сильніший бонусний потенціал, але дуже слабкі особисті якості, а другий має непогану явну відповідність за рахунок переліку технологій, але слабкий бонусний потенціал і помірні особисті якості.

Було відображено гістограму, яка показує, як різні кандидати набирають бали за трьома компонентами (рисунок 4.7). Наприклад, видно, що resume2_good має найвищі особисті якості серед усіх кандидатів, тоді як resume4_adjacent має найнижчі. Радарна діаграма демонструє профілі топ-3 кандидатів і підтверджує, що «найкращий» кандидат формується не за одним показником, а за збалансованістю.

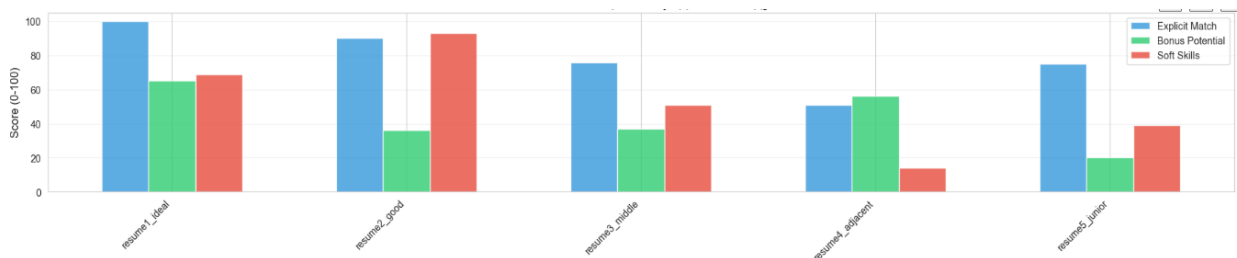


Рисунок 4.7 – Гістограма балів кандидатів за трьома компонентами

Також було зроблено матрицю кореляцій, яка показує, що фінальний рейтинг найтісніше пов'язаний з явною відповідністю, а також помітно залежить від особистих якостей (рисунок 4.8). Бонусний потенціал впливає менш стабільно, проте саме він може «підтягувати» кандидата із середньою явною відповідністю у випадках, коли додаткові компетенції є критично важливими для команди.



Рисунок 4.8 – Кореляція між компонентами

Також було візуалізовано три scatter-графіки, які демонструють залежність фінального рейтингу від кожної компоненти окремо. Вони важливі тим, що підтверджують: фінальний рейтинг не є прямою лінійною функцією лише явна відповідність (рисунок 4.9). Два кандидати можуть мати схожу явну відповідність, але різні особисті якості – і отримати різні підсумкові рішення.



Рисунок 4.9 – Scatter-графіки залежності фінального рейтингу від кожної компоненти окремо

4.3 Результати експерименту з базовими методами

Для об'єктивної оцінки якості гібридного підходу необхідно порівняти його з простішими підходами, які часто застосовуються на практиці. У роботі як базові використано Keyword Matching та Vector Semantic Analysis.

Таке порівняння дозволяє показати, що гібридний підхід не просто формує рейтинг, а забезпечує більш адекватну логіку ранжування кандидатів.

4.3.1 Результати методу Keyword Matching

Метод Keyword Matching реалізує найпростішу ідею: з вакансії виділяється набір ключових слів (топ-50 за частотою з фільтрацією коротких і службових слів), після чого для кожного кандидата рахується частка ключових слів вакансії, які зустрічаються в резюме.

Під час запуску методу система повідомляє, що з вакансії було отримано 50 ключових слів, а також виводить приклади таких слів (рисунок 4.10). Це важливо для інтерпретації результатів: користувач (рекрутер) може швидко побачити, що саме система вважає «ключовими сигналами» вакансії.

```
=====  
МЕТОД 1: KEYWORD MATCHING  
=====  
Витягнуто 50 ключових слів з вакансії  
Приклади: ['postgresql', 'спеціалізується', 'фінансового', 'процесів', 'lead', 'міжнародна', 'design', 'команда', 'python', 'професіоналів']
```

Рисунок 4.10 – Приклади ключових слів

Результати роботи есперименту з Keyword Matching можна побачити в таблиці 4.5.

Таблиця 4.5 – Результати експерименту з методом Keyword Matching

Ранг	Кандидат	Оцінка (%)
1	resume1_ideal	52.0
2	resume2_good	50.0
3	resume5_junior	46.0
4	resume3_middle	38.0
5	resume4_adjacent	30.0

Отримані результати є логічними лише частково. З одного боку, ideal-кандидат дійсно має найвищу оцінку. З іншого боку, junior-кандидат отримує доволі високе місце у рейтингу. Це пояснюється тим, що метод не аналізує глибину досвіду, а лише фіксує наявність термінів у тексті. Якщо junior-кандидат перерахував багато технологій, то метод може завищувати його оцінку, навіть якщо досвід є поверхневим.

Таким чином, Keyword Matching є корисним лише як швидкий фільтр, але не забезпечує якісного ранжування кандидатів у ситуаціях, де важлива інтерпретація контексту.

4.3.2 Результати методу Vector Semantic Analysis

Метод семантичного аналізу використовує модель векторних вкладень paraphrase-multilingual-MiniLM-L12-v2, що перетворює текст вакансії та резюме у числові вектори однакової розмірності. Після цього обчислюється семантична подібність між вакансією та кожним резюме.

Важливо, що такий метод вже здатен враховувати контекст і «схожі формулювання», однак залишається важко інтерпретованим, адже високий показник подібності не означає, що кандидат має саме ті компетенції, які критично потрібні вакансії.

Під час виконання методу система завантажує модель векторних представлень, формує embedding вакансії та повідомляє його розмірність, формує embeddings резюме і обчислює подібність для кожного кандидата (рисунок 4.11).

```

Завантаження моделі paraphrase-multilingual-MiniLM-L12-v2...
✓ Модель завантажено

=====
МЕТОД 2: VECTOR SEMANTIC ANALYSIS
=====

Створення embedding для вакансії...
Batches: 100%|██████████| 1/1 [00:00<00:00, 10.20it/s]
✓ Розмірність вектору: 384

Створення embeddings для резюме...
Batches: 100%|██████████| 1/1 [00:00<00:00, 6.01it/s]

Розрахунок семантичної подібності...
Оцінка кандидатів: 100%|██████████| 5/5 [00:00<00:00, 1292.38it/s]

✓ Оцінювання завершено

```

Рисунок 4.11 – Процес обчислень методу

Таблиця 4.6 – результати експерименту з методом Vector Semantic Analysis

Ранг	Кандидат	Оцінка
1	resume5_junior	85.796
2	resume1_ideal	84.230
3	resume3_middle	84.118
4	resume2_good	82.290
5	resume4_adjacent	81.850

Ключовий практичний висновок з цих результатів полягає у тому, що семантичний метод віддає перше місце junior-кандидату. Це пов'язано з тим, що об'ємне резюме з великою кількістю загальних технологічних описів часто

має високу семантичну близькість до вакансії як тексту, навіть якщо кандидат не відповідає позиції за рівнем. Отже, семантичний метод добре працює як інструмент пошуку схожих документів, але не як механізм рекрутингового рішення.

4.4 Порівняльний аналіз трьох методів та інтерпретація результатів

Після отримання результатів усіх методів було сформовано описову статистику оцінок (рисунок 4.12).

1. ОПИСОВА СТАТИСТИКА ОЦІНОК:			
	keyword_score	semantic_score	hybrid_rating
count	5.000000	5.000000	5.000000
mean	43.200000	83.652931	65.622719
std	9.121403	1.596630	10.491176
min	30.000000	81.852867	55.000000
25%	38.000000	82.294846	55.000000
50%	46.000000	84.117599	68.855071
75%	50.000000	84.203102	70.052632
max	52.000000	85.796219	79.205891

Рисунок 4.12 – Описова статистика оцінок

Описова статистика дозволяє оцінити «розкид» оцінок різних методів. У експерименті отримано такі узагальнення:

- keyword matching має середнє значення близько 43 та значний розкид оцінок;
- semantic analysis має середні значення близько 83–86 і дуже малий розкид, що вказує на слабку роздільну здатність методу у даному сценарії;
- hybrid method формує більш диференційовані оцінки, що відповідає реальному рекрутинговому ранжуванню.

Окрім того було проведено кореляційний аналіз (рисунок 4.13).



Рисунок 4.13 – Кореляційний аналіз методів

Кореляційний аналіз показує, наскільки схожі підходи між собою:

- кореляція між Keyword і Hybrid є помірною (близько 0.35), що логічно, оскільки обидва методи частково реагують на явну наявність термінів/вимог;

- кореляція між Semantic і Hybrid практично відсутня (близько -0.01), що підтверджує різну природу рішень: semantic similarity оцінює «схожість текстів», тоді як гібридний підхід оцінює «професійну відповідність»;

- кореляція між Keyword і Semantic є низькою (близько 0.08).

Отже, гібридний підхід забезпечує більш узгоджене з очікуваннями ранжування, оскільки враховує не лише «схожість текстів», а реальну відповідність вимогам та якісні характеристики кандидата.

ВИСНОВКИ

У магістерській кваліфікаційній роботі вирішено актуальне науково-прикладне завдання підвищення ефективності процесів рекрутингу в корпоративних інформаційних системах шляхом розробки інформаційної технології інтелектуального оцінювання кандидатів.

Основні наукові та практичні результати роботи полягають у наступному:

- проведено системний аналіз предметної області, який виявив суттєві обмеження існуючих ATS-систем (Applicant Tracking Systems). Встановлено, що традиційні методи (пошук за ключовими словами) та базовий векторний аналіз не здатні адекватно оцінювати контекст, особисті навички (Soft Skills) та прихований потенціал кандидатів, що призводить до помилок при наймі;

- обґрунтовано доцільність використання гібридного підходу, який поєднує можливості великих мовних моделей (LLM) для семантичного аналізу неструктурованого тексту резюме та апарат нечіткої логіки для прийняття формалізованих рішень в умовах невизначеності;

- запропоновано гібридний метод інтелектуального ранжування кандидатів, який, на відміну від існуючих аналогів, базується на двохетапній обробці даних: спочатку LLM видобуває структуровані метрики (явна відповідність, бонусний потенціал, soft skills), а потім система нечіткого виведення (Mamdani) агрегує їх у фінальний рейтинг. Це дозволяє моделювати експертну логіку рекрутера, враховуючи компенсацію одних критеріїв іншими. Окрім того, удосконалено формалізацію вхідних даних шляхом розробки спеціалізованого промпту, що забезпечує отримання від LLM строго типізованої JSON-структури з обґрунтуванням оцінок, що вирішує проблему «чорної скриньки» та забезпечує прозорість прийняття рішень;

- розроблено інформаційну технологію та програмну реалізацію системи мовою Python з використанням API OpenAI та бібліотеки scikit-fuzzy.

Архітектура системи включає модулі попередньої обробки документів (DOCX), інтелектуального аналізу та нечіткого логічного контролера. Створено базу правил нечіткої логіки, що формалізує типові сценарії оцінювання;

– виконано експериментальну перевірку розробленої технології на реальних даних (вакансія Python Developer та набір різнотипних резюме). Порівняльний аналіз показав переваги запропонованого підходу над базовими методами. Гібридний підхід забезпечив найбільш адекватне ранжування, коректно ідентифікувавши ідеального кандидата та відфільтрувавши кандидатів з низькими особистими якостями, незважаючи на їх технічну відповідність.

Практична цінність роботи полягає у створенні готового до інтеграції програмного модуля, який дозволяє автоматизувати рутинний скринінг резюме, скоротити час на первинний відбір та зменшити вплив людського фактору та когнітивних викривлень при оцінюванні.

Перспективами подальших досліджень є розширення бази правил нечіткої логіки, адаптація системи для аналізу портфоліо та GitHub-профілів, а також інтеграція розробленого модуля з популярними ERP/HRM-системами.

Результати роботи були опубліковані на XVI International Scientific and Practical Conference «Conceptual Framework and Dynamics of the Development of Science» (December 15, 2025). Munich, Germany [16].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи для (здобувачів вищої освіти усіх форм навчання другого (магістерського) рівня вищої освіти спеціальності 122 Комп'ютерні науки освітньо-професійної програми «Інформаційні управляючі системи та технології»). / Упоряд.: К.Е. Петров, С.Ф. Чалий, О.Д. Міхнова. – Харків: ХНУРЕ, 2024. – 19 с.
2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. Чинний від 2017-07-01. – Київ: ДП «УкрНДНЦ», 2016. – 26 с.
3. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. – Вид. офіц. Київ : УкрНДНЦ, 2016. 16 с.
4. Fuzzy Logic-based Recruitment System: An Innovative Solution for HR Departments – IJERT. IJERT – International Journal of Engineering Research & Technology. URL: <https://www.ijert.org/fuzzy-logic-based-recruitment-system-an-innovative-solution-for-hr-departments> (дата звернення: 09.11.2025).
5. G. Venkateshwaran,. Artificial intelligence in HR: transforming recruitment and selection in IT industry. Journal of information systems engineering and management. 2025. Vol. 10, no. 17s. P. 38–45. URL: <https://doi.org/10.52783/jisem.v10i17s.2705> (дата звернення: 09.11.2025).
6. Pah C. E. A., Utama D. N. Decision support model for employee selection based on data mining and fuzzy logic. Journal of computer science. 2020. Vol. 16, no. 5. P. 686–701. URL: <https://doi.org/10.3844/jcssp.2020.686.701> (дата звернення: 10.11.2025).
7. Role of artificial intelligence in employee recruitment: systematic review and future research directions / S. M. U. Dadaboyev et al. Discover global society. 2025. Vol. 3, no. 1. URL: <https://doi.org/10.1007/s44282-025-00246-w> (дата

звернення: 11.11.2025).

8. Amre Y., Ahmedabadwala M., Damania H. Decision making approach in recruitment using neuro-fuzzy system. International journal of computer applications. 2018. Vol. 180, no. 34. P. 29–33.

URL: <https://doi.org/10.5120/ijca2018916882> (дата звернення: 11.11.2025).

9. Dastin J. Amazon scraps secret AI recruiting tool that showed bias against women. Reuters. 2018. URL: <https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G> (дата звернення: 11.11.2025)

10. Enhancing person-job fit for talent recruitment / C. Qin et al. SIGIR '18: The 41st International ACM SIGIR conference on research and development in Information Retrieval, Ann Arbor MI USA. New York, NY, USA, 2018. URL: <https://doi.org/10.1145/3209978.3210025> (дата звернення: 12.11.2025).

11. Najjar A., Amro B., Macedo M. An intelligent decision support system for recruitment: resumes screening and applicants ranking. Informatica. 2021. Vol. 45, no. 4. URL: <https://doi.org/10.31449/inf.v45i4.3356> (дата звернення: 13.11.2025).

12. Walek B., Pektor O., Farana R. Decision support system for evaluating suitable job applicants. Mathematics. 2021. Vol. 9, no. 15. P. 1773. URL: <https://doi.org/10.3390/math9151773> (дата звернення: 13.11.2025).

13. Afenchenko D., Borysenko T. Дослідження підходів і моделей рекрутингу персоналу в корпоративній інформаційній системі// XVI International Scientific and Practical Conference “Conceptual Framework and Dynamics of the Development of Science” (December 15,2025). Munich, Germany. Pp.268-271. URL: <https://eu-conf.com/en/events/conceptual-framework-and-dynamics-of-the-development-of-science/>

14. Bray T. The JavaScript Object Notation (JSON) Data Interchange Format. IETF Request for Comments (RFC) 8259. 2017. URL: <https://tools.ietf.org/html/rfc8259> (дата звернення: 17.11.2025).

15. Warner J. D. et al. scikit-fuzzy: Fuzzy logic toolkit for SciPy. Zenodo. 2019. URL: <https://doi.org/10.5281/zenodo.3541386> (дата звернення: 18.11.2025).

16. OpenAI API Documentation. OpenAI Platform. URL: <https://platform.openai.com/docs/introduction> (дата звернення: 18.11.2025).