МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

СІМАК АНДРІЙ ЮРІЙОВИЧ

УДК 004.4:504.5:621.43.064

ДИСЕРТАЦІЯ

ІНТЕЛЕКТУАЛІЗОВАНА ПРОГРАМНА СИСТЕМА АНАЛІЗУ ТА РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

Спеціальність 121 – Інженерія програмного забезпечення
Галузь знань 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії.
Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело
_____ А.Ю. Сімак.

Науковий керівник: Пукас Андрій Васильович, доктор технічних наук, професор

Тернопіль – 2025

АНОТАЦІЯ

Сімак А.Ю. Інтелектуалізована програмна система аналізу та рейтингування діяльності науково-педагогічних працівників. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 121 «Інженерія програмного забезпечення» – Західноукраїнський національний університет, Тернопіль, 2025.

Підготовка здійснювалась на кафедрі комп'ютерних наук Західноукраїнського національного університету Міністерства освіти і науки України.

Дисертаційну роботу присвячено розв'язанню актуального науково-практичного завдання – розробки інтелектуалізованої програмної системи аналізу та рейтингування діяльності НПП, яка забезпечить комплексний аналіз, прогнозування та формування персоналізованих рекомендацій щодо покращення результативності науково-педагогічної діяльності для НПП.

У сучасних умовах глобалізації та посилення конкуренції у сфері вищої освіти, позиції закладів вищої освіти (ЗВО) у національних та міжнародних рейтингах безпосередньо залежать від сукупної результативності їхнього академічного персоналу. Ефективність діяльності кожного НПП стає ключовим фактором, що впливає на якість освітнього процесу, науковий потенціал та міжнародну конкурентоспроможність університету. Це зумовлює потребу впровадження дієвих інструментів для моніторингу, аналізу та, що найважливіше, стимулювання професійного розвитку працівників.

Більшість існуючих систем оцінювання та рейтингування, особливо в українських ЗВО, функціонують за традиційними принципами, що характеризуються високою трудомісткістю, значним адміністративним навантаженням, суб'єктивністю та обмеженими аналітичними можливостями. Такі системи здебільшого фіксують минулі досягнення, проте не надають дієвих інструментів для стимулювання професійного розвитку, або наявні механізми, як правило, є фрагментарними й обмежуються лише окремими видами діяльності

працівника. Виникає протиріччя між нагальною потребою у підвищенні результативності НПП та невідповідністю наявних засобів сучасним вимогам інтелектуалізації та автоматизації. Саме тому у дисертаційній роботі запропоновано комплексний підхід та його програмну реалізацію у вигляді інтелектуалізованої системи для аналізу та рейтингування НПП, яка зміщує акцент з простого контролю на надання персоналізованих рекомендацій для покращення результативності діяльності НПП.

У першому розділі дисертації проведено аналітичний огляд проблеми оцінювання, рейтингування та підвищення результативності діяльності НПП. Систематизовано теоретичні основи та критерії оцінювання, що охоплюють навчально-методичну, науково-дослідну, організаційно-виховну та міжнародну діяльність.

Після аналізу існуючих підходів до рейтингування було виокремлено традиційні системи, що базуються на ручному введенні даних, та інтелектуалізовані системи, що використовують автоматизацію та технології штучного інтелекту (ШІ). Розглянуто методи та засоби підвищення результативності, зокрема проаналізовано різні типи рекомендаційних систем, від колаборативної фільтрації до підходів на основі глибокого навчання, графових нейронних мереж та великих мовних моделей. Показано, що більшість існуючих систем, особливо в українських ЗВО, не повною мірою використовують потенціал інтелектуалізації. Саме необхідність подолання цих обмежень шляхом розробки відповідних методів та ефективних архітектурних рішень, які б інтегрували сучасні технології ШІ для надання персоналізованих рекомендацій, формує науковий контекст та обґрунтовує актуальність даного дослідження.

У завершальній частині розділу, виходячи з актуальності проблеми стимулювання професійного розвитку працівників і після аналізу методів та інструментів її вирішення, сформовано завдання для досягнення цілей дисертаційного дослідження.

У другому розділі дисертаційної роботи розроблено математичне забезпечення для побудови інтелектуалізованої системи та формування персоналізованих рекомендацій діяльності для НПП.

Розроблено модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі. Цей підхід дозволяє ефективно структурувати різнорідні та взаємопов'язані сутності (працівники, критерії, публікації, проєкти), що є ключовою передумовою для аналізу, виявлення прихованих закономірностей та реалізації процедури генерації персоналізованих рекомендацій. Також описано, як ця модель використовується для збору контекстних даних (наприклад, приклади успішних публікацій чи проєктів), щоб розширити та деталізувати рекомендації.

Запропоновано комбінований метод ранжування НПП для підвищення релевантності персоналізованих рекомендацій, що базується на комплексі метрик подібності для автоматизованого визначення працівників з максимально наближеними кращими показниками діяльності. Метод інтегрує метрику динаміки часового зміщення для аналізу часових рядів, косинусну подібність для порівняння співвідношень активностей та модифікований індекс Жаккара для врахування змістовної близькості виконуваних робіт.

Розроблено інтелектуалізований метод формування персоналізованих рекомендацій щодо діяльності працівників на наступний звітний період, який ґрунтується на архітектурі гетерогенних графових нейронних мереж (ГГНМ) з механізмами просторової та часової уваги. Для перетворення отриманих результатів прогнозування у розгорнуті текстові рекомендації, запропоновано використання великих мовних моделей, з динамічним уточненням структури промптів, що забезпечило їхню аргументованість, практичну цінність та скорочення часу їх формування.

У завершальній частині розділу представлено порівняльний аналіз рекомендованих критеріїв, сформованих запропонованим методом, та існуючих аналогів. Результати засвідчили, що запропонований підхід на основі ГГНМ, який враховує діяльність успішніших колег та семантику критеріїв, демонструє вищу

точність у прогнозуванні діяльностей, які НПП фактично виконують у наступному звітному періоді. Додатково було проведено апробацію системи на тестовій групі працівників. Аналіз її результатів засвідчив, що учасники взаємодіяли приблизно з половиною запропонованих рекомендацій. Важливо, що до 30% від загальної кількості рекомендованих діяльностей було виконано на прогнозований бал або вище, що сприяло реальному зростанню їхніх річних показників у середньому на 27% у порівнянні з попередніми результатами.

Третій розділ дисертаційної роботи присвячено опису спроєктованої архітектури інтелектуалізованої програмної системи. Вона базується на підході, що поєднує елементи модульної та сервіс-орієнтованої архітектур для забезпечення високого рівня гнучкості, масштабованості та незалежності функціональних компонентів. У розділі розглянуто структуру ключових компонентів: системи рейтингування, модуля персоналізованих рекомендацій, визначення відповідності НПП ліцензійним вимогам щодо здійснення освітньої діяльності та модуля збору даних. Також наведено особливості проектування основних модулів.

Особливу увагу приділено комбінованій системі зберігання даних, де реляційна база даних MySQL використовується для основного функціоналу, а графова Neo4j — для моделювання складних взаємозв’язків при формуванні рекомендацій. Такий підхід дозволив оптимізувати продуктивність та ефективно вирішувати задачі різного типу. Описано механізми взаємодії між модулями та зовнішніми сервісами, зокрема через REST API, токени доступу та спеціалізовані скрипти обміну даними. Такий підхід дозволив оптимізувати продуктивність системи.

Для візуалізації архітектурних рішень та процесів у розділі наведено UML-діаграми, такі як use-case діаграми, діаграми послідовності та ER-діаграми, що дозволяють визначити основні вимоги до системи та проілюструвати механізми взаємодії між модулями, життєвий цикл звіту та розподіл функціоналу між ролями користувачів.

У четвертому розділі розглянуто особливості програмної реалізації та апробації розробленої інтелектуалізованої системи. Описано стек технологій, що

включає фреймворк Laravel та мови програмування PHP для реалізації системи рейтингування, мову програмування Python з бібліотеками PyTorch та Transformers для розробки модуля персоналізованих рекомендацій, а також Node.js, Express.js та MongoDB для модуля збору даних. Продемонстровано реалізацію ключових класів та контролерів, що відповідають за інтелектуалізований функціонал, зокрема за взаємодію з великими мовними моделями, формування звітів про оцінку відповідності НПП ліцензійним вимогам та генерацію рекомендацій.

Представлено результати апробації програмної системи на даних Західноукраїнського національного університету. Наведено наявний розподіл балів у системі та середні кількості виконуваних критеріїв працівниками, а також, графік розподілу рекомендованих активностей за типом, який вказують що працівникам рекомендується більше половини нових діяльностей. Ефективність розроблених рішень підтверджено кількісними показниками: впровадження інструментів автоматизації дозволило скоротити середній час, необхідний для заповнення звітів, на 20%. Аналіз результатів діяльності тестової групи показав, що персоналізовані рекомендації сприяли реальному зростанню їхніх річних показників. Проілюстровано функціональні можливості та графічний інтерфейс системи для різних ролей користувачів, наведено приклади та проаналізовано згенеровані системою текстові рекомендації, звіти про оцінку відповідності ліцензійним вимогам та приклади визначення відповідності публікацій до дисциплін, які викладає працівник.

Практичне значення отриманих результатів полягає у створенні комплексної програмної системи, яка не лише може об’єктивно аналізувати та рейтингувати діяльність НПП, але й автоматизує ключові процеси та, що найважливіше, генерує персоналізовані, аргументовані рекомендації для підвищення їхньої результативності, що позитивно впливає на загальний рейтинг працівників та університету відповідно.

Ключові слова: інтелектуалізована система рейтингування, персоналізовані рекомендації, результативність НПП, графові нейронні мережі, великі мовні

моделі, графова база даних, модель представлення знань, метрики подібності, архітектура програмного забезпечення.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. Пукас А. В., Сімак А. Ю., Метод формування прогнозованих показників активності науково-педагогічних працівників для інтелектуалізованої системи аналізу їх діяльності та рейтингування. Наукові праці Вінницького національного технічного університету. – 2025. Вип. 2. – С. 1–15. (1,5 д.а. / 0,8 д.а.; особистий внесок: здобувачем запропоновано та обґрунтовано метод формування прогнозованих показників діяльності НПП).
ISSN: 2307-5376. https://praci.vntu.edu.ua/index.php/praci/article/view/813.

2. Сімак А., Юшко А. Інтелектуальний модуль збору відкритих даних в системі рейтингування діяльності науково-педагогічного персоналу. Measuring and computing devices in technological processes. – 2025 (1). – С. 353–361. (0,9 д.а. / 0,5 д.а.; особистий внесок: реалізовано модуль збору відкритих даних для автоматизації заповнення звітів в інтелектуалізованій системі з використання великих мовних моделей).
ISSN: 2219-9365. DOI: https://doi.org/10.31891/2219-9365-2025-81-45.

3. Юшко А. В., Сімак А. Ю., Інформаційна інтелектуальна система аналізу наукової та науково-педагогічної діяльності академічного колективу, Оптико-електронні інформаційно-енергетичні технології. 2024. Вип. 47(1). С. 7–16. (0,9 д.а. / 0,3 д.а.; особистий внесок: реалізовано архітектуру програмного забезпечення для аналізу наукової та науково-педагогічної діяльності академічного персоналу).
ISSN: 1681-7893. DOI: https://doi.org/10.31649/1681-7893-2024-47-1-7-16.

4. Pukas A., Simak A., Yushko S., Nadvynychnyy S., Shandruk S. and Vatslavskyi O., Automated Reporting Module in the Academic Staff Performance Appraisal System, in Proc. 2024 14th Int. Conf. Adv. Comput. Inf. Technol. (ACIT),

Ceske Budejovice, Czech Republic, 2024, pp. 824–827, (0,4 д.а. / 0,1 д.а.; здобувачем запропоновано архітектуру модуля збору та метод аналізу даних за допомогою великих мовних моделей).
ISSN: 2770-5218. DOI: https://doi.org/10.1109/ACIT62333.2024.10712573 (Scopus).

5. Pukas A., Simak A., Goncharuk-Cholach T., Konoplitska O. and Zabchuk V., Intelligent Analyzing Module in the Academic Staff Performance Appraisal System // CEUR Workshop Proceedings. – 2023. – Vol. 3624. – P. 423–429. (0,6 д.а. / 0,1 д.а.; особистий внесок: здобувачем запропоновано метод інтелектуального аналізу даних з відкритих джерел для перевірки відповідності інформації в системі).
ISSN: 1613-0073. https://ceur-ws.org/Vol-3624/Short_2.pdf (Scopus).

6. Pukas A., Simak A., Shandruk S., Bilovus L., Stepanenko V. and Demianiuk A., “Features of Implementation the Academic Staff Performance Appraisal System,” in Proc. 2022 12th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Ruzomberok, Slovakia, 2022, pp. 568–571, (0,4 д.а. / 0,1 д.а.; особистий внесок: здобувачем описано особливості реалізації системи оцінювання діяльності академічного персоналу).
ISSN: 2770-5218. DOI: https://doi.org/10.1109/ACIT54803.2022.9913180 (Scopus).

Наукові праці, які засвідчують апробацію матеріалів дисертації:

7. Мельниченко В.В., Сімак А.Ю., Юшко А.В. Інтелектуалізована система для підтримки реалізації принципів адаптивності навчання // Матеріали зимової школи-семінару молодих вчених і студентів “Комп’ютерні інформаційні технології” (CIT’2024). – Тернопіль: ЗУНУ, 2024. – С. 85–87. (0,2 д.а. / 0,07 д.а.; особистий внесок: здобувачем запропоновано архітектуру системи, яка забезпечує адаптивність процесу навчання).
https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf#page=92.

8. Пукас А.В., Сімак А.Ю., Юшко А.В., Мельниченко В.В. Модуль збору даних для автоматичного заповнення звіту в інтелектуалізованій системі аналізу діяльності та рейтингування викладачів // Матеріали зимової школи-семінару молодих вчених і студентів “Комп’ютерні інформаційні технології” (CIT’2024). – Тернопіль: ЗУНУ, 2024. – С. 69–71. (0,2 д.а. / 0,1 д.а.; особистий внесок: здобувачем

запропоновано метод аналізу і структуризації даних з використанням великих мовних моделей).
https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf#page=76.

9. Гуска Б. Д., Сімак А. Ю. Інтелектуалізована система обробки слабкоформалізованої інформації // Комп'ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів CIT'2023 (Тернопіль, 30 листопада 2023 р.). – Відп. за вип. А. В. Пукас. – Тернопіль: ЗУНУ, 2023. – С. 91. (0,1 д.а. / 0,06 д.а.; особистий внесок: здобувачем запропоновано механізми обробки неструктурованої інформації в рамках інтелектуалізованої системи).
https://dspace.wunu.edu.ua/bitstream/316497/49967/1/Гуска%20Б.Д..pdf.

10. Pukas A., Simak O., Syrnyk L., Horal V., Shyjko and Papa O., Software Module for Data Correctness and Completeness Control in the Academic Staff Performance Appraisal System, in Proc. 2019 9th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Czech Republic, 2019, pp. 277–280, (0,4 д.а. / 0,1 д.а.; особистий внесок: здобувачем розроблено модуль контролю повноти та коректності даних у системі оцінювання результативності НПП).
DOI: https://doi.org/10.1109/ACITT.2019.8779999 (Scopus).

11. Пукас А. В., Сімак А. Ю., Вітенко В. В., Інформаційна система рейтингування діяльності науково-педагогічних працівників закладів вищої освіти // Комп'ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів CIT'2019 (м. Тернопіль, 29 листопада 2019 р.). – Відп. за вип. М. П. Дивак. – Тернопіль: ТНЕУ, 2019. – С. 32–33. (0,1 д.а. / 0,04 д.а.; особистий внесок: здобувачем запропоновано архітектуру системи рейтингування науково-педагогічних працівників з додатковим рівнем перевірки даних).
https://dspace.wunu.edu.ua/jspui/bitstream/316497/36059/1/Пукас%20А.В.%2c%20Сімак%20А.Ю.%2c%20Вітенко%20В.В..pdf.

12. Пукас А. В., Сімак А. Ю., Сирник О. Й., Модуль контролю повноти та коректності даних інтелектуалізованої системи аналізу діяльності та рейтингування викладачів // Комп'ютерні науки та інформаційні технології: матеріали семінару CSIT'2018 (м. Тернопіль, 2 червня 2018 р.). – Відп. за вип. М.

П. Дивак. – Тернопіль: ТНЕУ, 2018. – С. 61–62. (0,1 д.а. / 0,04 д.а.; особистий внесок: здобувачем запропоновано процедуру перевірки повноти та коректності даних при їх обробці в системі рейтингування). https://dspace.wunu.edu.ua/bitstream/316497/31978/1/Пукас_Сімак_Сирник.pdf.

ANOTATION

Simak A.Y. Intellectualized software system for academic staff performance analysis and rating. – Scientific work on the rights of the manuscript.

Thesis for the degree of Doctor of Philosophy in the specialty 121 – Software Engineering – West Ukrainian National University, Ternopil, 2025.

The research was carried out at the Department of Computer Science of the West Ukrainian National University of the Ministry of Education and Science of Ukraine.

The dissertation work is devoted to solving an urgent scientific and practical problem - the development of an intellectualized software system for analyzing and rating the activities of NPE, which will provide a comprehensive analysis, forecasting and formation of personalized recommendations for improving the effectiveness of scientific and pedagogical activities for NPE. In modern conditions of globalization and increased competition in the field of higher education, the positions of higher education institutions (HEIs) in national and international rankings directly depend on the overall performance of their academic staff. The effectiveness of the activities of each NPE is becoming a key factor affecting the quality of the educational process, scientific potential and international competitiveness of the university. This necessitates the introduction of effective tools for monitoring, analysis and, most importantly, stimulating the professional development of employees. Most of the existing assessment and rating systems, especially in Ukrainian HEIs, operate according to traditional principles, which are characterized by high labor intensity, significant administrative burden, subjectivity and limited analytical capabilities. Such systems mostly record past achievements, but do not provide effective tools to stimulate professional development, or the existing mechanisms, as a rule, are fragmentary and limited only to certain types of employee activity. There is a contradiction between the urgent need to increase the effectiveness of academic staff and the inconsistency of existing means with modern requirements of intellectualization and automation. That is why the dissertation proposes a comprehensive approach and its software implementation in the form of an intellectualized system for analyzing and rating academic staff, which shifts the emphasis from simple control to

providing personalized recommendations to improve the effectiveness of academic staff activities. The first section of the dissertation provides an analytical review of the problem of evaluating, rating and improving the effectiveness of academic staff activities. The theoretical foundations and evaluation criteria covering educational and methodological, scientific and research, organizational and educational and international activities are systematized. After analyzing existing approaches to rating, traditional systems based on manual data entry and intellectualized systems using automation and artificial intelligence technologies were distinguished. Methods and means of increasing efficiency were considered, in particular, various types of recommender systems were analyzed, from collaborative filtering to approaches based on deep learning, graph neural networks and large language models. It is shown that most existing systems, especially in Ukrainian HEIs, do not fully use the potential of intellectualization. It is the need to overcome these limitations by developing appropriate methods and effective architectural solutions that would integrate modern AI technologies to provide personalized recommendations that forms the scientific context and justifies the relevance of this study. In the final part of the section, based on the relevance of the problem of stimulating the professional development of employees and after analyzing the methods and tools for its solution, tasks are formed to achieve the goals of the dissertation research. In the second section of the dissertation, mathematical support is developed for building an intellectualized system and forming personalized activity recommendations for academic staff. A model of representing knowledge about academic staff activities is developed based on a knowledge-oriented approach using a graph model. This approach allows for effective structuring of heterogeneous and interconnected entities (employees, criteria, publications, projects), which is a key prerequisite for analysis, identification of hidden patterns and implementation of the procedure for generating personalized recommendations. It is also described how this model is used to collect contextual data (for example, examples of successful publications or projects) in order to expand and detail recommendations. A combined academic staff ranking method is proposed to increase the relevance of personalized recommendations, which is based on a set of similarity metrics for automated identification of employees with the best performance

indicators as close as possible. The method integrates the time-shift dynamics metric for time series analysis, cosine similarity for comparing activity relationships, and the modified Jaccard index to account for the substantive proximity of the work being performed. An intellectualized method for generating personalized recommendations for employee activities for the next reporting period has been developed, which is based on the architecture of heterogeneous graph neural networks (GGNN) with spatial and temporal attention mechanisms. To transform the obtained forecasting results into detailed text recommendations, the use of large language models is proposed, with dynamic refinement of the prompt structure, which ensured their justification, practical value and reduced time for their formation. The final part of the section presents a comparative analysis of the recommended criteria generated by the proposed method and existing analogues. The results showed that the proposed approach based on GGNN, which takes into account the activities of more successful colleagues and the semantics of the criteria, demonstrates higher accuracy in predicting the activities that academic staff actually perform in the next reporting period. Additionally, the system was tested on a test group of employees. Analysis of its results showed that participants interacted with approximately half of the proposed recommendations. It is important that up to 30% of the total number of recommended activities was completed with a predicted score or higher, which contributed to a real increase in their annual indicators by an average of 27% compared to previous results. The third section of the dissertation is devoted to a description of the designed architecture of an intellectualized software system. It is based on an approach that combines elements of modular and service-oriented architectures to ensure a high level of flexibility, scalability and independence of functional components. The section considers the structure of key components: the rating system, the personalized recommendations module, determining compliance with licensing requirements for educational activities and the data collection module. The design features of the main modules are also presented. Special attention is paid to the combined data storage system, where the MySQL relational database is used for the main functionality, and the Neo4j graph database is used to model complex relationships when forming recommendations. This approach allowed optimizing performance and effectively solving problems of

various types. The mechanisms of interaction between modules and external services are described, in particular through REST API, access tokens, and specialized data exchange scripts. This approach allowed optimizing the system performance. To visualize architectural solutions and processes, the section provides UML diagrams, such as use-case diagrams, sequence diagrams, and ER diagrams, which allow determining the main requirements for the system and illustrating the mechanisms of interaction between modules, the report life cycle, and the distribution of functionality between user roles. The fourth section discusses the features of the software implementation and testing of the developed intellectualized system. The technology stack is described, which includes the Laravel framework and the PHP programming language for implementing the rating system, the Python programming language with the PyTorch and Transformers libraries for developing the personalized recommendations module, as well as Node.js, Express.js, and MongoDB for the data collection module. The implementation of key classes and controllers responsible for the intellectualized functional, in particular for interacting with large language models, generating reports on compliance with license requirements, and generating recommendations, is demonstrated. The results of testing the software system on data from the Western Ukrainian National University are presented. The current distribution of points in the system and the average number of criteria performed by employees are presented, as well as a schedule of the distribution of recommended activities by type, which indicates that more than half of the new activities are recommended to employees. The effectiveness of the developed solutions is confirmed by quantitative indicators: the implementation of automation tools allowed to reduce the average time required to fill out reports by 20%. Analysis of the test group's performance results showed that personalized recommendations contributed to a real increase in their annual indicators. The functional capabilities and graphical interface of the system for different user roles are illustrated, examples are given and text recommendations generated by the system, reports on compliance with license requirements and examples of determining the compliance of publications with the disciplines taught by the employee are analyzed. The practical significance of the results obtained lies in the creation of a comprehensive software system that can not only objectively analyze and rate the

activities of the academic staff, but also automate key processes and, most importantly, generate personalized, reasoned recommendations to increase their effectiveness, which positively affects the overall rating of employees and the university, respectively.

Keywords: intellectualized rating system, personalized recommendations, academic staff effectiveness, graph neural networks, large language models, graph database, knowledge representation model, similarity metrics, software architecture.

LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

Scientific papers in which the main scientific results of the dissertation were published:

1. Pukas A. V., Simak A. Yu., Method of forming predicted indicators of the activity of academic staff for an intellectualized system of analysis of their activity and rating. Scientific works of Vinnytsia National Technical University. – 2025. Issue 2. – P. 1–15. (1.5 d.a. / 0.8 d.a.; personal contribution: the applicant proposed and substantiated a method of forming predicted indicators of the activity of academic staff).
https://praci.vntu.edu.ua/index.php/praci/article/view/813.

2. Simak A., Yushko A. Intellectual module for collecting open data in the system of rating the activities of academic staff. Measuring and computing devices in technological processes. – 2025 (1). – P. 353–361. (0.9 d.a. / 0.5 d.a.; personal contribution: an open data collection module was implemented to automate the filling of reports in an intellectualized system using large language models).
DOI: https://doi.org/10.31891/2219-9365-2025-81-45.

3. Yushko A. V., Simak A. Yu., Information intellectual system for analyzing scientific and scientific-pedagogical activity of academic staff, Optical-electronic information-energy technologies. 2024. Issue 47(1). pp. 7–16. (0.9 d.a. / 0.3 d.a.; personal contribution: implemented software architecture for analyzing scientific and scientific-pedagogical activity of academic staff).
DOI: https://doi.org/10.31649/1681-7893-2024-47-1-7-16.

4. Pukas A., Simak A., Yushko A., Nadvynychnyy S., Shandruk S. and Vatslavskyi O., “Automated Reporting Module in the Academic Staff Performance Appraisal System,” in Proc. 2024 14th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Ceske Budejovice, Czech Republic, 2024, pp. 824–827, (0.4 p.s. / 0.1 p.s.; the applicant proposed the architecture of the collection module and the method of data analysis using large language models).
DOI: https://doi.org/10.1109/ACIT62333.2024.10712573 (Scopus).

5. Andriy Pukas, Andriy Simak, Tetiana Goncharuk-Cholach, Olena Konoplitska and Vadym Zabchuk, Intelligent Analyzing Module in the Academic Staff Performance Appraisal System // CEUR Workshop Proceedings. – 2023. – Vol. 3624. – P. 423–429. (0.6 p.s. / 0.1 p.s.; personal contribution: the applicant proposed a method of intellectual analysis of data from open sources to check the compliance of information in the system).
https://ceur-ws.org/Vol-3624/Short_2.pdf (Scopus).

6. Pukas A., Simak A., Shandruk S., Bilovus L., Stepanenko V. and Demianiuk A., “Features of Implementation the Academic Staff Performance Appraisal System,” in Proc. 2022 12th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Ruzomberok, Slovakia, 2022, pp. 568–571, (0.4 p.s. / 0.1 p.s.; personal contribution: the applicant describes the features of the implementation of the assessment of the activities of academic staff).
DOI: https://doi.org/10.1109/ACIT54803.2022.9913180 (Scopus).

Scientific works certifying the approval of the dissertation materials:

7. Melnychenko V.V., Simak A.Yu., Yushko A.V. Intellectualized system to support the implementation of the principles of adaptive learning // Materials of the winter school-seminar of young scientists and students “Computer information technologies” (CIT’2024). – Ternopil: ZUNU, 2024. – P. 85–87. (0.2 d.a. / 0.07 d.a.; personal contribution: the applicant proposed a system architecture that ensures the adaptability of the learning process).
https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf#page=92.

8. Pukas A.V., Simak A.Yu., Yushko A.V., Melnychenko V.V. Data collection module for automatic report filling in the intellectualized system of activity analysis and

rating of teachers // Materials of the winter school-seminar of young scientists and students “Computer information technologies” (CIT’2024). – Ternopil: ZUNU, 2024. – P. 69–71. (0.2 d.a. / 0.1 d.a.; personal contribution: the applicant proposed a method of data analysis and structuring using large language models).
https://dspace.wunu.edu.ua/bitstream/316497/52868/1/CIT%272024_Last.pdf#page=76.

9. Huska B. D., Simak A. Yu. Intellectualized system for processing weakly formalized information // Computer information technologies: materials of the school-seminar of young scientists and students CIT'2023 (Ternopil, November 30, 2023). – Corresponding to the issue of A. V. Pukas. – Ternopil: ZUNU, 2023. – P. 91. (0.1 p.s. / 0.06 p.s.; personal contribution: the applicant proposed mechanisms for processing unstructured information within the framework of an intellectualized system).
https://dspace.wunu.edu.ua/bitstream/316497/49967/1/Гуска%20Б.Д..pdf.

10. Pukas A., Simak A., Syrnyk O., Horal L., Shyjko V. and Papa O., “Software Module for Data Correctness and Completeness Control in the Academic Staff Performance Appraisal System,” in Proc. 2019 9th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Czech Republic, 2019, pp. 277–280, (0.4 p.s. / 0.1 p.s.; personal contribution: the applicant developed a module for controlling the completeness and correctness of data in the performance evaluation system).
DOI: https://doi.org/10.1109/ACITT.2019.8779999 (Scopus).

11. Pukas A. V., Simak A. Yu., Vitenko V. V., Information system for rating the activities of scientific and pedagogical workers of higher education institutions // Computer information technologies: materials of the school-seminar of young scientists and students CIT’2019 (Ternopil, November 29, 2019). – Corresponding to the issue of M. P. Dyvak. – Ternopil: TNEU, 2019. – P. 32–33. (0.1 p.s. / 0.04 p.s.; personal contribution: the applicant proposed the architecture of the system for rating scientific and pedagogical workers with an additional level of data verification).
https://dspace.wunu.edu.ua/jspui/bitstream/316497/36059/1/Пукас%20А.В.%2с%20Сімак%20А.Ю.%2с%20Вітенко%20В.В..pdf.

12. Pukas A. V., Simak A. Yu., Syrnyk O. Y., Module for controlling the completeness and correctness of data of the intellectualized system of analysis of activity

and rating of teachers // Computer science and information technologies: materials of the seminar CSIT'2018 (Ternopil, June 2, 2018). – Corresponding to the issue of M. P. Dyvak. – Ternopil: TNEU, 2018. – P. 61–62. (0.1 p.s. / 0.04 p.s.; personal contribution: the applicant proposed a procedure for checking the completeness and correctness of data when processing them in the rating system).
https://dspace.wunu.edu.ua/bitstream/316497/31978/1/Пукас_Сімак_Сирник.pdf.

Зміст

АНОТАЦІЯ 2
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ 21
ВСТУП 22
РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ОЦІНЮВАННЯ, РЕЙТИНГУВАННЯ ТА ПІДВИЩЕННЯ РЕЗУЛЬТАТИВНОСТІ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ 30
1.1. Теоретичні основи та критерії оцінювання діяльності науково-педагогічних працівників 31
1.2. Аналіз існуючих засобів рейтингування діяльності НПП та їх основні особливості 37
1.3. Аналіз методів і засобів підвищення результативності діяльності НПП 49
1.4. Постановка задачі дослідження 59
Висновки до розділу 1 61
РОЗДІЛ 2 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ 63
2.1. Метод формування рейтингової оцінки діяльності науково-педагогічних працівників 64
2.2. Модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі 68
2.3 Комбінований метод ранжування НПП на основі комплексу метрик подібності 80
2.4 Інтелекталізований метод формування персоналізованих рекомендацій 91
Висновки до розділу 2 117
РОЗДІЛ 3 АРХІТЕКТУРА ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ 120
3.1. Загальна архітектура системи 121
3.2. Особливості проектування модуля збору відкритих даних 131
3.3. Особливості проектування системи зберігання даних 136
3.4. Особливості проектування модулів формування персоналізованих рекомендацій та підтримки оцінювання відповідності НПП ліцензійним вимогам 147

Висновки до розділу 3 154
РОЗДІЛ 4 РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ ПРОГРАМНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ 156
4.1. Особливості реалізації системи 156
4.2. Апробація системи для визначення відповідності НПП ліцензійним умовам провадження освітньої діяльності 163
4.3. Функціональні можливості та графічний інтерфес програмної системи.....168
4.4. Апробація та оцінка ефективності роботи програмної системи 177
Висновки до розділу 4 188
ВИСНОВКИ.....190
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 193
ДОДАТОК А ОСНОВНІ DLL БАЗИ ДАНИХ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ 210
ДОДАТОК Б ЛІСТИНГ КОДУ ОСНОВНИХ МОДУЛІВ ПРОГРАМИ 220
ДОДАТОК В ER-ДІАГРАМА БАЗИ ДАНИХ 237
ДОДАТОК Г СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ 238
ДОДАТОК Д АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДІСЛІДЖЕННЯ.....240
ДОДАТОК Е МЕТОДИ ПОКРАЩЕННЯ РЕЗУЛЬТАТИВНОСТІ ДІЯЛЬНОСТІ НПП ТА ПОРІВНЯННЯ ПІДХОДІВ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ ...244
ДОДАТОК Ж РЕКОМЕНДОВАНІ КРИТЕРІЇ, ЯКІ ФОРМУЄ LLM ТА РЕЗУЛЬТАТ ЇЇ РОБОТИ 248

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ І ПОЗНАЧЕНЬ

AI	Artificial Intelligence
API	Application Programming Interface
DL	Deep Learning
FK	Foreign Key
GNN	Graph Neural Networks
KPI	Key Performance Indicators
LLM	Large Language Model
ML	Machine Learning
PK	Primary Key
БД	База даних
ЗВО	Заклад вищої освіти
НПП	Науково-педагогічний працівник
ПЗ	Програмне забезечення
СУБД	Система управління базами даних
ШІ	Штучний інтелект
МН	Машине навчання

ВСТУП

Актуальність теми дослідження. У сучасному освітньому середовищі ефективність науково-педагогічної діяльності все частіше оцінюється за допомогою формальних показників, що характеризують різні види науково-педагогічної активності, такі як: публікаційну активність, грантове забезпечення, педагогічний внесок, організаційна та інші види діяльності. Ці показники не лише відображають поточний стан дослідницької та педагогічної продуктивності, а й використовуються для внутрішнього рейтингування, атестації, преміювання та планування професійного розвитку НПП. Проте аналіз цих показників часто обмежується простим сумуванням балів без врахування контексту, динаміки, перспектив розвитку та можливостей покращення результатів [13, 14].

Зважаючи на сучасні вимоги та виклики, які ставлять перед вищою освітою міжнародні рейтингові системи, такі як Scopus, QS World University Rankings, Times Higher Education (THE) та Academic Ranking of World Universities (ARWU), стає очевидною необхідність модернізації внутрішніх механізмів оцінювання та стимулювання НПП [15]. Існуючі в більшості українських ЗВО системи рейтингування переважно є традиційними та характеризуються низкою суттєвих недоліків. До основних проблем можна віднести високу трудомісткість процесів збору та обробки даних, їх суб’єктивність через залежність від людського фактора, обмежені аналітичні можливості та, що найважливіше, відсутність персоналізованого зворотного зв’язку та рекомендацій для індивідуального розвитку.

Наразі в Україні та інших країнах світу активно розробляються системи внутрішнього рейтингування НПП, які мають на меті оптимізацію управління людськими ресурсами, стимулювання наукової та педагогічної активності, а також підвищення прозорості та об’єктивності оцінювання. Проте переважна більшість таких систем базується на статичних, адміністративно заданих метриках, які не враховують особливостей окремих кафедр, специфіки наукових напрямів чи індивідуального розвитку НПП.

За даними досліджень Організації економічного співробітництва та розвитку (OECP), з поширенням доступу до інструментів ШІ, у провідних країнах світу все ширше впроваджуються інтелектуалізовані системи аналізу діяльності наукових працівників у вищій освіті, які ґрунтуються на методах машинного навчання, графових моделях даних, системи моніторингу та аналізу динаміки діяльності, використанні різних інструментів ШІ для виконання певних процесів та рекомендаційних механізмах. Такі системи дозволяють: автоматично аналізувати діяльність НПП; порівнювати їх з подібними працівниками; прогнозувати динаміку розвитку; формулювати персоналізовані рекомендації щодо покращення результатів; інтегрувати дані з зовнішніх джерел; забезпечувати налаштування параметрів оцінювання залежно від специфіки і орієнтації університету [16].

Особливу актуальність набуває інтеграція графових баз даних в програмних системах, що дозволяє ефективно моделювати складні взаємозв'язки між НПП, їх науково-педагогічною діяльністю, інтересами, проєктами, кафедрами, факультетами тощо. Це забезпечує масштабованість, гнучкість та хорошу точність аналізу порівняльних характеристик.

Таким чином, розробка інтелектуалізованої програмної системи аналізу та рейтингування діяльності НПП, яка забезпечить комплексний аналіз, прогнозування та формування персоналізованих рекомендацій щодо покращення результативності науково-педагогічної діяльності для НПП, є актуальним науково-практичним завданням, що поєднує проблеми управління знаннями, аналізу великих даних, машинного навчання та оптимізації людського капіталу в освітніх закладах. Така система матиме потужний прикладний потенціал для впровадження в українські ЗВО, сприятиме підвищенню якості освіти та наукового потенціалу, а також забезпечить ефективне здійснення наукової діяльності.

Мета і завдання дослідження. Метою роботи є інженерія методів та архітектурних рішень для інтелектуалізованої програмної системи аналізу та рейтингування діяльності НПП, що забезпечує підвищення результативності їх діяльності.

Для досягнення зазначеної мети встановлено такі завдання дослідження:

- провести аналіз відомих методів та засобів для підвищення результативності діяльності НПП, виявити їх основні переваги та недоліки;
- розробити модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі, з інтеграцією їх часових портретів, персоналізованих рекомендацій та зворотного зв'язку;
- розробити комбінований метод визначення працівників з максимально наближеними кращими показниками діяльності, використовуючи прості та зрозумілі метрики подібності з врахуванням часових рядів та семантики виконуваної діяльності;
- розробити інтелектуалізований метод формування персоналізованих рекомендацій діяльності НПП, який ґрунтується на прогнозуванні показників діяльності, моделі представлення знань, використанні великих мовних моделей для генерації текстових персоналізованих рекомендацій з динамічним уточненням структури промптів;
- виконати реінженерію системи рейтингування, що забезпечить інтелектуалізацію її процесів та генерацію персональних рекомендацій;
- провести апробацію розроблених методів та моделей, процесів генерації персональних рекомендацій для підвищення результативності діяльності НПП в рамках системи рейтингування Західноукраїнського національного університету (ЗУНУ).

Об'єкт дослідження. Процеси оцінювання, аналізу, рейтингування та покращення результативності діяльності НПП у ЗВО.

Предмет дослідження. Математичне та програмне забезпечення інтелектуалізованої системи аналізу та рейтингування діяльності НПП.

Методи дослідження. Для досягнення поставленої мети та реалізації завдань дослідження використано комплекс наукових методів: методи системного аналізу для комплексного опису предметної області, формалізації взаємозв'язків між елементами системи рейтингування та проєктування її загальної архітектури; графового моделювання для представлення часових траєкторій діяльності науково-педагогічних працівників у вигляді структурованих графових моделей, що

дозволяє ефективно аналізувати динамічні патерни активності; метричні методи, зокрема алгоритм динамічного часового зміщення, косинусна подібність, індекс Жаккара та інші для кількісної оцінки подібності між траєкторіями діяльності; методи машинного навчання, а саме на основі гетерогенних графових трансформерів з механізмами просторової та часової уваги, для прогнозування майбутньої результативності та рекомендованої діяльності; методи обробки природної мови, зокрема великі мовні моделі, для синтезу структурованих текстових рекомендацій, а також для автоматизації збору та аналізу даних; методи об'єктно-орієнтованого програмування та теорії алгоритмів для програмної реалізації модулів інтелектуалізованої системи, а також методи експериментального дослідження для апробації розробленої системи та оцінки практичної ефективності сформованих рекомендацій.

Наукова новизна отриманих результатів. У межах дисертаційної роботи *вперше*:

– розроблено модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі, що дало змогу реалізувати процедури генерації персоналізованих рекомендацій для підвищення результативності науково-педагогічної діяльності;

– запропоновано комбінований метод ранжування НПП на основі комплексу метрик подібності, який враховує динаміку їх активності, семантичну подібність та кількісні характеристики результатів діяльності, а також його реалізацію у вигляді програмної моделі, що уможливило автоматизоване визначення працівників з максимально наближеними кращими показниками діяльності в умовах розрідженості та обмеженості даних;

– запропоновано інтелектуалізований метод та його програмну модель формування персоналізованих рекомендацій діяльності НПП, який ґрунтується на прогнозуванні показників діяльності, моделі представлення знань, використанні великих мовних моделей для генерації текстових персоналізованих рекомендацій з динамічним уточненням структури промптів, що забезпечує скорочення часу їх формування.

набули подальшого розвитку:

– архітектури інтелектуалізованих систем на основі розширення механізмів взаємодії з розподіленими джерелами даних, інтеграції з великими мовними моделями, розробки знання-орієнтованого підходу представлення знань, формування персоналізованих рекомендацій із вбудованими механізмами зворотного зв'язку, що забезпечило підвищення ефективності формування персоналізованих рекомендацій.

Особистий внесок здобувача. Результати, викладені в цій дисертаційній роботі, отримані автором самостійно. У друкованих працях, які опубліковані у співавторстві, автору належать такі результати:

- [1] проведено аналіз існуючих підходів до формування показників діяльності та запропоновано й обґрунтовано метод формування прогнозованих показників діяльності НПП для інтелектуалізованої системи аналізу їх діяльності та рейтингування;
- [2] запропоновано модуль збору відкритих даних для автоматизації заповнення звітів в інтелектуалізованій системі з використанням великих мовних моделей;
- [3] реалізовано архітектуру програмного забезпечення для аналізу наукової та науково-педагогічної діяльності академічного персоналу;
- [5] запропоновано метод аналізу і структуризації даних з використанням великих мовних моделей;
- [6] вдосконалено архітектуру модуля збору та метод аналізу даних використовуючи великі мовні моделі;
- [7] описано механізми обробки неструктурованої інформації в рамках інтелектуалізованої системи;
- [8] описано метод інтелектуального аналізу даних з відкритих джерел для перевірки відповідності інформації в системі;
- [9] описано особливості реалізації системи оцінювання діяльності академічного персоналу;

- [10] запропоновано модуль контролю повноти та коректності даних у системі оцінювання діяльності НПП;
- [11] запропоновано архітектуру системи рейтингування науково-педагогічних працівників з додатковим рівнем перевірки даних;
- [12] описано процедуру перевірки повноти та коректності даних при їх обробці в системі рейтингування.

Основні положення та результати дисертаційної роботи в наведених працях викладені в повному обсязі.

Апробація результатів дисертації. Основні положення і результати дисертаційної роботи презентовано на 7 конференціях, зокрема:

- 9th International Conference on “Advanced Computer Information Technologies” ACIT-2019 – Ceske Budejovice, Czech Republic, 5-7 Jun., 2019;
- 12th International Conference on Advanced Computer Information Technologies, ACIT-2022, Spisska Kapitula, Slovakia, 26-28 Sept., 2022;
- X International Conference “Information Technology and Implementation”, IT&I, Kyiv, Ukraine, 20-21 Nov., 2023;
- 14th International Conference on Advanced Computer Information Technologies, ACIT-2024, Ceske Budejovice, Czech Republic, 19-21 Sept., 2024;
- 15th International Conference on Advanced Computer Information Technologies, ACIT-2025, Šibenik, CROATIA 17-19 Sept., 2025;
- Наукові семінари кафедри комп’ютерний наук ЗУНУ, 2019-2024 рр.

Публікації. За результатами дисертаційного дослідження опубліковано 12 наукових праць (ДОДАТОК Г) загальним обсягом 64 сторінки, зокрема 6 статей у періодичних фахових виданнях [1, 2, 3, 4, 5, 6], 3 з яких статті у періодичних наукових виданнях України категорії “Б” [1, 2, 3] та 3 у виданнях, що індексуються наукометричною базою Scopus [4, 5, 6], а також 6 публікацій у матеріалах конференцій [7, 8, 9, 10, 11, 12], 1 з яких індексується міжнародною наукометричною базою Scopus [10].

Структура та обсяг роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 133 найменувань та 7

додатків. Загальний обсяг роботи складає 252 сторінки друкарського тексту, з них 171 сторінок основного тексту. Робота містить 72 рисунки і 6 таблиць.

Зв'язок роботи з науковими програмами, планами, темами.

Дисертаційна робота виконувалася в рамках пріоритетного напряму «Інформаційні та комунікаційні технології» відповідно до закону України «Про пріоритетні напрями розвитку науки і техніки» та пов'язана з науковими дослідженнями, які виконувалися за тематикою кафедри комп'ютерних наук Західноукраїнського національного університету.

У процесі виконання дисертаційного дослідження використано деякі результати молодіжного науково-дослідного проекту на тему «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (державний реєстраційний номер 0124U000076).

Практичне значення отриманих результатів полягає в створенні інтелектуалізованої програмної системи для формування персоналізованих рекомендацій діяльності НПП щодо покращення їх результативності. Архітектура програмного забезпечення складається з модулів, які використовуються для акумуляції даних, формування моделі представлення знань про діяльноість НПП, виконання обчислень, автоматизацію частини процесів та візуалізацію отриманих результатів.

Теоритичні та прикладні результати дисертаційної роботи використано:

- у Західноукраїнському національному університеті для удосконалення системи рейтингування ЗУНУ, формування персоналізованих рекомендацій для покращення результативності діяльності НПП (акт про впровадження результатів дисертаційної роботи від 1 вересня 2025 р.);
- в українському представництві IT компанії "WorldERP, LLC" (м. Тернопіль) в процесі реалізації програмно-технічних комплексів, які включають інтеграцію програмно-реалізованих засобів штучного інтелекту (акт про впровадження результатів дисертаційної роботи від 12 серпня 2025 р.);
- в НДЧ ЗУНУ при виконанні молодіжного науково-дослідного проєкту на тему «Математичне та програмне забезпечення прототипу біогазової установки з

підвищеною ефективністю функціонування» для розробки архітектури програмного забезпечення (державний реєстраційний номер 0124U000076) (довідка від 20 серпня 2025 р.);

- в освітньому процесі Західноукраїнського національного університету на кафедрі комп'ютерних наук при викладанні дисциплін: "Інтелектуальні інформаційні системи", "Інтелектаульні інформаційні системи", "Основи інформаційних технологій" (акт про впровадження результатів дисертаційної роботи від 1 вересня 2025 р.).

В

ДОДАТОК Д

АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДІСЛІДЖЕННЯдо дисертації подано акти впровадження результатів дисертаційного дослідження.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ОЦІНЮВАННЯ, РЕЙТИНГУВАННЯ ТА ПІДВИЩЕННЯ РЕЗУЛЬТАТИВНОСТІ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

Кожен заклад вищої освіти, в умовах динамічних змін у системі вищої освіти, прагне бути лідером, демонструвати високу результативність та прогресивний розвиток, що забезпечується науково-педагогічними працівниками (НПП), які є невід'ємною складовою забезпечення функціонування закладу. Діяльність академічних працівників охоплює широке коло напрямів [13], що безпосередньо впливає на якість підготовки фахівців, розвиток науки та, як наслідок, на міжнародну конкурентоспроможність університету [14]. Тому перед ЗВО виникає потреба відстежувати результативність роботи працівників, виявляти слабкі види активності, вмотивовувати до професійного розвитку та підвищення загальної ефективність праці [15], та впроваждувати певні специфічні інструменти для оцінювання і рейтингування діяльності НПП.

Різні міжнародні рейтинги, такі як рейтинг від Scopus, QS World University Rankings, Times Higher Education World University Rankings (THE) або Academic Ranking of World Universities (ARWU) та інші, вимагають модернізовувати внутрішні механізми оцінки та стимулювання працівників. Вони мають власні вимоги до яких потрібно адаптуватись кожному закладу освіти [17, 18, 19]. Адже, якщо розглядати діяльність ЗВО та її позицію на глобальній арені, саме індивідуальні та колективні досягнення науково-педагогічного складу мають суттєвий вплив на це [17]. Також стрімка цифрова трансформація освітнього простору вимагає формування у випускників університетів нових та актуальних навичок, які необхідні в сучасних професіях, а також постійне оновлення знань та технологій, що зумовлює пошук інноваційних підходів до підвищення результативності НПП, які виходять за межі традиційних, часто формалізованих, методів.

Цей розділ присвячений аналізу теоретичних основ, критеріїв оцінювання та існуючих підходів до інтелектуалізованих систем рейтингування НПП для підвищення їх результативності та позицій ЗВО в рейтингах.

Крім того, у розділі наведено існуючі методи та засоби підвищення результативності діяльності НПП, розглянуто традиційні та іноваційні підходи, які були порівняні між собою, включно з рекомендаційними системами, які ґрунтуються на використанні технологій ШІ. Після аналізу переваг і недоліків наявних підходів та врахування існуючого програмного забезпечення для рейтингування, було обґрунтовано необхідність розробки інтелектуалізованої рейтингової системи, яка б не лише оцінювала, а й активно сприяла зростанню показників кожного працівника через персоналізовані рекомендації.

Проаналізувавши предметну область дослідження, виконано постановку задачі дослідження та описано основні її завдання.

1.1. Теоретичні основи та критерії оцінювання діяльності науково-педагогічних працівників

Необхідність оцінювання діяльності НПП зумовлена потребою визначати рівень відповідності професійної діяльності працівника до встановлених вимог, стандартам та інституційним цілям, щоб здійснювати управління якістю освітніх послуг, що надаються у ЗВО. Тому вводяться відповідні системи для виконання таких завдань:

– накопичення аналітичної інформації [22] щодо кадрового розвитку, розподілу ресурсів та стратегічного планування;

– сприяння професійному та кар'єрному зростанню працівників, а також надання керівництва тим, хто вносить значний внесок у розвиток університету [23];

– підвищення результативність професійної активності НПП [20, 21].

Виходячи з цього, можна виділити основні принципи та підходи до ефективного оцінювання діяльності академічних працівників, які є міжнародно

визнаними й актуальними на сьогодні стандартами [24] та ґрунтуються на певних ключових принципах:

– інформація повинна бути перевіреною, достовірною та змістовною, підтвердженою копіями матеріалів в певному вигляді (електронному чи паперовому);

– об’єктивність оцінок гарантується цілісністю та повнотою даних. Для цього доступ до них має бути відкритим, а критерії — чіткими, конкретизованими та містити зрозумілий опис очікуваних результатів;

– систематичність впливає на постійний розвиток, тому оцінювання слід виконувати періодично.

Дослідники, такі як С. Мельник [20] та J. Smith [21], наголошують на необхідності врахування міжнародного досвіду, стандартизації критеріїв оцінювання та автоматизації процесів рейтингування НПП, що є невід’ємною частиною систем управління якістю освіти. Такі системи спрямовані на виконання встановлених стандартів професійної діяльності, де основним завданням є стимулювання професійного розвитку та покращення загальної результативності.

Поняття “результативності НПП”, зазвичай використовують при описі успішності виконання функцій викладача чи науковця, що охоплює широкий спектр досягнень. І зазвичай, сприймається як рівень досягнень у певних видах академічної діяльності, тобто здатність реалізовувати навчальні програми на високому рівні, здійснювати інноваційні наукові дослідження, активно залучатися до міжнародної співпраці. Як стверджують Zhao та співавт. [25], модель збалансованої системи показників та нечітка оцінка забезпечує достовірність, валідність, об’єктивність і систематичність через чітко визначені KPI. Reheilo [26] підкреслює роль наукометричних показників (WS/Scopus, h-індекс, гранти), які інтегруються в ключові показники для комплексного оцінювання. Hussain et al. [27], проаналізувавши тенденції за останні роки, визначив, що вони демонструють тенденцію до впровадження сучасних інтелектуалізованих систем, які включають AI-аналітику, дашборди з реальним часом та гібридні моделі, які поєднують кількісні KPI з якісним фідбеком. Шимон та ін. [28] підкреслюють, що

результативність діяльності НПП слід розглядати як інтегральний показник, що відображає якість їхньої роботи та вплив на розвиток університету. Він охоплює науково-дослідну активність, освітню роботу, методичні надбання, а також організаційний внесок, що є єдиним комплексним показником. Проте, слід зазначити, що трактування цього поняття може трохи відрізнятися залежно від контексту та критеріїв оцінювання.

Часто використовують ключові показники ефективності KPI [11, 29], для кількісних та якісних значень, що охоплюють різні аспекти діяльності НПП, серед яких можна виділити кілька основних, рисунок 1.1

Рис. 1.1. Основні аспекти діяльності НПП в ЗВО

Навчально-методична діяльність враховує не тільки саме педагогічне навантаження, а й якість розроблених навчальних матеріалів (програм, посібників, електронних курсів), використання інноваційних методик викладання, а також результати студентських опитувань щодо якості викладання.

Науково-дослідна діяльність в основному вимірюється кількістю та якістю наукових публікацій (статті у журналах, індексованих у наукометричних базах Scopus, Web of Science, Google Scholar, монографії, розділи монографій), цитованістю праць (наприклад h-індекси), участю у наукових конференціях (доповіді, тези), отриманням грантів та патентів на винаходи, керівництво науковою роботою студентів та аспірантів тощо.

Організаційно-виховна діяльність включає участь у роботі колегіальних органів університету (вчені ради, комісії), кураторство студентських груп, участь у

профорієнтаційній роботі, організацію наукових заходів, а також внесок у розвиток інфраструктури ЗВО.

Міжнародна діяльність відображає міжнародну академічну мобільність (стажування, викладання за кордоном), участь у міжнародних проєктах, публікації у міжнародних виданнях, залучення іноземних студентів та партнерство з закордонними ЗВО.

Щодо прикладів українських систем, можна назвати такі які включають критерії оцінювання та рейтингові положення Криворізького державного педагогічного університету, Дніпровський національний університет ім. академіка В. Лазаряна та Київський національний університет технологій та дизайну. В таких системах спостерігається прагнення до відповідності міжнародним стандартам наукової продуктивності, тому присутня інтеграція з різними міжнародними показниками, такими як індексація публікацій у Scopus та Web of Science, підготовка кадрів вищої кваліфікації тощо [30-32, 34, 35].

Своєю чергою, міжнародні системи оцінювання, на відміну від українських, часто мають ширший спектр, охоплюючи не лише безпосередню діяльність академічних працівників, а й інституційні показники, що відображають загальну ефективність та стратегічні цілі ЗВО, та можуть включати вищенаведені показники, та додаткові критерії [33], такі як: обсяг доходів та грантів; успішність, програм обміну, набір та утримання студентів; навантаження НПП; інфраструктура та ресурси закладу.

Певною мірою, міжнародні системи трохи ширше розглядають питання результативності НПП, тобто вони сприймають індивідуальні показники не як щось ізольоване, а як базову частину успіху університету в цілому.

Крім того, системи які використовують KPI, українські чи міжнародні, можуть забезпечувати автоматичний збір даних у звітний період (наприклад, з наукометричних баз даних, таких як: Scopus, Web of Science, Google Scholar) для автоматизації процесів, регулярність оцінювання та формування звітів, що забезпечить їх конкретність, наповненість, цілісність та актуальність для використання в ЗВО [2, 11, 29, 43, 73, 34, 35].

Варто сказати, що критерії оцінювання діяльності НПП є основою для формування KPI та забезпечують системний підхід до аналізу їхньої роботи. Ці критерії, як правило, охоплюють основні аспекти діяльності НПП [11, 29].

Слід зазначити, що між індивідуальною результативністю НПП та позиціями ЗВО у міжнародних і національних рейтингах існує безпосередній системний зв'язок: як показує Ali M. у своїй роботі [36], KPI-системи забезпечують покращення показників університету у грошових, репутаційних та наукометричних рейтингах; дослідження 2024 року підтверджує збіг внутрішніх KPI-метрик із критеріями університетських рейтингів, таких як THE, QS, ARWU, рейтинг Scopus, Digital Bibliography & Library Project (DBLP), "Топ-200 Україна" та інші, що прямо впливає на позицію в цих рейтингах [37]. Ці рейтинги оцінюють діяльність ЗВО в цілому, але вони не є ідеальними, кожен з них має різні критерії оцінювання та свої сильні та слабкі сторони, їхні методології опосередковано формують міжнародні стандарти оцінювання діяльності НПП, наприклад [19]:

– академічна репутація ЗВО (вагомий компонент у QS та THE) залежить від персонального наукового авторитету НПП, участь у міжнародних конференціях, членства у наукових асоціаціях, репутація роботодавців, у яких працевлаштовуються випускники [36];

– показник цитованості наукових публікацій виступає важливим критерієм у більшості світових рейтингах ЗВО (QS, THE, DBLP, ARWU);

– кількісне співвідношення студентів до викладачів (QS, THE), якість викладання та можливість індивідуально підходити до навчання студентів [36, 37];

– дохід від досліджень (THE), залучення грантового фінансування та комерціалізації наукових досліджень чи розробок [36];

– міжнародна інтеграція, частка іноземних студентів та викладачів, міжнародні публікації та співпраця [36, 37].

Відповідно, якщо працівники будуть покращувати свою результативність, це призведе до підвищення загального рівня, результативності ЗВО та його конкурентоспроможності на міжнародній арені, тому заклади продовжують активно розробляти, покращувати та використовувати різні програмні системи

оцінювання діяльності та рейтингування НПП. Вони розробляють власні положення щодо цього, які можуть відрізнятися від міжнародних стандартів оцінювання НПП або навпаки доповнювати їх, додаючи певні спеціальні критерії оцінювання. Наприклад, можна навести вітчизняні заклади: Київський політехнічний інститут імені Ігоря Сікорського (КПІ) [34], Західноукраїнський національний університет (ЗУНУ) [39, 40, 67], Чернігівський національний університет (НУЧП) [41], Харківський національний аерокосмічний університет (ХАІ) [62], Одеський національний морський університет [81] та інші. Якщо порівнювати їх визначені положення, то вони відрізняються кількістю критеріїв, групами активності які охоплює ЗВО, інтерпретацією опису критеріїв оцінювання, спеціальними видами діяльності, що враховуються певними університетами, та формою оцінювання критеріїв, хоча всі вони охоплюють суттєву частину активностей НПП, яка впливає на результативність ЗВО.

Системи оцінювання та рейтингування НПП, дозволяють проводити моніторинг, оцінку, порівняння діяльності, і відповідно, покращення їх показників безпосередньо впливають на позиції ЗВО у міжнародних рейтингах (рисунок 1.2).

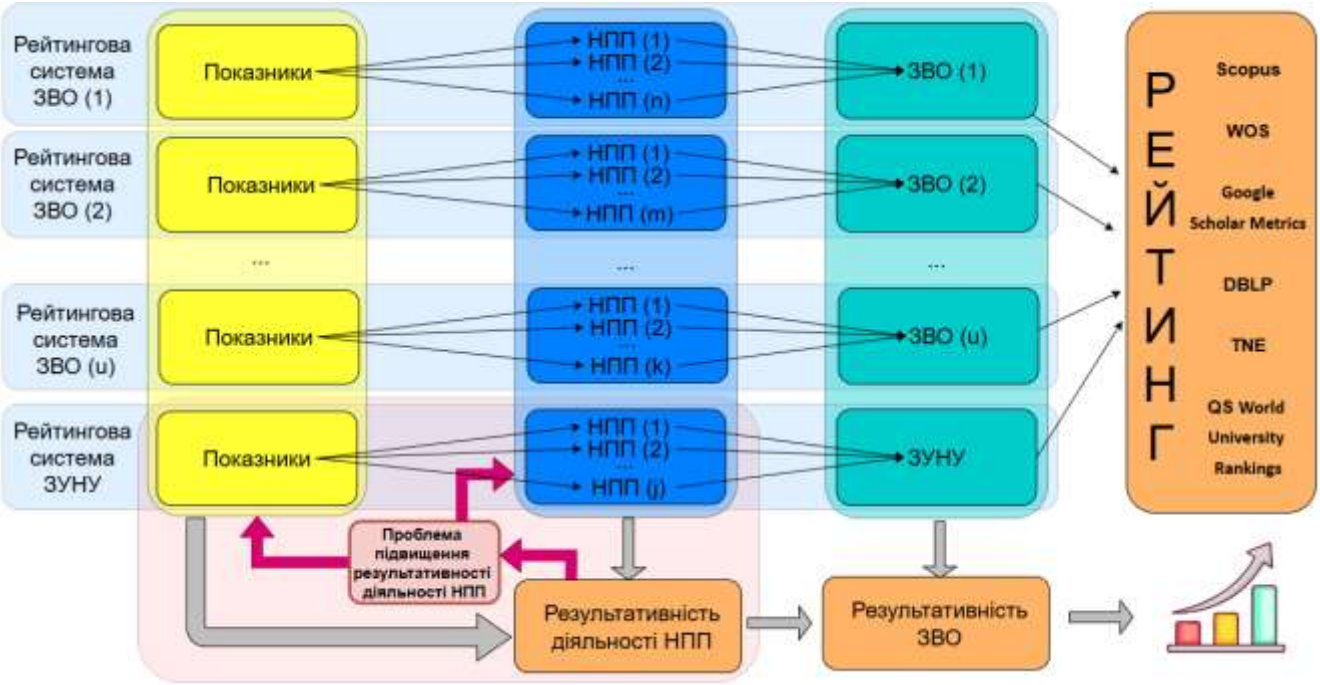


Рис. 1.2. Схема залежності загального рейтингу ЗВО від результативності діяльності його складових елементів

Це означає, що індивідуальна ефективність працівників є не просто внутрішнім кадровим питанням, а більш глобальним явищем, яке впливає на результативність та позиції університету в порівнянні з іншими [37, 42]. ЗВО вимушені узгоджувати свої внутрішні критерії оцінювання з міжнародними, що поріджує зворотний зв'язок, де висока результативність НПП призводить до вищих рейтингів університету, що, у свою чергу, збільшує привабливість для талановитих викладачів та студентів, сприяючи подальшому підвищенню результативності НПП. Тому можна сформувати проблематику дисертаційного дослідження, а саме в необхідності покращення результативності діяльності НПП (див. рисунку 1.2).

1.2. Аналіз існуючих засобів рейтингування діяльності НПП та їх основні особливості

Заклади освіти використовують власні рейтингові системи для відстеження та стимулювання до підвищення результативності діяльності працівників. Крім того, існують різні положення і закони, які вказують на необхідність виконувати оцінювання та рейтингування діяльності НПП у ЗВО, наприклад, закон України про вищу освіту, положення про рейтингування академічних працівників, міжнародні стандарти систем управління якістю освіти (ISO) або Європейська асоціація із забезпечення якості освіти (ESG) [39, 44, 45]. Як зазначалось раніше, результативність ЗВО є прямим наслідком сукупної діяльності його академічного персоналу. Кожен окремий внесок, чи то публікація наукової статті, чи успішно реалізований освітній проект, чи високі показники задоволеності студентів, у сукупності формує загальний рейтинг університету.

Більшість українських систем рейтингування НПП в ЗВО мають ієрархічну структуру, за якою рейтинги кожної структури складаються з рейтингу його елементів та основою виступає індивідуальний рейтинг кожного НПП [11, 12, 51, 53], візуалізація структури зображена на рисунку 1.3. Деякі системи можуть надавати ширший формат аналізу рейтингів факультетів та кафедр і їх результативності [9].

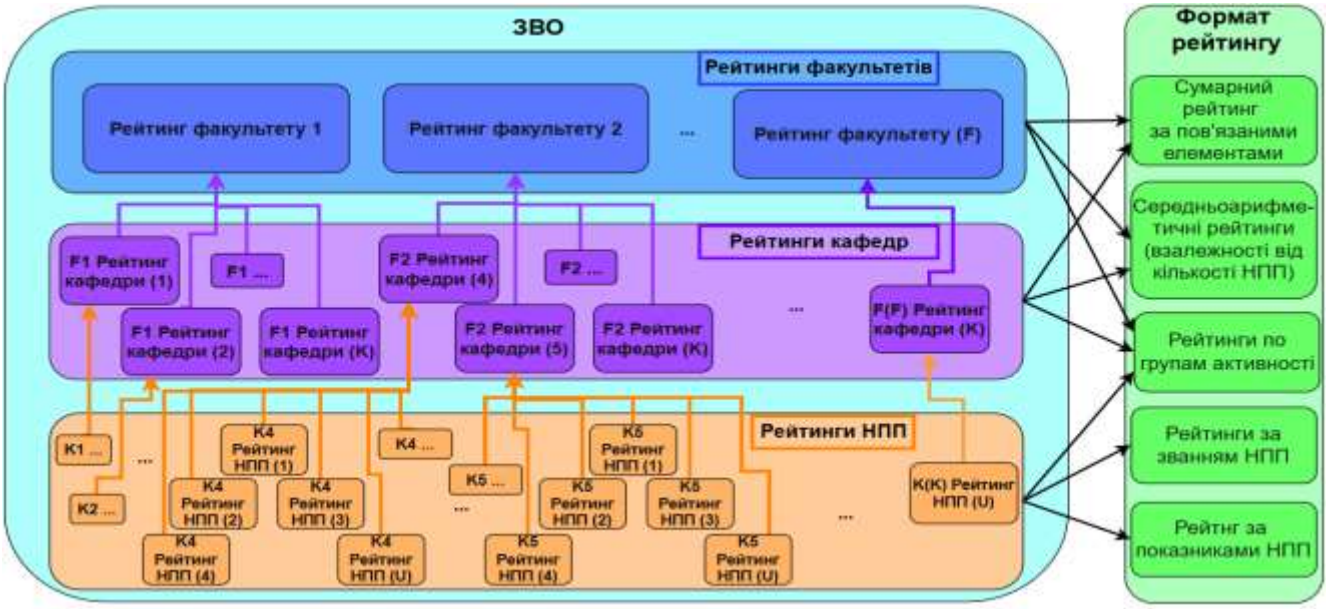


Рис. 1.3. Узагальнена ієрархічна структура систем рейтингування НПП

Дані, отримані в результаті рейтингування, мають широке застосування, наприклад, для ефективного розподілу ресурсів, прийняття обґрунтованих управлінських рішень, формування кадрової політики та ефективного стимулювання НПП [42]. Еволюція систем оцінювання також вказує на зростаючу інституційну та національну увагу до якості освіти та глобальної конкурентоспроможності, тому важливо проаналізувати динаміку цього процесу та визначити вектори його подальшого вдосконалення.

Досліджуючи праці, присвячені рейтинговим системам оцінювання НПП, можна виокремити дві умовні категорії підходів: традиційні системи, що базуються на ручному введенні даних та простій бальній оцінці. Такі процеси, досить трудомісткі та схильні до людських помилок. На противагу їм, інтелектуалізовані системи використовують автоматизацію, машинне навчання, інструменти ШІ, автоматичний збір метаданих для наповнення системи та адаптивний аналіз.

З одного боку, традиційні системи (наприклад, описані Дробахіним, Гуком і Ткаченком [46]), наштовхуються на очевидні обмеження: висока трудомісткість, суб’єктивність оцінювання, відсутність стандартизації заповнення звітів, неможливість своєчасного аналітичного супроводу та глибокого аналізу результативності працівників, враховуючи стрімкий розвиток технологій, такі

системи демонструють неефективність у сучасних умовах і спонукають до пошуку нових рішень. Недоліки традиційних систем підтверджуються спостереженнями Зеленського, Стельмащука та Луціва [47], які зазначали, що вони не стимулюють освітні інновації. Зарубіжний дослідник R. Johnson [55] у своїй праці, розглядає використання різних веб-платформ для аналізу та зберігання даних про діяльність НПП та впровадження рейтингових систем в різних відомих університетах світу, та акцентує увагу на тому, що зберігається необхідність відповідати сучасним вимогам, тому це вимагає систематичного внесення змін в наявні критерії оцінювання. Зазвичай оцінювання та рейтингування в таких системах відбувається за результатами звітного року та на основі загальних і спецефічних, для кожного із ЗВО, критеріїв оцінювання діяльності.

Поява інтелектуалізованих систем демонструє наступний етап розвитку в сфері рейтингування, які використовують технології ШІ, машинного навчання (МН) та аналітику великих даних для автоматизації та оптимізації різних процесів. Наприклад у роботі Cindrakasih R. [48] розглядається стратегія впровадження інтелектуалізованої системи з елементами ШІ для для оцінки ефективності працівників, що дозволить автоматизувати збір даних, аналізувати їх шляхом застосування машинного навчання і формувати рекомендації для персонального розвитку. Такі системи можуть підвищити об’єктивність, ефективність та глибину аналізу, що дозволяє перейти від простої оцінки минулої діяльності до активного заохочення постійного професійного зростання та розвитку, узгодження з ширшими інституційними стратегічними цілями, та надавання дієвих зворотних зв’язків, а не лише підсумкових балів.

Якщо розглядати переваги впровадження традиційних систем рейтингування, насамперед зазначимо, що традиційні методи оцінювання сприяють розвитку здорової конкуренції серед працівників, покращенню якості викладання та ефективності освітнього процесу [46, 47, 49], мотивації до продуктивної роботи та аналіз прийнятих рішень і керування ресурсами. Хоча й присутні певні проблеми, при використанні таких систем:

– трудомісткість та ресурсомність процесів, які спричиняють затримки та підвищене адміністративне навантаження [46, 47];

– оцінювання може бути суб’єктивним та упередженим [50];

– відсутні або обмежені інструменти для якісного відстеження довготермінових тенденцій у діяльності працівника, що може викликати ризик невідповідності НПП ліцензійним вимогам;

– перевага може надаватись кількісним показникам на противагу якісним, а також тим видам діяльності, що є легшими або швидшими в реалізації (залежно від критеріїв оцінювання), а також тим видам діяльності, що є легшими або швидшими в реалізації;

– відсутність механізму надання персоналізованих рекомендацій чи зворотного зв’язку.

Наприклад дослідження Sadiq, B. J. про застосування показників діяльності НПП у Багдадському університеті [63], виявило, що понад 90% НПП бути перевантажені навчальними годинами, та загальне співвідношення студентів до викладачів становило 27.51%, що є перевищеним показником за світовими стандартами. Відсоток отримання постійних посад був доволі низьким, персонал не брав активну участь у тренінгах (особливо в закордонних), практична відсутність активності в отриманні грантів на дослідження та нагород. Хоча публікаційна активність була достатньо високою, загальна незадоволеність персоналом залишилась низькою. Цей приклад свідчить про проблеми управління та необхідність інвестицій в академічний персонал ЗВО, а також про надмірну зосередженість на традиційних академічних метриках.

Закордонні підходи до рейтингування НПП більш різноманітні, і часто інтегровані в системи управління персоналом ЗВО, та мають такі особливості: підхід побудови профілю/портфоліо НПП для представлення досягнень та діяльності працівника; зворотний зв’язок не лише від керівництва, а й від колег та студентів, що ширше демонструє рівень наукової діяльності; періодичні атестації та співбесіди, зустрічі керівництва з НПП, планування роботи, постановка цілей.

Українські системи рейтингування в переважно є традиційними, але вони активно розвиваються і перетворюються на гібридні або частково інтелектуалізовані. Для прикладу, можна назвати Інформаційну систему “Електронний кампус” КПІ [51], яка виконує важливу роль у цифровій підтримці внутрішніх процесів університету, зокрема в оцінюванні роботи та рейтингуванні НПП, яке виконується відповідно до визначеного положення на рівні університету, яке регулярно оновлюється [38]. Через інтегрований модуль рейтингування працівники щороку подають звіти про результати своєї діяльності. Ця система тісно інтегрована з іншими модулями “Електронного кампусу”, що дозволяє автоматизувати підбір звітних документів, спрощує введення показників та генерує рейтингові списки користувачів. Система автоматизує облік досягнень у кількох напрямах: навчальному, науковому та організаційно-виховному. Кожен викладач самостійно вводить дані у свій електронний кабінет, після чого ці показники перевіряються кафедрою й затверджуються відповідними підрозділами. Формування підсумкового рейтингу базується на встановленій бальній моделі, які формують загальні рейтингові списки. Зібрані в системі дані дозволяють аналізувати ефективність роботи як окремих співробітників, так і цілих підрозділів. Інтерфейс платформи виглядає зручним та надає певні функціональні можливості: можна відстежувати статистику, переглядати отримані бали. Завдяки такій цифровій структурі КПІ вдалося створити прозорий механізм оцінювання, що поєднує автоматизацію, аналітику та контроль достовірності даних. Аналітичні звіти, які формуються на основі рейтингових результатів, використовуються для ухвалення управлінських рішень, визначення кандидатів на преміювання, формування кадрового резерву та оцінювання ефективності роботи кафедр. Процес рейтингування, в цій системі, не лише автоматизує рутинну звітність, а й створює передумови для побудови репутаційної системи з чітким розподілом ролей і функцій учасників процесу, що підвищує мотивацію до покращення власних досягнень. Щодо додаткових прикладів закладів які мають рейтингові системи, можна назвати: Донецький національний університет [46], Київський університет ім. Бориса Грінцевича, та їх система “Рейтинг прозорості” [53], Дніпровський

національний університет імені Олеся Гончара [54], Чернівецький національний університет, “Електрона система звітувань та оцінювання”, ЗУНУ “Рейтинг ЗУНУ” [11, 67]. Усі вони мають подібний функціонал: відслідковування певних аспектів наукової діяльності, побудова та заповнення звітів і формуванні рейтингів [56, 59].

Щодо іноземних вузів, які часто використовують не власні системи, а готові програмні рішення, можна назвати Каліфорнійський університет у Лос-Анджелесі (UCLA), Університет штату Каліфорнія в Лос-Анджелесі (Cal State LA), які використовують міжнародну систему оцінювання академічних працівників, яку у США часто називають “Retention, Tenure and Promotion (RTP) evaluations or reviews” [52]. Ці перевірки проводяться регулярно щороку, або рідше якщо працівник працює за контрактом, для різних категорій академічних працівників, включаючи лекторів, асистентів, доцентів та професорів. Оцінювання базується на звіті, яке розглядають комітети викладачів, керівники кафедр та адміністратори. Для управління цими процесами часто використовуються онлайн-платформи, такі як Interfolio. Ця платформа збирає і акумулює дані про діяльність НПП, формує профілі працівників по різним групам активності, допомагає іншим університетам дотримуватись внутрішніх політик та зовнішніх вимог.

В Оксфордському університеті передбачені щорічні атестації (Appraisals), які надають можливість для відкритого обговорення ролі, питань ефективності та узгодження “Плану особистого розвитку” [57]. Для академічно персоналу система передбачає самооцінку та формальне обговорення з призначеним оцінювачем щонайменше раз на три роки. Важливо, що в Оксфорді атестація не пов’язана безпосередньо з підвищенням, преміями чи дисциплінарними заходами, що підкреслює її фокус на розвитку НПП.

Технічний університет Мюнхена (TUM) також використовує систему рейтингування, найму та кар’єрного росту, яка є унікальною для Німеччини та базується на міжнародних стандартах, та пропонує прозорі критерії оцінювання ефективності, які враховують міжнародні стардарти [58]. Крім того, програма Faculty@TUM пропонує спеціалізовані програми розвитку, семінари та профілі

лідерства для професорів, спрямовані на вдосконалення їхніх лідерських та управлінських навичок.

На сучасному етапі активніше розвиваються інтелектуалізовані системи, в тому числі рейтингові системи для ЗВО [59]. Що дозволяє, в різноманітний спосіб, покращувати результативність НПП. Ілюстрація використання інстурментів ШІ в рейтингових системах, зображено на рисунку 1.4.

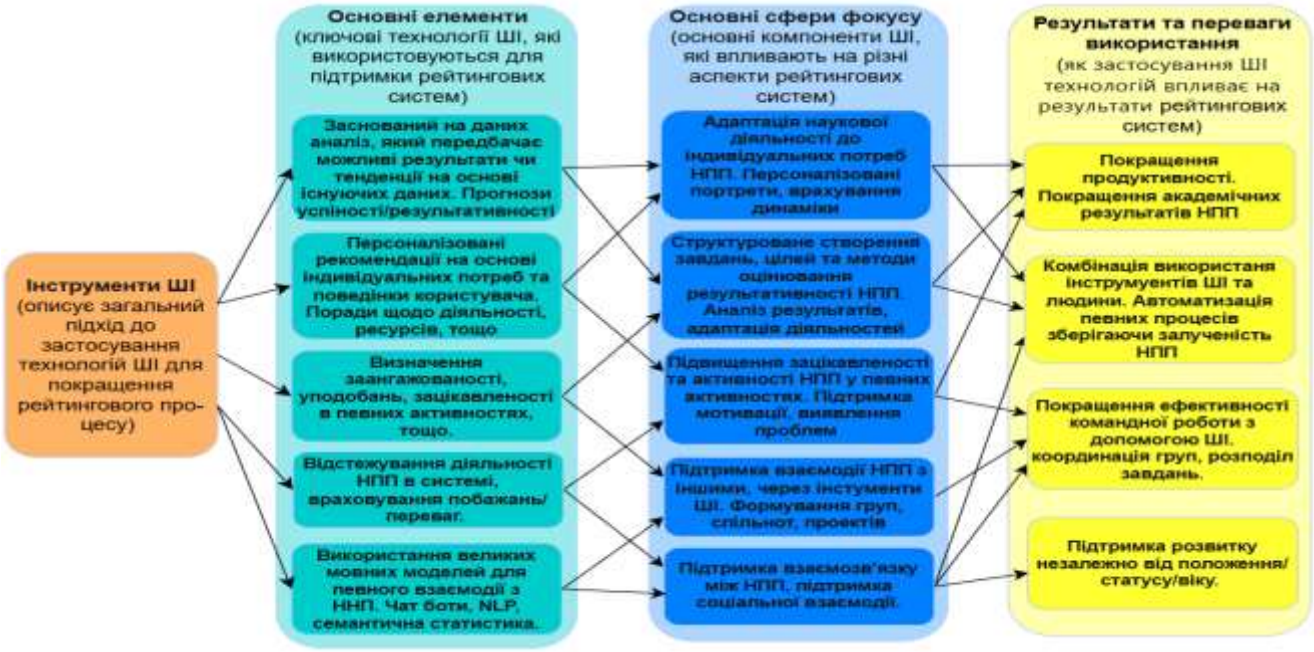


Рис. 1.4. Схематична візуалізація структури та взаємозв'язки між інстурментами ШІ

Можна сформувати такі переваги впровадження інтелектуалізованих систем у ЗВО [64-66]:

– підвищення ефективності, шляхом автоматизації різних процесів, прискорення циклів оцінювання та рейтингування працівників, звільнення ресурсів та фокус на важливих видах діяльності;

– інструменти ШІ можуть зменшувати упередження, забезпечувати об'єктивність оцінювання та рейтингування НПП, кафедр і факультетів;

– акумуляція та аналіз даних з різних джерел, формування аналітики, закономірності та тенденції в діяльності, персоналізованих рекомендації щодо покращення результативності діяльності НПП, забезпечення зворотного зв'язку;

– покращення прозорості оцінювання, рейтингування.

Водночас, впровадження таких систем має певні недоліки та виклики [66, 71]:

– інструменти ШІ мають упередженість, що може призвести до непослідовних або культурно невідповідних результатів її роботи та потребує глибини контексту і тематику даних, з якими вона взаємодіє;

– специфічні дані, вимагають здійснення критичного мислення, для забезпечення якості логіки та аргументації, що вимагає складнішого налаштування інструментів або використання допоміжних компонентів;

– потреба в наявності надійних та точних даних, великі обсяги даних для отримання більш якісних результатів чи висновків (гіпернагляд та конфіденційність даних);

– такі системи потребують певного людського контролю, тому зазвичай використовуються гібридні підходи;

– більшість таких систем є зарубіжними, відповідно існує проблема з локалізацією, інтеграцією з ЗВО, високою вартістю ліцензій або обмежений функціонал у безкоштовних рішенях.

Cflow, Watermark Faculty Success, Explorance Blue та SmartEvals можуть бути використані як частини інтелектуалізованих систем управління НПП, зокрема для процесів рейтингування, оцінювання ефективності, збору даних про якість викладання тощо [68-70]. Cflow може забезпечувати оцінювання ефективності викладачів, автоматизуючи збір даних та забезпечуючи відповідність стандартам ЗВО. Університет Міннесоти та Вісконсіна-Медісон використовують Watermark Faculty Success (раніше Digital Measures), який представляє онлайн систему звітності про діяльність викладачів, призначену для спрощення ведення обліку та звітування про щорічні види діяльності (викладання, дослідження, послуги). Вона використовується для оцінки ефективності, забезпечення кар’єрних рішень, звітування щодо акредитації. Creatrix Campus – це вже хмарна платформа, що надає рішення для підвищення кваліфікації викладачів, курування ефективністю та навантаженням працівників за допомогою автоматизації робочих процесів, інтеграцію із зовнішніми джерелами, чат-боти та технології ШІ для спрощення

адміністративних завдань та прискорення аналізу даних. Університет Піттсбурга та університет штату Арізона використовують відому платформу аналітики зворотного зв'язку нового покоління Explorance Blue, для проведення оцінювання курсів, передкурсових опитувань та централізованого збору зворотного зв'язку, з використанням автоматизації та ШІ-аналізу даних. SmartEvals використовує ШІ для оцінювання курсів, надання інтелектуальної аналітики (аналіз настроїв, виявлення тенденцій), підтримки для коучингу та врахування зворотного зв'язоку для професійного зростання. Однак важливо розуміти, що це не є автономними системами штучного інтелекту, а точніше інтелектуалізовані цифрові платформами з аналітичними функціями, які можуть підтримувати різні процеси для ЗВО та рейтингування.

У своєму дослідженні, Радкевич О. [71] говорить про те, що використання ШІ в освітньому середовищі в Україні розглядається як необмежене та інноваційне для всіх учасників освітнього процесу, включаючи викладачів, студентів, аспірантів та адміністративний персонал. Програмні системи на базі ШІ можуть оцінювати професійну діяльність у режимі реального часу, використовуючи дані з онлайн-платформ або електронних журналів та автоматизовувати різні процеси.

Для узагальнення відмінностей між зазначеними підходами, у таблиці 1.1 представлено порівняння традиційних та інтелектуалізованих систем оцінювання і рейтингування НПП, та детальніше описано в [46, 47, 49, 64-66, 71].

Таблиця 1.1

Порівняння традиційних та інтелектуалізованих систем рейтингування НПП

Критерій	Традиційні системи	Інтелектуалізовані системи
Об'єктивність та прозорість	Схильні до суб'єктивності та упереджень через людський фактор; відсутність стандартизації; критерії можуть обиратись за легкістю перевірки/виконання.	Підвищують справедливість, об'єктивність та послідовність процесів через автоматизований аналіз; зменшують упередження, складність оцінки вищих когнітивних діяльностей.

Продовження таблиці 1.1

Ефективність та автоматизація	Трудомісткі, ресурсоємні, ручні, недостатньо цифровізовані процеси; схильні до помилок; повільні цикли оцінювання.	Автоматизують збір даних (з наукометричних баз), оцінювання, генерування звітів; оптимізують робочі процеси; значно зменшують адміністративне навантаження та прискорюють цикли оцінювання.
Глибина аналізу даних та аналітичні висновки	Обмежені можливості для глибоких аналітичних висновків; складнощі з відстеженням довгострокової ефективності.	Агрегують дані з різних джерел; надають аналітику в реальному часі; виявляють тенденції, закономірності, сильні сторони та області для вдосконалення; підтримують предиктивні моделі.
Зменшення упереджень та суб'єктивності	Висока залежність від суб'єктивного зворотного зв'язку від студентів, колег, адміністраторів, що вносить упередження.	Алгоритми ШІ зменшують упередження та забезпечують послідовність оцінювання. Гібридні підходи, для нівелювання недоліків використання інстурментів ШІ.
Адаптивність та персоналізація	Часто жорсткі, стандартизовані рамки; обмежені можливості для зворотного зв'язку у реальному часі.	Адаптивність через зворотний зв'язок; зміна процесів функціоналу на основі аналізу даних; надають персоналізовані рекомендації; пропонують стратегії покращення результативності.
Адміністративне навантаження, інтеграції та стандарти	Високе навантаження; ризик невідповідності політиці ЗВО, немає інтеграцій з зовнішніми ресурсами.	Менше навантаження; відповідність інституційним політикам та стандартам акредитації; спрощують звітність; інтеграція з зовнішніми системами.

Дослідивши системи рейтингування та працю Mustafa Kayyali [60], можна вказати про певну складніть і багатовимірність взаємозв'язку між результативністю НПП та університетськими рейтингами [61]. Рейтинги, незважаючи на їхню поширеність у публічному дискурсі та освітньому процесі, не завжди забезпечують повноцінне відображення сутності академічного рівня. Вони,

як правило, базуються на обмеженому наборі кількісних показників, які можуть охоплювати не повний спектр характеристик освітньої та наукової діяльності, соціальний вплив чи ступінь залученості працівників. Крім того, надмірна орієнтація на позиції в рейтингах може спричиняти викривлення стратегічних пріоритетів ЗВО. У цьому контексті спостерігається тенденція до короткострокових дій, спрямованих на підвищення конкретних метрик, що зменшує увагу до довгострокового розвитку та сталості інституційних практик. Як наслідок, виникає ризик формування поверхневої моделі якості, що не враховує контекстуальної різноманітності та особливостей окремих закладів, а також стабільного покращення результативності НПП. Відповідно до цього, вдосконалення програмних рішень для рейтингування працівників залишається актуальним, та, на думку багатьох дослідників, необхідно формувати комплексні, багаторівневі, інтелектуалізовані системи, які включали б як кількісні, так і якісні індикатори, адаптовані до локального контексту та стратегічної орієнтації ЗВО, для покращення результативності НПП, при умові поєднання різноманітних підходів та інтелектуалізації [60, 61].

У контексті аналізу змін результативності діяльності НПП в ЗУНУ, на базі власної системи рейтингування [67], доцільним буде проаналізувати динаміку показників за міжрічними періодами, а також виявити тенденції у поведінці НПП. На рисунку 1.5 представлено частку працівників, які впродовж кожного року демонстрували покращення або погіршення своїх індивідуальних річних показників рейтингу та узагальнена класифікація працівників за їх динамікою. Варто зазначити, що хоча у певні роки (наприклад, 2018 та 2019) спостерігалося домінування позитивної динаміки (частка таких працівників перевищувала 70%), також це могло залежати від залученості користувачів до використання системи через нещодавній її запуск, проте в наступних роках ситуація суттєво змінювалася. У 2022 році, зокрема, вперше зафіксовано суттєву перевагу частки працівників із негативною динамікою. Надалі, тенденція трохи покращилася, хоча досі зберігається вагома частка працівників, які могли б підвищити свою результативність. Для узагальнення, працівників було поділено на три основні

групи, тих, хто: демонструє стабільне покращення; має ознаки сталого погіршення; мають нестабільну динаміку, яка демонструє коливання між зростанням і зниженням результативності. Як видно з даних, лише 19,9% працівників стабільно покращують свої результати, тоді як 9,4% – мають тенденцію до постійного зниження. Найчисельнішою є група з нестабільною динамікою (50,0% і більше), у межах якої частина НПП показує періодичне покращення (20,6%), а інша частина – погіршення (25,4%).

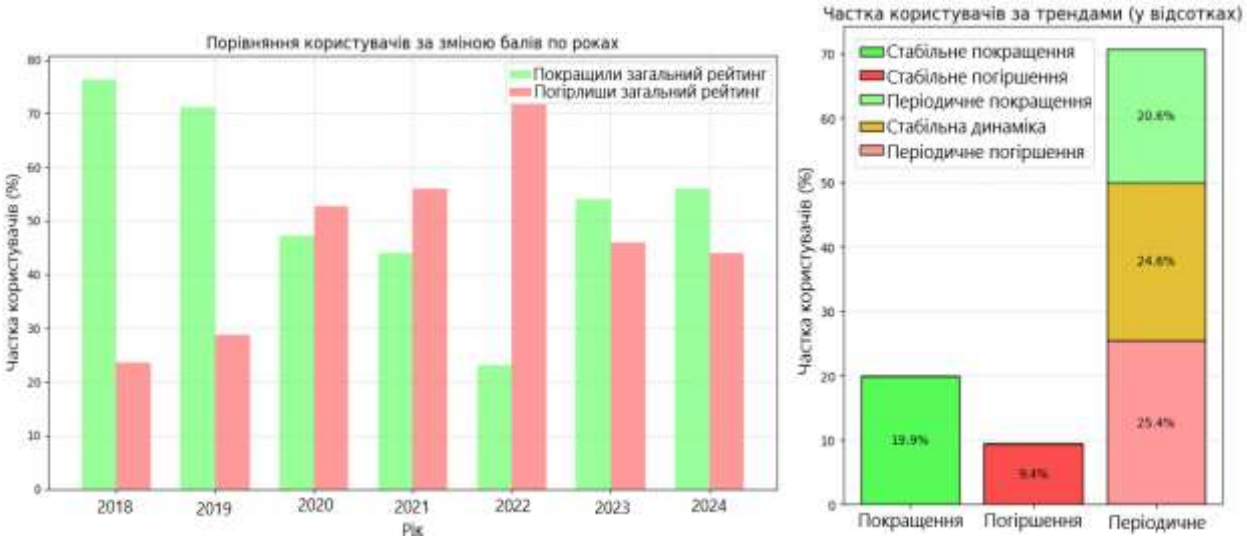


Рис. 1.5. Візуалізація динаміки зміни річних показників та частка НПП за їх трендом діяльності

Таким чином, виявлено, що значна частка НПП не демонструє сталого покращення результатів або більш стабільної діяльності. Це свідчить про потребу у впровадженні інструментів інтелектуального моніторингу та покращення результативності діяльності НПП, орієнтованих на індивідуалізований аналіз діяльності працівника, особливо для НПП з періодичною та негативною динамікою. Впровадження таких інструментів здатне зробити рейтингову систему більш відповідною до сучасних інтелектуалізованих систем оцінювання. Також доцільно врахувати семантичну складову діяльності та індивідуального профілю кожного працівника для надання персоналізованих порад щодо видів активності, що сприятимуть їх професійному зростанню.

1.3. Аналіз методів і засобів підвищення результативності діяльності НПП

На основі аналізу робіт Сергія Кубіцького [106], Сахарчук Ірини [76], та Джозефа Еваніка [77], можна виділити ряд ключових методів підвищення результативності НПП, з урахуванням сучасних викликів та інноваційних підходів. Академічному персоналу необхідно здійслювати впровадження освітніх інновацій у своїй діяльності, набувати нових знань та навичок для ефективної професійної реалізації в інформаційному суспільстві. Брати участь у міжнародних проєктах, проходити стажування, підвищувати рівень компетентності та якості досліджень, застосовувати інформаційно-комунікаційні технології (ІКТ), які сприяють цьому розвитку та бути готовим до інновацій. За даними Джозефа Еваніка, сучасні викладачі мають розвивати різні навички в освітніх технологіях, грамотності даних та культурної компетентності. Хоча це вимагає фінансових інвестицій, часу, тренерів або кураторів [77].

Оптимізація робочого навантаження є критично важливою для розподілу ресурсів, запобігання вигоранню та сприяння інноваціям. Значні зміни у робочому процесі потребують значних часових витрат. Інституції повинні визнавати ці вимоги, проводити організаційні зміни, індивідуальні консультації, та надавати підтримку, наприклад, через зміну чи перерозподіл частини робочого навантаження для НПП, що корисно впливає на індивідуальні результативність та продуктивність кадрів.

Як зазначається у роботі Сахарчук І., ефективними методами заохочення до безперервного професійного розвитку є рейтингові системи, які були розглянуті раніше. Також важливими чинниками підвищення результативності НПП є його мотивація та стимулювання. Матеріальні і нематеріальні фактори, кар’єрне зрістання може підвищувати задоволеність і залученість працівників до якісної роботи. Інституції повинні створювати культуру, що цінує та винагороджує інноваційні підходи [76].

Рекомендаційні системи в освітній сфері, з елементами інтелектуалізації, активно розвиваються у різних напрямках, зокрема, прогнозування чи поради для академічного персоналу, студентів, викладачів, щодо корисних джерел, матеріалів, сценаріїв розвитку та діяльності. Також розвиток таких систем є предметом особливої уваги міжнародних організацій. Зокрема, за даними досліджень Організації економічного співробітництва та розвитку (OECP) та у звітах і аналітичних матеріалах UNESCO, зазначено, що штучний інтелект та рекомендаційні системи дедалі активніше розробляють і впроваджують у вищій освіті для персоналізації навчання, аналізу великих масивів даних, підтримки викладачів і студентів, а також оптимізації управлінських процесів. У звітах OECP та аналітичних матеріалах підкреслюється, що AI-технології можуть забезпечити індивідуалізовані рекомендації, відстеження динаміки результативності, прогнозування академічної діяльності, а також аналітику для ухвалення управлінських рішень у закладах вищої освіти. Важливим напрямом вважається також застосування аналітики даних для моніторингу професійної активності НПП і формування систем підтримки їх професійного розвитку. У цьому контексті завдання розроблення інтелектуалізованої системи аналізу та рейтингування діяльності НПП відповідає міжнародним тенденціям розвитку цифрової освіти. [107, 16, 108, 111].

Враховуючи вищенаведені дані, можна сформувати загальні методи покращення результативності діяльності НПП (ДОДАТОК Е , таблиця Е.1).

Рис. 1.6. Узагальнена схема використання рекомендаційних систем

Враховуючи особливості різних методів підвищення результативності НПП, варто зупинитись на рекомендаційних системах, крім того, таку систему можна реалізувати на базі наявної системи оцінювання діяльності та рейтингування НПП в ЗУНУ.

На даний момент, рекомендаційні системи достатньо поширені в різних галузях, серед яких традиційно домінують електронна комерція, соціальні мережі, стрімінгові сервіси, фінансові послуги, медична сфера, новинні сайти тощо. Узагальнена візуалізація використання рекомендаційних систем зображена на рисунку 1.6 [8, 43, 73, 82, 97, 34, 35].

На данний момент, для академічної сфери, частіше зустрічаються рекомендаційні системи, які призначаються для студентів, викладачів, дослідників і загалом академічного персоналу, та можна сформувати загальні завдання, які покладаються на такі системи, серед яких є: формування рекомендацій щодо академічної співпраці та траєкторій розвитку в межах академічного середовища; пошук релевантної літератури, курсів, наукових публікацій та потенційних співавторів; вибір місця публікації результатів досліджень, відстеження метрик продуктивності, аналіз грантових можливостей та оцінка наукової ефективності.

Щоб створювати якісні рекомендації, потрібно орієнтуватись у специфіці науково-педагогічної сфери, та провести аналіз існуючих методів формування рекомендацій. Передусім, розглянемо традиційні підходи, які лягли в основу подальшого розвитку цієї галузі. До них належать, перш за все, колаборативна фільтрація (англ. Collaborative Filtering, CF), зокрема її різновиди Item-Based (IBCF) та User-Based (UBCF), і контентно-орієнтована фільтрація (англ. Content-Based Filtering, CBF). Як зазначають дослідники, зокрема, Peng [85], ці два методи є базовими при проектуванні рекомендаційних систем [82, 83].

Метрики подібності, такі як косинусна подібність та індекс Жаккара часто використовують для вимірювання подібності між об’єктами або працівниками в багатьох рекомендаційних системах, включаючи CF та CBF [82, 86]. Ці інструменти дозволяють кількісно оцінити ступінь відповідності між векторами ознак або множинами. У контексті діяльності НПП, вони можуть бути застосовані

для визначення схожості профілів викладачів, змісту навчальних матеріалів або наукових публікацій. Проте їх застосування широко розглядається в роботах Tran Phong [86], Missi Hikmatyar [87] присвячених CF та CBF та гібридним підходам з використанням графових нейронних мереж які досліджував Meng Liu [88]. Ці метрики досить гнучкі, але їхня ефективність залежить від якості та релевантності витягнутих ознак у контексті діяльності НПП.

Методи CF та CBF дозволяють виявляти неявні зв'язки між об'єктами рекомендації, що може бути корисним для розширення кругозору НПП та адаптуються до контексту діяльності базуючись на профілях та характеристиках НПП, але знижена чутливість до індивідуальних, унікальних потреб працівників, які можуть не мати "схожих" колег. Ці методи вимагають постійного перерахунку при оновленні даних, що також характерно для метрик подібності. Окрім того, виникає проблема "холодного старту" при появі нових об'єктів. Albert Paytaren у своєму дослідженні [99], пропонує персоналізовану систему рекомендацій матеріалів для викладачів ЗВО, яка використовує колаборативну фільтрацію, зіставлення профілів викладачів з доступними програмами розвитку, але без врахування зворотнього зв'язку, фокус на конкретній діяльності. Для CBF притамана висока інтерпретованість результатів, адже можна пояснити, чому саме цей елемент був рекомендований (наприклад, через схожість ключових слів, обраних метрик), а для CF – слабша, через більш складні взаємодії між працівниками. Для коректної роботи цим методам необхідна значна кількість даних, або детальний опис об'єктів взаємодії, але у вигляді компенсації, має не складну реалізацію [82, 83-88, 97, 99].

Методи матричної факторизації (MF) зазвичай використовують як доповнення до інших методів для покращення точності, проте зустрічаються і публікації, що базуються конкретно на них, наприклад, у роботі Thamaraiselvi [94]. Вони розкладають велику матрицю працівник-об'єкт на добуток двох менших матриць (латентних факторів), що дозволяє виявити приховані закономірності у даних. Сингулярне розкладання матриць SVD++ розширює класичний SVD, додаючи неявні зворотні зв'язки. Метод найменших квадратів (Alternating Least Squares,

ALS) є більш ефективним для великих розріджених матриць. Як доповнення до матричного підходу, використовують Bayesian Personalized Ranking (BPR), який може оптимізувати модель для ранжування. Ці методи широко використовуються, для поліпшення точності, та можуть показувати хорошу ефективність у виявленні прихованих інтересів та зв'язків між НПП та рекомендованими діяльностями. Щодо даних, то загалом робота ведеться зі статичними матрицями, тому виникає потреба перенавчання та доповнення матриці новими даними, та необхідний певний окремий спосіб для пердсталення профілю діяльності НПП. Дослідники цих методів зазначають що вони працюють на латентних факторах, які впливають на вподобання НПП, тим самим забезпечуючи хорошу персоналізацію, хоча інтерпретація результатів слабка, через абстрактні фактори без прямого змістового сенсу. Частково долають проблему "холодного старту", ніж CF, але все ще залишаються вразливими, та вимагають середньої або великої кількості даних для роботи [94].

Семантичні підходи (Term Frequency-Inverse Document Frequency (TF-IDF), Universal Sentence Encoder (USE), Sentence-BERT (SBERT), Content/Knowledge-Based (CB, KB)) які зосереджені на розумінні змістового значення даних. TF-IDF – класичний метод для вимірювання важливості слова в документі. Сучасніші підходи, такі як USE та SBERT, які використовували в своїх роботах Sangeetha N. [89], Attala Shadi [90], враховують семантичні значення речень, перетворюючи текст на числові вектори, придатні для обрахунку подібності. Це важливо для CB та KB систем, що працюють з текстовим змістом наукових публікацій, анотацій курсів або проєктних чи критеріальних описів діяльності НПП. Дослідники вказували про те що ці підходи, добре адаптуються до тематики науково-педагогічної діяльності, та систем, які мають суттєву кількість контентно-орієнтованих даних, хоча при необхідності враховувати часові зміни, необхідні допоміжні методи чи підходи. Персоналізація для рекомендацій може бути високою, на основі змісту профілю НПП, при умові його заповнення. Для роботи з цими методами необхідна помірна кількісті даних, та завантажені моделі, при умові

що їх не потрібно буде навчати самому. Для ефективного навчання сучасних моделей SBERT або аналогів, потрібні великі текстові дані [82, 89, 90].

Системи які базуються на секвенційно-орієнтованому (послідовним) підході, наприклад, Rule-based, Sequence-aware, рекурентні нейронні мережі (Recurrent Neural Network (RNN) чи Long Short-Term Memory (LSTM)), Transformers та інші, спеціально розроблені для аналізу послідовностей дій НПП з урахуванням динаміки. Це особливо актуально для НПП, які ведуть послідовну діяльність (публікації, участь у проєктах, педагогічна діяльність), та зазвичай відслідковується щорічно. Еволюція послідовних рекомендаційних систем перейшла від послідовностей та сесійних k-найближчих сусідів KNN (Hidasi B. [91]) до складних підходів глибокого навчання, включаючи RNN чи LSTM, механізми уваги та трансформаторні архітектури. Використання RNN та, особливо, трансформерів дозволяє моделювати довгострокові залежності та передбачати наступні дії на основі попередньої історії. Shoujin Wang [92] у своїй праці, вказує, що послідовні і сесійні методи використовують послідовності, враховують історію дій НПП для передбачення їх поведінки та майбутніх уподобань [93]. Такі підходи покращують персоналізацію та релевантність рекомендацій де спостерігається чітка послідовність дій НПП. Але вони вимагають велику кількість коректних даних для навчання, мають складнішу реалізацію, потребують обчислювальних ресурсів та експертних знань у галузі глибокого навчання, також присутня необхідність перенавчання, при зміні даних чи під час використання систем в реальному часі [91-93].

Гібридні підходи часто використовують при побудові рекомендацій, в загальному для того щоб перекрити індивідуальні недоліки певних підходів або для збереження певних особливостей систем (наприклад часто використовують CF з CB/CBF). В контексті даного дослідження, для НПП це може означати використання контент-орієнтованого аналізу даних разом із колаборативною фільтрацією, що враховує діяльність колег. Наприклад, у своїх дослідженнях Bablani D. [95], Azzam M. [96] розглядали гібридні рекомендаційні моделі для гетерогенної мережі медіа-інформації, та використовували методи на основі CF,

CBF та Deep Neural Network (DNN), що допомагає виправити проблему “холодного старту” та концептуально, такий підхід можна використати для опрацювання даних про діяльністі НПП, які також є гетерогенними (публікації, участь у проєктах, досягнення, кваліфікація тощо). Shi X. [98] пропонує послідовну рекомендаційну модель, яка інтегрує відгуки НПП, часові інтервали та послідовні патерни, використовуючи BiLSTM та CNN, що загалом можна розглянути як приклад реалізації рекомендацій діяльності НПП. Гібридних підходи дозволяють поєднувати переваги різних методів та перекривати їх недоліки, тому вони є більш гнучкими, краще охоплюють діяльності, можуть досягати високої персоналізації. Звичайно через комбінації методів реалізація стає складнішою, необхідні більші обсяги даних для ефективної роботи [95-98].

Далі, розглянемо підходи, які значною мірою ґрунтуються на машинному та глибокому навчанні ML/DL. Ці підходи, на думку дослідників, таких як Shaina Raza, Mizanur Rahman та інші [82], починають використовуватися частіше для розвитку рекомендаційних систем. Вони включають використання різноманітних алгоритмів, таких як метод опорних векторів, KNN, випадкові ліси, дерева рішень, лінійна регресія, а також рекурентні нейронні мережі та трансформери. Вони здатні виявляти складніші, нелінійні закономірності у даних. Samina Amin у своїй роботі [100] пропонує багатоагентний фреймворк рекомендаційних систем на основі DL з підкріпленням інформації, та використовує модель для персоналізації рекомендацій курсів в онлайн-навчанні, що можна інтерпретувати для діяльності НПП. Цей підхід враховує динаміку активності, персоналізацію рекомендацій, відгуки, високу адаптивність до різних контекстів, здатність моделювати складні взаємозв’язки в даних діяльності НПП. Щодо недоліків, орієнтація на послідовних взаємозв’язках, можлива складність інтепритації результатів, необхідність доступу до відкритих даних, відгуків та навчання моделі. Colton Walker із співавторами, розробили інтелектуальну систему рекомендацій [101] для персоналізації академічного розвитку студентів, використовуючи історичні дані про однолітків. Їхній гібридний підхід застосовує Марковські процеси прийняття рішень, Q-навчання та колаборативну фільтрацію для пропонування оптимальних траєкторій

навчання, що максимізують прогнозований середній бал студента. Щодо виявлених недоліків, вони зазначають: меншу точність через обмеженість даних, та відсутність зворотного зв'язку.

Відповідно, ML/DL підходи загалом мають різні алгоритми, можуть враховувати динаміку та виявляти приховані патерни. Зазвичай мають слабшу інтерпритацію (поняття "чорного ящика"), особливо серед складних моделей глибокого навчання, що ускладнює розуміння логіки рекомендації. Складніша реалізація із значними обчислювальними ресурсами та потребою в великих обсягах даних для тренування [82, 100, 101, 78].

Окремо слід зупинитися на графових нейронних та великих мовних моделях. Наприклад, Jiancan Wu у своєму дослідженні [103] використовував графові нейронні мережі, а саме Self-supervised Graph Learning для покращення графових згорткових мереж у рекомендаційних системах, які часто страждають від розрідженості даних, упередженості до популярних об'єктів та наявності шуму. Що свідчить про активне використання GNN в подібних системах, та сама структура графів підходить для моделювання непростих взаємозв'язків між працівниками та об'єктами рекомендації. У контексті НПП, це може бути мережа співпраці, наукових зв'язків або виконання певної діяльності в освітній сфері. Такі мережі враховують динаміку, через еволюцію зв'язків, підтримують високий рівень персоналізації та мають гнучкість у наповненні зв'язків, ребер та ознак. Необхідний набір даних для навчання може бути меншим ніж при використанні DL але збільшення даних зробить модель точнішою [103].

Dario Di Palma досліджував LLM [104] в рекомендаційній сфері. Дослідження показало, що мовні моделі можуть ефективно опрацьовувати списки, змістовні дані при формуванні певного результату без попереднього навчання, використовуючи рольовий промпт. Крім того, виявлено, що LLM має здатність виконувати відносно нескладну обробку даних, застосовуючи підходи, аналогічні CF та CBF, на рівні з класичними алгоритмами, що, як гіпотетично пояснює автор, через велику базу знань моделей. Щодо практичних недоліків використання LLM, автор вказує на обмеження токенів, що вимагало додаткової підготовки даних. Zhiyuan Ma у своїй

роботі [105] використовує LLM та метод Prompt Tuning щоб формувати рекомендації курсів в своїй інтелектуалізованій системі на основі тегів користувачів та зазначає перевагу у семантичному розумінні, оброблення текстових даних, можливість пояснення результатів та вирішення проблеми “холодного старту”. Проте існує проблема галюцинацій даних, неконтрольованості вихідних даних, вартості запитів та розуміння, що робота LLM залишається “чорним ящиком”. Є можливість використовувати готові моделі які систематично оновлюються та розвиваються [101, 104, 105, 109, 110].

Підсумкова таблиця порівняння підходів у рекомедаційних системах наведена в ДОДАТОК Е , таблиця Е.2.

Враховуючи проаналізовані методи та наявні структури та обсяги даних у системі рейтингування ЗУНУ, для реалізації персоналізованих рекомендацій для НПП використано графовий підхід, що дозволить застосовувати сучасні підходи машинного навчання, а саме графові нейронні мережі, без необхідності у великих наборах реальних даних, в поєднанні з метриками подібності та врахування семантики і динаміки діяльності, для підсилення роботи нейронної мережі, базуючись на подібних але кращих сценаріях розвитку. Для формування текстових рекомендацій, варіантів дій та покращення проблеми з “холодним стартом” використано великі мовні моделі, які поступово впроваджуються в рекомендаційних системах.

Проаналізувавши можливості поточної системи в ЗУНУ, подібні системи, особливо в українських ЗВО, та методи і засоби підвищення результативності діяльності, можна сказати, що вони не повністю відповідають сучасному рівню інтелектуалізації, тому можна виділити такі основні питання (рисунок 1.7), які можна вирішити:

– слабка або відсутня інтелектуалізація систем (тобто, відсутні інтеграції з іншими джерелами, автоматизація заповнення звітів тощо);

– обмежена або відсутня інтеграція з інструментами штучного інтелекту;

– відсутність персоналізованих рекомендацій для покращення результативності діяльності НПП;

– відсутність прогнозування результативності діяльності працівників.

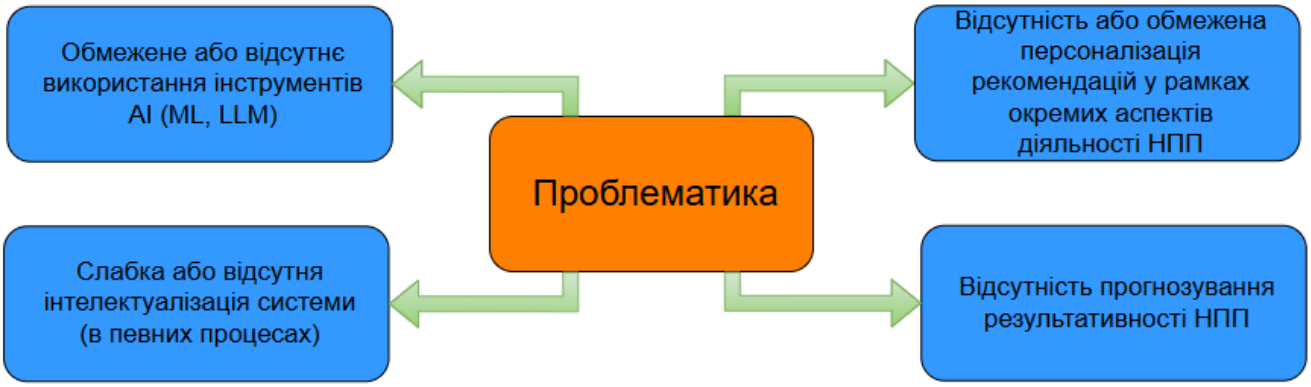


Рис. 1.7. Візуалізація проблематики систем рейтингування НПП

Якщо спробувати використати концепцію інтелектуальної системи для системи рейтингування ЗУНУ, тоді вона повинна сприймати інформацію з робочого середовища та взаємодіяти з іншими компонентами, зокрема з самими працівниками, пропонувати діяльності та поради щодо їх виконання, які будуть відповідати загальній сфері активності працівника. Тобто, система має надавати тільки рекомендації і дані щоб спростити цей процес, а працівник — самостійно приймає рішення щодо виконання або ігнорування рекомендації [117]. Під час цього, важливою буде довіра НПП до результатів роботи системи, тому також необхідна наявність пояснень, щодо сформуваних результатів, що тісно перетинається з напрямом пояснюваного штучного інтелекту (Explainable AI). При формуванні рекомендацій, це забезпечується завдяки використанню наявних даних, LLM та створених промптів, які будуть формувати приклади та аргументи щодо обґрунтовання і пояснення результатів [117, 109, 110].

Наприклад, Мартін Моліни [117] в своїй роботі пов’язаній з аналізом інтелектуальних систем та їх функціональних частин, виділяє такі основні функції: представлення предметної області (Representation subject area), опрацювання даних (Processing data) та представлення результатів та комунікація (Results & communicating). Схематичну ілюстрацію роботи інтелектуалізованої системи

рейтингування діяльності НПП наведено у функціональній блок-схемі на рисунку 1.8.

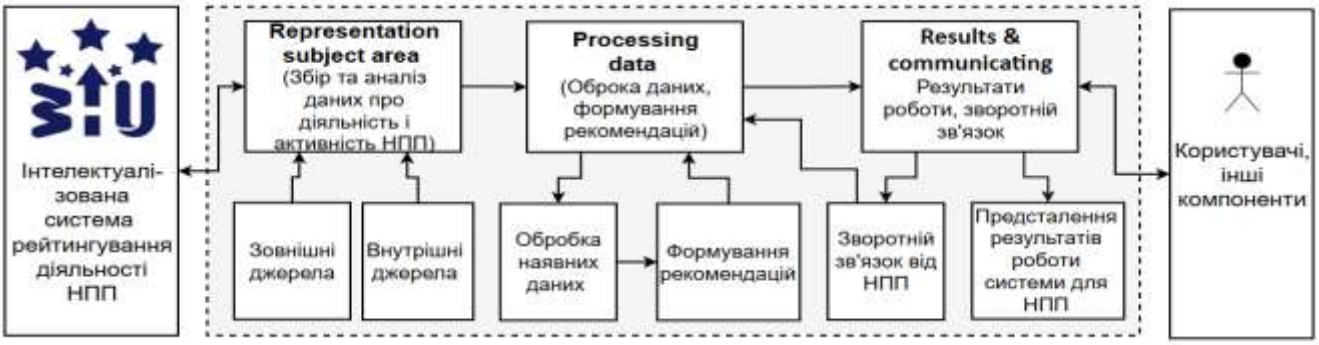


Рис. 1.8. Схематична ілюстрація роботи інтелектуалізованої системи рейтингування діяльності НПП

Варто зазначити, що функції опрацювання та комунікації є ключовими для рекомендаційної системи. Вона забезпечує взаємодію між системою та НПП, що дає можливість формувати та надавати рекомендації та отримувати зворотний зв’язок для покращення цього процесу.

1.4. Постановка задачі дослідження

В умовах динамічного розвитку вищої освіти та зростання конкуренції між закладами освіти, постає важливе завдання щодо підвищення результативності науково-педагогічних працівників. Проведений аналіз в цьому розділі, вказує, що індивідуальні досягнення НПП безпосередньо впливають на загальний рейтинг університету в міжнародних та національних рейтингах [13-18, 19]. Хоча багато університетів мають власні програмні системи для оцінювання та рейтингування діяльності НПП, однак, більшість з них, особливо в Україні, є традиційними, недостатньо інтелектуалізованими та характеризуються значним адміністративним навантаженням, обмеженими аналітичними можливостями, відсутністю автоматизацій, механізмів для покращення результативності діяльності працівників та, як наслідок, до недостатньо ефективного стимулювання професійного розвитку [33, 44, 45, 107, 16].

Проведений аналіз і виявлена проблематика засвідчують потребу у розробці інноваційних підходів та інтелектуалізованої системи, що ґрунтується на сучасних технологіях штучного інтелекту, зокрема на графових нейронних мережах і великих мовних моделях, з метою підвищення результативності діяльності НПП. Такі технології здатні не лише опрацьовувати суттєві обсяги даних, а й формувати персоналізовані рекомендації.

Таким чином, сформульовано таку задачу, спрямовану на підвищення результативності діяльності академічних працівників, на базі існуючої системи рейтингування НПП в ЗУНУ шляхом удосконалення та впровадження інтелектуалізації певних процесів в системі, яка зможе включати методи формування персоналізованих рекомендацій щодо їх діяльності.

Для вирішення поставленої задачі необхідно виконати такі завдання:

- провести аналіз відомих методів та засобів для підвищення результативності діяльності НПП, виявити їх основні переваги та недоліки;
- розробити модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі, з інтеграцією їх часових портретів, персоналізованих рекомендацій та зворотного зв’язку;
- розробити комбінований метод що уможливлює автоматизоване визначення працівників з максимально наближеними кращими показниками діяльності, а також його реалізацію у вигляді програмної моделі, використовуючи прості та зрозумілі метрики подібності з врахуванням часових рядів та семантики виконуваної діяльності;
- розробити інтелектуалізований метод та його програмну модель формування персоналізованих рекомендацій діяльності НПП, який ґрунтується на прогнозуванні показників діяльності, моделі представлення знань, використанні великих мовних моделей для генерації текстових персоналізованих рекомендацій з динамічним уточненням структури промптів, що забезпечує скорочення часу їх формування;
- виконати реінженерію системи рейтингування, що забезпечить інтелектуалізацію її процесів та генерацію персональних рекомендацій;

- провести апробацію розроблених методів та моделей, процесів генерації персональних рекомендацій для підвищення результативності діяльності НПП в рамках системи рейтингування ЗУНУ.

Висновки до розділу 1

1. Проаналізовано роль науково-педагогічних працівників як ключової ланки, що забезпечує життєдіяльність та прогресивний розвиток закладів вищої освіти. Їх різнопланова діяльність безпосередньо впливає на якість підготовки фахівців, формування інтелектуального потенціалу суспільства та міжнародну конкурентоспроможність університету. Обґрунтовано, що об’єктивне оцінювання і рейтингування діяльності НПП виступає потужним інструментом для підвищення якості освіти та наукової діяльності, а також для прийняття обґрунтованих управлінських рішень.
2. Розглянуто теоретичні засади та критерії оцінювання діяльності НПП, включаючи основні аспекти діяльності, такі як навчально-методична, науково-дослідна, організаційно-виховна та міжнародна активність. Визначено, що існує прямий системний зв’язок між індивідуальною результативністю НПП та позиціями ЗВО в міжнародних та національних рейтингах. Це підкреслює критичну важливість модернізації внутрішніх механізмів оцінки та стимулювання працівників.
3. Виявлено, що більшість існуючих систем рейтингування НПП, особливо в українських ЗВО, є переважно традиційними та характеризуються низкою суттєвих недоліків, таких як висока трудомісткість, суб’єктивність, обмежені аналітичні можливості та відсутність персоналізованого зворотного зв’язку. На противагу цьому, враховуючи сучасний розвиток цієї сфери, в інтелектуалізовані системи починають інтегрувати технології штучного інтелекту, які використовуються для автоматизації більшості процесів, забезпечення об’єктивності та створення персоналізованих рекомендацій.

4. Розглянуто методи та засоби підвищення результативності діяльності НПП, та на основі проведеного аналізу обґрунтовано необхідність розробки інтелектуалізованої системи, яка б не лише оцінювала діяльність НПП, а й активно сприяла підвищенню їхньої результативності шляхом формування персоналізованих рекомендацій щодо діяльності. Запропоновано використовувати гібридний підхід, який поєднує методи графових нейронних мереж, метрики подібності та великі мовні моделі, що дозволить враховувати динаміку, семантику та контекст діяльності НПП для надання обґрунтованих і зрозумілих рекомендацій.

РОЗДІЛ 2
МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

У цьому розділі розглядається комплексний підхід до формування персоналізованих рекомендацій для підвищення результативності діяльності працівників. Який включає в себе аналіз діяльності НПП, застосування метрик подібності, графових нейронних мереж, а також великих мовних моделей для генерації корисних текстових рекомендацій.

У межах дисертаційного дослідження постала потреба проаналізувати та зрозуміти як працює наявна система рейтингування НПП, що використовується в Західноукраїнському національному університеті [1, 11]. Важливо описати цей процес, щоб забезпечити подальше використання даних, інтеграцію з наявною системою, формування обґрунтованих рекомендацій та створити методологічну основу, яка відповідає політикам оцінювання ЗВО, що підтверджує практичну цінність системи.

Далі, розглянуто запропонований комбінований метод ранжування НПП, що ґрунтується на застосуванні різноманітних метрик подібності. Такий підхід дає змогу визначити НПП з наближеними кращими показниками діяльності, що посилить взаємозв’язки між ними в наступних обрахунках та, відповідно, підвищить ефективність формування рекомендацій. Окрім того, запропоновано та обґрунтовано використання гетерогенних графових нейронних мереж з механізмами уваги для моделювання динаміки діяльності НПП та прогнозування їх майбутніх досягнень. Цей метод дозволяє враховувати не лише індивідуальні показники, а й контекстні зв’язки між працівниками, кафедрами та критеріями оцінювання [2, 3].

Завершальним етапом запропонованого підходу є формування розгорнутих текстових рекомендацій за допомогою великих мовних моделей. У цьому процесі взаємодія з LLM здійснюється за допомогою створених промптів, які включають

результати роботи графової моделі, дані про діяльність працівника та його успішних колег. Завдяки інтеграції LLM з інструментами зовнішнього пошуку, рекомендації стають не лише більш персоналізованими, але й аргументованими та актуальними.

Отже, розділ розкриває концептуальні основи та практичні аспекти побудови інтелектуалізованої системи рекомендацій, яка використовує переваги сучасних методів аналізу даних для підвищення результативності НПП.

Основні результати цього розділу опубліковані в працях [1, 2, 5, 11].

2.1. Метод формування рейтингової оцінки діяльності науково-педагогічних працівників

Насамперед, розглянемо процес формування рейтингової оцінки діяльності НПП, який базується на систематичному зборі, обробці та агрегації даних, що відображають їхню професійну активність упродовж звітного періоду. Цей процес забезпечує об’єктивний моніторинг ефективності діяльності НПП та сприяє посиленню їхньої мотивації до професійного розвитку [11, 12, 39, 40, 67].

Загальна функціональність системи рейтингового оцінювання в ЗУНУ відповідає поширеним практикам, що застосовуються в інших українських ЗВО і передбачає щорічне заповнення звітів, що містять інформацію за різними критеріями діяльності. Кожен критерій характеризується описом діяльності, бальним значенням або формулою розрахунку, типом обрахунку балу та належністю до певної групи активності. В системі, діяльність поділена на шість основних груп активності: “Досягнута кваліфікація”; “Публікаційна активність”; “Грантова та патентно-ліцензійна діяльність”; “Науково-організаційна робота”; “Навчально-наукова робота”; “Додаткові критерії”, які в цілому охоплюють більшість видів діяльностей, що відстежуються в українських ЗВО.

Річний звіт НПП (далі – звіт) формується на основі критеріїв, кожен з яких асоційований з однією з вищезазначених груп. Обчислення балів заповнених критеріїв здійснюється відповідно до наявного встановленого типу обрахунку в

системі. Існує п’ять основних типів обчислень балів, які можуть бути застосовані до різних критеріїв:

– фіксований бал за критерій, застосовується для діяльності, яка має чітко визначене бальне значення, що нараховується одноразово за факт її виконання;

– вибір однієї/кількох заздалегідь визначених опцій. Критерій може містити декілька варіантів типів, класів, варіацій діяльності, кожен з яких оцінюється певною кількістю балів. Користувач може обрати одну або декілька опцій, і сума балів за обрані варіанти формує загальний бал за критерій. Максимальний бал за такий критерій визначається сумою балів за всі доступні варіанти;

– багаторазове виконання з фіксованим балом за одиницю. Для цього типу критерію бали нараховуються за кожне виконання певної діяльності, що дозволяє отримати необмежену кількість балів за критерієм залежно від частоти виконання;

– розрахунок за формулою. Бал за критерій визначається за допомогою математичної формули, яка включає одну або дві змінні, що заповнюються користувачем. Це забезпечує можливість динамічного розрахунку балу для якіснішої оцінки виконаної діяльності НПП на основі її кількісних показників;

– коефіцієнт, який застосовується як множник до загального річного балу НПП, що дозволяє враховувати певні особливі досягнення або адміністративні посади.

Тип заповнення та розрахунку критерію зазначає спеціалізована рада ЗВО. Тому додаткові вагові значення для критеріїв не обчислюються, оскільки їх балова оцінка вже включає важливість критерію для університету, адже вищий бал свідчить про пріоритетність відповідної діяльності.

Загальний річний бал (S) науково-педагогічного працівника обчислюється шляхом підсумовування балів, отриманих у кожній з шести груп активності. Спочатку бали за кожним критерієм у звіті (s_c) обчислюються відповідно до їхнього типу розрахунку. Далі ці бали сумуються в межах кожної групи активності (s_g), де g відповідає конкретній групі.

Річний бал, за кожною групою активності, математично розраховується як:

$$s_g = \sum_{c \in C(u)}^{n_g} s_{u,c,g}, \tag{2.1}$$

де: s_g – сумарний бал за групу активності g, n_g – кількість критеріїв у групі активності g, а $s_{u,c,g}$ – бал користувача u за критерій c у групі g.

Загальний річний бал НПП (S) формується шляхом підсумовування балів з усіх груп активності:

$$S = \sum_{g=1}^{n_g} s_g, \tag{2.2}$$

Формульний тип розрахунку балу за критерій виконується за формулою, яка формується адміністратором. Вона може мати одну чи дві змінні, та виконувати звичайні математичні операції (додавання, віднімання, ділення, множення). Загалом розрахунок критерію з таким типом обрахунку виглядає так:

$$s_g = eval(F_{str}), \tag{2.3}$$

де: F_{str} – рядкова формула, задана при створенні критерію; $eval$ – функція, яка перетворює стрічкову формулу в математичну та обчислює її. Її обчислення реалізовано з використанням механізму обробки виняткових ситуацій та попередньою валідацією на наявність атрибутів математичної формули.

Наприклад, для групи “Публікаційна активність” формула розрахунку балів за критерій “Публікації у співавторстві у міжнародних наукових виданнях, які входять до наукометричних баз Scopus чи Web of Science без квартилю” може виконуватися кілька разів та розраховується за формулою, що має такий вигляд:

$$s_g = \sum_{1}^{M} \frac{N_{atr} \cdot s_{base,g}}{K_{atr}}, \tag{2.4}$$

де: $s_{base,g}$ – фіксований показник балу за критерій g у формулі (це може бути базовий, максимальний або узагальнений бал, чи коефіцієнт); N_{atr} – перший атрибут, для змінної 1, у цьому прикладі відповідає за певний коефіцієнт для поточного критерію (наприклад, 10 чи 0.2); K_{atr} – другий атрибут, для змінної 2, в

даному прикладі відповідає за кількість співавторів, включаючи апліканта; M – кількість статей відповідного типу.

Для критерію “Рецензування дисертацій на здобуття наукового ступеня доктора наук”, використовується не формула, а просто можливість кількаразового виконання критерію, для якого використовується формула:

$$s_g = M \cdot s_{base,g}, \tag{2.5}$$

де: M – кількість рецензувань дисертацій на здобуття наукового ступеня доктора наук; $s_{base,g}$ – базовий бал (наприклад, 20 чи 50 балів).

Критерій, який має тип обрахування – коефіцієнт, розраховується за формулою:

$$S = S \cdot Koef, \tag{2.6}$$

де: S – загальний річний бал користувача; $Koef$ – введений коефіцієнт.

Після обрахунку річних балів для кожного НПП формується загальний рейтинг. Для забезпечення довгострокового аналізу динаміки результативності та стимулювання постійного розвитку, загальний рейтинг НПП враховує їхні річні бали за останні п’ять років. Цей підхід дозволяє нівелювати випадкові відхилення та представити кумулятивні досягнення. Візуалізація розрахунку річного рейтингу користувача, наведено на рисунку 2.1.

Рис. 2.1. Візуалізація розрахунку і формування рейтингу користувачів в інтелектуалізованій системі рейтингування НПП в ЗУНУ

Формула розрахунку загального рейтингу (TR) НПП може бути представлена таким чином:

$$TR = S_T \cdot W_0 + S_{T-1} \cdot W_1 + S_{T-2} \cdot W_2 + S_{T-3} \cdot W_3 + S_{T-4} \cdot W_4, \quad (2.7)$$

де: S_T – річний бал за поточний звітний рік T; S_{T-1} – річний бал за попередній рік, і так далі до S_{T-4} – річний бал за чотири роки до поточного; W – коефіцієнт ваги для відповідного року, який зменшується з плином часу, що відображає більшу значущість недавніх досягнень. В поточній реалізації системи, використовуються такі коефіцієнти: W_0=1, W_1=0.5, W_2=0.25, W_3=0.125, W_4=0.0625, які сформовані адміністраторами та відповідають внутрішній політиці університету [11].

Існуючий метод формування рейтингової оцінки забезпечує структурований підхід до аналізу та агрегації інформації про діяльність НПП, що слугує базою для подальшого розвитку інтелектуалізованої системи рейтингування та формування персоналізованих рекомендацій.

2.2. Модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі

Як уже зазначалося раніше, діяльність НПП є складною системою, що включає різноманітні та взаємопов’язані сутності: публікації, проєкти, навчальні дисципліни тощо. Ефективне моделювання цих даних є ключовою передумовою для побудови інтелектуалізованих систем та формування персоналізованих рекомендацій діяльності для НПП.

Для системи рейтингування ЗУНУ, на початкових етапах розробки, для збереження даних було обрано MySQL, який має певні переваги в порівнянні з аналогічними БД. Це дозволило швидко розгорнути базовий функціонал системи, що включає зберігання даних користувачів, звітів, критеріїв та механізмів їх затвердження. Реляційна модель добре справляється з даними, що мають чітко визначену структуру та передбачувані взаємозв’язки, що характерно для багатьох аспектів діяльності НПП.

Проте, при подальшому розвитку системи та реалізації інтелектуалізованих модулів, таких як формування персоналізованих рекомендацій, виникла потреба в ефективній обробці специфічніших взаємозв'язків та додаткових ознак між об'єктами. У реляційній моделі такі зв'язки часто реалізуються через множинні таблиці з проміжними ключами, що призводить до виконання великої кількості JOIN-операцій при запитах. Це, в свою чергу, може знижувати продуктивність та ускладнювати розробку складніших запитів, відповідно підтримка мережевих структур даних у MySQL не оптимальна, тому ця СУБД не повністю відповідає поставленим завданням. Зокрема, для ефективного пошуку подібних користувачів та виявлення неявних зв'язків у їхній діяльності потрібна модель, що природно відображає мережеві структури. Саме тому для опису моделі представлення знань про діяльність НПП та збереження проміжних даних і результатів, для процесу формування рекомендацій щодо діяльності для працівників, було прийнято рішення інтегрувати знання-орієнтований підхід на основі графової бази даних Neo4j для процесу формування рекомендацій [122]. Щодо основних переваг графових моделей можна назвати такі:

– більш природна структура даних на основі вузлів і зв'язків, яка ефективніше відображає реальні зв'язки між вузлами та простіша в додаванні нових, порівняно з реляційними таблицями;

– краща швидкодія досягається завдяки оптимізованішій роботі графових баз з вузлами та зв'язками, а це особливо важливо для таких задач, як пошук подібних НПП за спільними інтересами чи діяльністю;

– графові БД простіше модернізовувати та оновлювати, описуючи нові зв'язки, вузли тощо;

– спорідненість з реалізацією гетерогенної графової нейронної мережі (ГГНМ) у процесі формування рекомендацій.

Таким чином, було реалізовано гібридний підхід за якого MySQL використовується для основного функціоналу системи рейтингування, а Neo4j — для побудови графу знань, що лежить в основі рекомендаційного модуля, та додатково не навантажує і не ускладнює систему рейтингування.

Спочатку, було проведено експорт необхідних даних з MySQL (БД системи рейтингування) та імпортовано до Neo4j, завдяки скриптам, також цей процес включає не тільки імпорт існуючих сутностей, таких як користувачі, звіти та критерії, а й їхню структуризацію в модель представлення знань, яка є графом властивостей, що складається з типізованих вузлів та спрямованих зв'язків, які можуть мати власні атрибути [131]. Веб-інтерфейс СУБД Neo4j з відображенням частини структури моделі представлення знань, зображено на рисунку 2.2.

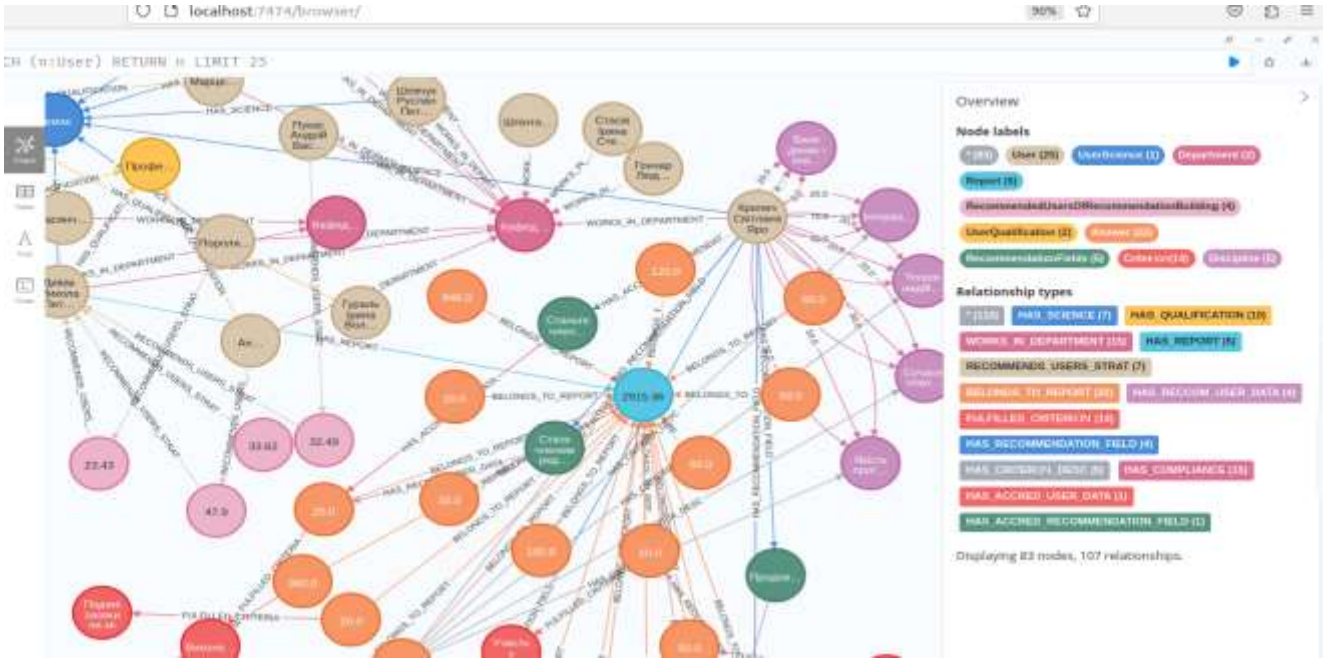


Рис. 2.2. Веб-інтерфейс для взаємодії з Neo4j з прикладом структури моделі представлення знань

Модель містить такі основні вузли:

- User, представляє користувача системи. R_{User} ={⟨id, name, qualification, science, interests⟩}.
- Report, загальна інформацію про річний звіт користувача. R_{Report} ={⟨id, user_id, template_id, totalPoints, year, section1, …, section6⟩}.
- Criterion, для опису окремого критерію оцінювання діяльності. $R_{\text{Criterion}}$ ={⟨id, text, notes, max_point, section_id, type⟩}.
- Answer, представляє дані, які надав користувач щодо конкретної виконаної діяльності. R_{Answer} ={⟨id, points, text, options, N, K, report_id, criteria_id⟩}.

- UserScience, стосується наукового ступеня користувача, наприклад, “кандидат наук”, “доктор наук”. $R_{\text{UserScience}}$ ={⟨title⟩}.
- UserQualification, аналогічно до UserScience, для вченого звання. $R_{\text{UserQualification}}$ ={⟨title⟩}.
- Department, вузол для об’єднання користувачів по кафедрах. $R_{\text{Department}}$ ={⟨id, name⟩}.
- RecommendStrategyBestUsers, цей вузол призначений для зберігання метрик подібності між користувачами. $R_{\text{RecommendStrategyBestUsers}}$ ={⟨userId, recommendUserId, year, totalUsPoints, totalReUsPoints, JaccardIndex, cosineSimilarity, TDWG, TDWS, totalAdcantage, performanceFactor⟩}.
- RecommendedUsersOfRecommendationBuilding, вузол для зберігання інформації про знайдених користувачів для підсилення зв’язків при використанні і навчанні графової нейронної мережі. $R_{RecUsOfRecBuid}$ ={⟨userId, recommendUserId, year, similarityPercentage⟩}.
- RecommendationFields, основний вузол для зберігання рекомендацій, які отримуються в результаті роботи графової нейронної мережі. $R_{\text{RecFields}}$ ={⟨user_id, report_id, rec_field_id, rec_group, variant, potential_score, probability, rec_details, type⟩}.

І такі зв’язки між ними:

- HAS_DISCIPLINE (зв’язує User – Discipline), дисципліна яку викладає користувач.
- HAS_COMPLIANCE (зв’язує User – Discipline), вказує на відповідність публікацій користувача певній дисципліні, яка визначається завдяки LLM.
- HAS_REPORT (зв’язує User – Report), звіт користувача.
- PART_OF_TEMPLATE (зв’язує Criterion – Template), критерій звіту.
- WORKS_IN_DEPARTMENT (зв’язує User – Department), кафедра користувача.
- FULFILLED_CRITERION (зв’язує Answer – Criteria), дані користувача за певний критерій, з властивістю набраного балу (points).

- BELONGS_TO_REPORT (зв’язує Answer – Report).
- PART_OF_REPORT (зв’язує Criterion – Report), показує, які критерії належать до конкретного звіту.
- HAS_SCIENCE, HAS_QUALIFICATION (зв’язує User – UserScience/ UserQualification), науковий ступінь і кваліфікація користувача.
- RECOMMENDS_USERS_STRAT (зв’язує User – User), відображає подібність користувачів, базуючись на загальному індексі подібності.
- HAS_RECOMMENDATION_FIELD (зв’язує User – RecommendationFields), вказує на рекомендовані критерії для покращення діяльності розраховані графовою нейронною мережею, де властивістю виступає потенційний бал.
- HAS_ACCRED_RECOMMENDATION_FIELD (зв’язує User – RecommendationFields), вказує на рекомендовані критерії, які потрібні для користувача, щоб відповідати ліцензійним вимогам.
- HAS_CRITERON_DESC (зв’язує User – Discipline), відображає відсоток відповідності публікації до дисципліни (розраховується з LLM).

Для підвищення семантичної насиченості моделі та забезпечення можливості формування більш глибоких та обґрунтованих рекомендацій, початкова структура була розширена новими типами вузлів та зв’язків. Ця розширена модель дозволяє декомпозивати загальні вузли працівників на конкретні, типізовані сутності, що відображають різні аспекти наукової та педагогічної діяльності [131].

Додаткові вузли можна згрупувати за наявними в системі групами активності для кращого розуміння їх ролі в графі знань:

- Вузол типу Publication зберігає інформацію про наукові публікації, $R_{\text{Publication}}$={⟨title, year, authors, doi, url, database, keywords⟩}.
- Journal, для опису наукових видань, R_{Journal} ={⟨title, abbreviation, quartile, issn, doi_prefix, publisher_country⟩}.
- Conference, для фіксації участі у наукових конференціях, $R_{\text{Conference}}$ ={⟨name, year, level, number, location⟩}.

- Monograph, вузол для монографій, $R_{\text{Monograph}}$ ={⟨is_indexed_sc_wos, year, publisher, language⟩}.
- EducationalMaterial, для навчально-методичних матеріалів, $R_{\text{EducationalMaterial}}$ ={⟨title, year, type⟩}.
- ScientificProject, описує наукові проєкти, $R_{\text{ScientificProject}}$ ={⟨title, scope, category, number⟩}.
- Grant, для представлення грантових заявок, R_{Grant} ={⟨title, type, number, status⟩}.
- Supervision, фіксує дані про наукове керівництво, $R_{\text{Supervision}}$ ={⟨speciality, degree_level, thesis_title, year⟩}.
- IntellectualProperty, для об’єктів інтелектуальної власності, $R_{\text{IntellectualProperty}}$ ={⟨title, number, year, type⟩}.
- AcademicCouncil, представляє членство у радах, $R_{\text{AcadCouncil}}$ ={⟨level, title⟩}.
- Organization, зберігає інформацію про організації, R_{Org} ={⟨title, type⟩}.
- Keyword, для тематичних ключових слів, що дозволяють об’єднувати різні види діяльності, наукові інтереси тощо, R_{Keyword} ={⟨term⟩}.

Для поєднання описаних вище вузлів у єдину мережу знань були визначені наступні семантичні зв’язки, які здатні моделювати процеси, дії та відносини між сутностями:

- AUTHORED (зв’язує User – Publication чи Monograph), встановлює зв’язок між НПП та його науковими роботами.
- CO_AUTHORED_WITH (зв’язує User – User), є похідним зв’язком, що моделює колабораційну мережу між науковцями.
- LEADS (зв’язує User – ScientificProject), визначає роль НПП як керівника наукового проєкту.
- PARTICIPATES_IN (зв’язує User – ScientificProject), визначає роль НПП як учасника наукового проєкту.
- APPLIED_FOR (зв’язує User – Grant), пов’язує НПП із поданою ним грантовою заявкою.

- SUPERVISED (зв'язує User – Supervision), відображає факт наукового керівництва дисертаційною роботою.
- MEMBER_OF (зв'язує User – AcademicCouncil), показує членство у спеціалізованій раді.
- MEMBER_OF_EDITORIAL_BOARD (зв'язує User – Journal), вказує на участь у редакційній колегії журналу, де властивість role деталізує характер участі.
- MEMBER_OF_PROGRAM_COMMITTEE (зв'язує User – Conference), фіксує участь у програмному комітеті конференції.
- COMPLETED_INTERNSHIP (зв'язує User – Organization), пов'язує НПП з організацією, де він проходив стажування.
- PUBLISHED_IN (зв'язує Publication – Journal), з'єднує видання публікації.
- PRESENTED_AT (зв'язує Publication – Conference), встановлює на якій конференції була представлена публікація.
- FUNDED_BY (зв'язує ScientificProject – Organization), вказує на організацію-фундатора наукового проєкту.
- HAS_RESEARCH_INTEREST, HAS_KEYWORD (зв'язує User – Keyword, Publication – Keyword), пов'язує НПП з його науковими інтересами та ключові слова до вузлів.
- EXTRACTED_FROM, зв'язує будь-який інший вузол, який в даний момент не визначений структурою моделі – Answer, забезпечує наповненість моделі даними, для об'єднання різних пов'язаних між собою вузлів.

Зв'язки в графовій моделі дозволяють формувати ланцюжки взаємопов'язаних між собою сутностей, які визначаються як кортежі знань, які представляють визначені факти або приховані закономірності в діяльності НПП. Формально, їх можна представити як шлях P у графі: $P = (v_0, e_1, v_1, e_2, \dots\ v_k, e_k)$, де v_k – вузли (суб'єкти або об'єкти), а e_k – спрямовані зв'язки між ними, k – кількість вузлів.

Дані для формування моделі представлення знань про діяльність НПП, беруться з наявних даних зі звітів працівників. Для частини зв'язків, використано структуровані дані з бази, для іншої частини – було описано проміжні підзапити для розмежування і виявлення сутностей та формування зв'язків між ними. Та

частину даних, яку краще описати у вигляді відповідних зв'язків, опрацьовуючи за допомогою ШІ інструментів, наприклад ключові слова стосовно критеріїв пов'язаних з проєктами або веденням проблемних груп тощо. Ця модель знань, дозволить ідентифікувати успішні патерни діяльності одних науково-педагогічних працівників, що представлені довгими, семантично насиченими кортежами та виявляти потенційні напрями для вдосконалення в інших співробітників, діяльність яких характеризується короткими або відсутніми кортежами.

Ця модель представлення знань буде використовувати побудовані кортежі знань, щоб сформувати уточнення даних для рекомендацій, а саме: розширення варіантів подібних рекомендованих критеріїв, отримання прикладів і варіантів виконання діяльності (журналів, проектів, конференцій та іншого), списки подібних за інтересами авторів з якими можна спробувати опублікувати статті тощо. Для того, щоб програмно реалізувати пошук даних для рекомендацій у графі знань, необхідно формалізувати логіку цих запитів, тому далі розглянемо математичний опис операцій, які дозволяють здійснити це.

Для розширення рекомендацій, обчислюємо подібність (*SimilarityScore*) для кожного потенційного колеги (u_2) відносно поточного користувача (u_1) і відбираємо топ-5 найкращих. Для пари користувачів (u_1, u_2) розраховуються наступні показники, що базуються на кількості спільних сутностей: співавтори (*CoAuthorScore*), проєкти (*ProjectScore*), наукові інтереси (*InterestScore*), ключові слова публікацій (*KeywordScore*), участь у програмних комітетах (*ConfScore*):

$$CoAuthorScore(u_1, u_2) = Count(N(u_1, CO_AUTHORED_WITH) \cap N(u_2, CO_AUTHORED_WITH)) \quad (2.8)$$

де: кількість елементів (*Count()*) перетину (∩) множин співавторів для користувачів u_1 та u_2; *N(u, R)* – множина сусідів вузла *u* за зв'язком *R*.

$$ProjectScore(u_1, u_2) = Count(N(u_1, PARTICIPATES_IN \cup LEADS) \cap N(u_2, PARTICIPATES_IN \cup LEADS)), \quad (2.9)$$

$$InterestScore(u_1, u_2) = Count(N(u_1, HAS_RESEARCH_INTEREST) \cap N(u_2, HAS_RESEARCH_INTEREST))\ HAS_RESEARCH_INTEREST)) \quad (2.10)$$

$$Keywords(u) = \pi_{term}(\sigma_{id} = u.id(R_{User}) \bowtie AUTHORED \bowtie R_{Publication} \bowtie HAS_KEYWORD \bowtie R_{Keyword}), \quad (2.11)$$

$$KeywordScore(u_1, u_2) = Count(Keywords(u_1) \cap Keywords(u_2)) \quad (2.12)$$

де: σ_{id} (...) – фільтрація, щоб знайти сутності за певним критерієм (наприклад, користувача за його id); π_{term} (...) – проєкція, конкретне поле із знайдених сутностей (наприклад, *term* – науковий інтерес); $\bowtie$ – об’єднання ланцюжка сутностей через зв’язки.

$$ConfScore(u_1, u_2) = Count(N(u_1, MEMBER_OF_PROGRAM_COMMITTEE) \cap N(u_2, MEMBER_OF_PROGRAM_COMMITTEE)) \quad (2.13)$$

Просумуємо отримані компоненти:

$$SimilarityScore(u_1, u_2) = ProjectScore + CoAuthorScore + ConfScore + InterestScore + KeywordScore \quad (2.14)$$

Множина *SimilarUsersSet* формується з *id* топ-5 користувачів (u_2), для яких *SimilarityScore* > 0, відсортованих за спаданням цього рейтингу.

Тепер опишемо етап генерації прикладів діяльності, на якому аналізуємо діяльність подібних користувачів у контексті бажаних критеріїв, які обрав користувач (*DesiredCriteriaSet*). Спочатку знаходимо всі факти виконання бажаних критеріїв подібними користувачами та пов’язані з ними сутності (публікації, проєкти тощо), та збережемо це в множину *FulfilledActivities*:

$$FulfilledActivities = \pi_{Entity,R_{Criterion},R_{User}}(\sigma_{R_{User}.id} \in SimilarUsersSet \wedge R_{Criterion}.id \in DesiredCriteriaSet(User \bowtie HAS_REPORT \bowtie R_{Report} \bowtie BELONGS_TO_REPORT \bowtie R_{Answer} \bowtie FULFILLED_CRITERION \bowtie R_{Criterion} \bowtie EXTRACTED_FROM \bowtie Entity)) \quad (2.15)$$

Ми об'єднуємо ланцюжок сутностей від R_{User} до *Entity* (де *Entity* — це будь-який вузол діяльності, як $R_{Publication}$, $R_{ScientificProject}$ тощо), фільтруємо (σ) результат за *id* користувачів та критеріїв, вибираємо (π) унікальні трійки (сутність, критерій, користувач) і повертаємо відсортовані результати по року (*Report.year*) та по балу (*Answer.points*), беручи тільки по 5 записів з кожного типу отриманих сутностей.

Тепер знайдемо інші подібні критерії, ґрунтуючись на моделі знань та варіативних критеріях які були обрані під час уточнення рекомендації (*SuggestedCriteriaSet*). Визначення релевантних типів сутностей, проводимо так:

$$RelevantEntityTypes = \pi_{EntityType}(\sigma_{R_{User}.id} \in SimilarUsersSet \wedge R_{Criterion}.id \in SuggestedCriteriaSet(R_{User} \bowtie \ldots \bowtie Entity)) \quad (2.16)$$

де: $R_{User} \bowtie \ldots \bowtie Entity$ – шлях між користувачем та *Entity* через побудовані зв'язки.

Таким чином отримуємо лише типи знайдених сутностей (*EntityType(Entity)*), наприклад, "*Publication*", "*ScientificProject*".

Тепер шукаємо інші критерії, які подібні користувачі виконували за допомогою тих самих типів діяльності:

$$CandidateCriteria = \pi_{R_{Criterion}}(\sigma_{R_{User}.id} \in SimilarUsersSet \wedge EntityType(Entity) \in RelevantEntityTypes(R_{User} \bowtie \ldots \bowtie R_{Criterion})). \quad (2.17)$$

Водночас, для розширення пошуку у випадках, коли зв'язок за типом діяльності є недостатнім, виконуємо пошук критерії з тих самих груп активності, що й початкові та ранжуємо їх за кількістю спільних ключових слів. Це дозволяє

ідентифікувати тематично найближчі альтернативи. Формально цей процес опишемо наступним виразом:

$$SimiCrit = Sort_{desc}(Count((\pi_{term}(\sigma_{id}\, c.id(R_{Criterion} \bowtie \ldots \bowtie R_{Keyword}))) \cap (\pi_{term}(\sigma_{id} = SuggestedCriteriaSet(R_{Criterion} \bowtie \ldots \bowtie Entity \bowtie HAS_{KEYWORD} \bowtie R_{Keyword}))))(\sigma_{section_{id}} \in (\pi_{section_{id}}(\sigma_{id} = SuggestedCriteriaSet(R_Criterion))) \wedge id \notin SuggestedCriteriaSet(R_Criterion)) \quad (2.18)$$

Ми отримуємо ключові слова які пов’язані з варіативними діяльностями, потім беремо критерії які знаходяться в подібних групах активності, ранжуємо їх за кількістю співпадінь по зв’язкам.

Та відбираємо унікальні критерії: *SuggestedSimilarCriteria = (CandidateCriteria* $\cup$ *SimiCrit)\DesiredCriteriaSet*, з обмеженням топ-3 за частотою їх згадки по зв’язкам. Кінцевий набір даних для рекомендацій є об’єднанням результатів попередніх етапів.

$$ExpansionDataset = FulfilledActivities \cup SuggestedSimilarCriteria \quad (2.19)$$

Цей формалізований опис відтворює логіку роботи Cypher-запиту, розбиваючи його на послідовні операції над графом знань.

Для прикладу, наведемо формування рекомендацій для користувача (id=45), який є доктором наук, професором, та має наукові інтереси в сферах “моделювання”, “онтологічний підхід”, “аналіз контенту” та “біогазові установки”, у вигляді кортежу. Аналіз його діяльності за попередні роки та порівняння з діяльністю його колег дозволяє сформувати йому рекомендацію щодо підвищення публікаційної активності у виданнях без квартилю, та запропонувати певний журнал для публікації. За допомогою створеного запиту на мові Cypher, на основі побудованих вузлів та зв’язків в моделі, де виконується пошук ланцюжків кортежів

щодо подібних сутностей, та формуємо датасет конкретних даних, в залежності від критеріїв які були сформовані користувачу. Це можуть бути приклади журналів, конференцій, проектів, авторів, подібні публікації, конкретні виконані діяльності які можна навести в приклад та додаткові критерії до виконання. Такий датасет буде використовуватись як допоміжні дані для формування уточненого тексту персоналізованих рекомендацій. Схематично, за структурою моделі знань, кортеж може виглядати так: (u1:User {id: 45})-[:PARTICIPATES_IN]->(project:ScientificProject)<-[:LEADS]-(u2:User)-[:AUTHORED]->(pub:Publication)-[:HAS_KEYWORD]->(kw:Keyword)<-[:HAS_RESEARCH_INTEREST]-(u1:User), (pub)-[:PUBLISHED_IN]->(journal:Journal). Той самий кортеж, але у графовій візуалізації з веб-середовища Neo4j (рисунок 2.3).

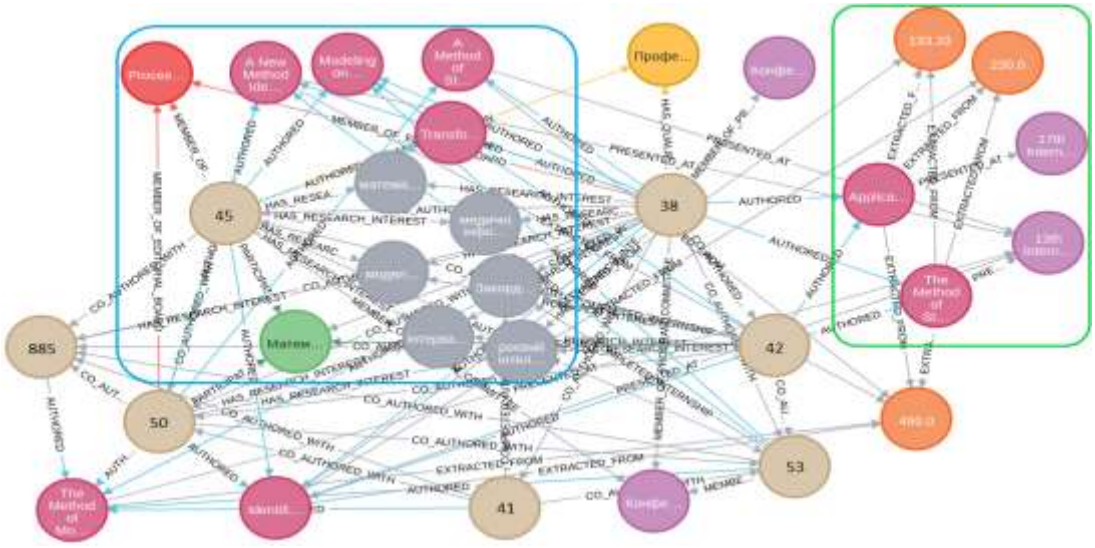


Рис. 2.3. Приклад графового представлення кортежу даних

З рисунка 2.3 видно, що пошук починався з користувача з id=45, для якого були знайдені подібні працівники (світло-коричневі) за інтересами/ключовими словами (сірі), публікаціями (темно-рожеві), журналами (червоні) та іншими зв’язками, та ця область обведена синім кольором. Подібні користувачі також мають певні зв’язки, а саме спільні діяльності, конференції (фіолетові), наукові проєкти (зелені), виконані критерії (оранжеві), співавторство тощо. Та завершується кортеж на області обведені зеленим кольором, де є подібні за тематикою публікації та журнали, які їх публікували. Таким чином, за допомогою

формування таких кортежів, можна взаємодіяти з побудованою моделлю представлення знань, отримувати бажану уточнюючу інформацію і використовувати її в наступних процесах. Поточний підхід, що поєднує MySQL для системи рейтингування та Neo4j для рекомендаційного модуля, дозволяє максимально використати переваги кожної СУБД, забезпечуючи оптимальну продуктивність та гнучкість для різних аспектів інтелектуалізованої системи рейтингування.

2.3 Комбінований метод ранжування НПП на основі комплексу метрик подібності

На початковому етапі цього комплексного процесу, необхідно визначити подібних НПП за допомогою різноманітних метрик подібності. Необхідність використання комбінації метрик полягає в тому, що вони дозволять ранжувати працівників, які демонструють подібні траєкторії розвитку або мають подібну структуру діяльності, але з кращими показниками. Тобто рекомендації враховують уподобання або модель поведінки подібних працівників, це співвідноситься з принципами колаборативної фільтрації. Окрім того, виявлення результативніших НПП, які досягли вищих показників у подібних сферах діяльності — дозволяє системі використовувати реальні активності працівників, як орієнтир для покращення. Таким чином, цей підхід забезпечить додаткову структуровану вхідну інформацію для подальшої обробки графовими нейронними мережами.

Розглянемо запропонований метод, який реалізується у декілька етапів, та спрямований на визначення працівників з максимально наближеними кращими показниками діяльності. Важливо, що цей метод адаптований для обробки часових рядів даних, що дозволяє оцінювати не лише поточний стан, але й прогрес працівників у часі та врахування семантики діяльностей. Загалом метод передбачає вісім ключових етапів, детальний розгляд яких наведено нижче.

1. Спочатку збираються та агрегуються дані щодо наукових досягнень кожного НПП, які представлені у їх звітах. Ці звіти містять інформацію,

категоризовану за різними групами активностей. Для аналізу динаміки діяльності НПП протягом визначеного звітного періоду застосовуються наступні метрики, представлені у вигляді часових рядів:

а) Вектор загального рейтингу користувача, який являє собою часовий ряд сумарних річних балів користувача *u*:

$$\overrightarrow{Stotal_u} = \{S_u^t, S_u^{t-1}, \dots, S_u^{t-n}\}, \tag{2.20}$$

де: S_u^t– загальний бал користувача *u* за рік *t*, який є номером найстаршого наявного року в послідовності років; а *n* – кількість наявних звітів користувача, без ураховування звітів, які старші за п'яти років.

б) Вектор балів за групами активностей користувача, це часовий ряд балів, агрегованих за кожною групою протягом наявних років:

$$\overrightarrow{Sga_u^t} = \{s_{g1,u}^t, s_{g2,u}^t, \dots, s_{g,u}^t\} \tag{2.21}$$

$$\overrightarrow{Sga_u} = \{s_{g1,u}^t, s_{g2,u}^t, \dots, s_{g,u}^t, \dots, s_{g1,u}^{t-1}, \dots, s_{g,u}^{t-n}\} \tag{2.22}$$

де: $s_{g,u}^t$ – бал користувача *u* за певну групу активності *g* в році *t*, а *g* – певна група активності, яких наразі є шість.

в) Середньорічний темп зростання активності, який відображає відносний темп розвитку активності користувача. Обчислення цієї метрики можливе за умови наявності мінімум двох заповнених річних звітів у користувача, в іншому разі цей показник не враховуватиметься:

$$Y_u = \frac{1}{\mathrm{T}-1} \Sigma_{t=2}^{T} \frac{(S_u^t - S_u^{t-1})}{S_u^{t-1}}, \tag{2.23}$$

де: S_u^t – загальний бал користувача *u* за рік *t*, а *T* – кількість років, за які користувач *u* має звіти. Слід зазначити, що недоцільно враховувати нульовий бал за попередній рік, коли звіт був відсутній (наприклад, для новоприбулих НПП або після перерви у діяльності). Такий підхід може некоректно вплинути на фінальні результати,

оскільки стартовий приріст балів у таких користувачів може бути високим. Низьке значення цього показника свідчить про низьку динаміку розвитку. Ця метрика дозволяє оцінити швидкість покращення результатів користувача протягом визначеного часу.

г) Показник наближеності двох користувачів за темпом динаміки активності, вимірює відносну відмінність у темпі зростання активності між двома користувачами:

$$Y_{u_1,u_2} = max\left(0, 1 - \frac{\left|Y_{u_1} - Y_{u_2}\right|}{max(Y_{u_1}, Y_{u_2}) + e}\right), \tag{2.24}$$

де: Y_{u_1}, Y_{u_2} – темп зростання за рік користувачів u_1 та u_2 відповідно, e – мале число, що запобігає діленню на нуль. Цей показник допомагає виявити користувачів, які розвиваються з подібною або відмінною швидкістю в діапазоні від 0 до 1, де 1 – найбільша схожість динаміки [1].

2. Для порівняння динаміки показників НПП у часі та виявлення шаблонів розвитку, що не фіксуються тільки статичними метриками, використано метрику динаміки часового зміщення (Dynamic Time Warping, DTW). Ця метрика дозволяє вимірювати подібність між двома часовими рядами (векторами показників користувачів), які можуть відрізнятися за швидкістю або фазою. Наприклад, на відміну від звичайної Евклідової відстані, яка чутлива до зсувів у часі та різної швидкості розвитку, DTW здатна знаходити оптимальне нелінійне зіставлення між двома послідовностями. Це дозволяє коректно порівнювати користувачів, активність яких, може бути різною або мати паузи.

Кожен користувач може бути представлений як багатовимірний часовий ряд, де елементи відображають їхні показники в певних групах активностей за послідовні звітні роки:

$$SGA_u = \left\{\overrightarrow{Sga_u^t}, \overrightarrow{Sga_u^{t-1}}, \dots, \overrightarrow{Sga_u^{t-n}}\right\}, \tag{2.25}$$

де: $\overrightarrow{Sga_u^t}$ – вектор балів за всі групи активностей користувача *u* за рік *t*; *n* – кількість звітів користувачів які обчислюються.

Формула для обрахунку DTW відстані між користувачами, враховуючи групи активності, визначається як:

$$DTWG_{u_1,u_2} = DTW\left(SGA_{u_1}, SGA_{u_2}\right) = min_W \sum_{k=1}^{K} d\left(\overrightarrow{Sga_{u_1,w_k}^t}, \overrightarrow{Sga_{u_2,w_k}^t}\right), \quad (2.26)$$

де: SGA_{u_1}, SGA_{u_2} – багатовимірні часові ряди показників двох користувачів u_1 та u_2; W=$(w_k)_{k=1}^K$ – шлях зміщення, що складається з *K* пар індексів показників, який мінімізує сумарну відстань між двома часовими рядами; $d\left(\overrightarrow{Sga_{u_1,w_k}^t}, \overrightarrow{Sga_{u_2,w_k}^t}\right)$ – локальна відстань між векторами показників користувача u_1 та користувача u_2 у момент w_k за рік *t*. Ця локальна відстань розраховується як Евклідова відстань між відповідними векторами балів за групи активності:

$$d\left(\overrightarrow{Sga_{u_1,w_k}^t}, \overrightarrow{Sga_{u_2,w_k}^t}\right) = \sqrt{\Sigma_{g=1}^{n}\left(s_{u_1,g}^t - s_{u_2,g}^t\right)^2}, \quad (2.27)$$

де: $s_{u_1,g}^t, s_{u_2,g}^t$ – бал користувача u_1 та u_2 за групу активності *g* за рік *t*, а *n* – кількість груп активностей.

Подібним чином, обчислюється DTW відстань, за загальним балом:

$$DTWS_{u_1,u_2} = DTW\left(S_{u_1}, S_{u_2}\right) = min_W \sum_{k=1}^{K} d\left(S_{u_1,w_k}^t, S_{u_2,w_k}^t\right), \quad (2.28)$$

де: S_{u_1}, S_{u_2}– часовий ряд загальних балів користувачів u_1, u_2 за всі звітні роки, які враховуються; $d\left(S_{u_1,w_k}^t, S_{u_2,w_k}^t\right)$ – локальна відстань між загальним балом користувача u_1 та користувача u_2 у рік *t*, що розраховується як абсолютна різниця $\left|S_{u_1}^t - S_{u_2}^t\right|$ річних балів.

В результаті обрахунку отримуємо число від 0 до необмеженого позитивного значення (залежить від величини відмінності). Низьке значення DTW вказує на

подібність шаблонів динаміки показників, навіть при наявності часових зсувів або різної інтенсивності активності, а велике — про суттєві відмінності у діяльності користувачів. Ці метрики дозволяють більш точно відображати подібність у динаміці професійного розвитку НПП, $DTWG_{u_1,u_2}$ – більш детально визначає подібність у професійному фокусі, а $DTWS_{u_1,u_2}$ – подібність за загальною успішністю і швидкістю розвитку, забезпечуючи глибший аналіз даних [1, 112].

3. Для виявлення подібності у загальних тенденціях або розподілі активностей між НПП, що не завжди очевидно за абсолютними значеннями, застосовується косинусна подібність, яка часто використовується у машинному навчанні, оскільки в багатовимірному просторі вона вимірює косинус кута між двома векторами. Таким чином, вона фокусується на напрямку векторів, а не на їхній довжині чи абсолютних величинах, що дозволяє оцінювати співвідношення між показниками [114]. Наприклад, коефіцієнт кореляції Пірсона підходить менше, оскільки він фокусується на лінійній залежності, що не зовсім відповідає діяльності НПП, та вимагає, щоб дані були числовими та нормально розподіленими. Коефіцієнт подібності Гауера можна розглядати більше як аналог, який працює з даними різних типів (числові, категоріальні), та усереднює їх. Незважаючи на те, що коефіцієнт Гауера краще працює з добре наповненими якісними ознаками (які, потребують завчасної підготовки та виділення), він частіше використовується для кластеризації та є менш придатною метрикою для поточної задачі. Це зумовлено його чутливістю до розрідженості даних, що може посилювати вплив відсутніх показників діяльності на подібність, а також вищою обчислювальною складністю порівняно з косинусною подібністю [114].

Формула для обрахунку косинусної подібності між векторами $\overrightarrow{Sga_{u_1}}$, $\overrightarrow{Sga_{u_2}}$ балів за групами активностей користувачів обчислюється так:

$$CosG_{u_1,u_2} = \frac{\left(\overrightarrow{Sga_{u_1}} \cdot \overrightarrow{Sga_{u_2}}\right)}{\left(\left\|\overrightarrow{Sga_{u_1}}\right\| \cdot \left\|\overrightarrow{Sga_{u_2}}\right\|\right)}, \tag{2.29}$$

де: $\overrightarrow{Sga_{u_1}}$, $\overrightarrow{Sga_{u_2}}$ – вектори балів за групами активностей користувачів u_1 та u_2 за роки, коли обоє проявляли активність. Використання даних за спільними роками, дозволяє зменшити розбіжності між “молодим” та “досвідченим” НПП, зосереджуючись на їхніх спільних періодах активності. Враховуючи різні розмірності між річними балами за групи активності, через виконувану діяльність користувачів, використання косинусної подібності залишається актуальнішим. Значення цього показника знаходиться в межах від 0 до 1, де чим вище значення, тим більш подібну діяльність ведуть користувачі. Розрахунок конкретно балів за критерії є недоцільним через те, що їх може бути кількісно більше, відповідно збільшена розмірність приведе до зменшення точності, потрібна буде додаткова нормалізація значень, з’являються шумові показники та необхідність встановлення нульових показників, що сильніше вплинуть на результати [1, 112, 114]. Косинусну відстань можна вважати подібною метрикою, яку краще не використовувати у випадках коли є чіткі метричні властивості, оскільки вона є неформальною метрикою. А кутова відстань між векторами є нормалізованою, формальною метрикою, але її використання більш обчислювально витратними [112].

4. Враховуючи, що різні користувачі виконують різні діяльності, порівняння їх діяльності потрібно буде здійснити з урахуванням спільних критеріїв та семантики. Тому використано індекс Жаккара, який традиційно застосовується для вимірювання подібності та різноманітності набору даних, шляхом ділення розміру перетину між ними на їх об’єднаний розмір [112].

Перед цим обрахунком, для кожного користувача формується список критеріїв, за якими вони отримали певні бали. Далі здійснюється сепарація спільних та унікальних критеріїв у звітному періоді для двох користувачів, що порівнюються.

Враховувати семантику критеріїв необхідно через специфіку формування звітів користувачів [1, 113]. Тому було додално матрицю SC, де $SC_{c_i c_j}$ представляє ступінь семантичної подібності між критерієм i та j, на основі їх векторів. Для отримання векторів, було обрано мультимовну модель SBERT, а саме “paraphrase-multilingual-mpnet-base-v2”, яка відрізняється високою швидкістю та якістю при

відносно невеликому розмірі. Перетворення текстового опису критерію на вектор ембеддингу здійснюється за формулою:

$$\overrightarrow{v_c} = SBERT(d_c), \tag{2.30}$$

де: c – це певний критерій (наприклад, [id:27; “Видання підручника, затвердженого вченою радою університету”]), d_c – текстовий опис критерію в системі (“Видання підручника, затвердженого вченою радою університету”), $\overrightarrow{v_c}$ – отриманий вектор опису критерію в числовому представленні у високорозмірному просторі. Він має фіксовану довжину (зазвичай 512–768 елементів) і кодує семантичне значення опису критерію таким чином, що семантично схожі критерії мають близькі вектори. Тепер визначемо подібність між ними, завдяки обчисленню косинусної подібності:

$$SC_{c_i c_j} = cos\left(\overrightarrow{v_{c_i}}, \overrightarrow{v_{c_j}}\right), \tag{2.31}$$

де: $\overrightarrow{v_{c_i}}, \overrightarrow{v_{c_j}}$ – це отримані від SBERT вектори критеріїв c_i та c_j, *cos()* – косинусна подібність [-1, 1].

Сформуємо масив попарно різних, семантично подібних критеріїв між двома користувачами, для зручності взаємодії з цими даними, $Csim_{u_1,u_2} = c_i \in C_{u_1}, c_j \in C_{u_2}$ при $SC_{c_i c_j} \geq \lambda_e, i \neq j$, де: λ_e – коефіцієнт подібності, який дорівнює 0.8 (значення було підібрано емпірично).

Тепер можемо описати модифікований індекс Жаккара, який розраховується як відношення зваженої суми балів за спільними та семантично подібними критеріями (чисельник) до зваженої суми балів за всіма унікальними для обох користувачів критеріями (знаменник):

$$J_{u_1,u_2} = \frac{\sum_{c \in C_{u_1} \cap C_{u_2}} min\left(s_{u_1 c},\ s_{u_2 c}\right) + \sum_{c' \in Csim_{u_1,u_2}} min\left(s_{u_1 c'},\ s_{u_2 c'}\right)}{\sum_{c \in C_{u_1} \cup C_{u_2}} max\left(s_{u_1 c},\ s_{u_2 c}\right) + \sum_{c' \in Csim_{u_1,u_2}} max\left(s_{u_1 c'},\ s_{u_2 c'}\right)}, \tag{2.32}$$

де: $s_{u_1 c}, s_{u_2 c}$ – бали користувачів u_1 та u_2 за певний критерій c; $C_{u_1} \cap C_{u_2}, C_{u_1} \cup C_{u_2}$ – множини спільних та унікальних критеріїв між користувачами; $Csim_{u_1,u_2}$ – масив

семантично подібних критеріїв c' між користувачами. Значення індексу Жаккара [0, 1], де більше значення показує більшу подібність за діяльністю [1, 112].

Щодо інших подібних показників, можна назвати коефіцієнт Дайса, який є іншою варіацією нормалізованого внутрішнього добутку, що вимірює подібність та різноманітність наборів шляхом ділення розміру перетину між наборами на їхній середній розмір, та порівняно з індексом Жаккара — не є метрикою. Коефіцієнт Сьоренсена часто використовується в екології, інформаційному пошуку, NLP, кластеризації та досить схожий на коефіцієнт Дайса. Натомість застосування наведеного модифікованого індексу Жаккара дозволяє зменшити вплив різниці у кількості критеріїв у множинах та здійснювати більш точне оцінювання подібності між НПП, виходячи за рамки простого збігу критеріїв та враховувати їх змістовне наповнення [112].

5. Для виявлення ступеня, наскільки один користувач перевершує іншого за спільними аспектами діяльності, використовується математична метрика різниці балів, яка буде обчислюватись на множині спільних критеріїв. Цей підхід дозволяє ідентифікувати, наскільки один користувач переважає над іншим за однаковими видами діяльності. Формула для обрахунку сумарної переваги балів за спільними критеріями визначається як:

$$P_{u_1,u_2} = max\left(0, \sum_{c \in C_{u_1} \cap C_{u_2}} \left(s_{u_2 c} - s_{u_1 c}\right)\right), \tag{2.33}$$

де: $\mathrm{C}_{u_1} \cap \mathrm{C}_{u_2}$ – множина спільних критеріїв між користувачами u_1 та u_2; $s_{u_2 c}, s_{u_1 c}$– бали користувачів відповідно за критерій c. Використання функції *max(0,…)* гарантує, що показник P_{u_1,u_2} відображає лише позитивну сумарну перевагу активностей користувача u_2 над u_1 у контексті спільних ознак. Для подальшої інтеграції в комплексний індекс подібності цей показник підлягає нормалізації, оскільки значення результату отримуємо в діапазоні від 0 до необмежено позитивного. Такий підхід корелює з концепцією Performance Difference та активно використовується в аналізі ефектів у задачах машинного навчання, та рекомендаційних системах. Наприклад, аналогічним є показник абсолютної

різниці, який вимірює загальну відмінність, але не вказує на напрямок переваги, а відсоткова різниця може оцінити зростання, хоча є нестабільною при малих значеннях у знаменнику і менш інтерпретованою у контексті сумарної переваги за багатьма критеріями [1].

6. Для інтеграції загальної ефективності потенційного рекомендованого користувача та визначення його пріоритетності введено фактор переваги F_{u_1,u_2}, який базується на порівнянні середньорічних балів користувачів і дозволяє динамічно регулювати вагу подібності залежно від загального рівня успішності рекомендованого НПП. Таким чином користувачі, які будуть мати менший фактор, отримуватимуть штраф до метрик подібності. Бінарне значення для цього було б менш інформативним, оскільки не враховувало б ступінь переваги або відставання. А відносна різниця дає відносне зростання, але вона не обмежується діапазоном і може призвести до негативних значень, що ускладнить її інтеграцію в загальний ваговий індекс. Тому формула для фактора переваги розраховуватиметься так:

$$F_{u_1,u_2} = min\left(1, \frac{Sa_{u_2}}{Sa_{u_1}}\right), \tag{2.34}$$

де: Sa_{u_1}, Sa_{u_2} – середньорічний бал користувачів u_1, u_2. За умови, що Sa_{u_2} менше за Sa_{u_1}, тоді фактор буде менше за 1, і користувач u_2 буде в меншому пріоритеті для рекомендації користувачу u_1, *min(1,…)* – якщо Sa_{u_2} більший або дорівнює Sa_{u_1}, фактор переваги залишається рівним 1, що підтримує нейтральний або позитивний вплив на подібність. Нам необхідно відфільтровувати користувачів, які мають середній бал нижчий ніж у поточного, подібні користувачі повинні мати кращі показники, тому даний фактор є більш гранульованим, дозволяючи враховувати вплив загальної ефективності [1].

7. Щоб звести, наведені вище, метрики в загальний індекс подібності, необхідно провести нормалізацію, оскільки різні показники подібності та переваги можуть мати різні діапазони значень та розмірності, їх пряме сумування не є коректним та може призвести до спотворення результатів. Після нормалізації всіх значень, вони будуть зведені до єдиного діапазону [0,100].

Процедура нормалізації включає дві основні формули:

$$\text{а) } X_{log} = log(x+1), \tag{2.35}$$

$$FN(x) = \frac{X_{log} - X_{log,min}}{X_{log,max} - X_{log,min}} * 100 \tag{2.36}$$

де: x – абсолютне значення показника (бал, різниця, подібність тощо); де $FN(x)$ – нормалізоване значення показника; X_{log} – логарифмічно перетворене значення; $X_{log,max}, X_{log,min}$ – максимальне і мінімальне логарифмічно перетворені значення на всій множині даних.

Логарифмічне перетворення (2.35) застосовується для показників, які можуть мати сильно асиметричний розподіл або великий діапазон значень (наприклад, бали або їх різниці). Це допомагає зменшити вплив викидів та зробити розподіл більш симетричним, що покращує стабільність подальшої нормалізації. Додавання одиниці до значення (x+1) дозволяє уникнути невизначеності при логарифмуванні нуля. Після логарифмічного перетворення значення масштабуються до бажаного діапазону з використанням лінійної нормалізації (2.36). Метрики DTWS і DTWG вказують на більшу подібність при менших значеннях, тому їх нормалізація передбачає додаткове обернення шкали [1].

8. Процес визначення подібних (але кращих) НПП завершується обрахунком загального індексу подібності $Sim_{u_1u_2}$. У машинному навчанні ансамблеві методи часто використовуються для об’єднання результатів кількох моделей або метрик з метою підвищення загальної точності та комбінації різних показників. Цей індекс є комплексним ваговим сумарним показником, що інтегрує всі метрики та фактори, обчислені на попередніх етапах, та розраховується за формулою:

$$\begin{aligned} Sim_{u_1u_2} = F_{u_1,u_2} \cdot \ (\lambda_1 \cdot FN\big(DTWS_{u_1,u_2}\big) + \lambda_2 \cdot \big(CosG_{u_1,u_2} \cdot 100\big) + \\ + \lambda_3 \cdot FN\big(DTWG_{u_1,u_2}\big) + \\ + \lambda_4 \cdot \big(J_{u_1,u_2} \cdot 100\big) + \ \lambda_5 \cdot \big(Y_{u_1,u_2} \cdot 100\big) + \lambda_6 \cdot FN\big(P_{u_1,u_2}\big)\big), \end{aligned} \tag{2.37}$$

де: $\lambda_1, \dots, \lambda_6$ – це вагові коефіцієнти, які в сумі дають 1, та емпірично було підібрано в розмірі = 0,2 для кожного, крім показника сумарної переваги і наближеності за темпом динаміки, вага яких дорівнює 0,1 (щоб подібні користувачі не були значно більш розвиненими ніж поточний), $FN()$ – функція нормалізації показника (2.35-2.36), для значень, які виходять за межі бажаного діапазону, а саме: DTW за загальним балом і групами активності, показник темпу динаміки та переваги [1, 112].

Загальний індекс подібності є важливим показником, який дозволяє проводити ранжування користувачів за ступенем їхньої подібності до поточного НПП. При цьому враховується його історичний прогрес та семантична подібність виконуваних діяльностей. Це дає можливість використати стратегії розвитку “більш успішних” НПП і підсилити роботу графової нейронної мережі. Чим вище значення цього індексу, тим вищий ранг рекомендованого користувача. Для прикладу, у таблиці 2.1 наведено дані для 10 найбільш подібних (з кращими показниками) користувачів до НПП з ID = 581, відсортованих за загальним індексом подібності.

Таблиця 2.1

Найбільш подібні користувачі, з кращими показниками діяльності, до користувача ID = 581 відсортовані за загальним індексом подібності

U ID	F_{u_1,u_2}	$DTWG_{u_1,u_2}$	$DTWS_{u_1,u_2}$	$CosG_{u_1,u_2}$	$J_{u_1,,u_2}$	Y_{u_1,u_2}	$P_{u_1,,u_2}$	$Sim_{u_1u_2}$
2	1	91.04	81.76	94	57.69	55.75	52.5	75.72
580	1	82.68	83.44	93.06	70	56.89	37.5	75.28
789	1	81.94	78.71	92.12	51.72	53.66	30	69.26
61	0.98	62.15	78	89.3	45.16	55.23	15	61.94

Продовження таблиці 2.1

855	1	66.55	65.01	94	48.39	44.33	25	61.72
300	1	54.46	61.83	92.22	45.16	42.16	27.5	57.68
563	1	52.79	55.98	89.3	51.61	39.76	20	55.91
720	1	61.9	48.39	86.48	33.08	26.17	52.5	53.84
57	0.95	45.11	65.67	81.78	45.16	44.77	15	53.52
117	0.99	74.17	41.05	82.24	48.48	27.99	14	51.99

Як видно з таблиці 2.1, підбір подібних користувачів здійснюється на основі метрик та формування персоналізованого рейтингу за подібною діяльністю, зокрема, за загальним індексом подібності.

2.4 Інтелекталізований метод формування персоналізованих рекомендацій

Тепер опишемо роботу з графовою нейронною мережею. Розроблений інтелектуалізований метод реалізовано у вигляді програмної моделі, що ґрунтується на гетерогенних графових трансформерів (Heterogeneous Graph Transformer, HGT) з механізмом часової уваги, що дозволяє враховувати часові портрети та зміни у діяльності НПП, їхній індивідуальний контекст, взаємозв'язки з іншими та семантику критеріїв оцінювання. Зазначений підхід базується на аналізі минулого досвіду, поведінки подібних, успішних колег та актуальних пріоритетів, забезпечуючи формування персоналізованих рекомендацій.

Для ефективного аналізу різних взаємозв'язків між НПП, критеріями їхньої діяльності та кафедрами, історичні дані системи рейтингування ЗУНУ структуруються у вигляді послідовності динамічних гетерогенних графів. Таке представлення дає можливість моделювати різнорідні сутності та їхні взаємодії, що краще розкриває приховані патерни активності та контекст діяльності.

Кожен часовий зріз даних, що відповідає певному звітному року t, формується як окремий гетерогенний граф $G_t=(V_t, E_t)$, де: V_t – множина вузлів, яка містить різні типи сутностей, в звітному році t, та включає такі типи вузлів ($u \in U_t$ – множина користувачів, $c \in C_t$ – множина критеріїв діяльності, $k \in K_t$ – множина кафедр); E_t – множина гетерогенних ребер, які встановлюються між сутностями "користувач – критерій", "користувач – подібний користувач", "користувач – кафедра" у рік t, та відображають різні типи взаємодій між вузлами.

Ребро між користувачем u і критерієм c $(u-c)$ додається, якщо користувач виконав критерій в певному році t. Вага ребра дорівнює отриманому балу, і зобразити це можна так:

$$e_{u,c}^{\mathrm{t}} = \begin{cases} s_{u,c}^{\mathrm{t}}, & \text{якщо } (u,c) \text{ виконано у році } t \\ 0, & \text{інакше} \end{cases}, \tag{2.38}$$

де: $s_{u,c}^{\mathrm{t}}$ – бал, отриманий користувачем *u* за виконаний критерій *c* у році *t*.

Ребро між користувачем u_1 та користувачем u_2 ($u_1 - u_2$) додається, якщо між ними існує подібність за діяльністю та u_2 має кращі показники (беремо топ 10 найбільш подібних користувачів за індексом подібності $Sim_{u_1 u_2}$ – який і буде виступати вагою ребра), це дозволяє орієнтуватись на сценарій діяльності більш “успішних” користувачів.

Ребро між користувачем *u* та кафедрою k (*u–k*) додається, якщо користувач належить до цієї кафедри в рік t та має бінарну вагу (вага = 1, якщо належить до кафедри) і вказує на факт приналежності.

Подібним чином додаються вузли для користувача: науковий ступінь (*at*) та вчене звання (*sd*). Також додано зворотні ребра (*c–u*, $u_2 - u_1$, *k–u*, *u–at/sd*) для вищенаведених, їх наявність забезпечує двостороннє поширення інформації, що сприяє кращому вивченню ембедингів вузлів (що є представленням об’єктів у вигляді векторів у просторі з низькою розмірністю) та зменшує проблему розрідженості даних.

Таке графове представлення дозволяє врахувати не лише індивідуальний вектор активності НПП, але й соціальний контекст, зокрема діяльність подібних працівників та наявні критерії, для формування більш точних та релевантних рекомендацій [130]. Візуалізація структури гетерогенного графа може бути представлена схематично на рисунку 2.4, де різні типи вузлів та ребра позначені різними кольорами або формами, ілюструючи їхні взаємозв’язки.

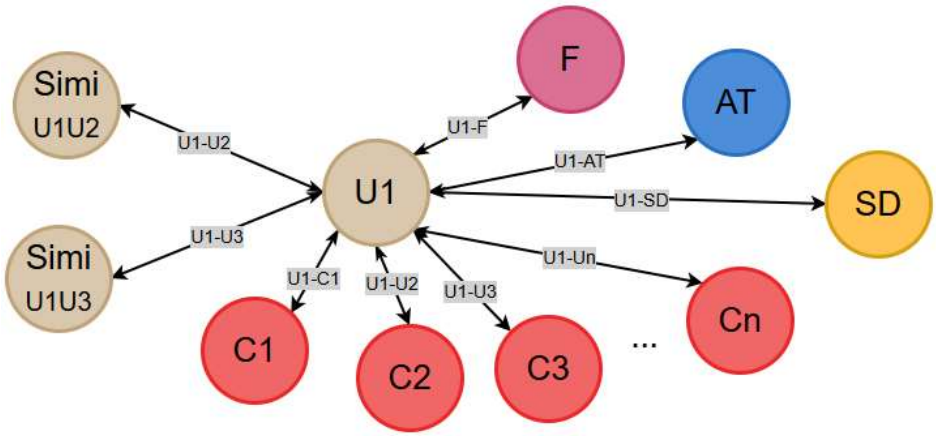


Рис. 2.4. Схема візуалізації гетерогенного графа з вузлами та ребрами

Для успішного навчання графової моделі кожен вузол графа потребує числового векторного представлення ознак. Ці ознаки формуються окремо для кожного типу вузлів, відображаючи їхні унікальні характеристики.

Для вузла “користувач” $u \in U_t$, де U_t – це множина користувачів у певному звітному році:

– Name – загальна ознака;
– AcademicTitle, ScientificDegree – науковий ступінь/вчене звання;
– ScientificInterests – коло наукових інтересів;
– MainWorkingSpecialty – основна робоча спеціальність;
– $CountC_{t(u)}$ – кількість виконаних критеріїв у рік t;
– $AvgC_{t(u)}$ – середній бал за критерій у рік t;
– $SumP_{t(u)}$ – загальна сума балів у рік t.

Для того щоб модель могла розрізняти річні звіти та зберігати їх коректний часовий порядок, використаємо темпоральне кодування (TE), як ознака вузла “користувач”. Це вектор, який в часовій послідовності річних звітів, вказує на позицію року в ній, та можна реалізувати на основі синусоїдального/косинусоїдального кодування [120]:

$$TE_{tp,I} = TE_f\left(\frac{tp}{10000^{\frac{I}{d_{tmp}}}}\right), \tag{2.39}$$

де: I – конкретний індекс розмірності в середині вектора позиційного кодування, який розраховується як $I = 2i = 2i+1$ для парних і непарних позицію елементів відповідно; tp – позиція певного року t в часовій послідовності, де 0 це найстаріший рік даних, які використовуються; TE_f – функції sin() або cos(), для парних і непарних елементів у векторі позиційного кодування; d_{tmp} – розмірність темпорального ембедингу. Таких підхід широко використовується в архітектурах трансформерів (і відповідно в GNN), для того щоб, у певній послідовності, враховувати порядок і позицію елементів. Значення 10000 є гіперпараметром, та було підібрано як стандартне значення для трансформерів, запропоноване у роботі Vaswani та ін. [133], його роль полягає в тому, щоб сформувати масштабуючий коефіцієнт для частот синусоїдальних/косинусоїдальних хвиль, чим більше значення — тим модель може краще визначати довгострокові залежності.

Ці значення ознак нормалізуються для стабільності навчання та формують вектор $x_{(u)}^t \in R^{d_u}$, де d_u – розмірність ознак користувача u. Надалі можна додавати інші ознаки користувача, які присутні зараз або з'являються в системі в майбутньому.

Для вузла "критерій" c $\in C_t$, де C_t – це множина критеріїв у певному звітному році:

– Title – загальні ознаки;

– Mandatory – бінарний показник обов'язковості виконання критерію;

– SbertC – вектор семантичного ембеддингу критерія, отриманого через Sentence-BERT, на основі його опису, що дозволить моделі краще розрізняти семантично подібні критерії між собою;

– DefaultPoints – загальний бал критерію (при ідеальному виконанні, який демонструє його важливість чи складність).

– ActGroup – one-hot кодування групи активності критерію.

Ознаки критеріїв об'єднуються у вектор $x_c^t \in R^{d_c}$, де d_c – розмірність ознак критерію.

Для вузлів “кафедра” $k \in K_t$, де K_t – це множина кафедр у певному звітному році. В даний момент, вузли кафедр будуть мати тільки ознаки: Name – назва кафедри. По аналогії, ознаки кафедр будуть об’єднані у вектор $x_k^t \in R^{d_k}$, де d_k – розмірність ознак.

Тепер, коли ознаки сформовані, відповідно до класичного підходу, що застосовується у ГГНМ, створені ознаки вузлів проектуються в прихований спільний простір розмірності, умовно назвемо d_h [102]. В формуванні цього простору, використовуються нелінійні шари (проекційні шари), які виконують уніфікацію представлення різних типів вузлів в графі, перед подачею їх до графових шарів:

$$X_v^t = ReLU\left(W_{p,v} x_v^t + b_v\right), \tag{2.40}$$

де: v – тип конкретного наявного вузла (наприклад: “користувач”, “критерій”, “кафедра”), $W_{p,v}, b_v$ – відповідно, матриця ваг та вектор зсуву проекційного шару, $ReLU$ – відома функція активації, яка, незважаючи на свій лінійний вигляд, широко використовується в нейронних мережах і може допомогти уникнути проблеми зникнення градієнта, який виникає для дуже великих або дуже малих значень. Проблемою ReLU є потенційна можливість мертвих нейронів, LeakyReLU вирішує цю проблему, оскільки він залишає невеликий ненульовий потік сигналу через активацію, що в деяких випадках є необхідним.

Для врахування часової динаміки змін у діяльності НПП та ефективного збору інформації у гетерогенному середовищі, створимо послідовність графів $G_{t=1}^T$, де T – індекс поточного звітного року, та кожен рік подається як окремий граф (портрет діяльності працівників в певному році) та використовуємо архітектуру графової нейронної мережі, що поєднує два рівні механізмів уваги: просторову та часову увагу. Цей підхід дозволяє моделі краще фокусуватися як на релевантних зв’язках у структурі графа, так і на часових послідовностях, для формування адаптивних рекомендацій.

На кожному часовому кроці t, для кожного вузла $v \in V_t$, модель формує просторове векторне представлення $h_v^t \in R^{d_h}$ шляхом агрегації ознак сусідів з

урахуванням їхньої важливості, тобто, ваг уваги. Цей процес реалізується за допомогою шарів HGTConv (Heterogeneous Graph Transformer Convolution).

Обчислюємо коефіцієнти уваги $\alpha_{v,v'}^{t}$ для кожного ребра (v, v'), який визначає важливість зв'язку між вузлами. Це досягається шляхом обчислення сумісності між трансформованими ознаками вузлів v та v', з урахуванням типу ребра:

$$\alpha_{v,v'}^{t} = \mathrm{Softmax}_{v}\left(\mathrm{LeakyReLU}\left(a_r{}^{\top}\left[W_r X_v^t || W_r X_{v'}^t\right]\right)\right), \tag{2.41}$$

де: a_r – транспонований вектор ваг уваги для скалярного продукту, для r типу ребра; W_r – матриця ваг для типу вузла, асоційована з типом ребра r; $\left[W_r X_v^t || W_r X_{v'}^t\right]$ – трансформовані конкатиновані вектори ознак вузлів v та v', після проекції у прихований простір; $LeakyReLU$ – активаційна функція; $Softmax_v$ (нормована експоненційна функція) – використовується для нормалізації ваг уваги, для всіх сусідів вузла u, щоб перетворити ваги в імовірнісні коефіцієнти.

Після обчислення коефіцієнтів уваги, просторове векторне представлення h_v^t формується шляхом агрегації ознак сусідів з урахуванням їхніх ваг уваги:

$$h_v^t = ReLU\left(\sum_{r \in R_a} \sum_{v \in N(v',r)} \alpha_{v,v'}^{t} \cdot W_{agr} W_r X_v^t\right), \tag{2.42}$$

де: h_v^t – векторне представлення ембеддингу вузла v у році t після агрегації; $N(v, r)$ – множина сусідніх вузлів до v у графі G_t з типом ребра r; $W_r X_v^t$ – трансформований вектор ознак вузла v у році t; W_{agr} – матриця ваг, що трансформує ознаки до простору агрегації; $\alpha_{v,v'}^{t} \in [0, 1]$ – коефіцієнт уваги між вузлом v та його сусідом v'.

Такий процес відбувається для кожного з шарів HGTConv у моделі, що дозволяє ембедингу вузлів акумулювати інформацію з багатошарового сусідства між вузлами в графі.

Враховуючи, що актуальність звіту з часом — втрачається, а рекомендації для покращення результативності діяльності НПП формуються на наступний рік, і деякі критерії або певна активність користувачів в конкретних групах чи

діяльностях може бути вже не актуальною або складною в реалізації на поточний момент, тому додамо фактор застарівання інформації (decay factor) [114].

Відповідно, після обчислення просторового ембедингу h_v^t для кожного року t, отримується послідовність векторів, що відображає річну динаміку діяльності вузла v. Для агрегації цих векторів у єдиний фінальний ембеддинг f_v застосовується комбінований механізм часової уваги, який поєднує контекстну важливість та хронологічне застарівання даних:

$$f_v = \sum_{t=1}^{T} \gamma_t \cdot \beta_t h_v^t, \tag{2.43}$$

де: T – індекс поточного звітного року в аналізованому періоді; $\beta_t \in [0,1]$ – коефіцієнт контекстної уваги, який модель обчислює самостійно за допомогою шару уваги та функції *Softmax*. Цей коефіцієнт відображає, наскільки рік t був значущим у діяльності вузла v з огляду на виявлені патерни (наприклад, аномальна активність чи зміна траєкторії розвитку); γ_t – фактор застарівання інформації з діапазоном значень [0.1,1], що розраховується як $\delta^{\mathrm{T}-t}$. Він примусово знижує вплив старіших даних, надаючи перевагу новішим звітам. Параметр $\delta \in (0,1)$ підібрано так, щоб вага найстарішого року становила 0.1, а найновішого – 1. Тому загальна формула буде $\delta = 0.1^{\frac{1}{T-1}}$. Наприклад, без фактору γ_t модель могла б надати високу вагу β_t звіту за 2021 рік через його унікальність чи аномальну активність, хоча для формування актуальних рекомендацій на 2025 рік дані за 2024 рік є значно релевантнішими. Поєднання обох коефіцієнтів дозволяє моделі враховувати як вивчену важливість кожного року, так і загальне припущення про більшу значущість свіжих даних, що робить прогнози більш адаптивними до поточних пріоритетів НПП [114].

Після підготовчих етапів агрегації даних, наступним завданням розробленої програмної моделі є прогнозування майбутньої активності НПП. Для цього в межах моделі визначено два ключових завдання, що вирішуються одночасно:

– прогнозування виконання критерію користувачем в наступному році;

– прогнозування очікуваного балу за виконання конкретного критерію в наступного році.

Ці задачі реалізовуються в межах моделі, яка навчається одночасно на обидві цілі, що дозволяє підвищити загальну ефективність прогнозування та персоналізації рекомендацій.

Для кожної пари (u,c), де: $u \in U$ – користувач; $c \in C$ – критерій діяльності; T – останній відомий рік з історичних даних, модель формує 2 прогнози: ймовірність виконати критерій та очікуваний бал за нього.

Ймовірність виконання критерію у наступному році T+1, можна описати такою формулою:

$$\hat{y}^{T+1}_{clf\ u,c} = \sigma\left(MLP_{clf}([f_u \| f_c])\right), \tag{2.44}$$

де: f_u,f_c – підсумковий вектор користувача u та критерію c, отриманий після просторової та часової агрегації; MLP_{clf} – багатошаровий перцептрон (нейронна мережа з нелінійністю), що відповідає за класифікацію; σ – сигмоїдна функція активації, яка переводить значення у діапазон [0, 1], та інтерпретується як ймовірність. Якщо отримане значення ближче до 1 – користувач з більшою ймовірністю виконає критерій, ближче до 0 – критерій малоймовірний або не притаманний для користувача, тому ймовірність виконання низька.

Очікуваний бал за виконання критерію (тільки для позитивних випадків) в наступному році:

$$\hat{y}^{T+1}_{reg\ u,c} = ReLU\left(MLP_{reg}([f_u \| f_c])\right), \tag{2.45}$$

де: $ReLU$ – функція max(0,x), яка обмежує прогнозоване значення до нуля, x – може бути будь-яким значенням (залежить від інших балів в графі в межах кожного критерію і йому подібних); MLP_{reg} – багатошаровий перцептрон (нейронна мережа з нелінійністю), що відповідає за регресію. Перед тим як передати значення в MLP_{reg} вони масштабуються за допомогою логарифмічної функції, а на виході

виконуємо обернене перетворення. В результаті обрахунку, значення може бути від 0 до максимально можливого (або отриманого) балу.

Кожен із двох компонентів (класифікатор і регресор), реалізується у вигляді окремої гілки багатошарового перцептрону, де вхідний вектор для них формується шляхом конкатенації векторів користувача та критерію.

Для ефективного навчання моделі графової нейронної мережі із зазначеними завданнями прогнозування, необхідно побудувати навчальну вибірку, яка буде містити як бінарну мітку (факт виконання критерію), так і реальне значення балу (за виконання критерію). Такий підхід дозволяє моделі не лише передбачати ймовірність виконання, але й оцінити якість та інтенсивність виконання через балову оцінку, що критично важливо в контексті рейтингування та планування діяльності користувача.

Кожна навчальна пара (u, c), де $u \in U$ – користувач, $c \in C$ – критерій діяльності, має форму: $(u,\ c,\ y_{u,c}, y_{p\ u,c})$, де: $y_{u,c} \in \{0, 1\}$ – бінарна мітка факту виконання критерію у році t; $y_{p\ u,c}$ – фактичний бал за виконання критерію c користувачем u.

Позитивні приклади формуються на основі реальних записів у звітних даних за рік t. Негативні приклади моделюють ситуації, коли користувач не виконував певний критерій. Оскільки такі зв'язки не фіксуються у звітності, їх створюватимемо штучно: для кожного користувача u, вибираються критерії c', які він не виконував у році t, але які виконувались подібними користувачами, де для кожного з них створюється пара: $y_{u,c'} = 0, y_{p\ u,c'} = 0$. Для кращого узгодження зі структурою графа використаємо негативні приклади серед критеріїв, які не виконувались подібними та поточним користувачами. Перевага такого стратегічного "семплінгу" полягає в тому, щоб отримати негативні приклади, базуючись на діяльності подібних. Тобто, щоб надавати моделі очевидні дії (наприклад, "користувач не публікував статті, хоча він керує багатьма науковими дослідженнями"), даємо їй релевантні приклади від подібних користувачів. Це змушує модель глибше вивчати тонкі відмінності та краще розуміти, чому певний користувач не виконав критерій, який, здавалося б, підходить йому, та підвищує

ефективність навчання і якість рекомендацій. Після побудови позитивних і негативних прикладів формується єдина навчальна вибірка.

Для відстеження та одночасного навчання обох задач, використаємо загальну функцію втрат, яка об'єднає втрати за обома завданнями:

$$L_t = L_{clf} + L_{reg_n}, , \tag{2.46}$$

де: L_{clf} – функція втрат для задачі класифікації (це бінарна крос-ентропія); L_{reg_n} – нормалізована функція втрат для задачі регресії.

Функція втрат для задачі класифікації:

$$L_{clf} = -\sum_{(u,c)}\left[y_{u,c} \cdot log\left(\hat{y}_{clf\ u,c}\right) + \left(1 - y_{u,c}\right) \cdot log\left(1 - \hat{y}_{clf\ u,c}\right)\right], \tag{2.47}$$

де: $y_{u,c} \in \{0, 1\}$ – бінарна мітка факту виконання критерію у році t; $\hat{y}_{clf\ u,c}$ – передбачена ймовірність виконання критерію.

Функція втрат для задачі регресії (використовується тільки для позитивних прикладів, оскільки для невиконаних критеріїв бал дорівнює 0):

$$L_{reg_n} = \sum_{(u,c)|y_{u,c}=1}\left(y_{p,norm\ u,c} - \hat{y}_{reg,norm\ u,c}\right)^2, \tag{2.48}$$

де: $y_{p,norm\ u,c} = y_{p\ u,c}/\mathrm{maxp}_c$ – фактичний бал за виконання критерію c, з нормалізовацією відносно максимального балу maxp_c, користувачем u; $\hat{y}_{reg,norm\ u,c}$ – передбачене значення балу за критерій, з урахуванням нормалізації, аналогічно до попереднього значення.

Під час навчання моделі, на основі поточних даних, які були розділені на тестові та валідаційні, було достатньо 40 епох щоб навчити модель. Продовження навчання моделі може призвести до її перенавчання та погіршення результатів. Графік значень втрат та якості моделі (Area Under the Curve, AUC) наведено на рисунку 2.5, та в таблиці 2.2.

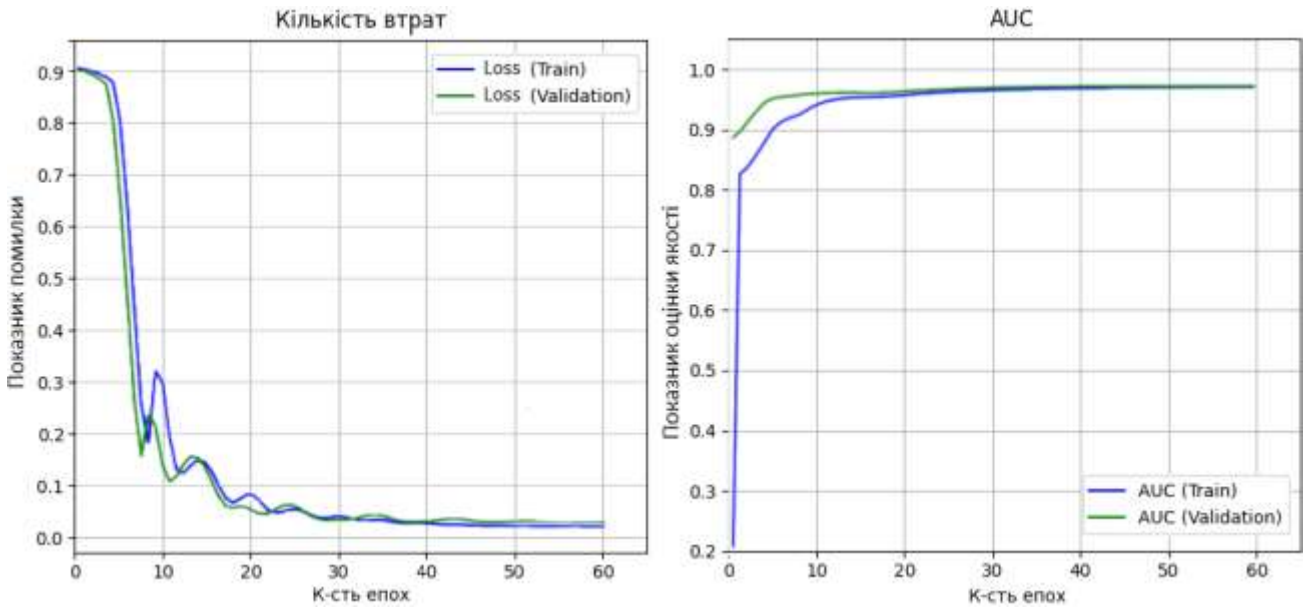


Рис. 2.5. Графік втрат та AUC під час навчання моделі в підході з графовою нейронною мережею

Таблиця 2.2

Значення втрат та AUC підчас навчання моделі для різних епох

Кількість епох	Втрати (Train)	Втрати (Validation)	ACU (Train)	ACU (Validation)
1	9.078	9.010	0.213	0.892
5	7.905	6.312	0.901	0.951
10	2.916	1.370	0.945	0.959
15	1.452	1. 259	0.955	0.960
20	0.882	0.511	0.958	0.963
25	0.519	0.592	0.963	0.966
30	0.444	0.407	0.964	0.970
35	0.323	0.489	0.965	0.971
40	0.284	0.291	0.967	0.972
45	0.242	0.287	0.970	0.973
50	0.224	0.282	0.973	0.973

Після завершення етапу навчання моделі, нейронна мережа формує прогнозований персоналізований список Top-N критеріїв для рекомендації користувачу, що включає найбільш релевантні, досяжні та результативні критерії.

Для всіх користувачів та критеріїв ($u \in U, c \in C$) модель генерує два прогнози ($\hat{y}^{T+1}_{clf\ u,c}$ та $\hat{y}^{T+1}_{reg,norm\ u,c}$) та формується матриця прогнозів, з якої фільтруються неактуальні критерії (ті критерії, які недоступні в наступному році), залишаються тільки множина актуальних критеріїв C^{T+1}_{valid}. Ранжування відбувається на основі комбінованої оцінки, яка враховує: прогнозовану ймовірність виконання критерію ($\hat{y}^{T+1}_{clf\ u,c}$), яка включає евристичне підсилення, на основі критеріїв, які успішно виконували працівники з максимально наближеними кращими показниками діяльності, а також потенційні “нові” критерії або “покращення” вже існуючих; прогнозований бал за виконання критерію ($\hat{y}^{T+1}_{reg,norm\ u,c}$), нормалізований відносно максимального балу за критерій.

Комбінований рейтинг для рекомендацій у вигляді формули можна зобразити так:

$$CR_{u,c} = \frac{\hat{y}^{T+1}_{clf\ u,c} + (\hat{y}^{T+1}_{reg,norm\ u,c})}{q}, \tag{2.49}$$

де: q – кількість параметрів які комбінуються, наразі вона включає два параметри.

Тепер, значення прогнозованого балу, в нормалізованому форматі, нам не потрібно, тому відновлюємо оригінальне значення, перемножуючи на максимальне значеня, та отримуємо $\hat{y}^{T+1}_{reg\ u,c}$. На основі відфільтрованих критеріїв, формується персоналізований список Top-N критеріїв (за комбінованим рейтингом) для користувача:

$$R_u = TopN_{c \in C^{T+1}_{valid}} \left(CR_{u,c}\right), \tag{2.50}$$

де: R_u – персоналізовані рекомендовані критерії для користувача *u* на рік *T+1*.

Візуально, процес формування рекомендацій ГГНМ зображено на рисунку 2.6.

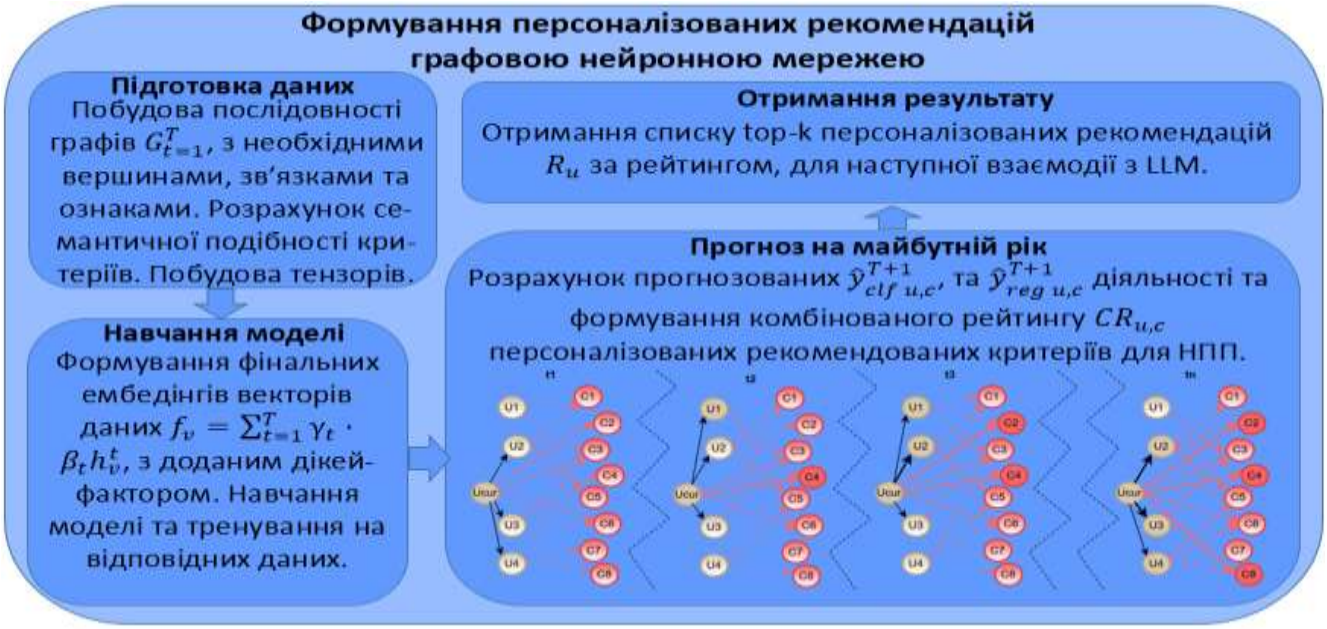


Рис. 2.6. Візуалізація формування персоналізованих рекомендацій графовою нейронною мережею

На попередньому етапі було отримано результат, сформований за допомогою програмної моделі, яка реалізована на основі графової нейронної мережі, ранжований список критеріїв з прогнозованими балами, однак він не містить інтерпретації, контексту чи рекомендаційної компоненти, необхідних для безпосереднього використання кінцевим користувачем. Тому ключовим завданням стає трансформація цих даних у зрозумілі, аргументовані та дієві поради. Саме на цьому етапі, наступним кроком роботи програмної моделі є залучення великих мовних моделей, які здатні створювати якісний, персоналізований текст рекомендацій.

Нещодавні досягнення в сфері LLM відкрили нові можливості для рекомендаційних систем. Замість того, щоб розглядати рекомендації виключно як засіб зіставлення або ранжування, LLM забезпечують більш гнучкий підхід, перетворюючи завдання рекомендацій на проблеми розуміння та генерації природної мови, як це зазначав Geng et al., у своїй публікації [116]. Це дозволяє рекомендаційним системам використовувати багаті можливості семантичного розуміння та генерації мовних моделей, одночасно забезпечуючи єдину структуру для кількох рекомендаційних завдань. LLM можуть виконувати численні ролі у

рекомендаційних системах, що описано в роботі Xu et al., [118]. Вони можуть виконувати інженерію ознак, шляхом генерації багатьох текстових характеристик, отримувати семантичні ембеддінги для елементів або користувачів для вирішення сценаріїв “холодного запуску”, діяти як прямі рекомендатори або виступати в ролі контролерів, що забезпечує більш інтерактивні та зрозумілі поради.

Ці можливості можна використовувати за допомогою ретельно побудованих дій у промптах або шляхом точного налаштування даних для рекомендацій, специфічних для даної предметної області. Ключовою перевагою рекомендаційних систем на основі мовних моделей є їхня здатність об’єднувати різноманітні типи інформації та завдань в рамках єдиної структури. Як продемонструвала система M6-Rec, яку описував Cui et al., у своїй роботі [119], модель якої може підтримувати кілька завдань рекомендацій, вирішуючи завдання розуміння задач, їх реалізації та генерації тексту. Це включає як класичні завдання, такі як пошук та ранжування, а також складніші аспекти, такі як генерація пояснень та діалогові рекомендації. Підхід до текстового представлення дозволяє системі обробляти дані про поведінку користувача, метадані елементів та взаємодію між ними в узгодженому форматі, уникаючи обмежень ідентифікаторних систем при роботі з раніше невідомими елементами [115-119].

Відтак, після етапу ранжування, система здійснює перехід до LLM для формування розгорнутих рекомендацій. Цей процес відбувається шляхом конструювання спеціалізованого запиту (промпту), який передає LLM всю необхідну контекстну інформацію. Конкретно для задачі формування тексту рекомендації користувачу на наступний рік, промпт можна описати у вигляді формули:

$$Prompt_{rec,u}^{T+1} = Template_{rec}\begin{pmatrix} UserMainChar_u^T, R_u, \\ UserContext_u^T, UserContext_u^{T-1}, SimiUserContext_u^T \end{pmatrix}, \quad (2.51)$$

де: $Template_{rec}$ – розроблений шаблон сценарію формування рекомендацій, що визначає загальну структуру та мету генерації тексту (наприклад, “Формування персоналізованих рекомендацій користувачу”, “Уточнення рекомендацій

користувача” тощо); $UserMainChar_u^T$ – включає загальні характеристики поточного користувача u, такі як його спеціальність, кафедра, наукові вподобання, інтереси та інші описові дані профілю; R_u – рекомендовані критерії для поточного користувача (з очікуваним балом за нього, отриманого від графової нейронної мережі); $UserContext_u^T, UserContext_u^{T-1}$ – відображає контекст діяльності користувача u за поточний та минулий роки, який включає повний звіт про виконані критерії, що дозволяє LLM глибше розуміти предметну область, його сильні та слабкі сторони, а також динаміку активності (хоча б за останні роки); $SimiUserContext_u^T$ – охоплює контекст діяльності подібних користувачів і використовується як приклад для LLM; T – поточний звітний рік. Для цього промпту додаються повні звіти двох найбільш подібних НПП (за загальним індексом подібності), що мають кращі показники. Процес побудови промпту передбачає інтеграцію цих даних у відповідні секції заздалегідь визначеного шаблону. Крім того, в промпті явно вказується інструкція для LLM, що охоплює такі аспекти:

– аналіз діяльності користувача та виявлення тенденцій розвитку в кожній групі активності та порівняння їх з показниками подібних;

– визначаються ключові завдання, які LLM має реалізувати, наприклад, щодо формування наданих даних і рекомендованих криеріїв у певний визначений результат тощо;

– генерація реалістичних та персоналізованих рекомендацій, включаючи певні вимоги до них, такі як варіативність, відповідність прикладам та ресурсам профілю користувача, реалістичність виконання з урахуванням минулої активності та часових обмежень тощо;

– деталізація рекомендацій за допомогою LLM, тобто формування аргументів доцільності діяльності, приклади успішного виконання подібними користувачами, чітке пояснення шляхів реалізації критерію (включаючи приклади статей, журналів, конференцій, грантових програм, веб-ресурсів, авторів тощо), а також потенційний приріст балів за виконання критерію;

– визначений, структурований формат результату, який LLM має генерувати, що включає текст рекомендації та JSON-об’єкт (recommendationDB) з певними полями. Це забезпечує можливість подальшої програмної обробки та зберігання інформації;

– обмеження та вимоги до реалізму, тобто врахування даних і динаміки поточного користувача, додаткові обмеження чи вимоги від ЗВО тощо. Особлива увага приділяється наданню конкретних прикладів та ресурсів, доступних для українських вчених.

Також, для забезпечення актуальнішої та достовірнішої інформації, яка надається у рекомендаціях, LLM інтегрована з інструментом пошуку (наприклад, Google Search). Це дозволяє моделі виходити за межі своїх внутрішніх знань, які можуть бути обмежені датою останнього оновлення її тренувальних даних. Шляхом використання такого інструменту пошуку від Google, та його включення в конфігукацію клієнта мовної моделі, LLM може динамічно отримувати більш нову інформацію для формування релевантних та своєчасних рекомендацій [129]. Візуалізація формування персоналізованих рекомендацій для користувачів з використанням LLM та пошуковим інструментом зображено на рисунку 2.7.

Рис. 2.7. Загальна візуалізація формування текстових рекомендацій для користувачів з використанням LLM

Зокрема, інтеграція з пошуковим інструментом дозволяє LLM:

– знаходити поточні конференції, відкриті грантові програми, інформацію про видавництва та журнали з їхніми актуальними квартилями, посилання на останні статті за певною тематикою або контактні дані ключових осіб та організацій;

– частково верифікувати інформацію, підтверджувати достовірність певних фактів або наявність можливостей, що дозволяє уникнути застарілих або неточних рекомендацій;

– збагачувати контекст, надаючи додаткові деталі та приклади, які можуть бути відсутні у знаннях моделі, але важливими для повноцінного розуміння та реалізації рекомендації користувачем.

Ця синергія між LLM та пошуковим інструментом є важливою для забезпечення того, що згенеровані рекомендації не лише відповідають індивідуальному профілю НПП та прогнозам графової моделі, але й стають більш точними, практичними та актуальними для поточної науково-освітньої та грантової діяльності в Україні та світі. Таким чином, система може формувати рекомендації, що містять не просто загальні фрази, а включають конкретні приклади, аргументи, ресурси, поради та посилання на джерела та діяльність інших успішних НПП, а також актуальну інформацію з пошукових систем. Такий підхід забезпечує кращу якість, релевантність та практичну цінність рекомендацій для кінцевого користувача, забезпечуючи імітацію написання тексту рекомендацій у ролі академічного працівника, завідувача кафедри тощо [115-119].

При формуванні користувачу тексту рекомендації, система надає 3 варіанти діяльності на вибір. Основна частка рекомендованих критеріїв, за типом та рейтингом, зберігається спільною для всіх, а інші розподіляються по окремим варіантах, дозволяючи надати користувачу можливість уточнити отримані результати. В результаті виконання промпту для формування рекомендацій $Prompt_{rec,u}^{T+1}$, отримаємо JSON модель з трьома варіантами тексту $Rec_{u,1}^{T+1}, Rec_{u,2}^{T+1}, Rec_{u,3}^{T+1}$, які зберігаємо в БД та виводимо в системі для працівника.

Також окремо зберігаємо критерії, які містяться в цьому тексті, для прикладу наведемо результат отримали від LLM у форматі json (рисунок 2.8).


```
"recommendationDB1": [
  {
    "uid": 241,
    "potential_score": 20,
    "details": "Членство у спеціалізованій вченій раді."
  },
  {
    "uid": 325,
    "potential_score": 100,
    "details": "Участь в освітньому міжнародному проекті."
  }
],
```

```
"recommendationDB2": [
  {
    "uid": 325,
    "potential_score": 100,
    "details": "Участь в освітньому міжнародному проекті."
  },
  {
    "uid": 305,
    "potential_score": 10,
    "details": "Підготовка одноосібної монографії українською мовою."
  },
  {
    "uid": 265,
    "potential_score": 15,
    "details": "Публікація одноосібної статті у вітчизняному фаховому
      виданні категорії "Б"."
  }
],
```

```
"recommendationDB3": [
  {
    "uid": 346,
    "potential_score": 10,
    "details": "Виконання обов'язків члена редакційної колегії зарубіжного
      наукового видання."
  },
  {
    "uid": 352,
    "potential_score": 150,
    "details": "Керівництво програмним комітетом конференції, що
      індексується Scopus/WoS."
  },
  {
    "uid": 350,
    "potential_score": 10,
    "details": "Рецензування дисертації на здобуття наукового ступеня
      доктора наук."
  }
]
```


Рис. 2.8. Приклад отриманих варіативних рекомендованих критеріїв для працівника

Тепер працівник може ознайомитись з основними рекомендаціями, обрати більш бажаний для нього варіант діяльності, та отримати уточнений текст рекомендацій, який також опрацьовується подібним чином як і в попередніх етапах, але з використанням моделі представлення знань та динамічним уточненням структури промпту. У вигляді формули це можна зобразити так:

$$Prompt_{recClar,u}^{T+1} = Template_{recClar}\left(Rec_{u,k}^{T+1}, ExpansionDataset_u\right), \quad (2.52)$$

де: $Template_{recClar}$ – розроблений шаблон сценарію уточнення рекомендацій, що приймає обрані рекомендації користувача, та додає додаткову інформацію; $Rec_{u,k}^{T+1}$ – обраний k варіант рекомендації для користувача u; $ExpansionDataset_u$ – дані із додатковими подібними критеріями та додатковою інформацією для рекомендації, які зібрані з моделі представлення знань, детальніше описаної в підрозділі 2.2, що дозволяє розширити корисну інформацію при формуванні тексту рекомендації за допомогою LLM; T – поточний звітний рік. Такий підхід дозволяє користувачу не зосерджуватись на самостійному аналізі області знань, не вишукувати описи профілів інших працівників для порівняння діяльності та не формувати чи

уточнювати промпти або запити самостійно (при використанні інструментів AI), а працювати конкретно з результатами сформованих, персонально для нього, варіантів рекомендацій, з можливістю їх уточнення. Якщо розглядати попередні приклади, то дані в $ExpansionDataset_u$ – отримуємо як структурований набір кортежів знань. Цей набір містить як і розширену інформацію конкретних, перевірених прикладів реалізації діяльності (журнали, проекти, співавтори тощо), так і розширення обраного варіанту діяльності, у вигляді додаткових подібних критеріїв. Наприклад, розглянемо кортеж знань, сформований для користувача u_1 (ID 45), який отримав варіант рекомендації щодо монографії, та публікації у фаховому виданні та участь у міжнародному проекті. Схематично, за структурою моделі знань, такий об'єднаний кортеж, що одночасно надає приклад і знаходить подібну діяльність, може виглядати так: (u1:User {id: 45})-[:PARTICIPATES_IN]->(project:ScientificProject {title: 'Математичне'})<-[:LEADS]-(u2:User {id: 38}), (u2)-[:AUTHORED]->(mono:MonographChapter)-[:PART_OF]->(book:Monograph {publisher: 'Springer'}), (mono)-[:HAS_KEYWORD]->(kw:Keyword {term: 'Міжнародна'}), (u2)-[:HEAD_OF_COMMITTEE]->(conf:Conference {name: 'ACIT'})-[:HAS_KEYWORD]->(kw). Тобто, спочатку знаходимо працівників з наближеними кращими показниками діяльності по наявним зв'язкам та вузлам, відносно них, обираємо реалізовані діяльності та їх вузли з описами, щоб продемонструвати приклад їх реалізацій. Подібним чином виконуємо пошук для подібних критеріїв, шукаючи спільні зв'язки через інші сутності які виконували подібні користувачі.

Крім того, система самостійно формує промпти до LLM, для опрацювання різних завдань, а користувач взаємодіє з результатами роботи системи з можливістю впливати на них. Замість ітеративного процесу, що вимагав би декількох послідовних звернень до мовної моделі для поступового збагачення тексту рекомендації або уточнень щодо певних діяльностей, система консолідує необхідну інформацію в один комплексний запит. Цей процес включає початковий текст рекомендації, дані, отримані на основі вибору користувача, та релевантну інформацію з моделі знань. Тому, промпти формуються динамічно і адаптовано до

конкретного працівника, що дозволяє генерувати розширену та деталізовану рекомендацію за одну операцію.

Ефективність такого підходу підтверджується сучасними дослідженнями у сфері взаємодії людини з LLM. Sofia Eleni Spatharioti та її колеги [132] у своєму дослідженні, експериментально довели, що використання LLM може значно підвищити продуктивність користувачів. Учасники дослідження, які використовували пошук на основі LLM, виконували завдання приблизно вдвічі швидше, ніж ті, хто користувався традиційними пошуковими системами. Основною причиною такого прискорення стало те, що користувачі формулювали меншу кількість запитів, але робили їх значно складнішими та комплексними. LLM-системи ефективно обробляли такі складні запити, самостійно синтезуючи інформацію, що позбавляло користувачів необхідності витрачати час на аналіз та об’єднання даних з різних джерел.

Таким чином, застосований в системі механізм динамічного уточнення промптів, підтримує перевагу застосування LLM, автоматизує процес збору та інтеграції даних в єдиний складний запит, який суттєво скорочує загальний час, необхідний для формування персоналізованої рекомендації. Відповідно мінімізується кількість звернень до моделі та перекладаємо завдання зі синтезу інформації на саму LLM, яка виконує це значно ефективніше за часом, ніж багатоетапна генерація або ручна обробка даних.

Для оцінки роботи системи та безпосередньо формування персоналізованих рекомендацій, було обрано тестову фокус-групу з 10 науково-педагогічних працівників. У вибірці охопилено працівників з різним рівнем продуктивності, високими та низькими річними рейтинговими балами. Для них було сформовано та представлено тексти рекомендацій на 2024 рік, на основі їх діяльності з 2018 по 2023 роки з можливістю уточнити свій вектор активності та збагатити обрані рекомендації даними з внутрішньої бази знань системи. Цей підхід дозволив не лише автоматизувати збір і структуризувати контекст даних, а ще підвищити деталізацію та практичну цінність сформованої рекомендації, але й скоротити час на ручне формування складних, багатокомпонентних промптів. Майже через рік,

коли відбувалося заповнення наступного річного звіту, було розглянуто їхню діяльність та співставили з рекомендованими критеріями. Результати цього аналізу візуалізовано на рисунках 2.9 та 2.10.

Рис. 2.9. Динаміка загального річного балу та вплив виконаних рекомендацій

На рисунку 2.9 представлено порівняльну динаміку загальних балів науковців за звітні роки. Стовпець за 2024 рік має двокомпонентну структуру: синім кольором позначено базовий бал, а зелений сегмент візуалізує приріст, отриманий безпосередньо за рахунок виконання тих критеріїв, що були рекомендовані системою. Такий розподіл дозволяє наочно оцінити прямий вплив рекомендацій на підвищення результативності працівника.

Рис. 2.10. Статистика виконання рекомендованих критеріїв тестовою групою

Рисунок 2.10 деталізує статистику взаємодії користувачів із запропонованими рекомендаціями. Світлим стовпцем позначено кількість рекомендованих критеріїв, які науковець намагався виконати, що свідчить про рівень сприйняття та відповідність рекомендацій системи до діяльності, яку веде працівник. Зеленим стовпцем відображено кількість критеріїв, за якими показник 2024 року перевищив або дорівнював прогнозованому показнику.

Отже, проведене тестування продемонструвало, що більшість користувачів з тестової групи спробували виконати приблизно половину рекомендацій, з яких більшу частину виконали на прогнозований бал або вище, що становить до 30% від усіх сформованих рекомендацій. Згідно з кількісними показниками, річні результати користувачів, в середньому, покращилися на величину, що перевищує 100 одиниць в абсолютному значенні. У відсотковому відношенні середнє покращення склало 27%, в порівнянні з попередніми показниками працівників. Можна зробити висновок, що запропонований підхід дозволяє не лише генерувати персоналізовані та релевантні рекомендації, але й позитивно впливати на наукову продуктивність співробітників, що підтверджується кількісними показниками.

Для оцінки ступеня відповідності рекомендованих критеріїв, що генеруються розробленою системою на основі графових нейронних мереж, до фактичної діяльності НПП, було проведено порівняльний аналіз з іншими наявними підходами. Враховуючи, що використання LLM, як доповнюючий етап для покращення персоналізованих рекомендацій, вирішення проблеми “холодного старту” та формування варіативності — можна застосувати до кожного з методів, тому порівняння проводитмемо на результатах сформованих списків рекомендованих критеріїв без етапу опрацювання мовними моделями. В цілому її задача полягає в додатковій фільтрації критеріїв та наповнення прикладами їх реалізації, особливо для користувачів з меншою активністю.

Відтак, для порівняльного аналізу та валідації запропонованої моделі були задіяні як класичні, так і гібридні рекомендаційні підходи, які використовуються в

подібних рекомендаційних системах в академічній сфері та були розглянуті в попередньому розділі.

Почнемо з підходу, який базується на контентно-орієнтованій фільтрації, яка більше зосереджуються на внутрішніх характеристиках самих критеріїв та персональному профілі діяльності користувача, який формується на основі його попередніх взаємодій з критеріями. Для його реалізації, було використано семантичні ембединги описів критеріїв, які отримаємо за допомогою мультимовної моделі SBERT (*bert_model.encode(*c_{desc}*)*), де c_{desc} – опис критерію c. Профіль користувача формується шляхом об'єднання ембедингів критеріїв, які користувач вже виконував раніше. Визначення релевантних рекомендацій діяльності для користувача здійснюється шляхом обчислення косинусної подібності між його профілем (вектором діяльності) та ембедингом потенційного критерію (вектор представлення опису критерія). Ранжування проводимо з урахуванням подібності опису критерію до користувача, та нормалізованого базового балу критерія, тому більш важливіші для ЗВО критерії будуть займати вище місце в списку рекомендованих. Для оцінки потенційного покращення балів за критеріями використано евристичну стратегію, яка ґрунтується на мінімальному покращенні з урахуванням результатів інших користувачів. Це покращення визначалося як найменший серед тих балів, які досягали інші користувачі за той самий критерій, що перевищували рівень поточного користувача. Серед мінусів підходу які можна сформувати на основі робіт інших дослідників, це акцент на діяльності поточного користувача та ігнорування сценаріїв інших, зосередженість на наявних подібних діяльностях, слабша пропозиція нового або нечітка відповідність рекомендованих критеріїв до реальної спроможності користувача виконати їх [89].

На противагу звичайного СВ, розлянемо ще матричну факторизацію, за якої припускається, що взаємодії між користувачами та елементами (критеріями) можуть бути пояснені невеликою кількістю латентних факторів. Таким чином, формуємо матрицю взаємодій "користувач-критерій" (умовно, Sm, де $Sm_{u,c}$ – бал користувача u за критерій c), яка через невелику кількість взаємодій та більш об'ємних можливих варіантів діяльності — вийде досить розрідженою. Потім

розкладаємо її на дві менші матриці низького рангу (матрицю користувачів Um та матрицю критеріїв Cm), що представляють приховані ознаки користувачів та критеріїв відповідно. Математично це мало б такий вигляд: $Sm \approx Um \cdot Cm^{\top}$, де Um – матриця латентних факторів користувачів, Cm – матриця латентних факторів критеріїв, а $Cm^{\top}$ – транспонована матриця Cm.

Технічно за основу можна взяти алгоритм SVD, зокрема його урізану версію *TruncatedSVD* з бібліотеки *sklearn.decomposition* на мові Python, який зможе виконати факторизацію матриці взаємодій, заповнюючи відсутні значення (а саме, прогнозуючи бали) на основі виявлених латентних факторів [100, 121].

Це класичний підхід, але для поточного дослідження було обрано його більш досконалу версію SVD++, реалізовану в бібліотеці *surprise*, оскільки вона додатково інтегрує дані про неявні зворотні зв'язки. Для підсилення цього методу було поєднано його з контентно-орієнтованим підходом, отримавши таким чином гібридну модель. Прогнозовані бали від кожної моделі ($s_{svdpp,c}$ від SVD++ та $s_{cb,c}$ від CB) зважуються рівномірно з коефіцієнтами $w_{svdpp} = w_{cb} = 0.5$, та формують єдиний комбінований бал (s_{com}), та у вигляді формули це можна зобразити так як $s_{com} = w_{svdpp} \cdot s_{svdpp,c} + w_{cb} \cdot s_{cb,c}$. Реалізація гібридного підходу здійснюється за допомогою об'єднання результатів цих методів та ранжування списку критеріїв по балу [85, 86, 100].

Для всебічної кількісної оцінки ефективності розробленої моделі GNN та її порівняння з аналогічними підходами було обрано низку метрик, які охоплюють різні аспекти якості рекомендацій та зазвичай застосовуються в рекомендаційних системах [119]. Враховуючи що формується певна кількість персоналізованих рекомендованих критеріїв зі списку, тому можемо використовувати як і 5, так і 10 критеріїв для рекомендації, тому можна використати підхід Top-K, та метрики точності та релевантності рекомендацій, які дозволяють оцінити якість перших k елементів у списку рекомендацій [97, 103, 119]:

– Precision@k, метрика яка визначає частку критеріїв, які реалізував або спробував реалізувати користувач, серед k перших рекомендованих критеріїв (R_u).

Цей показник вкаже нам, наскільки підбір рекомендацій відповідає діяльності користувача;

– Recall@k, який подібний до Precision, тільки обчислює відносно усіх фактично релевантних критеріїв які має користувач у звітний період, тобто, враховуємо відношення реалізованих рекомендації, відносно усієї виконаної діяльності користувача;

– Hit Rate@k, простий показник, який вкаже на факт наявності хоча б однієї рекомендації в наступному звітньому році у користувача, тобто чи була хоча б одна релевантна рекомендація;

– Coverage@k, використовуємо для визначення охоплення рекомендацій, відносно всієї доступної множини критеріїв, а саме відсоток унікальних критеріїв, які були рекомендовані системою в перших k позиціях. Високе значення цієї метрики свідчить про здатність системи надавати більш різнопланові рекомендації та не зосереджуватись на стандартних діяльностях.

Враховуючи загальну кількість критеріїв в системі, та кількісні середні показники їх виконання користувачами, для тестування, максимальне значення k буде дорівнювати 10. Враховуючи, наявні дані за минулі роки, можна сформувати рекомендації на поточний звітний рік і співставити їх з реальними даними.

Ступінь відповідності рекомендованих критеріїв фактичній діяльності працівників можна розрахувати на даних, за минулі роки, де можна сформувати рекомендації на наступний звітний рік і співставити їх з реальними даними. Вищенаведені метрики були розраховані як для існуючих аналогічних підходів, так і для запропонованого методу, результати наведено на рисунках 2.11-2.12. Для тестування використано дані діяльності НПП за 2024 рік включно. Розрахунки метрик точності та релевантності результатів, були проведені на основі сформованих рекомендацій на 2024 рік, а також порівняні з фактичними даними за цей же рік.

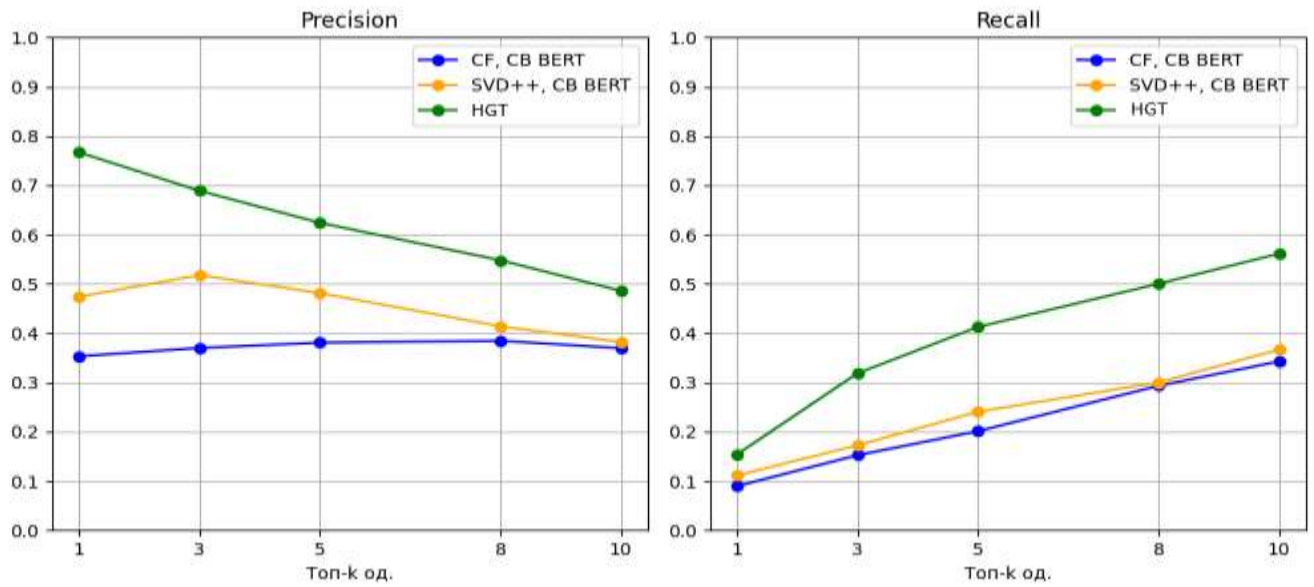


Рис. 2.11. Графік показників метрик Precision, Recall при різних Top-K рекомендацій для різних підходів

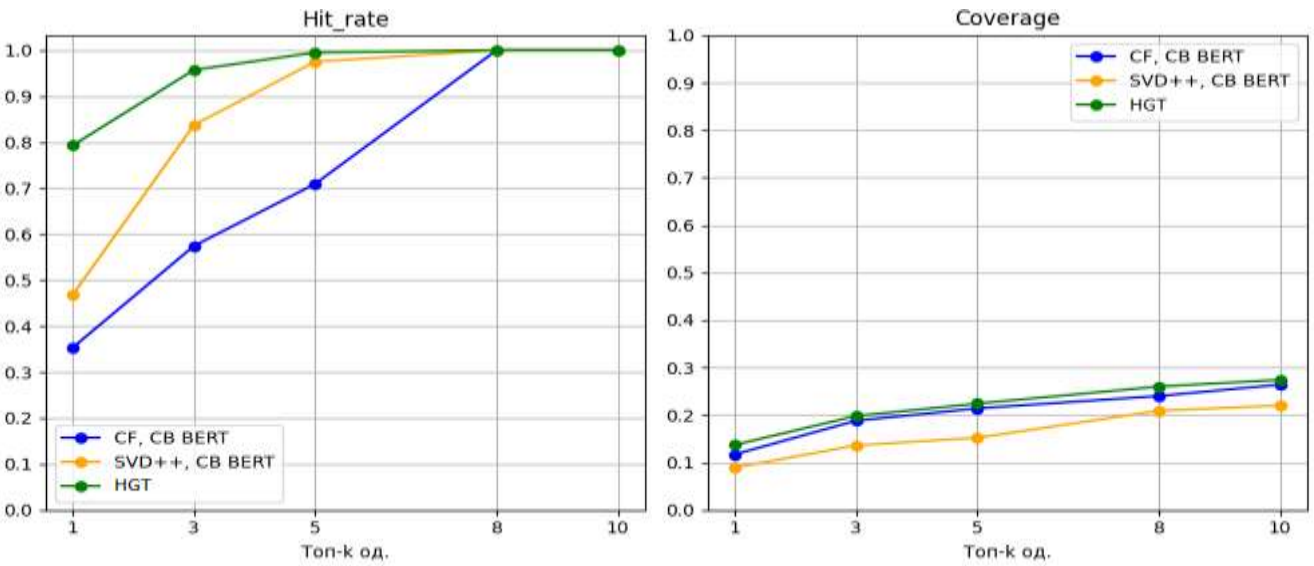


Рис. 2.12. Графік показників метрик HitRate, Coverage при різних Top-K рекомендацій для різних підходів

Показники порівняння різних підходів, за вищеописаними метриками наведено в таблиці 2.3 (Precision@k, Recall@k) та таблиці 2.4 (Hit Rate@k, Coverage@k).

Проаналізувавши наведені результати різних методів, можна зробити висновок, що запропонований підхід з використанням ГГНМ, з підсиленням від

Таблиця 2.3

Таблиця порівняння показників Precision@k та Recall@k для різних підходів

k	Precision			Recall		
	CB	SVD++	HGT	CB	SVD++	HGT
1	0.353	0.473	0.767	0.089	0.111	0.154
3	0.369	0.517	0.689	0.152	0.172	0.314
5	0.380	0.481	0.624	0.201	0.240	0.410
8	0.385	0.413	0.548	0.293	0.299	0.500
10	0.369	0.381	0.485	0.342	0.366	0.561

Таблиця 2.4

Таблиця порівняння показників Hit Rate@k та Coverage@k для різних підходів

k	Hit Rate			Coverage		
	CB	SVD++	HGT	CB	SVD++	HGT
1	0.353	0.468	0.793	0.117	0.089	0.137
3	0.574	0.837	0.957	0.188	0.136	0.198
5	0.709	0.975	0.995	0.214	0.152	0.224
8	1	1	1	0.240	0.209	0.260
10	1	1	1	0.264	0.220	0.274

більш результативних НПП та врахування семантики критеріїв, демонструє кращу точність визначення релевантних рекомендацій до поточної динаміки діяльності НПП, які виконують користувачі в наступному звітному періоді.

Висновки до розділу 2

1. Розроблено інтелектуалізований метод формування персоналізованих рекомендацій для підвищення результативності діяльності НПП, що ґрунтується на комбінованому підході, що включає збір та аналіз даних, застосування метрик подібності, генерацію рекомендацій за допомогою ГГНМ, а також формування текстових рекомендацій із залученням великих мовних моделей з динамічним уточненням структури промптів.

2. Розроблено модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі, що дало змогу

реалізувати процедури генерації персоналізованих рекомендацій для підвищення результативності науково-педагогічної діяльності.

3. Обґрунтовано використання комбінованого методу ранжування НПП на основі комплексу метрик подібності, таких як часове зміщення динаміки, косинусна подібність, індекс Жаккара та інші, з урахуванням семантики діяльностей. Цей підхід дозволяє автоматизовано визначати НПП з максимально наближеними кращими показниками діяльності, що допомагає при формуванні рекомендацій. Для врахування семантичного наповнення діяльності було інтегровано аналіз критеріїв за допомогою мультимовної моделі SBERT.

4. Запропоновано та обґрунтовано використання архітектури ГГНМ з механізмами просторової та часової уваги для прогнозування майбутньої активності НПП. Таке представлення даних у вигляді послідовності динамічних графів дозволяє моделювати різнорідні сутності (НПП, критерії, кафедри) разом з їхніми взаємозв’язками та врахуванням застарівання інформації через введений фактор.

5. Визначення прогнозованих показників діяльності НПП реалізовано як двозадачна модель, що одночасно вирішує задачі класифікації (прогнозування ймовірності виконання критерію) та регресії (прогнозування очікуваного балу). Комбінований рейтинг для рекомендацій враховує прогнозовану ймовірність та очікуваний бал. Навчання графової моделі здійснюється на основі позитивних та негативних прикладів, що покращує цей процес і якість рекомендацій.

6. В рамках даної системи для формування розгорнутих, аргументованих та актуальних рекомендацій запропоновано використовувати великі мовні моделі, завдяки створеним промптам, що включають контекст діяльності користувача, його успішних колег та результати роботи графової нейронної мережі. Крім того, інтеграція LLM з інструментом пошуку даних у мережі, дозволяє надавати актуальнішу інформацію, а уточнення рекомендації від працівника дозволяє розширити результат даними з побудованої моделі представлення знань. Процес динамічного уточнення структури промпту забезпечує скорочення часу їх формування та отримання більш релевантних і бажаних персоналізованих

текстових рекомендацій. Учасники тестової групи спробували виконати приблизно половину рекомендованих критеріїв, з яких більшу частину виконали на прогнозований бал або вище, що складає майже 30% від усіх рекомендованих їм діяльностей. Це сприяло отриманню ними вищих показників порівняно з попереднім роком в середньому на 100 одиниць, а у відсотковому відношенні, середній приріст склав 27% порівняно з попередніми результатами. Також, проведено порівняльний аналіз із класичними та гібридними підходами, який продемонстрував, що запропонований метод на основі графової нейронної мережі демонструє вищу точність у прогнозуванні критеріїв, які науково-педагогічні працівники фактично виконують у наступному звітному періоді, що підтверджується вищими показниками метрик Precision та Recall.

РОЗДІЛ 3
АРХІТЕКТУРА ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

Цей розділ присвячено архітектурі інтелектуалізованої програмної системи рейтингування діяльності НПП, а також обґрунтуванню й опису її структурних компонентів, що забезпечують підвищення результативності користувачів.

В розділі представлено опис загальної архітектури, яка базується на модульній та сервісно-орієнтованій архітектурі, що об’єднує в собі декілька ключових функціональних компонентів. Сформована архітектура інтелектуалізованої системи дозволяє досягти синергетичного ефекту, де кожен компонент, виконуючи свої специфічні завдання, взаємодіє з іншими для досягнення спільної мети. Далі розглянуто три основні компоненти: рейтингова система, модулі персоналізованих рекомендацій та збору даних.

Окрім цього, в розділі наведено особливості проєктування системи зберігання даних, обґрунтовано вибір технологій СУБД та описано їх використання для різних аспектів системи. Зокрема, продемонстровано, як поєднується реляційна та графова база даних, що дозволяє ефективно обробляти частково структуровані дані системи рейтингування та різнотипні взаємозв’язки, необхідні для формування рекомендацій.

Також описано особливості проєктування ключових модулів, що забезпечують інтелектуалізований функціонал: модуль формування персоналізованих рекомендацій, модуль підтримки оцінювання відповідності НПП ліцензійним вимогам та модуль збору даних. Наприкінці розділу представлено діаграми, що ілюструють взаємодію між основними компонентами системи, життєвий цикл звіту, ролі користувачів та інші аспекти програмної реалізації.

Результати цього розділу опубліковано автором у таких працях [2, 3, 6, 8, 10, 11, 12].

3.1. Загальна архітектура системи

Розглянемо основні аспекти розробленої системи, яка базується на модульній архітектурі, де окремі компоненти реалізовано як незалежні сервіси, що відповідає принципам сервіс-орієнтованого підходу та частково мікросервісної архітектури і дозволяє об’єднати декілька ключових функціональних компонентів [2]. Для такої архітектури притаманна: відокремленість функціоналу, взаємодії через API, незалежність компонентів та загальна гнучкість системи. Основна ідея полягає у створенні цілісного інформаційного середовища, що автоматизує процеси збору, обробки, аналізу даних, рейтингування, генерації персоналізованих рекомендацій та звітності про діяльність НПП.

Загальна архітектура інтелектуалізованої системи, представлена на рисунку 3.1, включає ключові модулі, які забезпечують збір, обробку, аналіз даних, надання рекомендацій та оцінювання відповідності НПП ліцензійним вимогам. Фундаментом системи є наявна система рейтингування діяльності НПП в ЗУНУ, яка взаємодіє з власною базою даних для зберігання системних даних, звітів, налаштувань, даних користувачів тощо.

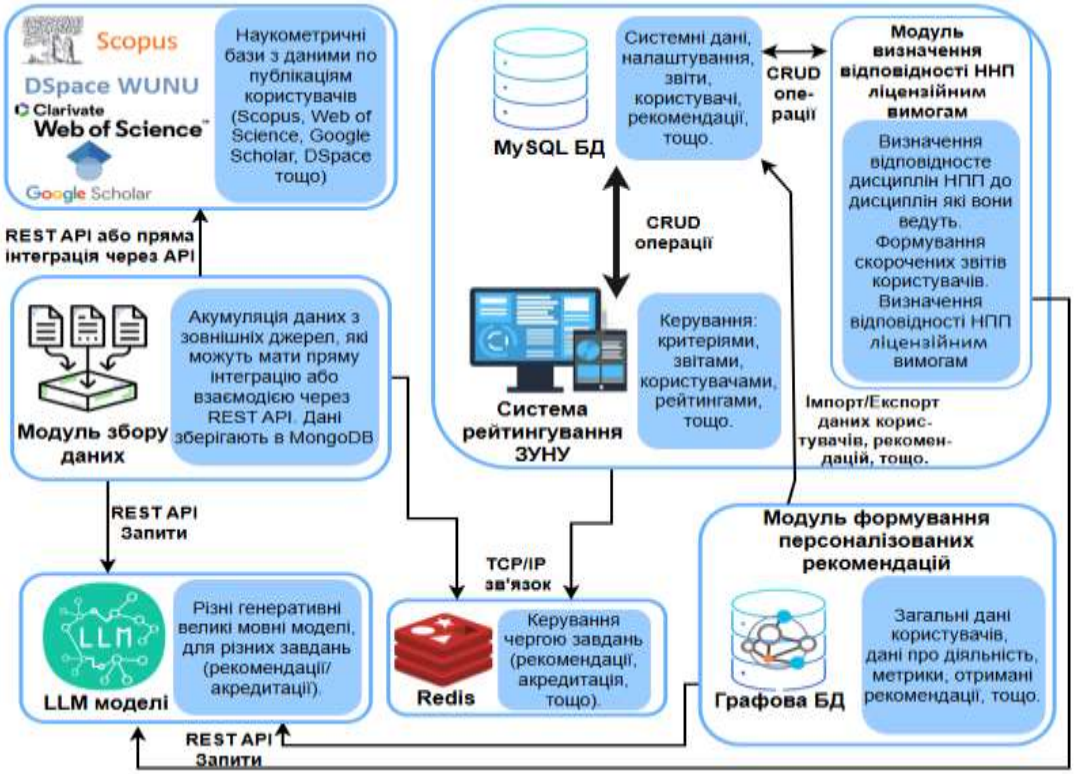


Рис. 3.1. Загальна архітектура системи рейтингування з ключовими модулями

Принцип побудови системи обумовлений необхідністю обробки переважно текстових даних та забезпечення високої продуктивності. Розподіленість системи досягається завдяки винесенню окремих функціональних блоків у незалежні модулі, які взаємодіють між собою за допомогою API, спеціальних компонентів або з через проміжні скрипти. Наприклад, модуль збору даних та модуль LLM функціонують як окремі сервіси, що забезпечує гнучкість та можливість паралельної обробки даних. Процес формування рекомендацій не впливає на роботу системи рейтингування, що дозволяє уникнути додаткового навантаження на її базу даних. Графова база даних, яка використовується для роботи з даними для формування персональних рекомендацій та необхідними для цього обчисленнями, відокремлена від рейтингової, що дозволяє швидше обробляти суттєві обсяги даних та використовувати більше інструментів і бібліотек для аналізу. Безпека системи забезпечується через використання токенів для взаємодії між модулями та API з ключами доступу для LLM, а також реалізацією ролей користувачів (“Адміністратор”, “НПП”, “Зав. кафедри”, “Рецензент”) з чітко розмежованим доступом до функціоналу [1, 2, 3, 4, 5, 11].

Архітектуру системи можна поділити на кілька ключових модулів, кожен з яких відповідає за певний аспект функціонування (рисунок 3.2).

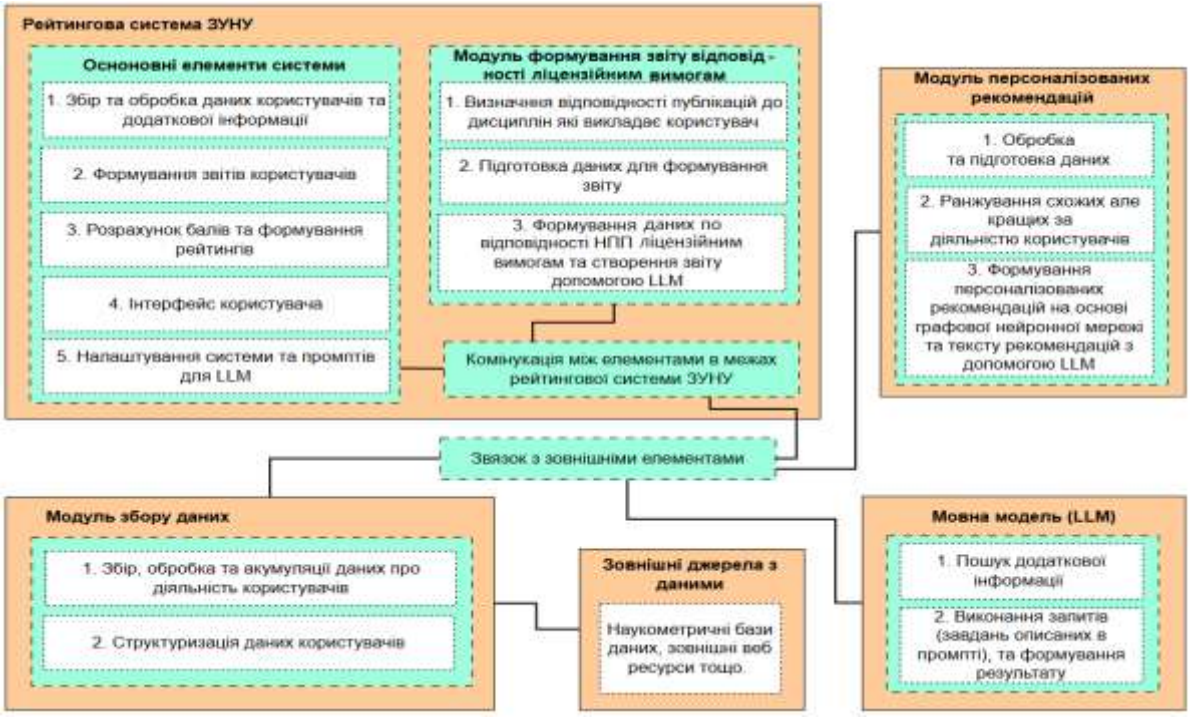


Рис. 3.2. Функції системи з основними її модулями

1. Система рейтингування діяльності НПП є базовою, її було розширено та інтелектуалізовано, і вона відповідає за основний функціонал. Вона включає такі основні елементи:

– Збір та обробка даних, які стосуються профілю користувачів, кафедр, факультетів тощо, та акумуляція даних із зовнішніх джерел.

– Формування звітів користувачів відповідає за щорічне заповнення звітів, автоматичне дозаповнення даними з профілю користувача та зовнішніх джерел з використанням LLM [2, 5].

– Розрахунок балів та формування рейтингів, відповідає за життєвий цикл звіту, починаючи з його заповнення користувачем та завершуючи затвердженням, розрахунком загального річного балу і формуванням рейтингу користувачів. Також є можливість переглядати рейтинг за певний рік, за останні 5 років із застосуванням коефіцієнтів.

– Інтерфейс користувача відповідає за взаємодію користувачів з системою, надаючи зручний для використання інтерфейс. Кожна роль користувача в системі має доступ до відповідного функціоналу, наприклад, НПП може заповнювати звіти, керувати профілем, своїми дисциплінами тощо.

– Налаштування системи та промптів для LLM. Дозволяє адміністратору керувати системою, редагувати промпти, які використовуються для обробки даних мовними моделями для різних функціональностей. Це дозволяє гнучко налаштовувати результати цих операцій, вносити додаткові обмеження, умови, правила, специфічні конструкції тощо.

2. Модуль персоналізованих рекомендацій для досягнення покращення результативності діяльності НПП [1]. Він включає:

– Обробку та підготовку даних для аналізу, обрахунків та формування рекомендацій.

– Ранжування користувачів з максимально наближеними кращими показниками діяльності. Використовуючи метрики подібності (DTW, Cosine Similarity, Jaccard Index та інші) із врахуванням семантики, що дозволяє

використовувати траєкторії їхньої діяльності як орієнтир для наступних процесів [1].

– Формування персоналізованих рекомендацій та тексту рекомендацій за допомогою LLM. Як було розглянуто раніше, цей модуль використовує графові нейронні мережі (HGT з механізмами просторової та часової уваги, фактором застарівання інформації, евристичним підсиленням від користувачів з наближеними кращими показниками діяльності, прогнозуванням виконання критерію та потенційного балу) для підсилення зв’язків з ідентифікованими НПП з подібними характеристиками, але вищими показниками результативності. На основі аналізу профілів користувачів користувачів, їх взаємодію з критеріями — графова нейронна мережа формує список критеріїв які можна порекомендувати виконати користувачам наступного року, щоб покращити їх показники, а LLM генерує деталізовані рекомендації з конкретними прикладами, аргументами, порадами та посиланнями на корисні ресурси у вигляду тексту. Цей модуль також може використовувати Google Search для отримання більш актуальних даних з мережі і, відповідно, покращувати рекомендації [1].

3. Модуль визначення відповідності НПП ліцензійним вимогам, який допомагає визначити, чи користувачі відповідатимуть вимогам до освітньої діяльності в наступному році та сформувати звіт і поради для відповідності вимогам. Він складається з таких етапів та враховує дані за останні 5 років:

– визначення відповідності публікацій дисциплінам, які викладає користувач. Необхідність цієї функції зумовлена тим, що наукові праці НПП мають тематично відповідати його викладацькій діяльності. Для автоматизації цього процесу застосовано LLM, як ефективний інструмент для вирішення подібних завдань. Взаємодія з LLM через спеціалізовані промпти дає змогу надійно виконувати зіставлення тематики публікацій та змісту дисциплін в автоматичному режимі;

– підготовка даних для формування звіту про відповідність НПП ліцензійним вимогам. Скорочується звіт користувача, щоб отримати тільки

конкретні дані, які відповідають критеріям в ліцензійних умовах, без зайвої інформації, використовуючи для цього LLM та створені промпти;

– формування даних про відповідність НПП до вимог та формування звіту про це за допомогою LLM. Після цього підготовлені звіти передаються в інший промпт який призначенний для визначення повної відповідності до критеріїв ліцензійних умов. В результаті аналізу, мовна модель повертає в заданому форматі статус користувача, щодо відповідності ліцензійним вимогам, зі статистикою та порадами. До цього промпту додаються річні персональні рекомендації користувача, сформовані раніше рекомендаційним модулем.

Щодо зовнішніх елементів, то взаємодія з ними, забезпечує повноту та актуальність даних для формування звітів користувачів та виконує низку важливих для системи завдань:

– модуль збору даних, який взаємодіє з наукометричними базами (Scopus, Web of Science, Google Scholar, DSpace) через API інтеграції, а також парсить веб-сторінки з доменів інших джерел, які додані в “while list”. Це перевірені домени, що дозволяють не валідувати двні рецензентами, а позначати їх як «перевірені». Отриманий текст із веб-сторінок, парситься, очищується від зайвих тегів, елементів тощо, та передається для аналізу та структуризації до LLM [2, 5, 8]. Результати роботи LLM формують списки відповідних даних (в даному випадку, списки публікацій конкретного користувача) та зберігаються в внутрішню базу даних MongoDB. Використовується мажоритарний спосіб перевірки валідності даних, тобто якщо дані кількісно підтверджується у більшій половині джерел, система позначає їх як актуальні. Після акумуляції даних по публікаціям користувачів, вони комбінуються і очищуються від дублікатів записів. Redis використовується для реалізації безперервного виконання збору даних для користувачів у вигляді черг. Це дозволяє ефективно обробляти асинхронні процеси та знижувати навантаження на основну систему. Взаємодія відбувається через TCP/IP зв’язок;

– великі мовні моделі використовуються безпосередньо для опрацювання текстових даних для різних сценаріїв. Наприклад, під час формування тексту рекомендацій для користувачів за умовами та правилами, які описуються в

промпті, з використанням інструменту Google Search, який дозволяє моделі отримувати актуальну інформацію з мережі під час виконання запитів мовною меделлю, використовуючи клас Client з бібліотеки google.genai для створення чат-сесії з моделлю Gemini, де налаштований доступ до інструменту пошуку даних в мережі. Або підготовка і попередня обробка даних, для наступного аналізу чи формування конкретної інформації для виконаня функціоналу по автоматичному визначенню відповідності користувача до критеріїв ліцензійних умов тощо. Взаємодія з LLM відбувається через ключі доступу та API;

– наукометричні бази даних та зовнішні веб-ресурси, які у цій реалізації є основними джерелами даних для модуля збору даних.

Комунікація між модулями в межах системи рейтингування ЗУНУ здійснюється через внутрішні механізми системи (наприклад, виклики функцій, обмін даними між модулями) та через спільну базу даних MySQL. Взаємодія зі зовнішніми модулями (модуль рекомендацій – через скрипти та графову базу даних, модуль збору даних — через токени доступу до системи та запитами до внутрішньої бази даних, LLM — через API) та зовнішніми елементами (зовнішні джерела даних та інтегровані через API) відбувається через REST API запити та TCP/IP зв’язок.

Якщо говорити про функціональність, ефективність чи стійкість системи, то це забезпечується її способом побудови, який ґрунтується на наступних принципах:

– система складається з незалежних одне від одного, логічно відокремлених модулів, що спрощує розробку, тестування, підтримку та оновлення. Кожен модуль відповідає за певну функціональність, що дозволяє легко додавати нові можливості або модифікувати існуючі без впливу на інші частини системи;

– окремі елементи системи можуть бути розгорнуті на різних серверах, що підвищує продуктивність та надійність. Наприклад, обробка даних у графовій БД та використання LLM відбувається поза системи рейтингування, що розвантажує її основні обчислювальні ресурси, що дозволяє працювати паралельно та не залежати від інших елементів;

– данна архітектура, передбачає можливість легкого масштабування для обробки даних звітів користувачів, які з часом зростатимуть (збільшення кількості користувачів, дисциплін, кафедр, та інших сутностей в системі). Використання Redis для черг завдань та графову БД для обчислень, сприяє цьому;

– взаємодія між модулями та зовнішніми системами здійснюється через чітко визначені інтерфейси (REST API), що забезпечує гнучкість та інтероперабельність, що відповідає принципам сервіс-орієнтованого підходу. Це дозволяє простіше інтегрувати нові сервіси, моделі та адаптуватися до змін у зовнішньому середовищі;

– безпека, яка забезпечується на кількох рівнях: авторизація, використання токенів та налаштування LLM.

Авторизація та аутентифікація в самій системі рейтингування підтримує різні ролі користувачів з відповідним рівнем доступу до різного функціоналу системи та до самих даних (користувачів, звітів, кафедр, рекомендацій тощо).

Взаємодія з зовнішнім модулем збору даних, реалізована через доступ по спеціальному токену, який сформований для системи та використовується у POST запитах.

Під налаштуванням LLM мається на увазі можливість редагувати промпти, які виконує LLM, це дозволяє контролювати та адаптувати взаємодію з моделлю, зменшуючи ризики некоректної генерації контенту. Дані, які передаються в промпти як датасети користувацьких даних, використовуються без прямих формулювань відносно конкретних користувачів, замість цього, використовуються заглушки, наприклад “поточний користувач”, “користувач1”, “користувач2” тощо. Крім того, система використовує тільки описові дані зі звітів, де відсутня якась персональна інформація, ігноруючи вкладення чи посилання в промптах.

Для ілюстрації взаємодії модулів та ролей у системі, доцільно розглянути послідовність кроків у життєвому циклі звіту, що веде до нарахування річного балу та формування рейтингу, який наведено у вигляді діаграми послідовності життєвого циклу звіту на рисунку 3.3.

Цей життєвий цикл звіту включає процес заповнення, затвердження та розрахунку балів звіту НПП. Основні учасники процесу: користувач (НПП), завідувач кафедри, рецензент, сама система рейтингування як інструмент та інтерфейс для користувачів, модуль заповнення даних в звіт, модуль розрахунку балів, модуль формування рейтингу користувачів [11, 12].

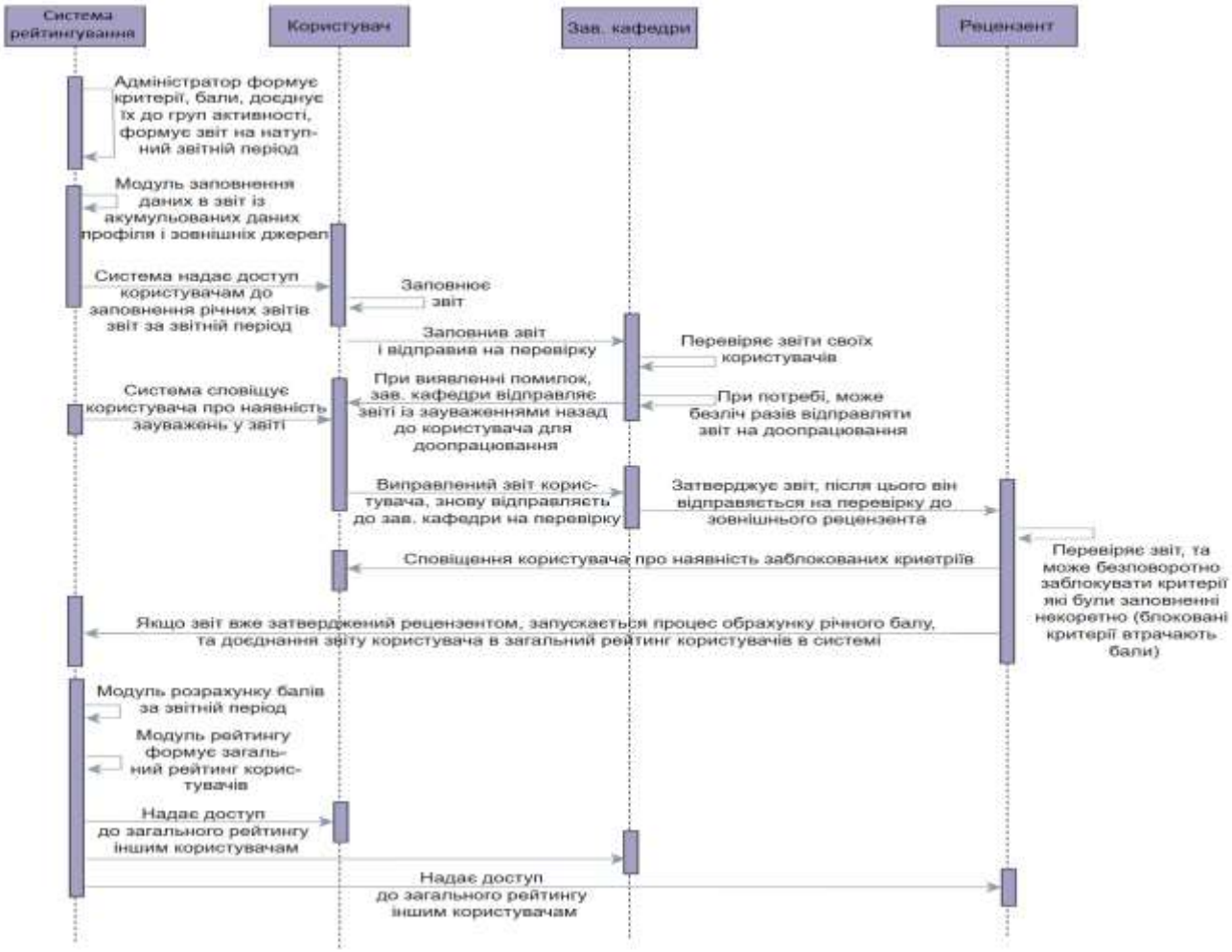


Рис. 3.3. Діаграми послідовності життєвого циклу звіту

Якщо описувати детальніше, то життєвий цикл звіту складається з таких основних етапів:

– Адміністратор системи, створює критерії оцінювання, описуючи їх, додаючи базовий бал, тип формування оцінки, формат заповнення критерію, додає групу активності тощо. Після цього, формує звіт на наступний звітний період з певної кількості критеріїв, та налаштовує часові рамки старту, завершення заповнення звіту, перевірки завідувачами кафедр та рецензентами.

– Користувач (НПП) заповнює річний звіт, вносячи дані за критеріями діяльності. Модуль автоматичного заповнення звіту може допомогти, використовуючи дані з профілю та зовнішніх джерел, та дозаповнити частину звіту. Коли звіт буде готовий, користувач може відправити звіт на наступні етапи обробки.

– Після цього, система автоматично відправить звіт на перевірку до завідувача кафедри поточного користувача.

– Завідувач кафедри може переглядати звіти своїх працівників, та перевіряти коректність заповнення. У випадку виявлення некоректного або неповного заповнення одного із критеріїв, він може сформувати зауваження та повернути звіт користувачеві для внесення правок, в іншому випадку він затверджує звіт.

– Якщо звіт був повернутий до користувача, він знову може оновити свій звіт, виправивши зауваження. Система повідомляє користувачам про зауваження шляхом підказок в інтерфейсі та сповіщенням на електронну пошту. Користувач може внести зміни у свій звіт і відправити знову на перевірку до завідувача кафедри. Цей процес може відбуватись без кількісних обмежень.

– Далі, за умови, що завідувач кафедри затвердив звіт користувача, система рейтингування відправляє його до рецензентів на перевірку.

– Рецензент — це кінцевий рівень перевірки звіту. Він може блокувати некоректно заповнені критерії (бал за них буде ігноруватися в загальному рейтингу). Цей процес безповоротній, повернути назад звіт до користувача (без затвердження адміністратора) вже не можна. Якщо користувач та завідувач кафедри припустились помилки та неправильно заповнили дані в звіті, вони уже будуть заблоковані. Рецензент після перегляду звіту також затверджує його.

– Якщо завідувач кафедри та рецензент вже затвердили звіт, система запускає процес розрахунку балів.

– Модуль розрахунку балів обчислює бали за виконані критерії в залежності від їх типу і способу калькуляції, сумує їх за групами активності для отримання набраного річного балу користувача.

– Модуль рейтингу формує загальний рейтинг користувача в конкретному році та рейтинг користувачів (за 5 років).

– Тепер загальний рейтинг користувачів стає доступними для перегляду.

Діаграма 3.3 дозволяє чітко прослідкувати послідовність дій та взаємодію між різними ролями та модулями системи в процесі обробки звітів та формування рейтингу.

Діаграма варіантів використання системи, наведена на рисунку 3.4, наочно демонструє взаємодію кожної ролі з різними функціональними блоками системи, підкреслюючи розподіл обов'язків та прав доступу. Для забезпечення ефективного функціонування системи та розмежування прав доступу, передбачено декілька ролей користувачів, кожна з яких має свій специфічний набір функцій. Розглянемо їх детальніше [8, 12].

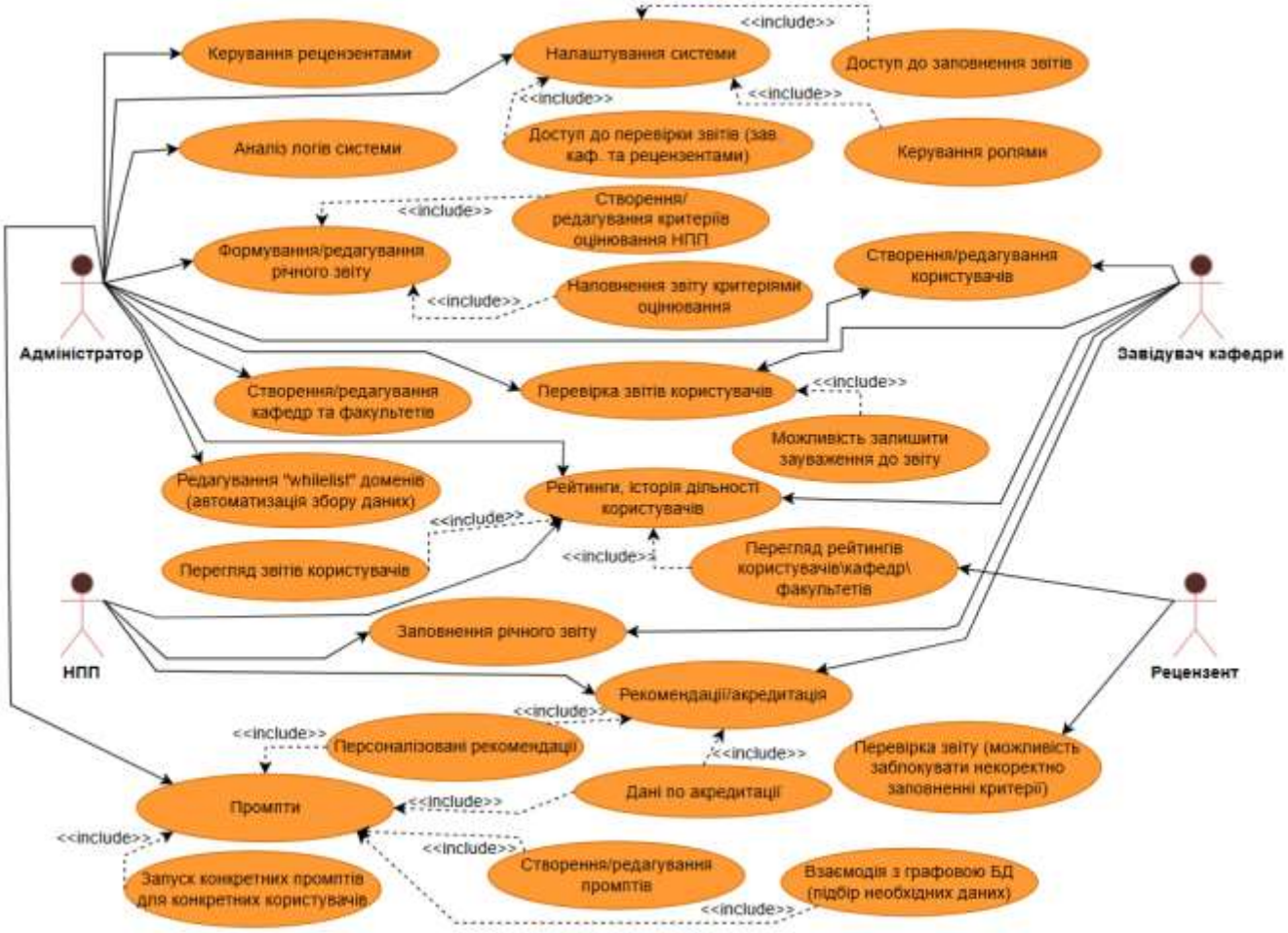


Рис. 3.4. Діаграма варіантів використання інтелектуалізованої системи рейтингування НПП

1. Адміністратор. Має повний доступ до всіх налаштувань системи. Функції включають: налаштування системи і процесів пов'язаних з рейтингуванням, перевірки звітів, CRUD операції (Create, Read, Update, Delete) для всіх сутностей у системі (користувачів, звітів, критеріїв, дисциплін, кафедр та факультетів), рейтинги, доступ до налаштувань LLM, промпти сценаріїв, логи системи, керування ролями, редагування "white list" доменів для автоматизації збору даних. Критерії оцінювання можуть затверджувати на іншому рівні (спеціалізованою групною, вищим керівництвом університету), але доступ до внесення змін до критеріїв оцінювання є тільки у адміністратора.

2. Користувач, який і є НПП. Це ключова роль, на якій базується система і для якої реалізовано основний функціонал. Він звітує про свою науково-педагогічну діяльність та заповнює річні звітів, також має доступ до керування власним профілем в системі, своїми дисциплінами. Крім того, доступний перегляд рейтингів, даних по відповідності їх публікацій до дисциплін які вони викладають, своїх персоналізованих рекомендацій та звітів про ліцензійні вимоги до НПП.

3. Завідувач кафедри. Керує користувачами своєї кафедри, має обмежений доступ до керування профілями своїх користувачів, може перевіряти звіти користувачів в межах кафедри, перевіряти та затверджувати звіти. Ця роль, практично є розширеною версією ролі "користувача".

4. Рецензент — це людина, яка може бути або не бути працівником поточного ЗВО, але має доступ до процесу рецензування, а саме може перевіряти звіти користувачів (після затвердження завідувачами кафедр), заблокувати критерії та затверджувати звіт, якщо його заповнено коректно. Якщо це користувач з поточного ЗВО, він також буде мати доступ до функціоналу ролі "користувача".

Отже, розподіл ролей у системі виконано залежно від ключових користувачів, з розподілом доступу до різного функціоналу відповідно до потреби. Налаштування ролей користувачів в системі дозволяє зручно керувати їх доступом до різних елементів системи та керувати працівниками.

3.2. Особливості проектування модуля збору відкритих даних

Розробка інтелектуалізованої системи рейтингування діяльності НПП, що була розглянута у попередніх підрозділах, вимагає надійного та ефективного механізму для збору релевантної інформації та процесу автоматичного заповнення звіту. Цей аспект реалізується через окремий модуль збору відкритих даних, який забезпечує більшу актуальність та повноту даних для системи рейтингування.

Модуль збору відкритих даних спроектовано на принципах модульної автономності, що дозволяє йому функціонувати незалежно від системи рейтингування, акумулювати дані та виконувати роль зовнішнього дата-центру. Такий підхід мінімізує вплив процесу збору даних на загальну продуктивність системи, оскільки цей процес може бути тривалим. Зазначений модуль покликаний автоматизувати збір частини інформації про наукову діяльність працівників, на поточний момент, це статті, публікації та інші роботи з цієї сфери, які походять з різноманітних зовнішніх джерел та наукометричних баз.

Архітектура модуля побудована за клієнт-серверною моделлю та включає декілька ключових компонентів, що забезпечують його функціональність та масштабованість [2, 3]. Графічне представлення компонентів цього модуля наведено на рисунку 3.5.

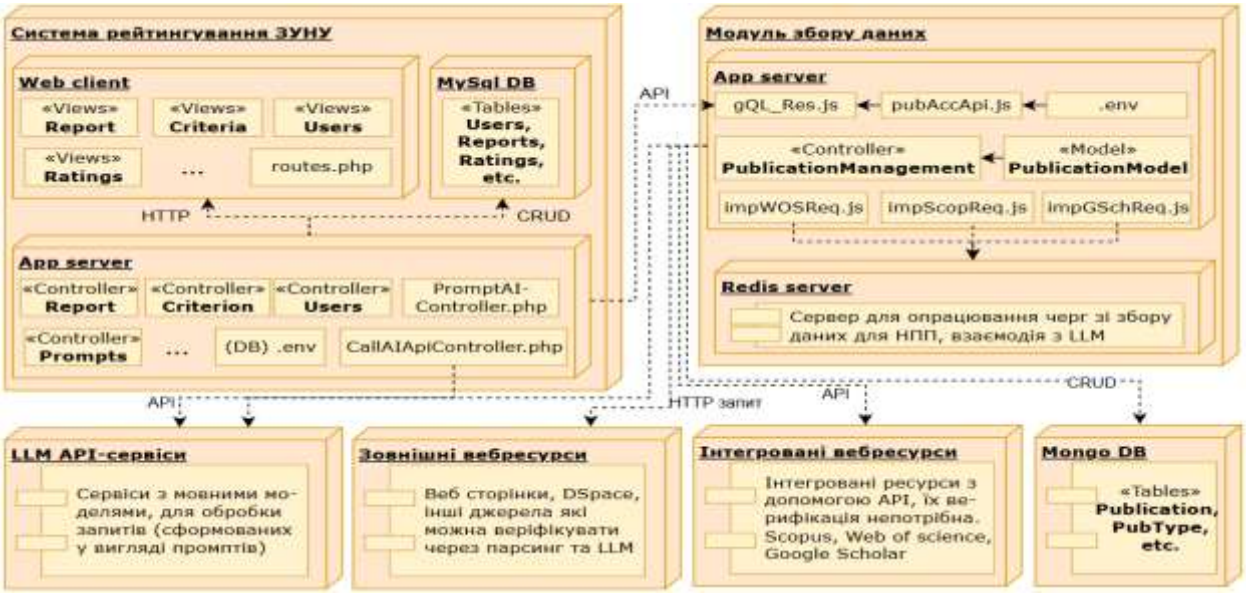


Рис. 3.5. Діаграма компонентів модуля збору відкритих даних разом із системою рейтингування ЗУНУ

Основна серверна частина модуля може виконувати асинхронну обробку запитів та має базову модель з контролером для роботи з даними та власну БД для збереження проміжних результатів. Вона побудована на основі Node.js та фреймворку Express.js. Вибір Node.js обґрунтований його здатністю до асинхронної обробки запитів, що сприяє значному підвищенню продуктивності системи при високому навантаженні, а Express.js, будучи мінімалістичним веб-фреймворком, спрощує розробку серверної логіки та управління маршрутизацією модуля. Для зберігання даних використовується MongoDB, що надає гнучкість у роботі з неструктурованими та напівструктурованими даними, які характерні для наукових публікацій та їх метаданих. Загалом такі елементи часто комбінуються при створені подібних систем чи модулів [2, 3].

Взаємодія між сервером та базою даних реалізується за допомогою GraphQL API. Він має переваги над традиційним REST API, оскільки надає можливість точного контролю над обсягом даних (якими можна маніпулювати), що дозволяє оптимізувати використання ресурсів та зменшити кількість надлишкових запитів вибираючи тільки ті дані, які необхідні в поточний момент. Є загальні js файли які описують API для взаємодії з БД, а саме для доступу системи рейтингування до зібраних даних із захистом за персональним токеном [2, 3].

Для керування чергами завдань, зокрема імпортом публікацій з різних джерел, застосовується Redis Server, що продемонстрував свою ефективність у системі рейтингування, тому для поточного модуля він також виконує функцію обробки черг завдяки зберіганню даних в оперативній пам’яті, що додатково пришвидшує його роботу. Конфігураційні параметри сервера зберігаються у файлі .env, а точки входу для API визначені у файлі pubAccApi.js. Бізнес-логіка моделі описується у відповідних контролерах, а обробка запитів GraphQL здійснюється через налаштування у файлі gqlRes.js [2, 3].

Модуль збирає дані, взаємодіючи із зовнішніми веб-ресурсами, такими як DSpace, а також з провідними наукометричними базами, включаючи Web of Science, Google Scholar та Scopus. Це дозволяє збирати актуальну інформацію про наукову діяльність [2, 3].

Крім того, цей модуль включає процес парсингу веб-сторінок для отримання даних з зовнішніх джерел, інтеграцію з системами штучного інтелекту для автоматизації розпізнавання і класифікації інформації, та автоматичне дозаповнення звіту користувачів зібраними даними (рисунок 3.6).

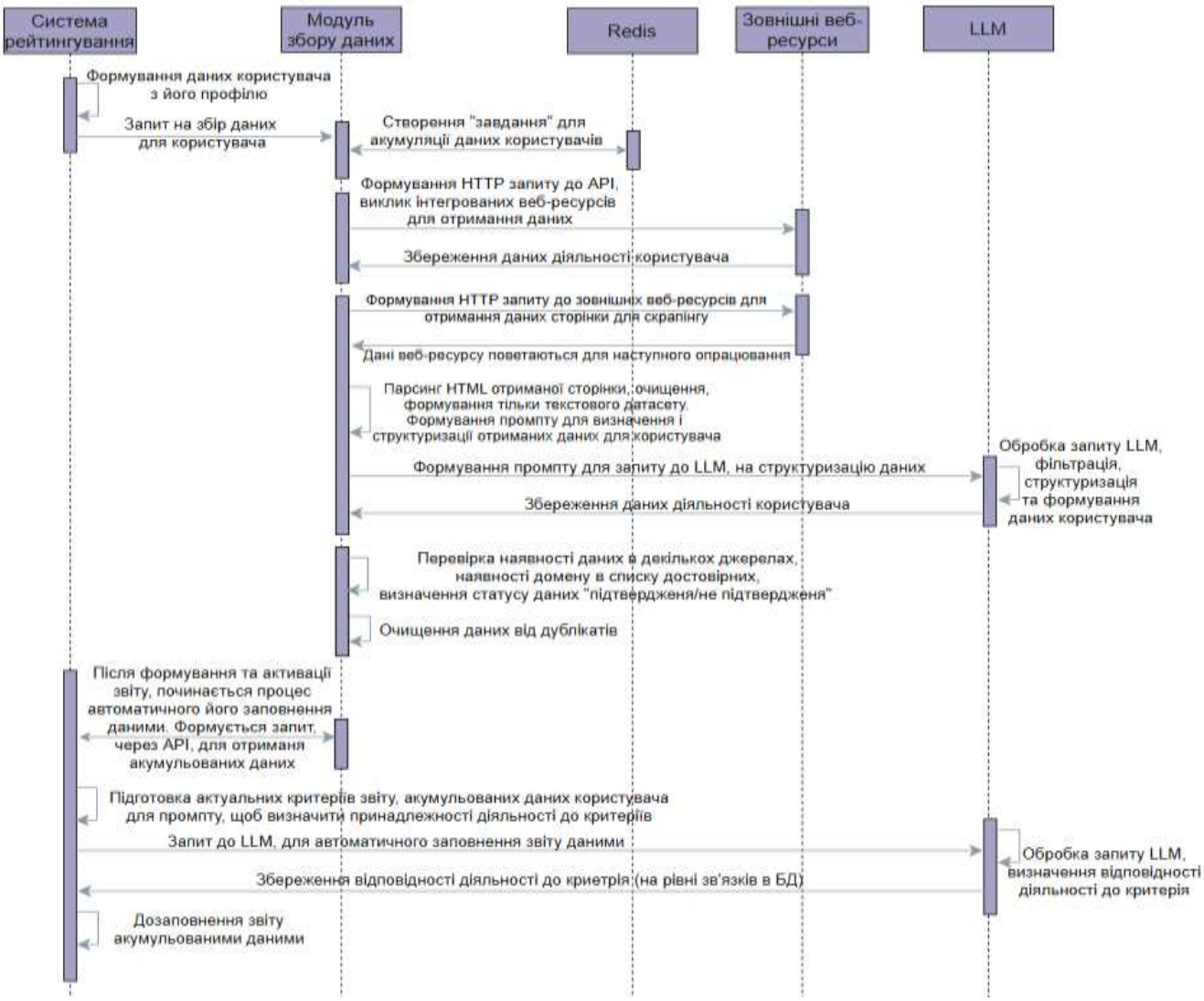


Рис. 3.6. Діаграма послідовності роботи модуля збору відкритих даних та автозаповнення звіту

Щоб зберегти зібрані дані під певним користувачем, виконується API-запит до системи рейтингування, щоб звірити дані по користувачам, та отримати дані, необхідні для роботи модуля (ПІБ, ORCID, ScopusId, GoogleScholarID) та зберігаємо їх в локальну БД. Отримані дані передаються в нове “завдання” в Redis для паралельної обробки. Тепер використовуються актуальні дані користувача та отримані дані з веб-ресурсів для їх опрацювання. Під час цього процесу

відбувається визначення основних структур та елементів веб-сторінки, з яких збираються дані по публікаціям (назви, ключові слова, автори, тип/категорія, контекст тощо). З інтегрованих сервісів акумулюються дані користувачів напряму, а з інших ресурсів — з допомогою парсингу веб-сторінок, очищуючи їх від всіх HTML тегів (посилань, зображень, кнопок, header/footer тощо) залишаючи тільки текст сторінки [5, 8, 127, 128]. Отриманий датасет аналізується з допомогою штучного інтелекту (за аналогією до того, як це працює з рекомендаціями). В результаті отримуємо основні дані публікацій, які стосуються конкретного користувача (назви статей, ПІБ авторів, ключові слова тощо). Це досягається завдяки контекстуальному розпізнаванню, що базується на трансформерних архітектурах та механізмі самоуваги (self-attention). Моделі здатні розрізняти синоніми та різні варіанти написання слів, включаючи кирилицю та латиницю, забезпечуючи коректне зіставлення авторів чи назв статей, наприклад, “Мельник Андрій” та “Andrii Melnyk” [5, 8, 127, 128].

Взаємодія з LLM також виконується через підготовлені промпти, у яких описано, що саме та як саме потрібно проаналізувати, а також який результат очікується [2]. Датасет, який опрацьовується, не має чіткої структури даних і в залежності від вебсервісу, датасет може мати різний вигляд, але всіх їх об’єднує подібна структура даних, а саме списки чи послідовності публікацій разом із їх контекстом. Для мовних моделей, структуризація таких даних, не є великою проблемою, вони чітко виявляють структуру датасету, та відокремлюють дані, які відносяться до одного сигмента. Адаптивність промптів дозволяє не залежати від формату вхідних даних та дозволяє формувати бажаний результат (наприклад, в форматі JSON) в чітко зазначену модель, що забезпечує зручність подальшої роботи та збереження в БД. Крім того, дані отримані з доменів, які містяться у таблиці з «white list», система позначає як достовірні. Якщо джерело було зовнішнє і немає додаткового підтвердження у вигляді таких самих даних в іншому джерелі, система вважає дані не підтвердженими. Вони залишаються в системі але використовуються для ручного заповнення, натомість підтверджені дані автоматично дозаповнюються в звіт працівника, а саме в групи активності

пов'язаної з публікаціями, включаючи дані про кількість сторінок, співавторство, тип публікації та ключові слова. Коли процес збору даних для користувача завершився, система очищує всі дублікати записів, які були зібрані, по назві роботи.

Впровадження цього модуля частково автоматизує процес заповнення звітів, зменшуючи адміністративне навантаження на НПП та мінімізуючи ймовірність помилок. Система надає користувачам попередньо заповнений звіт з основними даними, зібраними модулем, залишаючи можливість для редагування та додавання детальнішої інформації. Ті дані, які не були підтверджені, користувачі можуть вручну імпортувати в систему і доєднати до звіту в кілька кліків.

У подальших дослідженнях та розробках передбачається розширення функціоналу модуля, а саме збір більшої кількості даних про діяльність користувачів з інших груп активностей та покращувати процес підтвердження даних для автоматичного заповнення.

3.3. Особливості проектування системи зберігання даних

Проєктування ефективної та надійної системи зберігання даних — це важлива частина комплексної розробки інтелектуалізованої системи рейтингування діяльності НПП. Вона має забезпечити не тільки збереження великого обсягу різнорідних даних, а й швидкий доступ до них, гнучкість у маніпулюванні інформацією та підтримку специфічних запитів, щодо різного функціоналу системи. Вибір архітектури бази даних, а також її реалізація, прямо впливають на продуктивність усієї системи, її масштабованість та можливості для подальшого оновлення та розвитку.

Раніше було зазначено, що система рейтингування почала працювати в ЗУНУ в 2017 році, та її основна, базова структура бази даних була закладена ще тоді. Для реалізації функціоналу системи рейтингування було обрано реляційну базу даних MySQL. Таке рішення було зумовлене поширеністю даної СУБД, її відносною простотою в освоєнні та широким спектром доступних інструментів та

документації. MySQL зарекомендувала себе як надійна система управління базами даних, що забезпечує високу продуктивність завдяки оптимізованим алгоритмам обробки запитів, планувальникам та механізмам кешування, дозволяючи ефективно працювати з великими обсягами даних та високонавантаженими операціями. Її надійність та стабільність, підкріплена вбудованими механізмами відновлення та резервного копіювання, сприяє запобіганню можливих втрат даних [11, 12].

Проте, варто зазначити, що, незважаючи на ці переваги, специфіка реляційних баз даних може створювати певні складнощі при роботі зі складними, взаємопов'язаними даними, характерними для сфери діяльності НПП. Маніпулювання та аналіз даних, що мають багаторівневі зв'язки, може вимагати виконання ресурсоємних операцій об'єднань таблиць, що потенційно впливає на продуктивність та ускладнює написання запитів. Основна частина даних, що стосується діяльності НПП та формування звіту, була сформована специфічним чином і організована у не оптимальну структуру, для виконання складних аналітичних запитів. Тому не всі бажані аналітичні дані або запити можна легко, без навантаження на БД, і відповідно систему, отримати з поточної реалізації архітектури бази даних. З огляду на це, для певних модулів системи було прийнято рішення щодо використання інших підходів до зберігання та маніпулювання даними.

Fig. 1. Для того щоб краще розуміти структуру системи збереженя необхідно мати коректні зв'язки між сутностями в БД і їх опис в системі базою даних. Щоб оформити це, зазвичай використовують мову опису Definition Language (DDL), який формує метадані щоб виконувати CRUD над структурними об'єктами бази даних, у

ДОДАТОК А представлено DDL основної частини системи, включаючи рекомендації користувачів та рецензування.

Реляційна БД побудована на основі декількох ключових сутностей, що відображають діяльність НПП, їхні дані, звіти, і пов’язані з цим сутності, публікації, дані по рекомендаціям та відповідність ліцензійним вимогам. Загальна ER-діаграма усієї системи наведена в ДОДАТОК В, де показано всі наявні сутності та їх взаємозв’язки.

Всю структуру БД можна поділити на кілька блоків. Почнемо з блоку який пов’язаних з користувачем та його профілем в системі, ER-діаграма наведена на рисунку 3.7.

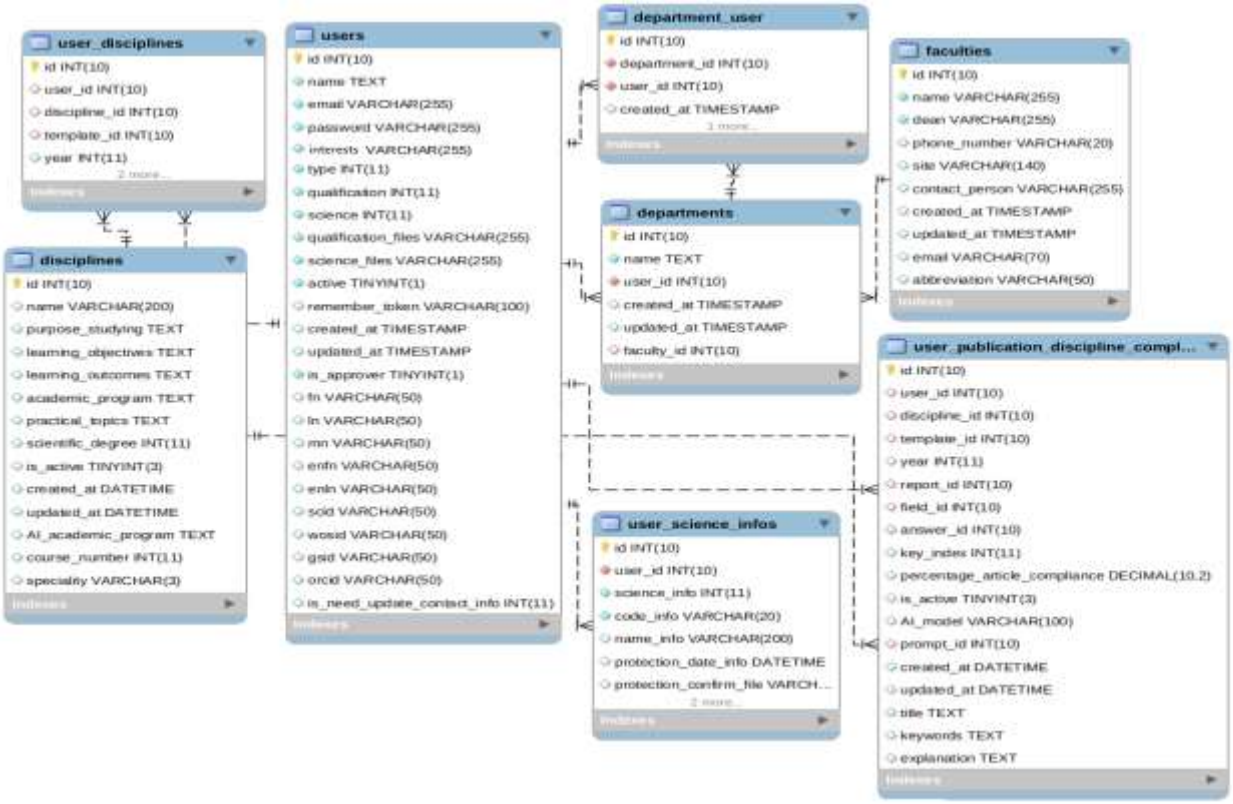


Рис. 3.7. ER-діаграма користувача і даних його профілю в системі

Отже, в системи є таблиця “*users*” (користувачі), яка зберігає основну інформацію про НПП та інших працівників які доєднані до системи рейтингування (рисунок 3.5). З корисувачем пов’язані: “*departments*” – де містяться основні кафедри які існують в ЗУНУ, “*faculties*” – містить факультети, “*disciplines*” – списки дисциплін з навчальної програми ЗУНУ по всіх групах та рівнях, “*user_disciplines*”

– фіксує дисципліни, що викладаються користувачем, та “*user_science_infos*” – містить додаткові наукові відомості про користувача.

Далі, розглянемо більш детально основні таблиці системи. Таблиця “*users*” (рисунок 3.8) має такий набір основних атрибутів:

– id (PK): користувача;

– name, fn, ln, mn: повне ім’я користувача, скорочені ім’я, прізвище, по батькові, та англомовна версія полів також;

– email, password: пошта, хешований пароль користувача для входу в систему;

– interests: список інтересів користувача (для опису профілю);

– type: роль користувача (наприклад, адміністратор, користувач тощо) ;

– qualification, science, qualification_files, science_files: ключ кваліфікації/наукового ступеня користувача та підтверджуючих файлів;

– active: статус активності користувача;

– is_approver: вказує на можливість рецензувати;

– scid, wosid, gsid, orcid: ключі користувача в наукометричних базах даних (Scopus, Web of Science, Google Scholar, ORCID) ;

– dis_science_info1, dis_code_info1, dis_name_info1, dis_protection_date_info1, dis_science_info2, dis_code_info2, dis_name_info2, dis_protection_date_info2: інформація про захищені дисертації (науковий ступінь, код, назва, дата захисту);

– is_need_update_contact_info: прапорець, що вказує на необхідність оновлення контактної інформації.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int		UNSIGNED	No	None		AUTO_INCREMENT
2	name	text	utf8mb3_unicode_ci		No	None		
3	email	varchar(255)	utf8mb3_unicode_ci		No	None		
4	password	varchar(255)	utf8mb3_unicode_ci		No	None		
5	type	int			No	3		
6	qualification	int			No	4		
8	qualification_files	varchar(255)	utf8mb3_unicode_ci		No	None		
9	interests	varchar(255)	utf8mb3_unicode_ci		No	None		
10	active	tinyint(1)			No	1		

14	is_approver	tinyint(1)	
15	fn	varchar(50)	utf8mb3_unicode_ci
16	ln	varchar(50)	utf8mb3_unicode_ci
20	scid	varchar(50)	utf8mb3_unicode_ci
21	wosid	varchar(50)	utf8mb3_unicode_ci
24	dis_science_info1	int	
29	dis_code_info2	varchar(20)	utf8mb3_unicode_ci
30	dis_name_info2	varchar(100)	utf8mb3_unicode_ci
31	dis_protection_date_info2	datetime	
32	is_need_update_contact_info	int	

Рис. 3.8. Структура таблиці “users”

Таблиця “*departments*” (рисунок 3.9) має такий набір основних атрибутів:

– id (PK): кафедри;

– name: назва кафедри;

– user_id (FK): посилання користувача, завідувача кафедри;

– faculty_id (FK): посилання на факультет, з яким пов’язана кафедра.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**name**	text	utf8mb3_unicode_ci		No	*None*		
3	**user_id**	int		UNSIGNED	No	*None*		
4	**created_at**	timestamp			Yes	*NULL*		
5	**updated_at**	timestamp			Yes	*NULL*		
6	**faculty_id**	int		UNSIGNED	Yes	*NULL*		

Рис. 3.9. Структура таблиці “departments”

Таблиця “*disciplines*” (рисунок 3.10) має такий набір основних атрибутів:

– id (PK): дисципліни;

– name: назва дисципліни;

– purpose_studing, learning_objectives, learning_outsomes, academic_program, practical_topics_scientific_degree, AI_academic_program: основні поля по опису дисципліни, головною колонкою є AI_academic_program, яка використовується для взаємодії з LLM;

– course_number, spesiality: номер курсу і спеціальність дисципліни;

– is_active: вказує чи активна дисципліна.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**name**	varchar(200)	utf8mb3_general_ci		Yes	*NULL*		
3	**purpose_studying**	text	utf8mb3_general_ci		Yes	*NULL*		
4	**learning_objectives**	text	utf8mb3_general_ci		Yes	*NULL*		
5	**learning_outcomes**	text	utf8mb3_general_ci		Yes	*NULL*		
6	**academic_program**	text	utf8mb3_general_ci		Yes	*NULL*		
7	**practical_topics**	text	utf8mb3_general_ci		Yes	*NULL*		
8	**scientific_degree**	int			Yes	*NULL*		
9	**is_active**	tinyint		UNSIGNED	Yes	0		
10	**AI_academic_program**	text	utf8mb3_general_ci		Yes	*NULL*		
11	**course_number**	int			Yes	*NULL*		
12	**speciality**	varchar(3)	utf8mb3_general_ci		Yes	*NULL*		

Рис. 3.10. Структура таблиці “disciplines”

Таблиця “*user_publication_discipline_compliances*” (рисунок 3.11) має такий набір основних атрибутів:

– id (PK): таблиць для зв’язків;

– user_id, discipline_id, template_id, report_id, field_id, answer_id: ключ користувача, дисципліни, шаблону звіту, самого звіту, критерію, відповіді на критерій, з яким пов’язаний запис;

– year: рік, до якого відноситься запис;

– key_index: внутрішній індекс запису, який допомагає ідентифікувати дані;

– percentage_article_compliance: десяткове число, що відображає відсоток відповідності статті до дисципліни (в % від 0 до 100);

– is_active: статус активності запису;

– AI_model: текстове поле, що містить назву або ключ моделі штучного інтелекту, яка використовувалась;

– prompt_id: ключ промпту, застосованого в контексті AI-моделі;

– title, keywords: текстове поле для зберігання заголовка та ключових слів публікації;

– explanation: текстове поле для надання додаткових пояснень від ШІ щодо значення percentage_article_compliance (корисно щоб аналізувати результат).

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int		UNSIGNED	No	None		AUTO_INCREMENT
2	user_id	int		UNSIGNED	Yes	NULL		
3	discipline_id	int		UNSIGNED	Yes	NULL		
4	template_id	int		UNSIGNED	Yes	NULL		
5	year	int			Yes	NULL		
6	report_id	int		UNSIGNED	Yes	NULL		
7	field_id	int		UNSIGNED	Yes	NULL		
8	answer_id	int		UNSIGNED	Yes	NULL		
9	key_index	int						
10	percentage_article_compliance	decimal(10,2)						
11	is_active	tinyint						UNSIGNED
12	AI_model	varchar(100)	utf8mb3_general_ci					
13	prompt_id	int						UNSIGNED
14	title	text	utf8mb3_general_ci					
15	keywords	text	utf8mb3_general_ci					
16	explanation	text	utf8mb3_general_ci					

Рис. 3.11. Структура таблиці “user_publication_discipline_compliances”

Наступним розглянемо блок структури звітів користувачів, які зберігаються в таблиці “*reports*”, які є основним джерелом даних для рейтингової оцінки, ER-діаграма цього блоку наведено на рисунку 3.12.

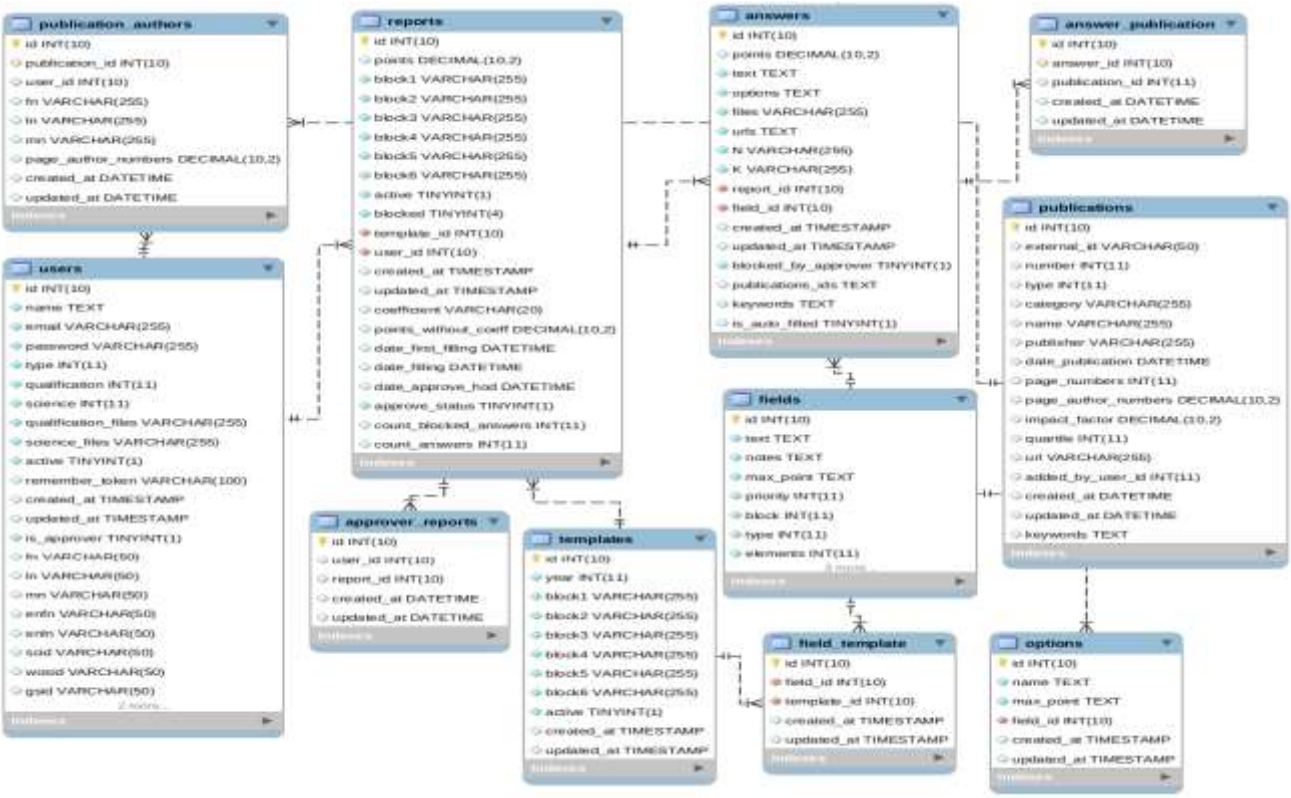


Рис. 3.12. ER-діаграма звіту користувачів в системі

Звіт складається з “*answers*” (відповідей) на певні критерії “*fields*” (критерії), що представляють собою опис діяльності, базовий бал або формулу розрахунку, а також номер групи активності. Та пов’язані зі структурою критерії, таблиці: “*templates*”, “*options*” – які вказуєть на відповідність звіту до певного року та додаткові опції критеріїв, які залежить від типу критерію. Звіти проходять процес затвердження через “*approver_reports*” (таблиця затвердження звітів), що включає перевірку завідувачами кафедр та рецензентами.

Таблиця “*reports*” (рисунок 3.13) має такий набір основних атрибутів:

– id (PK): звіту;

– points: загальна кількість балів за звіт;

– block1 – block6: текстові поля для зберігання балів за різні групи активності (спрощує обрахунки в системі);

– active, blocked: статус активності чи блокування звіту;

– template_id (FK), user_id (FK): зовнішнє посилання для зв’язку із шаблоном звіту та користувачем;

– coefficient: коефіцієнт для річного балу (множник);

– points_without_coeff: бали без урахування коефіцієнта;

– date_first_filling, date_approve_hod: дата заповнення\затвердження звіту;

– approve_status: статус звіту;

– count_blocked_answers: кількість заблокованих критеріїв у звіті.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int		UNSIGNED	No	None		AUTO_INCREMENT
2	points	decimal(10,2)			Yes	NULL		
3	block1	varchar(255)	utf8mb3_unicode_ci		No	0		
4	block2	varchar(255)	utf8mb3_unicode_ci		No	0		
5	block3	varchar(255)	utf8mb3_unicode_ci		No	0		
6	block4	varchar(255)	utf8mb3_unicode_ci		No	0		
7	block5	varchar(255)	utf8mb3_unicode_ci		No	0		
8	block6	varchar(255)	utf8mb3_unicode_ci		No	0		
9	active	tinyint(1)			No	1		
10	blocked	tinyint			No	0		
11	template_id	int						
12	user_id	int						
13	coefficient	varchar(20)	utf8mb3_unicode_ci					
14	points_without_coeff	decimal(10,2)						
15	date_first_filling	datetime						
16	date_filling	datetime						
17	date_approve_hod	datetime						
18	approve_status	tinyint(1)						
19	count_blocked_answers	int						
20	count_answers	int						

Рис. 3.13. Структура таблиці “reports”

Таблиця “*answers*” (рисунок 3.14) має такий набір основних атрибутів:

– id (PK): відповіді користувача на критерій;

– points: кількість балів, нарахованих за критерій;

– text, keywords: текстовий опис даних для критерію від користувача;

– options: обрані опції (залежить від типу критерія);

– files, urls: посилання на прикріплені файли, зовнішні ресурси;

– N, K: змінні для формул розрахунку балів;

– report_id (FK), field_id (FK): зовнішнє посилання для зв’язку з відповідним звітом, критерієм;

– blocked_by_approver: чи заблоковано відповідь рецензентом;

– publications_ids: ключі пов’язаних публікацій;

– is_auto_filled: чи була відповідь автоматично заповнена системою.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int		UNSIGNED	No	None		AUTO_INCREMENT
2	points	decimal(10,2)			Yes	NULL		
3	text	text	utf8mb3_unicode_ci		No	None		
4	options	text	utf8mb3_unicode_ci		No	None		
5	files	varchar(255)	utf8mb3_unicode_ci		No	None		
6	urls	text	utf8mb3_unicode_ci		No	None		
7	N	varchar(255)	utf8mb3_unicode_ci		No	0		
8	K	varchar(255)	utf8mb3_unicode_ci					
9	report_id	int						
10	field_id	int						
11	blocked_by_approver	tinyint(1)						
12	publications_ids	text	utf8mb3_unicode_ci					
13	keywords	text	utf8mb3_unicode_ci					
14	is_auto_filled	tinyint						

Рис. 3.14. Структура таблиці “answers”

Таблиці “*templates*”, “*options*” (рисунок 3.15) пов’язані зі звітом і критерієм та мають такі атрибути:

– id (PK): шаблону і опції критерія;

– year: рік, для якого призначений шаблон, відповідно і звіт;

– block1 – block6: текстові поля, що містять назви кожної з шести груп активності (для можливості додавання кастомних назв);

– active: статус активності шаблону;

– name: назва, опис опції;

– max_point: максимальний бал, який надається за вибір цієї опції;

– field_id (FK): посилання для зв’язку з відповідним критерієм.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default
1	id	int		UNSIGNED	No	None
2	year	int			No	None
3	block1	varchar(255)	utf8mb3_unicode_ci		No	None
4	block2	varchar(255)	utf8mb3_unicode_ci		No	None
5	block3	varchar(255)	utf8mb3_unicode_ci		No	None
6	block4	varchar(255)	utf8mb3_unicode_ci		No	None
7	block5	varchar(255)	utf8mb3_unicode_ci		No	None
8	block6	varchar(255)	utf8mb3_unicode_ci		No	None
9	active	tinyint(1)			No	1

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments
1	id	int		UNSIGNED	No	None	
2	name	text	utf8mb3_unicode_ci		No	None	
3	max_point	text	utf8mb3_unicode_ci		No	None	
4	field_id	int		UNSIGNED	No	None	
5	created_at	timestamp			Yes	NULL	
6	updated_at	timestamp			Yes	NULL	

Рис. 3.15. Структура таблиці “templates” та “options”

Важливим компонентом діяльності НПП є публікаційна активність. Дані про публікації зберігаються у таблиці “*publications*”, яка містить інформацію про саму публікацію (рисунок 3.12). Зв’язок між публікаціями та користувачами здійснюється через таблицю “publication_authors”.

Таблиця “*publications*” (рисунок 3.16) має такий набір основних атрибутів:

– id (PK): публікації;

– external_id: зовнішній ключ публікації з модуля збору даних;

– type: тип публікації (наприклад, стаття, тези, монографія);

– category: категорія публікації (наприклад, стаття в журналі, конференція, категорія “Б” тощо) (якщо є);

– name, keywords: назва, ключові слова публікації;

– publisher, date_publication: видавець, дата публікації (якщо є);

– page_numbers: кількість сторінок (якщо є);

– page_author_numbers: кількість сторінок, написаних автором (якщо є);

– impact_factor, quartile: імпакт-фактор, квартиль журналу (якщо є);

– url: URL-адреса публікації (якщо є);

– added_by_user_id (FK): посилання на користувача, який додав публікацію.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int		UNSIGNED	No	None		AUTO_INCREMENT
2	external_id	varchar(50)	utf8mb3_general_ci		Yes	NULL		
3	number	int			Yes	NULL		
4	type	int			Yes	NULL		
5	category	varchar(255)	utf8mb3_general_ci		Yes	NULL		
6	name	varchar(255)	utf8mb3_general_ci		Yes	NULL		
7	publisher	varchar(255)	utf8mb3_general_ci		Yes	NULL		
8	date_publication	datetime						
9	page_numbers	int						
10	page_author_numbers	decimal(10,2)						
11	impact_factor	decimal(10,2)						
12	quartile	int						
13	url	varchar(255)	utf8mb3_general_ci					
14	added_by_user_id	int						
15	keywords	text	utf8mb3_general_ci					

Рис. 3.16. Структура таблиці “publications”

Наступний блок безпосередньо приймає участь у формування рекомендацій, це промпти для взаємодії з LLM. Для інтелектуалізованих функцій системи, таких як формування рекомендацій, звіту про діяльність НПП та ліценійні вимоги, відповідність публікацій до дисципліни, які веде користувач, чи автоматичне заповнення звітів — використовуються великі мовні моделі. Управління промптами здійснюється за допомогою таблиці “*prompts*”, яка містить опис, тип, текст промпту, параметри та статус активності промпту (рисунок 3.17). Промпти групуються за типами та моделями LLM за допомогою “*prompt_group_model*” та “*prompt_type_model_groups*”, що забезпечує систематизацію та ефективне

управління. Можливість налаштовувати різні промпти, для різниз завдань, та використовувати спецефічні моделі або проксі надає гнучкості цьому функціоналу.

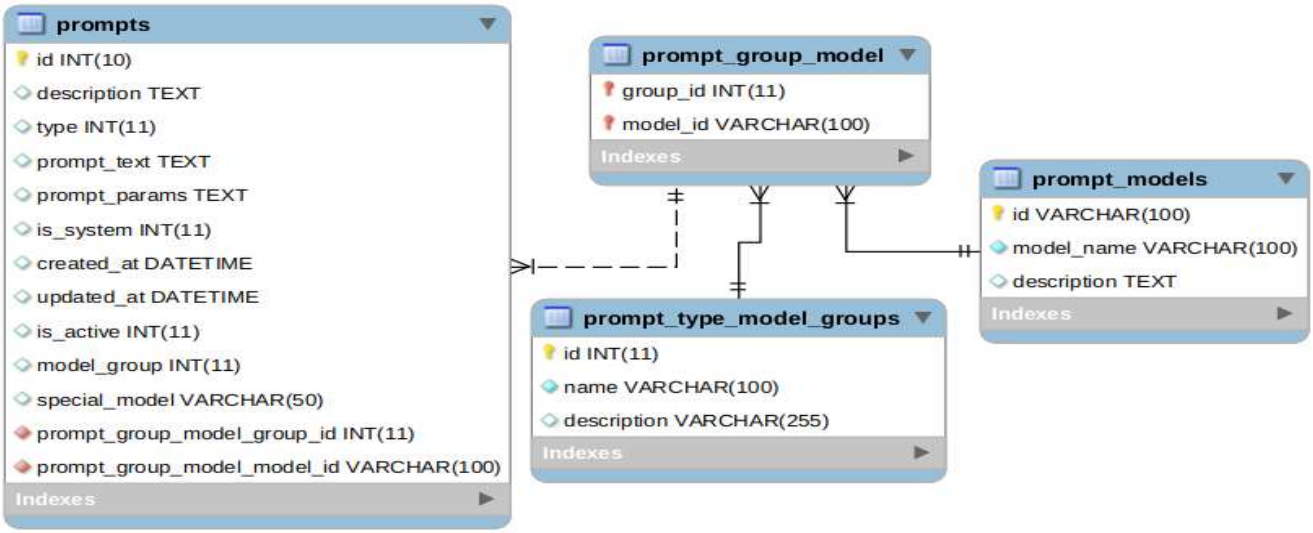


Рис. 3.17. ER-діаграма структури промптів

Таблиця “*prompts*” (рисунок 3.18) має такий набір основних атрибутів:

– id (PK): промпту;
– description: опис призначення промпту;
– type: тип промпту;
– prompt_text, prompt_params: текст/параметри промпту;
– is_system, is_active: статус промпт системним/активний;
– model_group: ключ групи моделі;
– special_model: спеціальна модель LLM для опрацювання.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**id**	int		UNSIGNED	No	*None*		AUTO_INCREMENT
2	**description**	text	utf8mb3_general_ci		Yes	*NULL*		
3	**type**	int			Yes	*NULL*		
4	**prompt_text**	text	utf8mb3_general_ci		Yes	*NULL*		
5	**prompt_params**	text	utf8mb3_general_ci		Yes	*NULL*		
6	**is_system**	int			Yes	0		
7	**created_at**	datetime			Yes	*NULL*		
8	**updated_at**	datetime			Yes	*NULL*		
9	**is_active**	int			Yes	0		
10	**model_group**	int			Yes	0		
11	**special_model**	varchar(50)	utf8mb3_general_ci		Yes	*NULL*		

Рис. 3.18. Структура таблиці “prompts”

Промпти підв’язані під різні проксі сервери, які надають моделі для використання, та різні конкретні моделі на цих серверах. Є офіційні міжнародні платформи які розробляють свої мовні моделі, зазвичай вони надають доступ до них через API або свої власні клієнти з доступом через персональні ключі, для прикладу OpenAI (GPT-v[version]), Anthropic (Claude [version]), Google Cloud AI/Gemini (Gemini [version]), Meta (Llama), Tongyi Lab (Qwen[version]) та інші. Або проксі сервіси, які надають доступ до різних моделей в одній екосистемі, наприклад OpenRouter [5, 124, 125].

3.4. Особливості проектування модулів формування персоналізованих рекомендацій та підтримки оцінювання відповідності НПП ліцензійним вимогам

Якщо розглядати особливість проектування модуля рекомендацій і оцінювання відповідності НПП ліцензійним вимогам, варто описати їх представлення в БД системи. Для них передбачені окремі таблиці “*user_report_recommendations*” у якій зберігаються згенеровані тексти рекомендацій для користувачів які формуються від запитів до LLM, ER-діаграма цього блоку наведено на рисунку 3.19. В таблиці “*user_report_accreditations*” містяться дані, що використовуються для визначення статусу відповідності вимогам, включаючи інформацію про відповідності діяльності користувачів критеріям ліцензійних умов за останні п’ять років, а “*user_report_field_recommend*” деталізує персоналізовані рекомендовані критерії для користувачів, з прив’язкою до сутностей системи, потенційних балів та варіативністю, яка визначається мовними моделями.

Таблиця “*user_report_recommendations*” (рисунок 3.20), яка містить тексти рекомендацій або звіти про виконання ліцензійних умов, враховуючи зворотний зв’язок (коментар від НПП з оцінкою рекомендації), та має такий набір основних атрибутів:

– id (PK): рекомендації;

– user_id (FK), report_id (FK): посилання на користувача та його звіту, якому згенерована рекомендація;

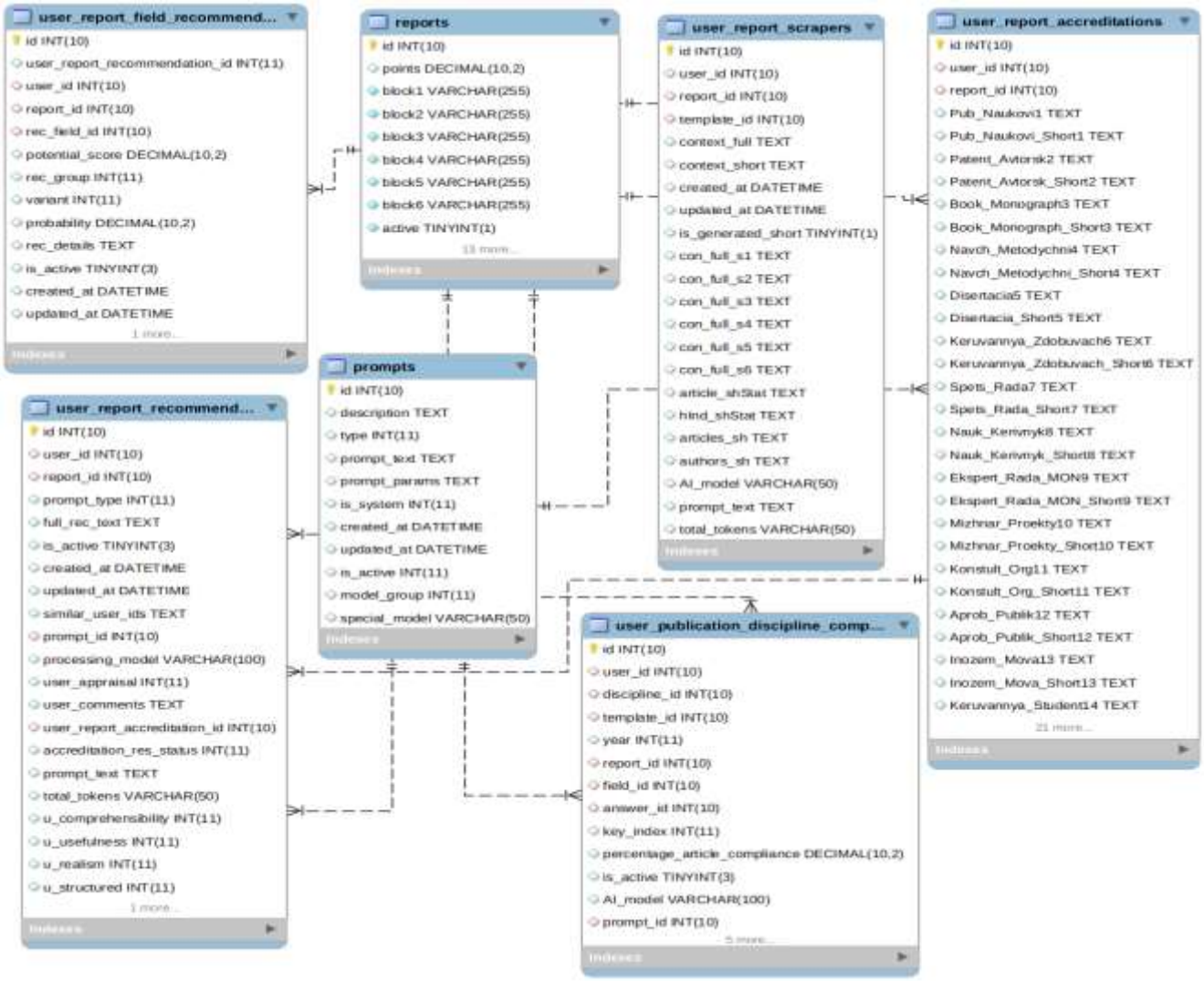


Рис. 3.19. ER-діаграма структури даних для рекомендацій та оцінювання відповідності НПП ліцензійним вимогам

– full_rec_text: повний, сформований текст рекомендації;

– is_active: статус активності (активна чи ні);

– similar_user_ids: ключі наближено кращих за діяльністю користувачів, які використовувалися для формування рекомендацій/порівнянь діяльності;

– prompt_type, processing_model: тип та модель, що використовувалася для обробки промпта;

– user_comments: коментар користувача щодо рекомендації;

– user_appraisal: оцінка рекомендації від користувача;

– user_report_accreditation_id (FK): посилання на звіт про виконання користувачем ліцензійних умов;

– accreditation_res_status: статус відповідності НПП ліцензійним вимогам;

– prompt_text, total_tokens: текст промпту та кількість токенів, використаних підчас запиту до LLM (системні стовпці, для тестування та аналізу результатів);

– u_comprehensibility, u_usefulness, u_realism, u_structured: колонки для оцінок користувача за критеріями: зрозумілість, корисність, реалістичність та структурованість рекомендації чи звіту про відповідність вимогам.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default
1	id	int		UNSIGNED	No	None
2	user_id	int		UNSIGNED	Yes	NULL
3	report_id	int		UNSIGNED	Yes	NULL
4	prompt_type	int			Yes	NULL
5	full_rec_text	text	utf8mb3_general_ci		Yes	NULL
6	is_active	tinyint		UNSIGNED	Yes	0
7	similar_user_ids	text	utf8mb3_general_ci		Yes	NULL
8	prompt_id	int		UNSIGNED	Yes	NULL
9	processing_model	varchar(100)	utf8mb3_general_ci		Yes	NULL
10	user_appraisal	int				
11	user_comments	text	utf8mb3_general_ci			
12	user_report_accreditation_id	int				
13	accreditation_res_status	int				
14	prompt_text	text	utf8mb3_general_ci			
15	total_tokens	varchar(50)	utf8mb3_general_ci			
16	u_comprehensibility	int				
17	u_usefulness	int				
18	u_realism	int				
19	u_structured	int				
20	u_relevance	int				

Рис. 3.20. Структура таблиці “user_report_recommendations”

Таблиця “*user_report_field_recommendations*” (рисунок 3.21) має такий набір основних атрибутів:

– id (PK): рекомендованого критерія;

– user_id (FK), user_report_recommend_id (FK), rec_field_id (FK): посилання на користувача, відповідної рекомендації та критерія;

– potential_score: потенційний бал, який може отримати користувач;

– rec_group, variant: група та варіант рекомендації;

– rec_details: деталі рекомендації (текст, який сформувала LLM);

– is_active: статус активності запису;

– type, prompt_type, total_tokens, special_model: системні колонки, тип рекомендації і промпту, кількість токенів, та модель.

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int		UNSIGNED	No	None		AUTO_INCREMENT
2	description	text	utf8mb3_general_ci		Yes	NULL		
3	type	int			Yes	NULL		
4	prompt_text	text	utf8mb3_general_ci		Yes	NULL		
5	prompt_params	text	utf8mb3_general_ci		Yes	NULL		
6	is_system	int			Yes	0		
7	created_at	datetime			Yes	NULL		
8	updated_at	datetime			Yes	NULL		
9	is_active	int			Yes	0		
10	model_group	int			Yes	0		
11	special_model	varchar(50)	utf8mb3_general_ci		Yes	NULL		

Рис. 3.21. Структура таблиці “user_report_field_recommendations”

Таблиця “*user_report_accreditations*” (рисунок 3.22) має такий набір основних атрибутів:

– id (PK): ліц. вимоги;

– user_id (FK), report_id (FK), prompt_id (FK): посилання на користувача, відповідного звіту та промпту;

– Pub_Naukovi1 – Prakt_Robot17: текстові поля, що містять скорочені звіти або інформацію про діяльність, яка відповідає певним критеріям ліцензійних умов;

– is_active: статус активності звіту про ліцензійні вимоги;

– generate_status: статус формування звіту про відповідність НПП ліцензійним вимогам;

– AI_model: назва мовної моделі, що використовувалася;

– prompt_text: текст промпту, який виконувався в запиті (для аналізу і тестувань).

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default
1	id	int		UNSIGNED	No	None
2	user_id	int		UNSIGNED	Yes	NULL
3	report_id	int		UNSIGNED	Yes	NULL
4	Pub_Naukovi1	text	utf8mb3_general_ci		Yes	NULL
5	Pub_Naukovi_Short1	text	utf8mb3_general_ci		Yes	NULL
6	Patent_Avtorsk2	text	utf8mb3_general_ci		Yes	NULL
7	Patent_Avtorsk_Short2	text	utf8mb3_general_ci		Yes	NULL
8	Book_Monograph3	text	utf8mb3_general_ci		Yes	NULL
9	Book_Monograph_Short3	text	utf8mb3_general_ci		Yes	NULL
10	Navch_Metodychni4	text	utf8mb3_general_ci		Yes	NULL

31	Keruvannya_Student_Short14	text	utf8mb3_general_ci
32	Keruvannya_Shkolar15	text	utf8mb3_general_ci
33	Keruvannya_Shkollar_Short15	text	utf8mb3_general_ci
34	Boiov_Dii16	text	utf8mb3_general_ci
35	Boiov_Dii_Short16	text	utf8mb3_general_ci
44	is_active	tinyint(1)	
45	generate_status	int	
46	AI_model	varchar(100)	utf8mb3_general_ci
47	prompt_id	int	
48	prompt_text	text	utf8mb3_general_ci
49	total_tokens	varchar(50)	utf8mb3_general_ci

Рис. 3.22. Структура таблиці “user_report_accreditations”

Щодо опису процесу побудови рекомендацій для користувачів (зокрема, процес пошуку подібних користувачів, розрахунку метрик та побудови ГГНМ і взаємодії з LLM), його можна розділити на кілька етапів: від початкового збору та визначення подібності даних до фінальної генерації тексту рекомендації з використанням великих мовних моделей. Цей процес, представлений на діаграмах послідовності на рисунку 3.23-3.24, дозволяє ефективно інтегрувати різні технологічні підходи та забезпечити бажаний кінцевий результат.

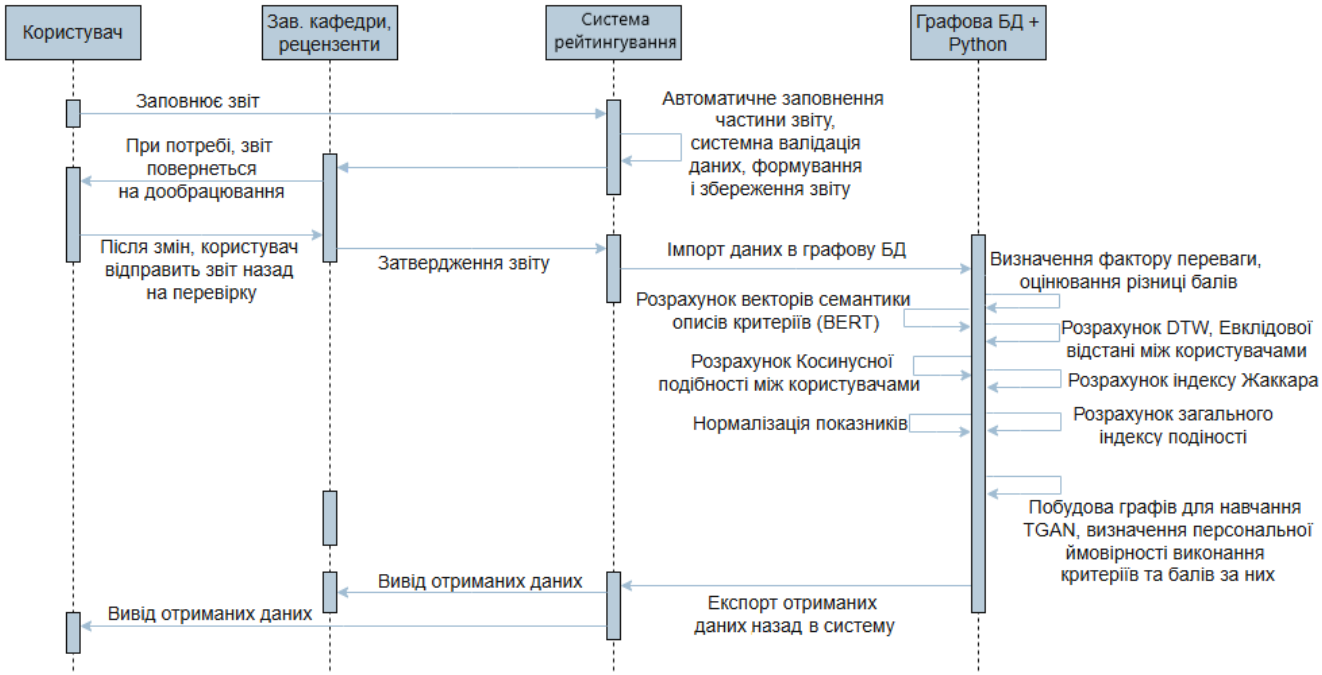


Рис. 3.23. Діаграма послідовності знаходження подібних але кращих за показниками користувачів

Почнемо зі збору даних та визначення подібності між НПП, цей процес починається після того, як всім користувачам буде затверджено їх звіти. Отримані дані імпортуються у графову базу даних Neo4j, для подальшого аналізу взаємозв’язків.

У середовищі графової БД, що функціонує з Python та бібліотеками *sklearn*, *dtaidistance*, *sentence-transformers* та інших, здійснюється комплексний розрахунок метрик подібності та розрахунок семантичних описів критеріїв за допомогою SBERT. Після цього, відбувається нормалізація показників, що забезпечує коректне порівняння даних, представлених у різних масштабах та формування загального індексу подібності. Отримані дані зберігаються в графову БД, у

відповідні колокни в вузлі “RecommendStrategyBestUsers” та використовуються для подальшого аналізу.

Наступним, розглянемо етап побудови графової нейронної мережі та безпосереднє формування персоналізованих рекомендацій. Цей процес використовує графову БД Neo4j як джерело даних та в подальшому, отримані результати, опрацьовуються з LLM (діаграма послідовності зображена на рисунку 3.24).

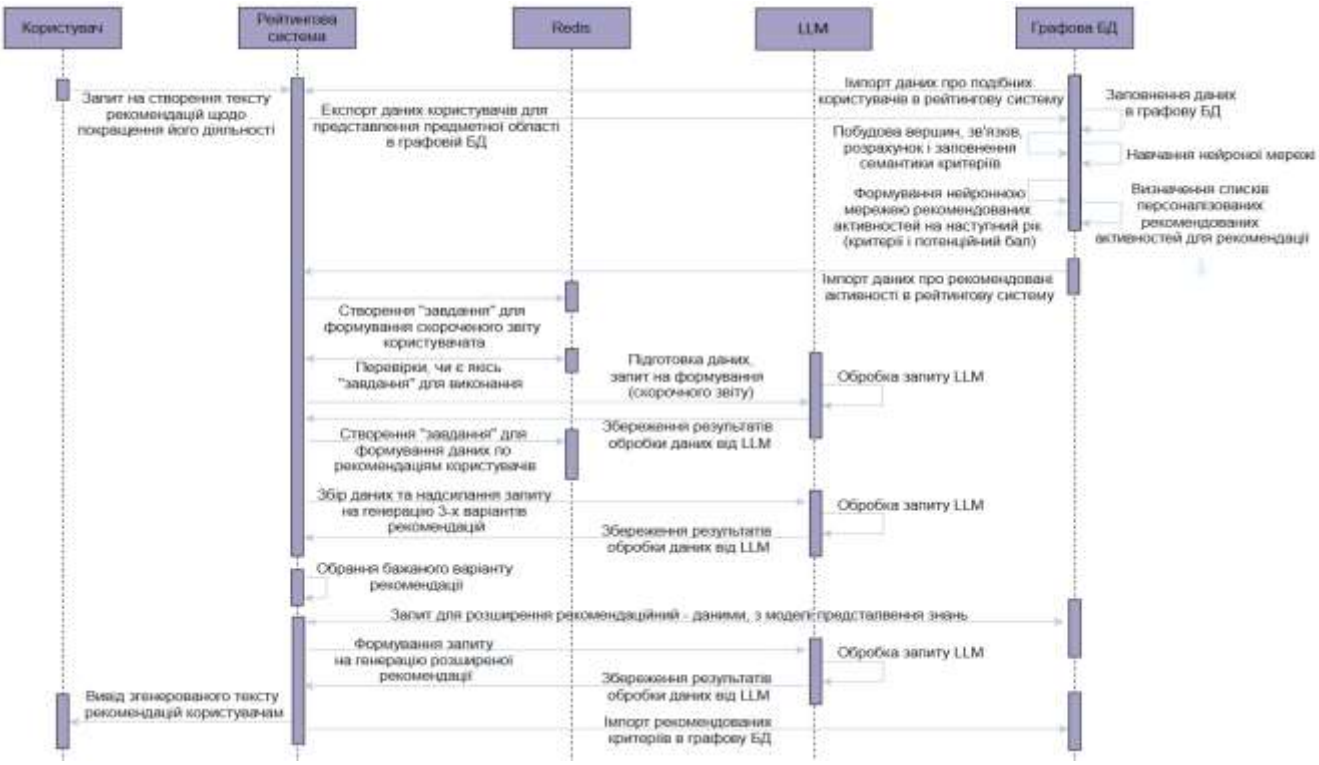


Рис. 3.24. Діаграма послідовності формування персоналізованих рекомендацій

Спочатку відбувається імпорт даних про подібних користувачів до системи рейтингування, а потім ініціюється запит до формування персоналізованих рекомендацій для НПП. У графовій базі даних здійснюється побудова вершин та зв’язків, а також розрахунок та заповнення семантичного ембедингу ознак для критеріїв яке використовується для структурованого представлення даних, необхідне для навчання графової нейронної мережі.

Для генерації рекомендацій використовується HGT з механізмами просторової та часової уваги, використанням фактору застарівання інформації, евристичним підсиленням та прогнозуванням одночасно і виконання критерію, і можливого балу [1]. Ця архітектура дозволяє аналізувати динаміку змін у діяльності НПП та

виявляти кращі шляхи для покращення діяльності. Після навчання моделі здійснюється визначення списків персоналізованих рекомендованих активностей для виконання НПП, а також прогнозування потенційних балів, які могли б отримати користувачі за них.

На фінальному етапі цього процесу, взаємодія з LLM забезпечується інфраструктурою системи рейтингування. Для підтримки динамічних запитів до реляційної бази даних імпортується дані про рекомендовані активності, отриманих з моделі знань.

При заповненні звітів частина НПП вказує надлишкову інформацію. Хоча ці дані можуть бути корисними як контекст для перевірки валідності, вони є надмірними для подальшої автоматизованої обробки даних та займають надлишкові ресурси. Тому виникає потреба “очищувати” звіти, перед наступними процесами. Перед безпосереднім формуванням тексту рекомендацій, кожного працівника, за кожен з останніх п’яти звітних періодів, виконується скорочення річного звіту діяльності за допомогою LLM. Це дозволяє оптимізувати обсяг даних для подальшої обробки мовними моделями та економити обчислювальні ресурси (токени) при запитах (формування рекомендацій, статусу відповідності вимогам тощо).

Формування тексту рекомендацій можна зобразити у вигляді наступних кроків:

– система рейтингування, через Redis формує собі чергу з запитів до мовних моделей і при готовності, створює відповідний промпт та виконує запит;

– формується виклик сценарію який скоротить звіт користувача. Він має не складний шаблон, який можна редагувати, та параметри для даних користувача;

– підготовка даних та запуск запиту для формування 3-х варіантів рекомендацій (векторів діяльності) для працівника. Цей промпт містить основні дані про поточного користувача (звання, ступінь, спеціальність, інтереси тощо), скорочений звіт поточного користувача за останні 2 роки, та звіти двох найбільш подібних користувачів (це додає додаткового опису для промпту, про діяльність поточного і подібних користувачів) та критерії які були рекомендовані нейронною

мережею (наразі формуємо топ-10 критеріїв). Кожен побудований промпт містить дані про те, що необхідно виконати, додаткові умови, зауваження, правила тощо. За допомогою цього можна генерувати якісні та змістовні тексти рекомендацій та виконувати інші операції;

– після представлення рекомендацій працівнику та після обрання ним одного з 3-х варіантів, система виконує розширення та уточнення даних для фінального тексту рекомендацій. Обрані варіантивні критерії розширюються подібними діяльностями, та надаються приклади реалізацій діяльностей з моделі представлення знань, за допомогою запиту написаного на Cypher, який отримує дані об'єднанні вз'язками, ключовими словами, тематикою, інтересами, підібними користувачами тощо.

– формується наступний запит до LLM, який розширює рекомендації, даними з моделі знань, який зберігається в базу даних для відображення користувачам.

Процес формування даних про відповідність НПП ліцензійним вимогам, реалізований за схожою структурою. Тільки, попередньо, виконується сценарій, який визначає відповідність публікацій дисципліні, яку викладає працівник. Тепер система має публікації, які можна враховувати при визначенні відповідності критеріям ліцензійних умов. Адміністратор системи має можливість додавати ці критерії до промпту пов'язаного з цим процесом, що забезпечує гнучкість у налаштуванні актуальних вимог. Результатом роботи цього модуля є звіт щодо відповідності користувача ліцейзійним вимогам, поточний статус користувача та поради щодо діяльностей, яких бракує і які він може спробувати реалізувати найближчим часом. Трохи детальніше цей процес описано в розділі 2.

Висновки до розділу 3

1. У цьому розділі було представлено архітектуру інтелектуалізованої системи рейтингування діяльності працівників, яка спроєктована на основі гібридної архітектури, що дозволило відокремити ключові функціональні

елементи, забезпечити їхню незалежність та гнучкість. Це забезпечує ефективність та масштабованість системи, а також дозволяє досягти синергетичного ефекту між її компонентами.

2. Було описано гібридну архітектуру зберігання даних, що поєднує реляційну базу даних MySQL для основних функцій системи та графову базу даних Neo4j для модуля формування персоналізованих рекомендацій. Такий підхід дозволив використати переваги кожної СУБД: надійність та простоту MySQL для структурованих даних, а також оптимізовану швидкість та гнучкість Neo4j для аналізу взаємозв'язків між користувачами та їхньою діяльністю.

3. Описано основні компоненти та модулі: інтелектуалізована система рейтингування, модулі рекомендацій, підтримки оцінювання відповідності НПП ліцензійним вимогам провадження освітньої діяльності та збору даних. Розглянуто принцип їх взаємодії між собою та з великими мовними моделями для реалізації формування персоналізованих рекомендацій, часткового автоматичного заповнення звітів і визначення відповідності ліцензійним умовам.

4. Також було наведено діаграму послідовності життєвого циклу звіту, розписані ролі користувачів у вигляді діаграми компонентів та їхній функціонал, а також механізми взаємодії з зовнішніми елементами та модулями. Загалом, розроблена архітектура забезпечує ефективне, автоматизоване та гнучке управління рейтингуванням та науково-педагогічною діяльністю.

РОЗДІЛ 4
РЕАЛІЗАЦІЯ ТА АПРОБАЦІЯ ІНТЕЛЕКТУАЛІЗОВАНОЇ ПРОГРАМНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

У цьому розділі представлено опис процесу програмної реалізації інтелектуалізованої системи рейтингування діяльності НПП, а також аналізу її ефективності.

Спершу розглядаються особливості архітектурних рішень та програмних засобів, використаних для побудови системи, що забезпечують її масштабованість, продуктивність та гнучкість. Описано функціонування ключових модулів, таких як основна система рейтингування, модуль збору даних, модуль персоналізованих рекомендацій та модуль визначення відповідності НПП ліцензійним вимогам, з акцентом на їхню взаємодію та інтеграцію з великими мовними моделями.

Далі представлено результати апробації та оцінки ефективності розробленої програмної системи, на основі даних про діяльність НПП, що підтверджують дієвість системи та її позитивного впливу на результативність працівників. Проаналізовано такі ключові показники, як обсяг даних в системі, динаміка часу заповнення звітів, потенційний приріст рейтингу НПП завдяки виконанню рекомендацій, а також продемонстровано результати тестувань автоматизованих процесів, зокрема визначення відповідності публікацій дисциплінам, що є необхідним для встановлення відповідності НПП ліцензійним умовам.

У завершальній частині розділу описано функціональні можливості та графічний інтерфейс системи з погляду взаємодії користувачів, і продемонстровано, що розроблена система є комплексним рішенням, яке об’єднує веб-технології, графові бази даних, машинне навчання та LLM для покращення результативності НПП.

4.1. Особливості реалізації системи

Процес реалізації інтелектуалізованої системи рейтингування передбачав обрання оптимальних програмних засобів та архітектурних підходів, що дозволяють забезпечити високу продуктивність, масштабованість та гнучкість функціоналу. З огляду на поставлені завдання, розробка системи здійснювалася з використанням сучасних веб-технологій, які дозволяють створити багатокомпонентну та розподілену архітектуру.

Система функціонує з 2017 року. На початковому етапі для її розробки було обрано фреймворк Laravel, який залишається актуальним і продовжує активно розвиватися й оновлюватися. Наразі використовується версія 8. Цей фреймворк працює на базі мови програмування PHP (версія 7.3). Обрання Laravel було зумовлене його архітектурою Model-View-Controller (MVC), що сприяє чіткому розділенню логіки, представлення даних та їхньої обробки. Це забезпечує легкість підтримки, розширення та тестування кодової бази. Як система управління базами даних для системи рейтингування використовується MySQL (8.0).

Взаємодія між різними модулями системи реалізована з використанням стандартизованих інтерфейсів. Наприклад, взаємодія між системою та модулем, що відповідає за персоналізовані рекомендації, відбувається за допомогою скриптів, для імпорту та експорту даних з графовою базою даних Neo4j, тоді як модуль збору даних — через авторизаційні токени. Для інтеграції з великими мовними моделями застосовується API, що забезпечує гнучкість у виборі та використанні різноманітних моделей. Наразі використовуються відомі провайдери та проксі-сервіси, що надають доступ до своїх мовних моделей через API, однак, за потреби, можливе розгортання власної моделі на локальному сервері. Реалізована інфраструктура вимагає лише наявності робочого ендпоінта до мовної моделі.

Для підвищення ефективності аналізу даних та швидкості обчислень для деяких ключових модулів було прийнято рішення використати графову базу даних та бібліотеку для машинного навчання PyTorch, замість традиційної реляційної. Це дозволило отримати значні переваги у швидкості обробки запитів, будувати власні додаткові зв’язки, сегментувати дані та використовувати розширений набір

інструментів і бібліотек для наукових обчислень, що необхідне для реалізації завдань дисертаційної роботи.

Графова база даних Neo4j відіграє роль внутрішнього середовища даних у модулі рекомендацій, що використовується для зберігання зв'язків між користувачами, критеріями, проміжними результатами математичних обчислень, взаємозв'язків з іншими сутностями тощо. Це дозволяє ефективно будувати та обробляти граф, необхідний для роботи ГГНМ.

Діаграма розгортання програмного забезпечення, яку наведено на рисунку 4.1, демонструє, що архітектура системи побудована як розподілена та модульна.

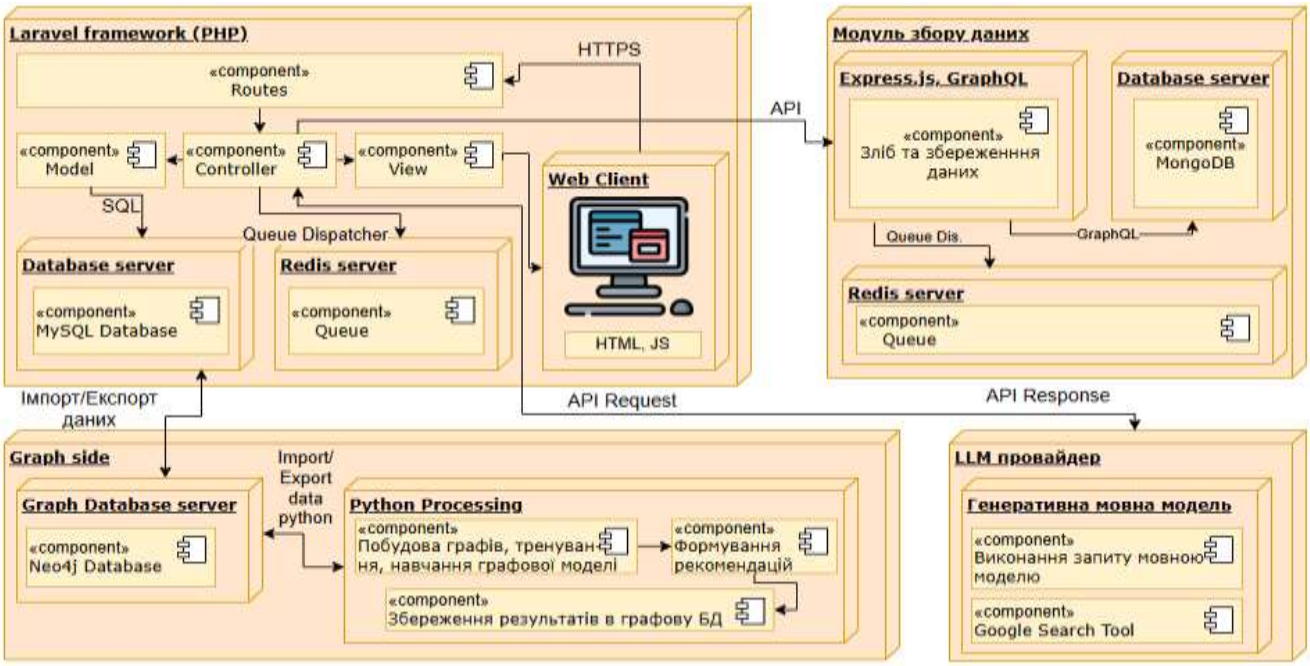


Рис. 4.1. Діаграма розгортання інтелектуалізованої програмної системи рейтингування діяльності НПП

Як видно з рисунка 4.1, система складається з наступних ключових компонентів та їх взаємодії:

– Laravel framework, PHP, який працює за принципом MVC, та включає в себе моделі (Model), контролери (Controller) та представлення (View), що реалізують основну логіку системи рейтингування. Взаємодія з базою даних MySQL здійснюється через SQL-запити. Для роботи з чергами завдань, з

рекомендаціям, звітами про відповідність вимогам чи інших процесів, використовується Redis server;

– модуль збору даних (Express.js, GraphQL, MongoDB), який відповідає за збір та збереження даних з зовнішніх джерел. Він взаємодіє з основним фреймворком Laravel через HTTPS та API за допомогою токенів для забезпечення безпечного з'єднання. Збереження акумульованих даних відбувається спочатку у базі даних MongoDB і після опрацювання вони використовуються основною системою, а для управління чергами завдань також застосовується Redis server;

– графовий сегмент, який є базою рекомендаційної частини системи, що поєднує графову БД Neo4j та реалізований функціонал мовою програмування Python. Він охоплює побудову графів, тренування графової моделі та формування персоналізованих рекомендацій із потенційними балами. Збереження результатів відбувається у Neo4j (версія 5.26.1) та взаємодія з основною системою здійснюється через імпорт/експорт даних за допомогою Python скриптів;

– провайдери LLM, які використовуються для генерації тексту рекомендацій для користувачів та інших процесів. Цей модуль відповідає за виконання запитів до великих мовних моделей та використання пошукового інструменту. Взаємодія з ним здійснюється через його API та за наявності сформованого персонального ключа;

– веб-клієнт, який є кінцевим користувацьким інтерфейсом, що відображається у браузері (за допомогою HTML, JS).

Розглянемо детальніше реалізацію ключових функціональних модулів, спираючись на наявні фрагменти коду та діаграми класів (рисунок 4.2), яка ілюструє взаємозв'язки між основними сутностями та їх атрибутами.

На діаграмі класів представлені наступні ключові класи:

– User: представляє науково-педагогічного працівника, тобто користувача системи.

– Report: відповідає за щорічний звіт користувача, що містить інформацію про його діяльність за критеріями.

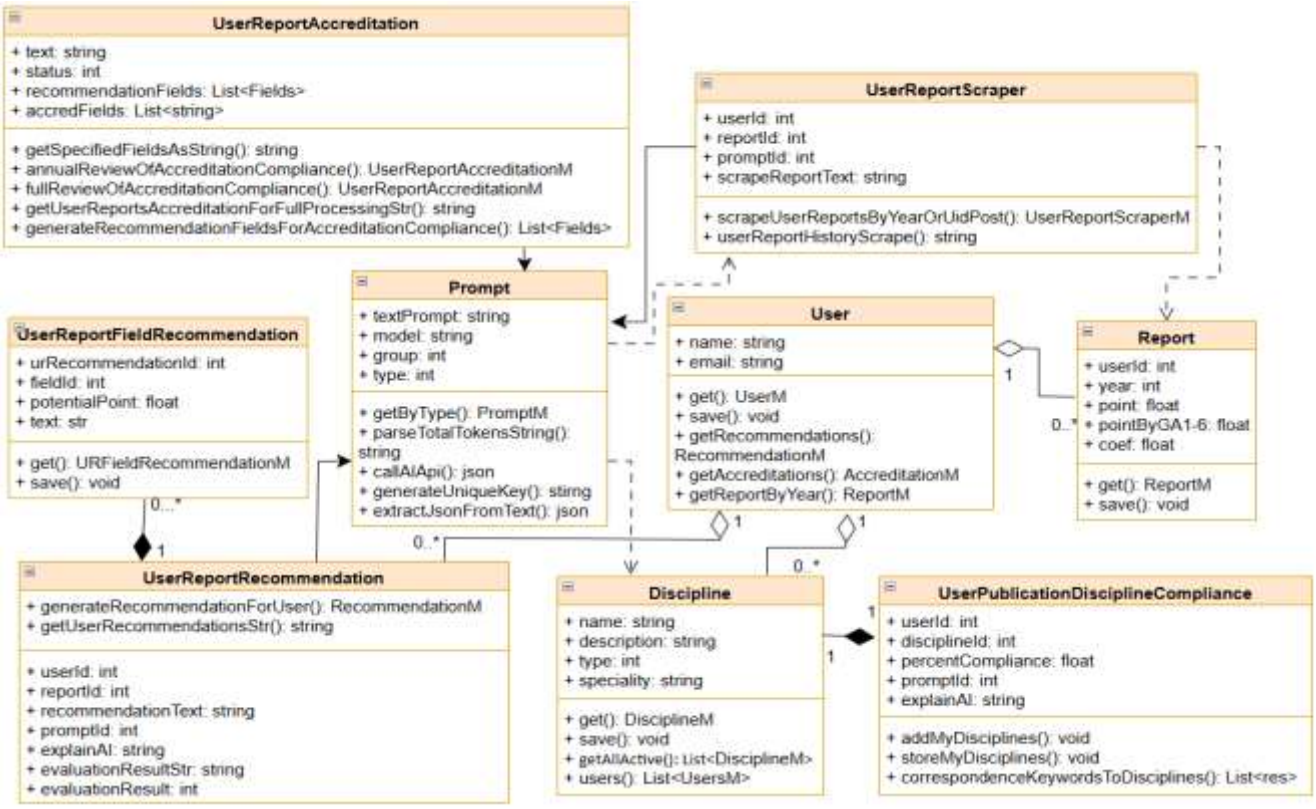


Рис. 4.2. Діаграма основних класів інтелектуалізованої програмної системи рейтингування діяльності НПП

– Discipline: описує навчальну дисципліну, включаючи її назву, мету вивчення, результати навчання, академічну програму тощо. Клас Discipline також має метод users(), що реалізує зв’язок “один до багатьох” з користувачами, які її викладають, через проміжну таблицю user_disciplines. Це необхідно для визначення відповідності статей, які публікує користувач.

– UserReportAccreditation: клас моделі, що зберігає інформацію, пов’язану зі звітом користувача про відповідність ліцензійним вимогам. Він містить такі поля, як user_id, report_id, generate_status, AI_model, prompt_id, prompt_text, total_tokens та скорочені дані по критеріях ліцензійних умов, наприклад, Pub_Naukovi_Short та Patent_Avtorsk_Short та інші, в яких міститься інформацію стосовно відповідності певним критеріям, які опрацьовуються за допомогою LLM. Для реалізації цих завдань призначений контролер AccreditationController, який містить методи для річного та повного (на основі 5 років) аналізу відповідності НПП освітній діяльності. Зокрема, він включає метод

getSpecifiedFieldsAsString, що формує представлення діяльності користувача у вигляді рядкового датасету, та ключовий метод fullReviewOfAccreditationCompliance, який відповідає за фінальну генерацію текстового звіту про відповідність ліцензійним вимогам. Приклад реалізації частини останнього наведено на рисунку 4.3.

```
Discipline.php    PromptAiController.php 1, M
app > Http > Controllers > UserReportAccreditationController.php > PHP > PromptAiController > fullReviewOfAccreditationCompliance()
25  class UserReportAccreditationController extends Controller
        0 references | 0 overrides
1087    public function fullReviewOfAccreditationCompliance(Request $request): JsonResponse|mixed
1088    {
1089        $authUser = Auth::user();
1090        if($authUser->isAdmin()) {
1091            $UserReportScraperController = new UserReportScraperController();
1092            $idRec = $request->get(key: 'idRec'); $year = $request->get(key: 'year');
1093            $userToRec = User::query()->where(column: 'id', operator: '=', value: intval(value: $idRec))->first();
1094
1095            // Processing
1096            $curLastReport = null;
1097            if($userToRec && $userToRec->id > 0)
1098            {
1099                $userRecReports = $userToRec->getReportByPeriod(yearFrom: $year-5, yearTo: $year, reportApproveStatus: ReportApproveStatus::A
1100                $userReportsDataLast5YearContextStr = $this->getUserReportsAccreditationForFullProcessingStr(userRecReports: $userRecReports)
1101                $curLastReport = $userRecReports[0];
1102
1103                if(strlen(string: $userReportsDataLast5YearContextStr) > 0)
1104                {
1105                    $userRecommendationStr = $this->getUserRecommendationsStr(curLastReport: $curLastReport);
1106                    $CallAIApiController = new CallAIApiController();
1107                    $prompt = Prompt::getByType(type: PromptTypes::AccreditationFinal);
1108                    $promptText = $prompt->prompt_text;
```

Рис. 4.3. Приклад частини реалізації методу fullReviewOfAccreditationCompliance

– UserReportRecommendation: модель, яка зберігає згенеровані персоналізовані рекомендації для користувача. Вона включає посилання на користувача та повний текст рекомендації з іншими метаданими, такими як: prompt_type, prompt_id, prompt_text, total_tokens, processing_model, is_active, user_appraisal, user_comments, user_report_accreditation_id та accreditation_res_status. Клас також має зв’язок “один до багатьох” з моделдю UserReportFieldRecommendation, що представляє окремі критерії рекомендацій. Контролер PromptAIController відповідає за генерацію рекомендацій (для прикладу фрагмент частини реалізації методу generateRecommendationForUser наведено на рисунку 4.4).

– UserReportScraper: модель, що використовується для зберігання скороченого звіту користувача. Вона працює з LLM та виконується після того, як звіт користувача повністю затверджений та розрахований річний бал.

```
app > Http > Controllers > PromptAiController.php > ...
37          if($userToRec && $userToRec->id > 0)
38          {
39              $userRecReports = $userToRec->getLastReports(2, $selectedTemplate->year, ReportApproveStatus::ApprovedByReviewers);
40              $userRecReportTextData = [];
41              foreach ($userRecReports as $key => $uRecReport) {
42                  if($curLastReport == null)
43                      $curLastReport = $uRecReport;
44                  $usText = $this->getUserDataText($uRecReport, false, $key);
45                  if($usText && $usText != null)
46                  $userRecReportTextData[] = $usText;
47              }
48              $userSimiReportTextData = [];
49              $prompt = Prompt::getByType(PromptTypes::GenerateRecommendationPerUserV1);
50              $promptText = $prompt->prompt_text;
51              $promptText = str_replace("{p1_CurrentUserSectionData_curYear}", "", $promptText); $promptText = str_replace("{p2_CurrentUse
52              $promptText = str_replace("{p3_CurrentUserSectionData_prevYear}", "", $promptText); $promptText = str_replace("{p4_CurrentUs
53              $promptText = str_replace("{p5_SimiUser1SectionData_curYear}", "", $promptText);
54              $promptText = str_replace("{p6_SimiUser1ReportData_curYear}", str_replace("\n\n", "\n", $userSimiReportTextData[0]), $prompt
55              $promptText = str_replace("{p7_SimiUser2SectionData_curYear}", "", $promptText);
56              if($promptText) {
57
58                  $CallAIApiController = new CallAIApiController();
59                  $resultCallAIApi = $CallAIApiController->callAIApi($prompt, $promptText);
60                  $response = $resultCallAIApi[0];
61                  $errors = $resultCallAIApi[1];
62                  if($response !== null) {
63                      $responceIsOk = false;
64                      $jsonResNoDecod = $response['choices'][0]['message']['content'];
```

Рис. 4.4. Приклад фрагменту реалізації методу generateRecommendationForUser

– Prompt: модель, яка зберігає інформацію про промпти (в системі — сценарії), що використовуються для виконання певних операцій над даними з LLM. Модель містить дані про провайдера, тип операції, назву і текст промпту. Метод getByType дозволяє отримати активний промпт за його типом. CallAIApiController містить функціонал для виклику мовних моделей через API з конкретним промптом.

Спосіб формування рекомендацій був описаний в розділі 2, та він складається з трьох основних етапів: збір даних і визначення працівників з максимально наближеними кращисм показниками діяльності (метод compute_similarity_index), графові нейронні мережі для генерації персоналізованих рекомендацій на наступний рік (метод buildModelWithRecommendations), та використання великих мовних моделей для формування тексту рекомендацій (реалізація на стороні системи рейтингування) з уточненням даних в моделі представлення знань. Програмний код цього процесу написаний на Python, він досить об’ємний та його основна частина наведена в ДОДАТОК Б.

Загалом, інтелектуалізована система рейтингування є комплексним програмним рішенням, що об’єднує в собі переваги сучасних веб-технологій,

графових баз даних, машинного навчання та великих мовних моделей для досягнення цілей дисертації.

4.2. Апробація системи для визначення відповідності НПП ліцензійним умовам провадження освітньої діяльності

Розглянемо послідовність кроків, що лежать в основі автоматизованого формування звіту про відповідність ліцензійним вимогам, який реалізовано в інтелектуалізованій системі рейтингування в ЗУНУ, оскільки у ЗВО постійно виникає потреба визначати, чи відповідає НПП вимогам для провадження освітньої діяльності, оскільки це безпосередньо впливає на їхню професійну легітимність, успішність функціонування та відповідність встановленим освітнім стандартам.

Відповідність ліцензійних умов визначається за сукупністю досягнень у професійній діяльності, які встановлені на державному рівні та яка враховується за останні п’ять років. Ця сукупність включає як загальні кваліфікаційні вимоги, так і конкретні показники, формалізовані у вигляді 20 критеріїв, з яких 17 — відповідають діяльності ЗУНУ. Додатковою умовою виступає відповідність освітньої або професійної кваліфікації НПП освітньому компоненту [123].

Ця відповідність може бути представлена бінарною змінною $A_u \in \{0,1\}$, де 1 означає відповідність, а 0 – невідповідність. За умовами про відповідність освітньої або професійної кваліфікації, НПП повинен відповідати наступним параметрам, які зберігаються у його профілі в системі рейтингування: наявність вищої освіти; наявність наукового ступеня та відповідність спеціальності (предметної спеціальності, спеціалізації) захищеної дисертації спеціальностям освітніх компонентів, що викладаються; наявність досвіду роботи за межами діяльності закладу не менше 5 років, або бути керівником або консультувати здобувача дисертації на здобуття певного наукового ступеня.

Для перевірки відповідності коду спеціальності дисертації та кодів спеціальностей викладацьких предметів застосовується мапінгова таблиця, що

містить історичні та актуальні коди спеціальностей. Формально, функція відповідності спеціальності $Ac_{spec,u}(D_{diss,u}, Spec)$ може бути визначена як:

$$Ac_{spec,u}\left(D_{diss,u}, Spec\right) = \begin{cases} 1, & \text{якщо } \left(d_i \in D_{diss,u}, sp_j \in Spec\right), \text{то } map(d_i) = map(sp_j) \\ 0, & \text{інакше} \end{cases}, \quad (4.1)$$

де: $D_{diss,u}$ – множина спеціальностей з дисертацій, які були захищені користувачем u; d_i – дисципліна i; $Spec$ – множина спеціальностей освітніх компонентів, які викладає u; sp_j – спеціальність j; а map() – функція, що приводить історичні/актуальні коди спеціальностей (які змінювались кілька разів) до єдиного нормалізованого формату.

Наявність хоча б однієї з перелічених вище умов, визначає відповідність НПП за цим пунктом, та позначимо це як $Ac_{spec,st}$ – яка буде відповідати бінарній змінній.

Загалом, в ліцензійних умовах [123], описані критерії, що враховують різні можливі типи діяльностей НПП (зобразимо їх як $C_{ac,1}, C_{ac,2}, \ldots, C_{ac,17}$). Кожен критерій $C_{ac,u,i}$ представлений бінарною змінною, та вказує чи користувач u виконав відповідну вимогу i.

Відповідно до ліцензійних умов, якщо $Ac_{spec,st}$ = 1, НПП повинен мати не менше 4 досягнень з 17 критеріїв, в інакшому випадку $Ac_{spec,st}$ = 0, та необхідно мінімум 8 досягнень.

Нехай $N_{u,ac}$ – загальна кількість виконаних критеріїв для відповідності ліцензійним вимогам. Тоді загальна умова може бути сформульована як:

$$A_u = \begin{cases} 1, & \text{якщо } \left(Ac_{spec,st} = 1 \;\&\; N_{u,ac} \geq 4\right) \,||\, \left(Ac_{spec,st} = 0 \;\&\; N_{u,ac} \geq 8\right) \\ 0, & \text{інакше} \end{cases} \quad (4.2)$$

де: $N_{u,ac}$ – виконані критерії, можна зобразити наступною формулою $N_{u,ac} = \sum_{i=1}^{17} C_{ac,u,i}$, де $C_{ac,u,i}$ – бінарний індикатор виконання i-го критерію працівником u.

Одним з критеріїв вимог, який найчастіше виконується та потребує додаткового аналізу даних, це наявність наукових публікацій, що відповідають профілю діяльності НПП, а саме його спеціальності. Для автоматизованого визначення відповідності тематики публікацій дисциплінам, які викладає працівник, застосовується аналіз на базі LLM. Цей підхід дозволяє проводити семантичний аналіз та оцінювати ступінь релевантності публікації до опису навчальної програми дисципліни.

Процес визначення відповідності публікації до дисципліни складається з наступних етапів:

– збір вхідних даних (ми беремо всі публікації за звітний період, їх назви, ключові слова та короткий опис (якщо він наявний), та назви дисциплін з описами їх навчальних програм, які викладав НПП;

– формування промпту для LLM. Агреговані зібрані дані використовуються у створені структурованого промпту, який передається до LLM для виконання. Цей промпт містить вхідні дані та завдання, яке полягає в тому, щоб визначити відповідність публікацій до дисципліни, на основі контексту, ключових слів та тематики цих даних [109, 110]. Зазначимо, що промпт може приймати всі публікації за звітний період для одночасної обробки та їх порівняння з однією дисципліною, тому кількість запитів до LLM буде залежати від кількості дисциплін НПП;

– LLM обробляє наданий промпт, використовуючи свої семантичні можливості для порівняння контексту публікації та опису дисципліни. За умовою створеного промпту, на виході LLM генерує числове значення, що відповідає відсотку відповідності. Результат обробки можна описати як $S_{u,accord}(Pub_j, D_k)$, де Pub_j – j публікація працівника u, D_k – k дисципліна яку має працівник. Значення $S_{u,accord}\left(Pub_j, D_k\right)$ лежить у діапазоні від 0 до 100 [104, 105];

– фільтрація публікацій, які вважаються релевантними для врахування в критеріях ліцензійних умов за певною дисципліною, якщо їхній відсоток відповідності перевищує встановлений поріг, що дорівнює 60% (це значення було встановлено емпірично, залежно від опису аналізу відповідності в промпті, це

значення можна замінити). Тобто, публікація Pub_j зараховується до D_k, якщо $S_{u,accord}(Pub_j, D_k) \geq 60$. Ці відповідності зберігаються у базі даних для подальшого використання.

Для ілюстрації процесу аналізу відповідності наведено схему, яка зображена на рисунку 4.5, що демонструє потік даних від вхідних джерел, взаємодії в LLM та до моменту збереження результатів.

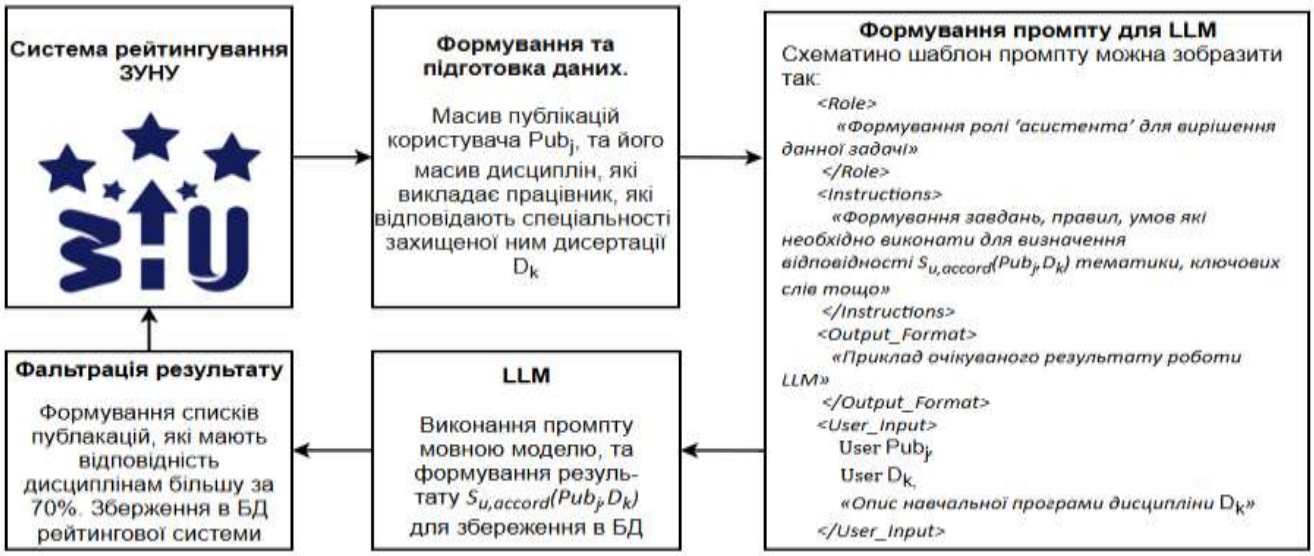


Рис. 4.5. Схема процесу аналізу відповідності публікацій НПП його дисциплінам

Після таких процесів, починається формування агрегованого звіту для оцінювання відповідності НПП ліцензійним вимогам, який охоплює аналіз діяльності НПП за останні п'ять років, інтеграцію персоналізованих рекомендацій (R_u) для прогнозування відповідності вимогам у наступному звітному періоді, та сформовані дані щодо відповідності цим вимогам та поточний статус працівника.

Детальніше опишемо цей метод, який включає наступні етапи:

1. Для кожного річного звіту ($Rep_{u,t}$), по академічній діяльності, мовна модель визначає, які з 17 ліцензійних критеріїв були виконані НПП протягом відповідного року та в якій кількості. Умовно, процес можна зобразити як $Rep_{ac,u,t} = LLM(Prompt_{accred}(Rep_{u,t}, C_{ac}))$, де: $LLM()$ – це мовна модель, а саме опрацювання запиту; $Prompt_{accred}()$ – створений промпт для визначення чи НПП відповідає вимогам; C_{ac} – це множина критерів для дотримання ліцензійних умов $\{C_{ac,1}, C_{ac,2},$

…, $C_{ac,i}\}$, i – загальна кількість критеріїв. В результаті отримуємо звіт ($Rep_{u,t}$) з діяльністю, яка відповідає ліцензійним критеріям за кожний річний звіт $t \in [T-4, T]$, де T – поточний річний звіт.

2. Зібрані за 5 років звіти виконання критеріїв агрегуються для формування повного звіту освітньої діяльності, що буде відображати загальну картину досягнень НПП за вказаний період. Ці звіти формують фінальний звіт за визначений період, та це можна представити як логічне "І": $Rep_u = Rep_{u,T-4}$ & $Rep_{u,T-3}$ & $Rep_{u,T-2}$ & $Rep_{u,T-1}$ & $Rep_{u,T}$, де Rep_u – агрегований звіт про діяльність працівника, що враховується у ліцензайних умовах.

3. Тепер, на основі агрегованого звіту (Rep_u), виконаних діяльностей та згенерованими персоналізованими рекомендаціями (R_u) щодо діяльності на наступний рік, система визначає статус відповідності лійензійним вимогам для НПП. Цей етап також використовує LLM для комплексного аналізу та формування текстового висновку. Промпт для LLM, включає дані поточного працівника u, ліцензійні вимоги (A_u), персональні рекомендації на наступний рік (R_u), агрегований звіт (Rep_u) та опис того як потрібно врахувати умови та виконувати аналіз даних працівника. Формулою цей етап можна представити так: $AccStatus_u = LLM(Prompt_{totalAccred}(Rep_u, C_{ac}, R_u))$. В результаті формується один з трьох статусів:

– "Відповідає ліцензійним вимогам": A_u=1 вже на даний момент;

– "Може відповідати ліцензійним вимогам": A_u = 0, але ($AccStatus_u$) повертає R_u для виконання користувачеві. Це означає, що при виконанні рекомендованих видів діяльності, які були попередньо сформовані, НПП буде відповідати ліцензайним вимогам в наступному році;

– "Не відповідає ліцензійним вимогам": A_u = 0, і навіть якщо користувач виконає всі рекомендовані йому діяльності, він не задовольнить вимоги, тому йому потрібно буде виконути додаткові діяльності.

Крім того, за запитом в промпті, LLM формує поради та пояснення щодо того, яких саме критеріїв бракує НПП для задоволення необхідних вимог, надаючи

інформацію про конкретні невиконані пункти, діяльності, яких простіше досягти в наступному звітному році [104, 105].

Для візуалізації цього методу, на рисунку 4.6, наведено діаграму послідовності процесу формування даних по відповідності цим вимогам, яка відображатиме взаємодію елементів в системі, включаючи взаємодію з LLM.

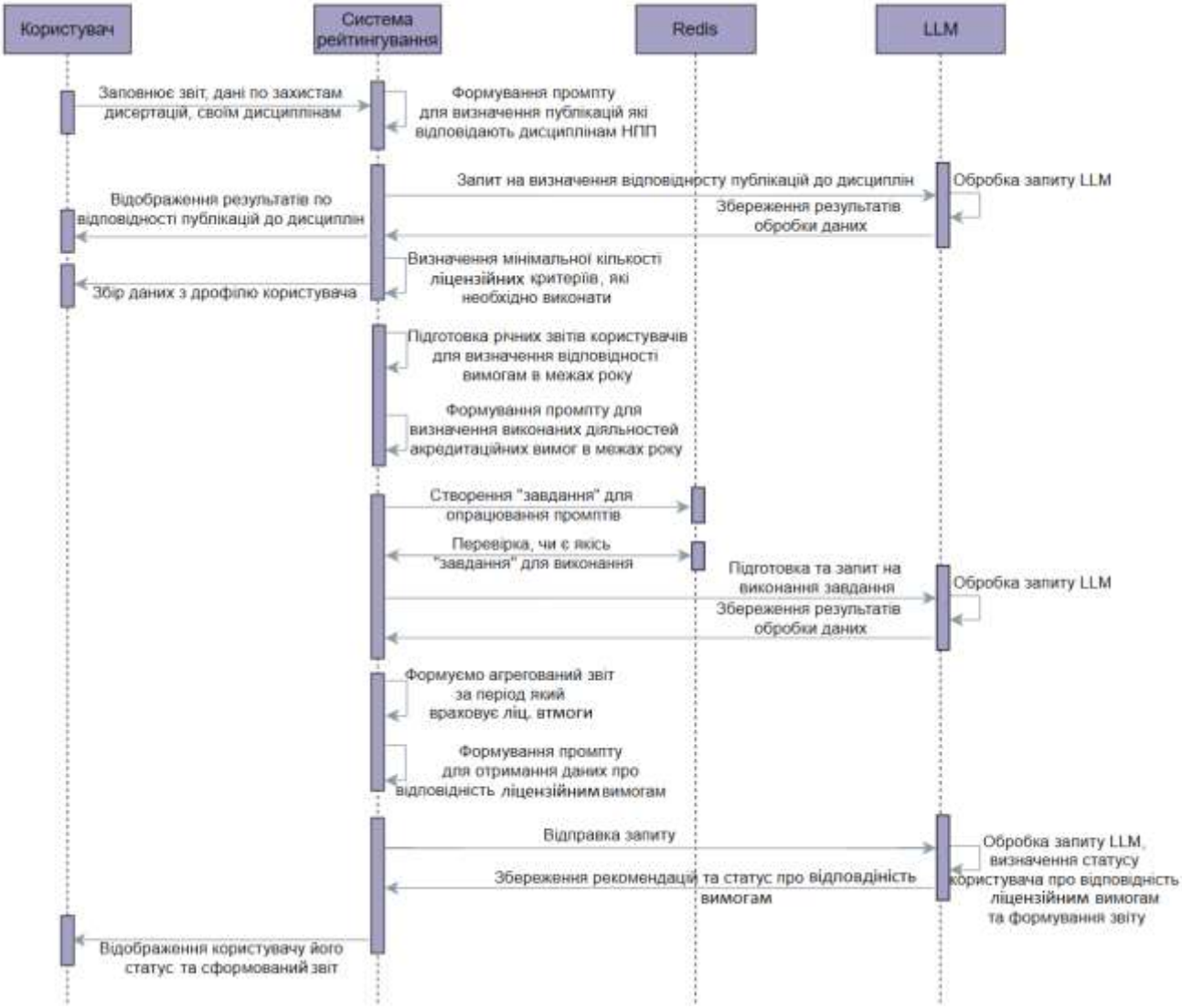


Рис. 4.6. Діаграма послідовності формування даних про відповідність НПП ліцензійним вимогам

4.3. Функціональні можливості та графічний інтерфес програмної системи

Тепер можна розглянути організацію інтерфейсу та функціонування системи. Важливо зазначити, що деякі модулі, такі як модуль збору даних та графова база

даних, функціонують переважно у фоновому режимі та не мають прямого користувацького інтерфейсу, тому у цьому підрозділі акцент буде зроблено на взаємодії з інтерфейсом інтелектуалізованої системи рейтингування.

Щоб скористатися системою, скористувачам необхідно пройти авторизацію, яка доступна за адресою https://rating.wunu.edu.ua, де, залежно від налаштувань акаунта, відбувається розмежування доступу до функціоналу системи [12, 67]. Після успішної авторизації (користувачу достатньо ввести тільки емейл та пароль до свого акаунта), інтерфейс системи адаптується до рівня доступу користувача, відображаючи відповідні елементи меню та функціональні можливості, що можна побачити на рисунку 4.7. Система передбачає декілька ключових ролей: адміністратор (1), завідувач кафедри (2), рецензент (3) та користувач чи НПП (4).

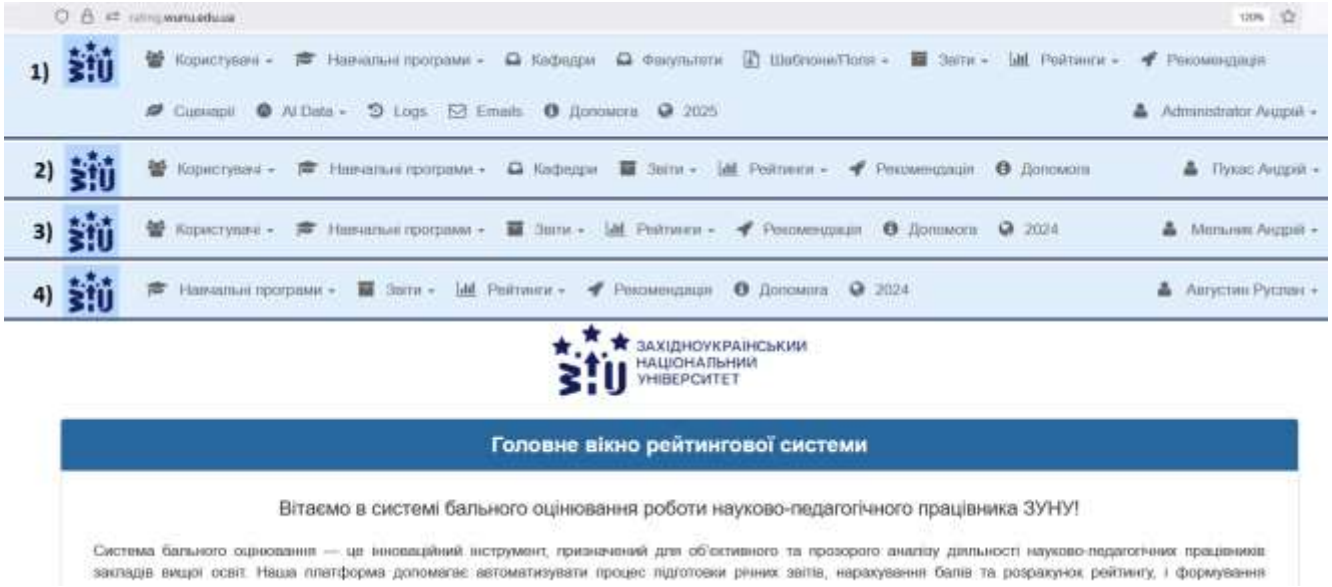


Рис. 4.7. Порівняння інтерфейсу меню для різних ролей

Користувачі мають можливість заповнювати та оновлювати свої персональні публічні дані, інформацію про освіту, наукові ступені, вчені звання, а також визначати перелік дисциплін, які вони викладають, у своєму профілі (на вкладці “Профіль користувача”), що наведено на рисунку 4.8. Ці дані важливі для подальшого формування звітів відповідність освітній діяльності та персоналізованих рекомендацій.

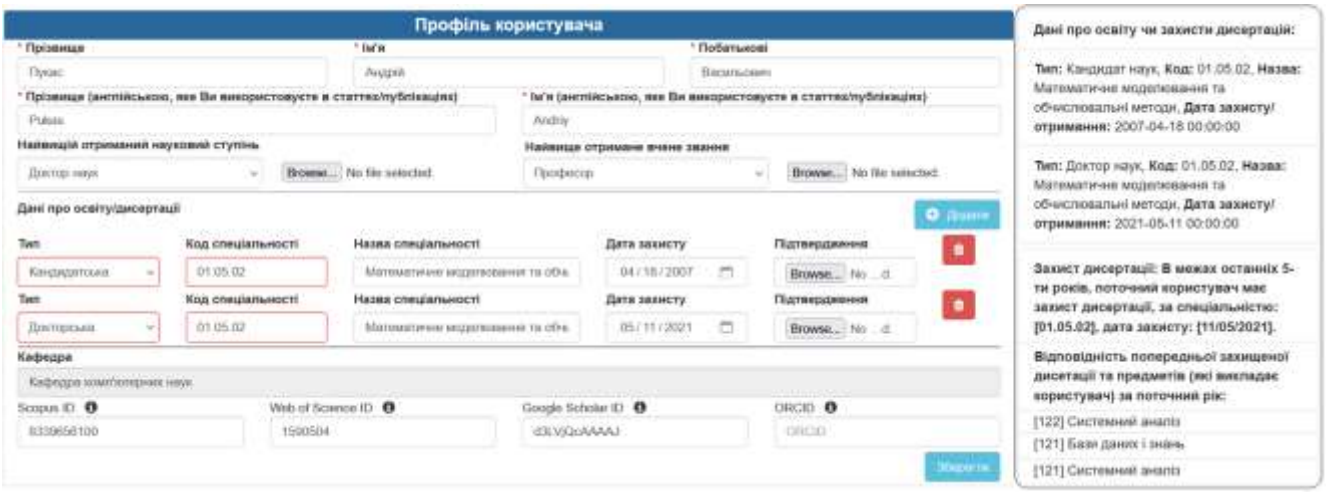


Рис. 4.8. Заповнений профіль користувача, з даними про дисертації

Процес створення звіту розпочинається з визначення критеріїв оцінювання. Адміністратор може здійснювати налаштування критеріїв оцінювання діяльності: наповнювати їх даними, обирати спосіб обрахунку балів, підв'язувати до групи активності, додавати опис та інше. Процес створення нового критерію, наведено на рисунку 4.9. Після формування критеріїв та створення звіту, користувачі можуть їх заповнювати, додаючи діяльність, яку вони виконали протягом звітного року, це доступно в вкладці "Звіти", рисунок 4.10.

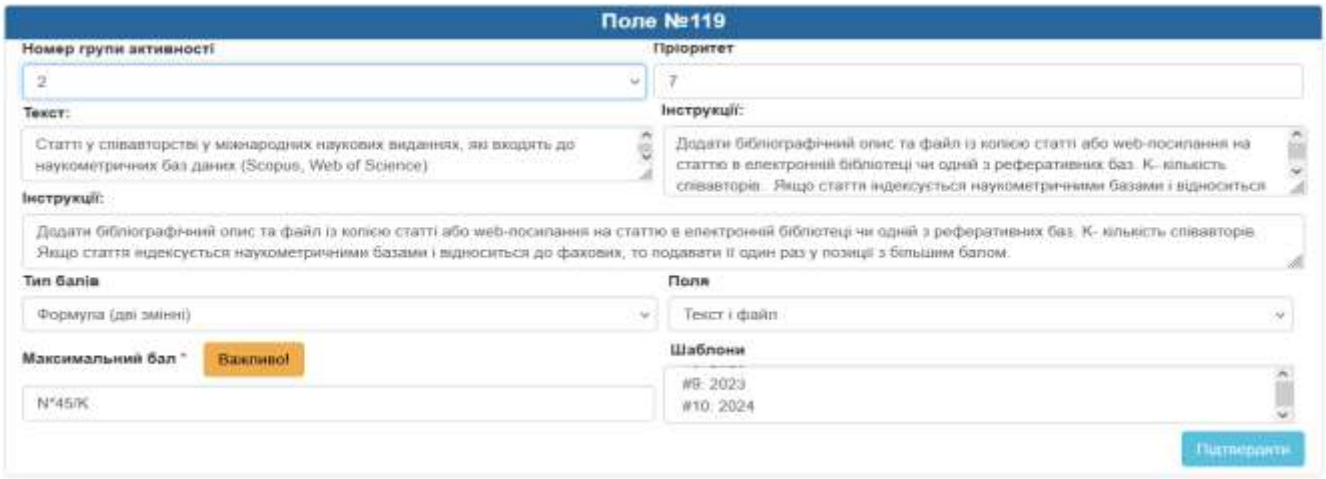


Рис. 4.9. Приклад створення критерію з різними типами та формами нарахування балів

Система, шляхом використання даних з профілю користувача, модуля збору даних та мовних моделей, може автоматично заповнювати частину критеріїв звіту.

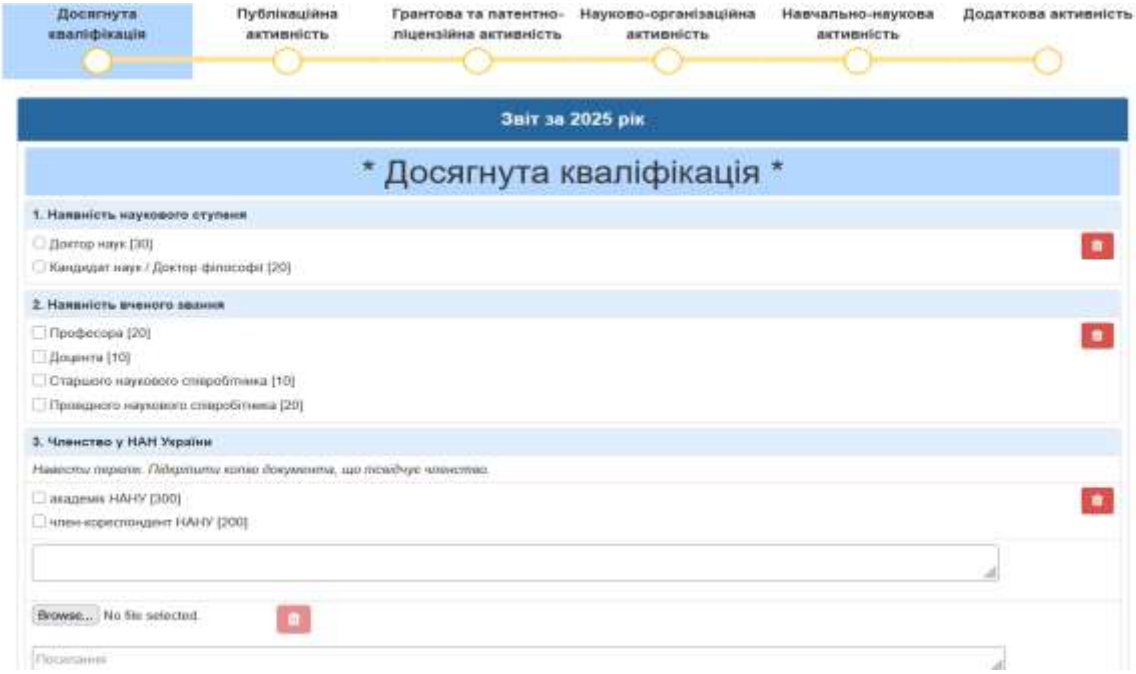


Рис. 4.10. Створений річний звіт, готовий до заповнення користувачем

Більш детально цей процес було описано раніше. Такі автоматично заповнені критерії позначаються окремо, що спрощує їх перевірку завідувачем кафедри та рецензентами (приклад автозаповнення зображено на рисунку 4.11).

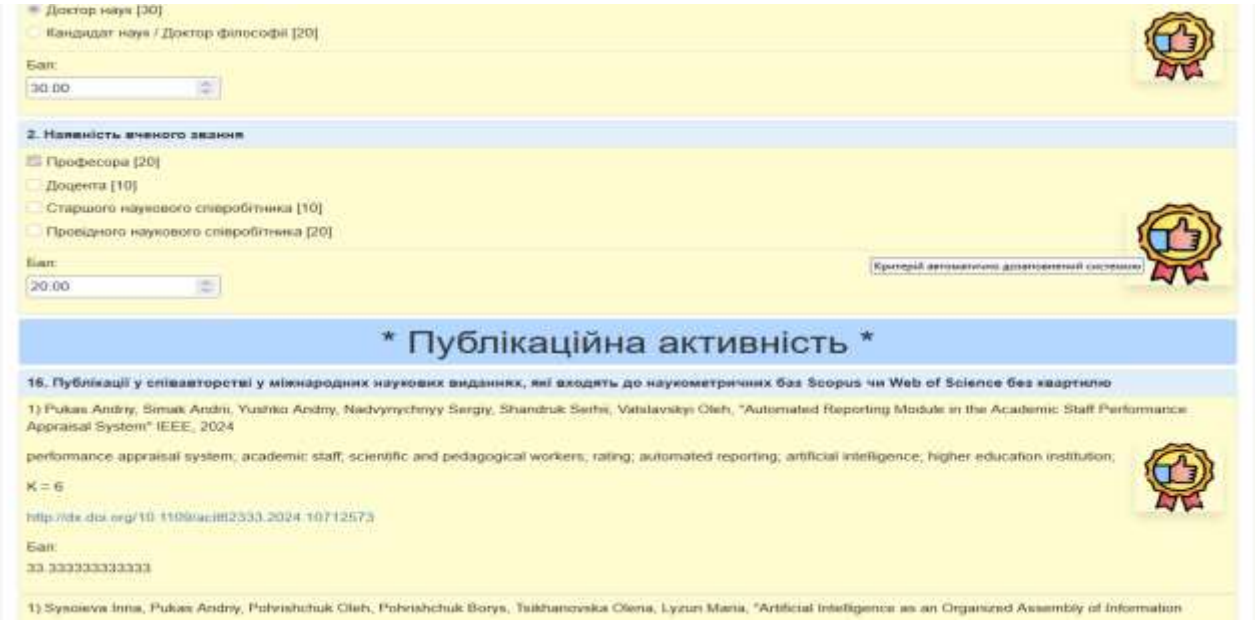


Рис. 4.11. Приклад автозаповнених критеріїв у звіті користувача

Крім того, дані, які система не змогла сама точно ідентифікувати, можна додати до звіту вручну (рисунок 4.12).

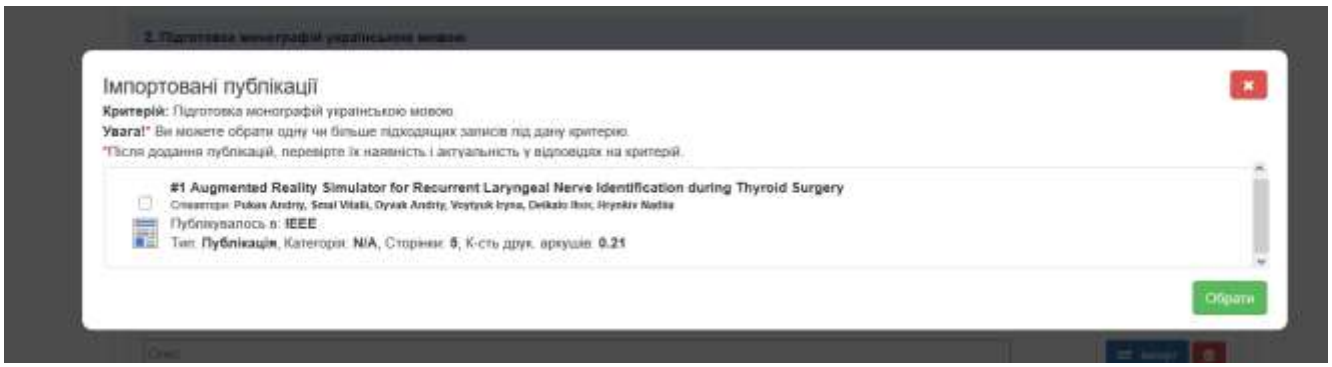


Рис. 4.12. Доповнення звіту акумульованими даними користувача

Як уже було розписано раніше в життєвому циклі звіту, після заповнення користувачами, вони відправляються на перевірку до завідувача кафедри і рецензентів. Вони обидва можуть їх переглядати, додавати зауваження і затверджувати звіт. Необхідність у ролі рецензента виникла після кількох років використання системи, оскільки частина користувачів (НПП, зав. кафедри) допускали помилки при заповненні і перевірці звіту, і щоб частково покращити цю ситуацію — було додано проміжний варіант валідації, у вигляді рецензентів. На рисунку 4.13 зображено сторінку перевірки звіту рецензентом. У завідувачів кафедр схожий вигляд інтерфейсу (без можливості блокувати критерії).

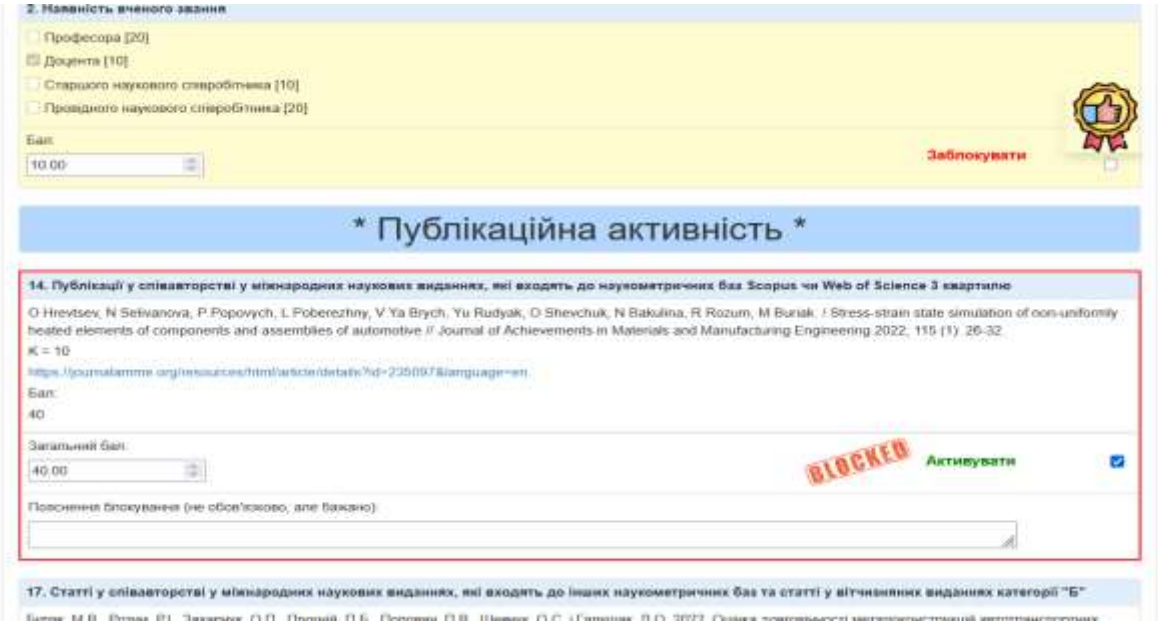


Рис. 4.13. Інтерфейс сторінки перевірки звіту рецензентом

Після повного заповнення звіту і його затвердження, обчислюється річний бал

користувача та результати відправляються в загальний рейтинг, рисунок 4.14. Він доступний у вкладці “Рейтинги”, з можливістю переглядати їх за різними роками, за формулою в межах 5-ти років (детальніше розглядали в розділі 2). Крім загального прогресу у вигляді відсоткового приросту балів (рисунок 4.14), є можливість у вигляді візуальних графіків, відстежувати бали за групами активності відносно максимально набраних балів по користувачам. Також є статистика та розрахунок співвідношення кількості заблокованих критеріїв між користувачами, що дозволяє візуально простіше аналізувати діяльність НПП. Приклад вигляду статистики, наведено на рисунку 4.15. Структура зв’язків між користувачами, кафедрами та факультетами забезпечує можливість аналізувати результативність НПП — як на індивідуальному рівні, так і в розрізі підрозділів. (рисунок 4.16).

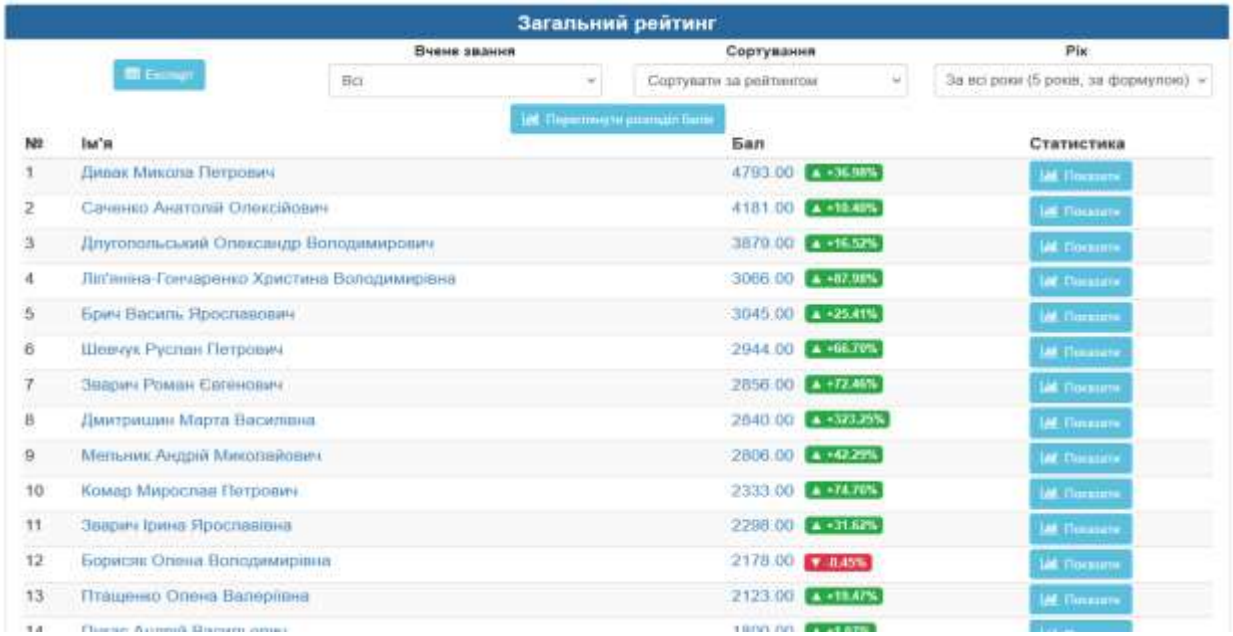


Рис. 4.14. Загальний рейтинг користувачів

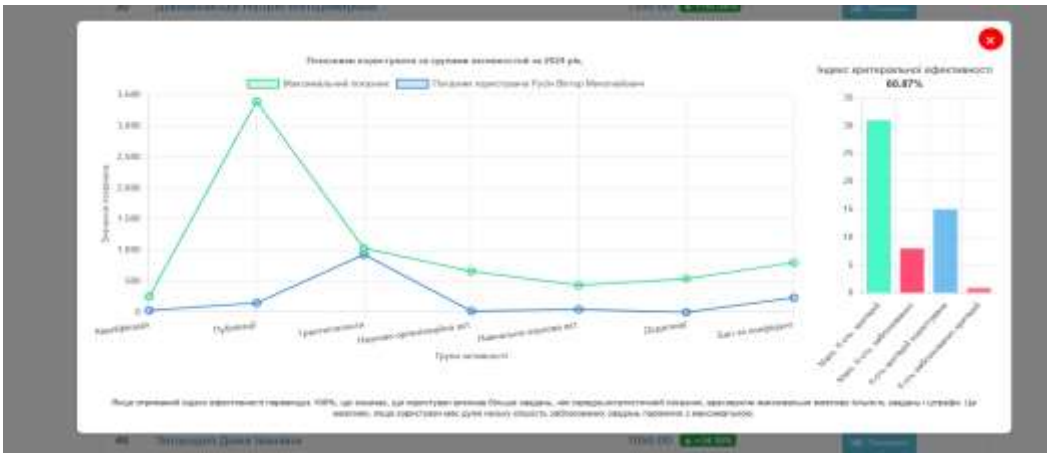


Рис. 4.15. Приклад більш детальної статистики по групам активності користувача

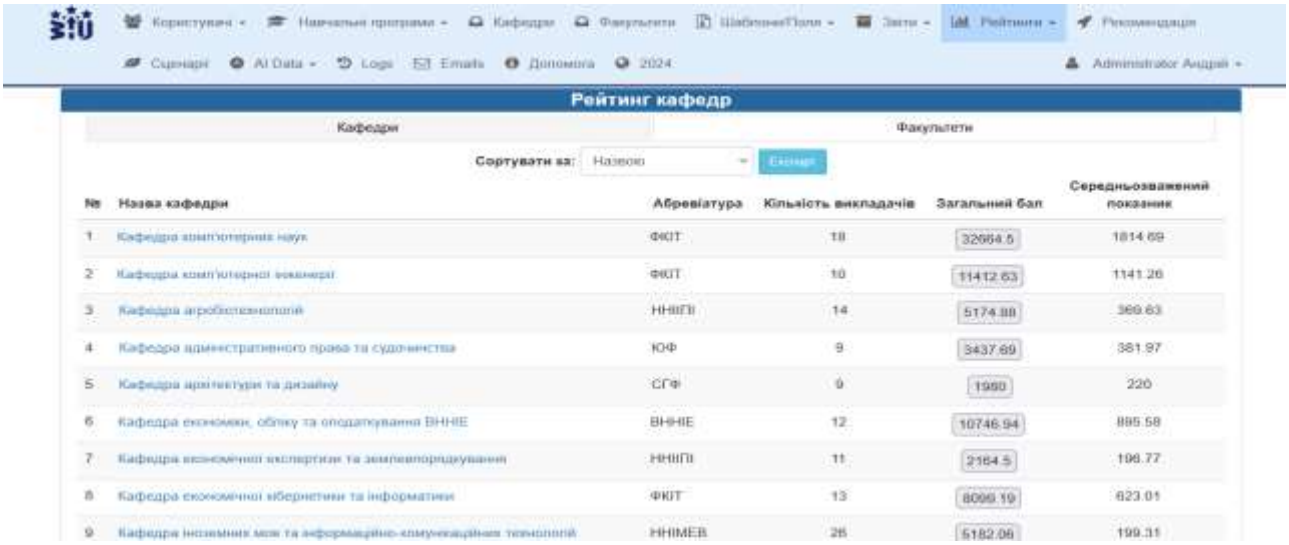


Рис. 4.16. Сторінка з рейтингами кафедр

Персоналізовані рекомендації формуються після заповнення всіх даних на початку нового звітного року та запускається процес формування рекомендацій для користувача та інших процесів, імпорт даних в графову БД тощо, (рисунок 4.17).

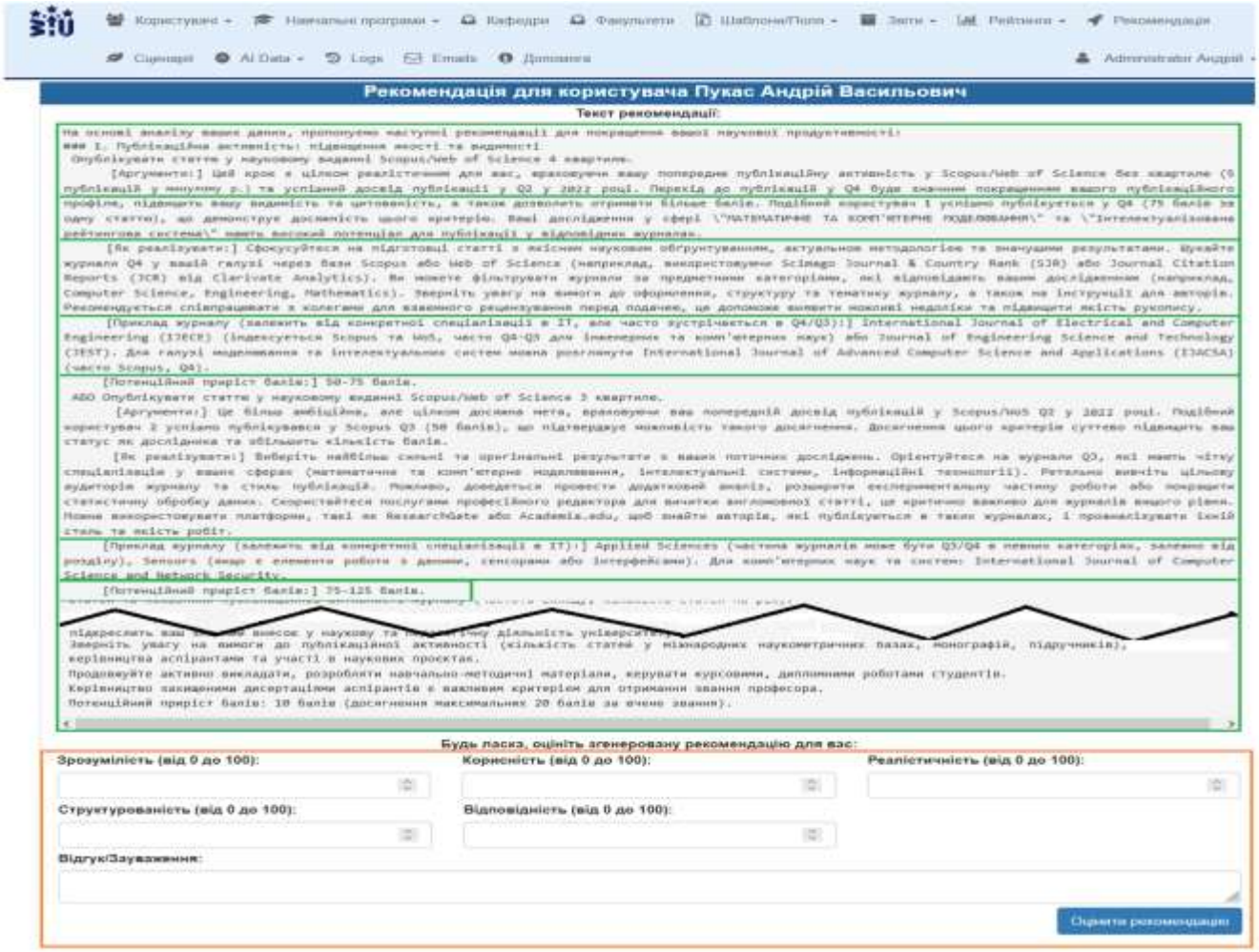


Рис. 4.17. Персональні рекомендації для користувача на наступний рік

Це відбувається у фоновому режимі, з використанням черг та додаткових модулів. Після цього, результати стають доступні на вкладці “Рекомендації”, приклад сторінки наведено на рисунку 4.18.

Рис. 4.18. Відображення частини звіту про відповідність користувача ліцензійним вимогам

Як видно з вищенаведених рисунків, система виводить текст рекомендацій для користувача (зелений блок, рисунок 4.17), та надає можливість залишити оцінку та коментар для зворотного зв’язку (помаранчевий). Це додатковий функціонал для аналізу результатів формування рекомендацій. Звіт про відповідність НПП ліцензійним вимогам має схожий інтерфейс (рисунок 4.18). Стає доступним для користувача, після формування рекомендацій з можливістю залишити окремий відгук про нього.

Додатково, система автоматизує процес визначення відповідності публікації до дисципліни користувача у відсотковому значенні, що є результатом опрацювання даних за допомогою LLM. Публікації, що мають 60 відсотків відповідності й вище, враховуються у звіті про освітню діяльність НПП (рисунок 4.19).

Адміністратор системи має розширені можливості для налаштування роботи LLM, а саме створення і редагування сценаріїв промптів, що зображено на рисунках 4.20-4.21. Це забезпечує гнучкість у адаптації системи до нових вимог та

Рис. 4.19. Демонстрація відповідності публікації дисципліні користувача

оптимізації якості генерованих рекомендацій та звітів. Також адміністратор може переглядати статистику за сформованими логами щодо роботи LLM за різними сценаріями: “скорочення звіту”, “формування рекомендацій”, “визначення відповідності предмету до дисципліни” та “звіт про відповідність ліцензійним умовам” тощо, в вкладці “Logs”. Це системна вкладка, як і “AI Data” та “Emails”, до яких є доступ тільки у адміністратора.

Рис. 4.20. Вкладка перегляду активних сценаріїв

Підсумовуючи, функціональні можливості розробленої інтелектуалізованої системи рейтингування охоплюють повний цикл управління діяльністю працівників: від збору та структуризації даних, часткового автоматизованого заповнення звітів, багаторівневої системи їх затвердження, та до формування

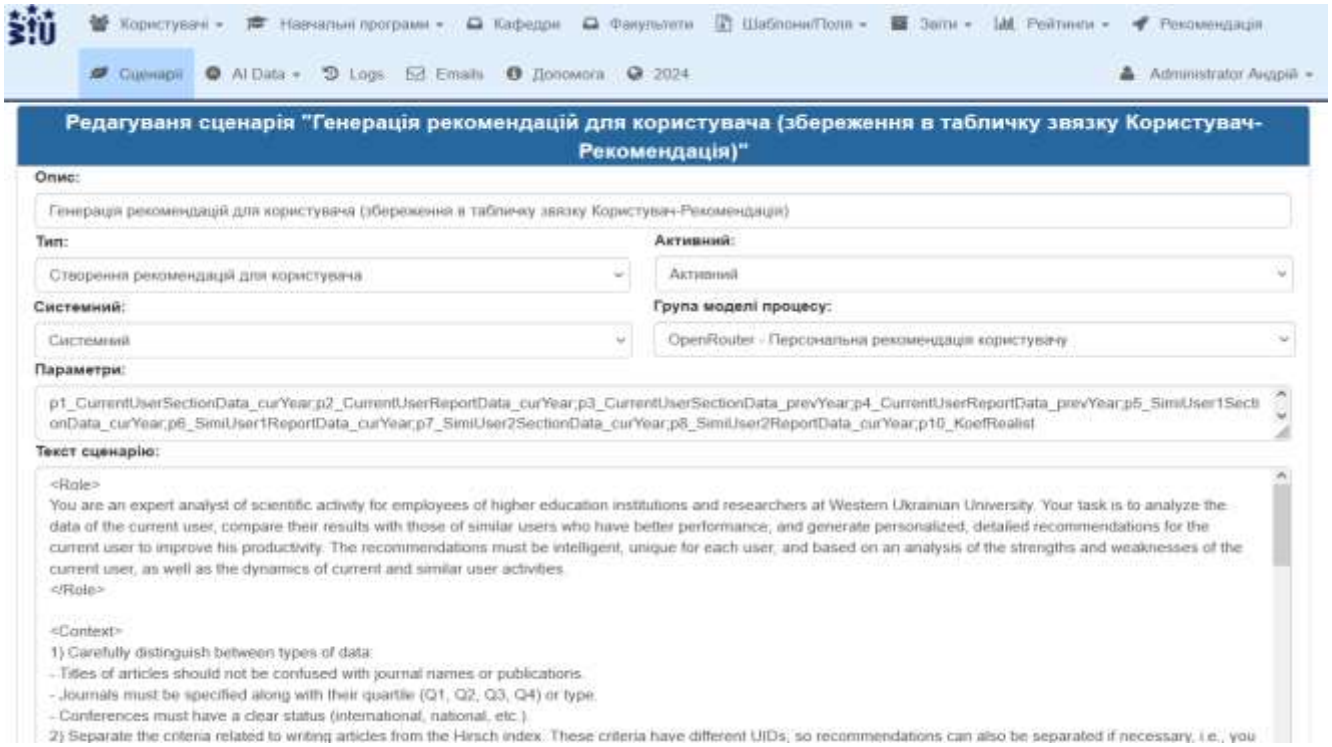


Рис. 4.21. Сторінка з редагуванням промпту для формування персоналізованих рекомендацій

комплексних рейтингів, персоналізованих рекомендацій та інших звітів про діяльність працівників. Інтерфейс системи розроблений з акцентом на зручність використання для різних ролей, а інтеграція передових технологій, таких як LLM та графові нейронні мережі, забезпечує високу ефективність та точність аналізу та прогнозування. Це дозволяє суттєво покращити результативність діяльності НПП, автоматизувати рутинні процеси та допомогти у прийнятті стратегічних рішень на різних рівнях управління.

4.4. Апробація та оцінка ефективності роботи програмної системи

Процес апробації та оцінки ефективності розробленої програмної системи здійснювався на базі Західноукраїнського національного університету, де виникла потреба вдосконалити наявну систему рейтингування шляхом її інтелектуалізації, автоматизації окремих процесів та, як наслідок, підвищення результативності НПП. Період апробації охоплював діяльність НПП з 2016 по 2024 роки, хоча

безпосередньо для реалізації процесу формування персоналізованих рекомендацій використовувались дані системи рейтингування з 2018 року. На момент запуску системи, вона мала вкрай обмежений функціонал, та частина звітів в перші роки були заповнені та перевірені некоректно, тому наявні не зовсім коректні дані. Тому, щоб зменшити кількість неправильної інформації під час навчання нейронної мережі, кілька перших років були пропущені, приблизно з 2018 року заповнені звіти є переважно коректними.

Якщо говорити про кількісні показники, то наразі в системі є понад 900 акаунтів користувачів, з яких активних практично дві третини. Понад сотня професорів, понад три сотні доцентів, та інші наукові співробітники налічують користувачів системи рейтингування ЗУНУ. Інфраструктура з кафедрами та факультетами нараховує понад пів сотні кафедр, та 14 факультетів, які мають своїх деканів і зав. кафедрів та наявних працівників, які відносяться до відповідних підрозділів. Цією структурою можна вільно керувати з інтерфейсу системи, при наявності відповідних прав доступу.

Кількість критеріїв, які враховувались і оцінювались в університеті також змінювалась протягом часу. Наприклад на початку роботи системи, було додано понад 60 критеріїв, а в деякі роки, їх кількість була більша за 100. Зміна кількості критеріїв оцінювання в системі за роками наведено в таблиці 4.1.

Таблиця 4.1

Кількість критеріїв в різні звітні роки

Рік	2016	2017	2018	2019	2020	2021	2022	2023	2024
Кількість критеріїв	64	83	100	105	105	100	72	72	70

Також варто вказати як змінилася середня кількість виконаних критеріїв за роками, щоб уявляти кількість діяльності яка виконується працівниками, обсяги даних та їх розрідженість. Як видно з рисунка 4.22, після періоду зростання (2018-2019) та наступного спаду (2020-2023), у 2024 році спостерігається незначне, але позитивне зростання середньої кількості виконаних критеріїв.

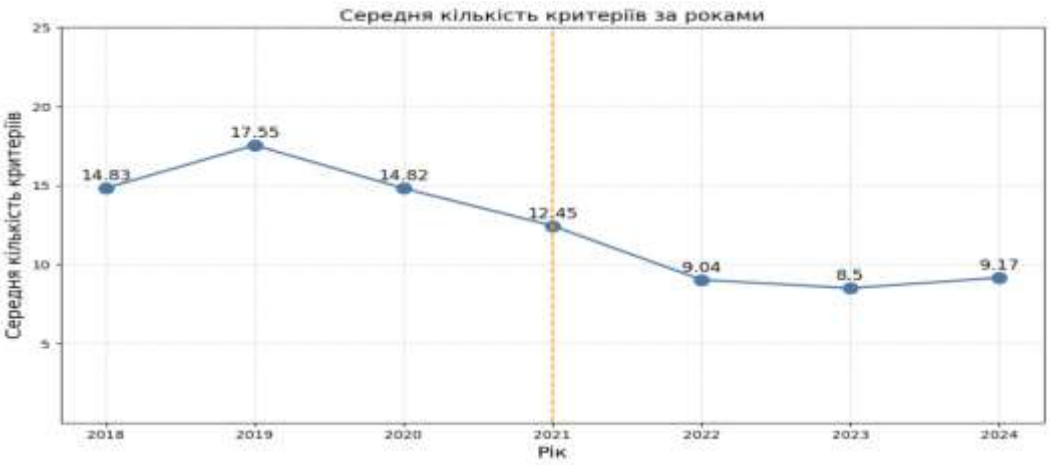


Рис. 4.22. Середня кількість виконаних критеріїв за період використання системи

Щоб продемонструвати кількість наявних даних, було проаналізовано обсяг зібраних та оброблених даних за період існування системи. На рисунку 4.23 зафіксовано динаміку кількості поданих звітів. Зокрема, спостерігається майже постійне зростання кількості звітів з 2016 по 2024 рік, що вказує на збільшення активності НПП та про сталу інтеграцію системи в академічне середовище.

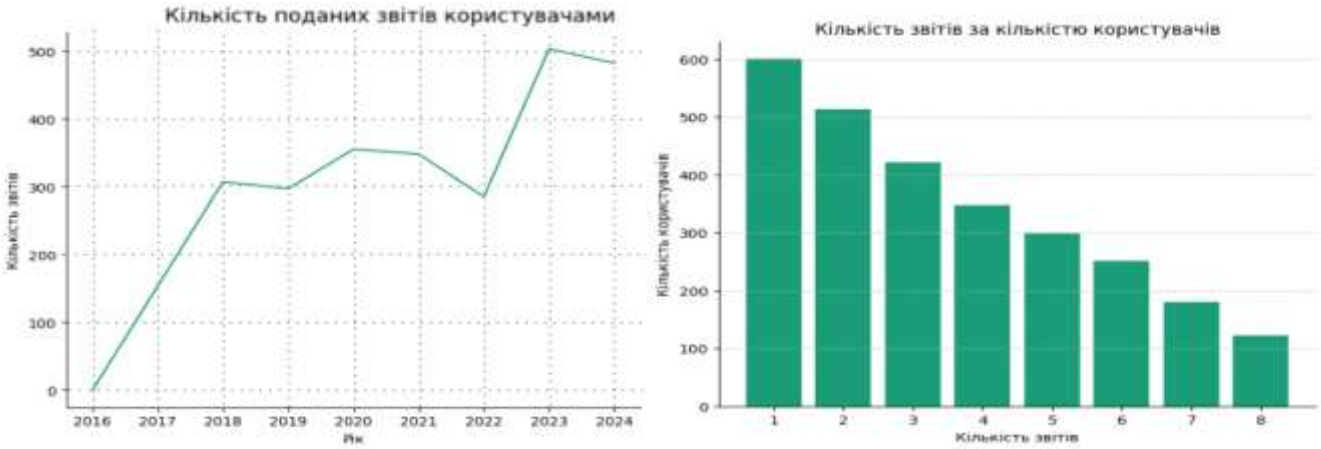


Рис. 4.23. Динаміка кількості поданих та наявних звітів користувачами за час існування системи рейтингування

Аналіз кількості звітів, поданих одним користувачем, показав, що значна частина НПП активно користується системою протягом декількох років. Наприклад, 600 активних користувачів мають мінімум 1 поданий звіт, понад 300 подали по 5 звітів, а трохи більше 100 — мають по 8.

Також, щодо розрідженості даних, можна навести графік розподілу балів між користувачами (рисунок 4.24), який демонструє, що більшість користувачів мають низькі та середні бали (оскільки виконують простіші або менш важливі діяльності для ЗВО), а невелика група має значно вищі показники, що підтверджує необхідність впровадження інструментів для підвищення результативності працівників. Персональні рекомендації мають потенціал сприяти “підтягуванню” показників менш активних працівників та підтримувати високі результати у лідерів рейтингу.

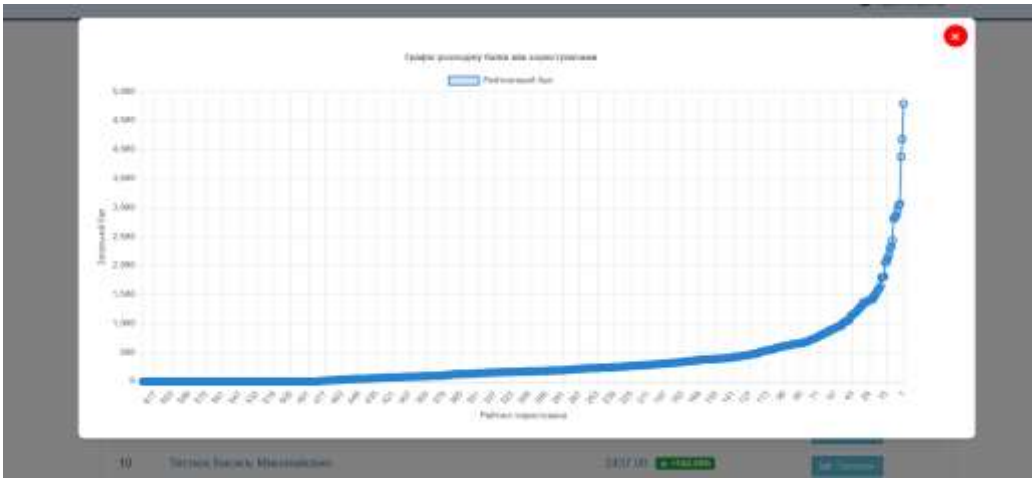


Рис. 4.24. Розподіл балів між користувачами в системі

Проаналізуємо вплив модуля збору та автозаповнення даних в звіті, на часові витрати користувачів, зокрема на необхідний час заповнення звіту. Відповідно, розраховано середній показник часу, який витрачають користувачі для заповнення річного звіту та результати навели на рисунку 4.25.

Рис. 4.25. Динаміка середнього часу заповнення звітів (у хвилинах)

Відповідно до цього графіка, можна зробити висновок, що завдяки автоматизації заповнення звітів, час, необхідний для виконання цього процесу, скоротився приблизно на 20%, що є прямим наслідком впровадження інструментів, які спрощують роботу НПП (і це статистика з урахуванням того, що частина користувачів все-таки вводили дані вручну, хоча у них була можливість підтягнути акумульовані дані в кілька кліків) та автоматизація наразі охоплює лише одну з шести груп активності. Якщо у 2018 році середній час заповнення звіту становив 53.5 хвилини, після зменшення кількості критеріїв у 2022 – до 32.5, а у 2024 році, з автоматизацією, цей показник знизився до 24.7 хвилин.

Наведемо відсоткове співвідношення рекомендованих критеріїв та їх типу для користувача (новий критерій, покращення тощо) на наступний звітний рік, при Top-K=10, можна сказати, що приблизно 54.5% рекомендацій були “новими” для користувача, 37.2% — відповідали за “покращення” наявних активностей, 8.3% — “відновленням активності”, тобто діяльностей, які користувач виконував раніше, але припинив. Такий розподіл рекомендацій вказує на те, що система зосереджується на рекомендації нових видів діяльності, та меншою мірою пропонує покращити поточні показники чи відновити раніше виконувані (рисунок 4.26).

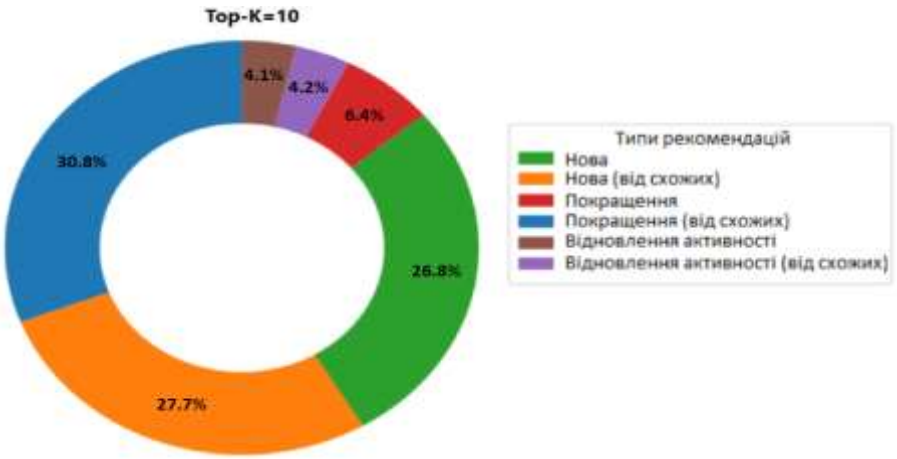


Рис. 4.26. Розподіл рекомендацій за типом у 2025 році

Для демонстрації ефективності рекомендацій, розглянемо роботу системи для деяких працівників з тестової групи. Система, аналізуючи діяльність користувача, може сформувати текст з персоналізованими рекомендаціями діяльності, що

містить аналіз минулих досягнень та конкретні поради з поясненнями. Після всіх підготовчих процесів та старту формування рекомендацій, для кожного НПП система формує 3 варіанти тексту рекомендацій щодо покращення їх діяльності на наступний рік. Для прикладу, наведемо сторінку яку бачить користувач системи у розділі “Рекомендації” (рисунок 4.27).

Рис. 4.27. Сторінка з можливістю обрати уточнення для рекомендації

НПП отримує 3 варіанти рекомендацій, які відрізняються кількома діяльностями, в залежності від їх типу, складності, групи активності (рисунок 4.27 – зелені блоки). При обранні бажаного варіанту, працівник підтверджує свій вибір,

та запускається процес формування корисних даних, при наступному запиті до мовних моделей. Для обраних діяльностей, виконується пошук подібних активностей та приклади їх реалізацій з моделі представлення знань. Після цього, отриманий датасет і сформований попередньо рекомендаційний текст, додаються в промпт на формування фінальної рекомендації. Під час цього етапу, мовна модель використовується для семантичного зв'язування корисних даних з рекомендаціями та розширення повного тексту рекомендацій. В результаті роботи цього запиту, працівник отримує фінальний текст персоналізованих рекомендацій діяльності, для прикладу, наведемо деякі частини такого тексту на 2024 рік, для користувача з вищесереднім річним балом:

"*[…] Опублікувати статтю у науковому виданні Scopus/Web of Science 4 квартилю.*

Цей крок є цілком реалістичним для вас, враховуючи вашу попередню публікаційну активність у Scopus/Web of Science без квартилю (5 публікацій у 2023 році) та успішний досвід публікації у Q2 у 2022 році. Перехід до публікацій у Q4 АБО Q3, або вище, буде значним покращенням вашого публікаційного профілю, підвищить вашу видимість та цитованість, а також дозволить отримати більше балів. Подібний Користувач1 успішно публікується у Q4 (75 балів за одну статтю), що демонструє досяжність цього критерію. Ваші дослідження у сфері "МАТЕМАТИЧНЕ ТА КОМП"ЮТЕРНЕ МОДЕЛЮВАННЯ" та "Інтелектуалізована рейтингова система" мають високий потенціал для публікації у відповідних журналах.

Як реалізувати: виберіть найбільш сильні та оригінальні результати з ваших поточних досліджень. Орієнтуйтеся на журнали Q4-Q3, які мають чітку спеціалізацію у ваших сферах (математичне та комп"ютерне моделювання, інтелектуальні системи, інформаційні технології). […]. Шукайте журнали Q4-Q3 у вашій галузі через бази Scopus або Web of Science (наприклад, використовуючи Scimago Journal & Country Rank (SJR) або Journal Citation Reports (JCR) від Clarivate Analytics). […]Можна використовувати платформи, такі як ResearchGate або Academia.edu, щоб знайти авторів, які публікуються в таких журналах, і проаналізувати їхній стиль та якість робіт.

Приклад журналу (залежить від конкретної спеціалізації в IT, але часто зустрічається в Q4/Q3): International Journal of Electrical and Computer Engineeing (IJECE) (індексується Scopus та WoS, часто Q4-Q3 для інженерних та комп"ютерних наук) або Journal of Engineering Science and Technology (JEST). […], Applied Sciences (частина журналів може бути Q3/Q4 в певних категоріях, залежно від розділу). Для комп"ютерних наук та систем: International Journal of Computer Science and Network Security.

Потенційний приріст балів (Q4): 50-75 балів АБО (Q3): 75-125 балів.

Опублікуйте одноосібну статтю у вітчизняному фаховому науковому виданні категорії "Б". […]. Опублікуйте статтю, що відображає ваші поточні дослідження або практичний досвід. Знайдіть відповідні журнали категорії "Б" з інформаційних технологій, наприклад, "Український журнал інформаційних технологій" або "Телекомунікаційні та інформаційні технології". […]. Потенційний приріст балів: 20".

Ще приклад, з посиланнями:

"*[…] Приклад платформ та ресурсів: Funding & Tenders Portal (Європейська Комісія): [https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/home] (https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/home) (основний ресурс для грантів ЄС). Erasmus+ Programme Guide: https://erasmus-plus.ec.europa.eu/resources/programme-guide (для детального вивчення можливостей Erasmus+). DAAD: https://www.daad.ua/uk/ (для програм обмінів та грантів з Німеччиною) […].*

Потенційний приріст балів: 20-40 балів (за подання заявок, у разі отримання – значно більше, залежно від умов гранту). […]".

Або ще такий приклад рекомендації щодо участі у наукових заходах:

"*[…] Приклад видань (якщо доступні інші публікації в університеті, розгляньте участь у редагуванні їх матеріалів, або спробуйте долучитися до зовнішніх видань): "Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка" (якщо є секції, що відповідають вашій галузі), "Вісник Тернопільського національного технічного університету імені Івана Пулюя", Продовжуйте брати активну участь у роботі програмного комітету конференцій, які ви вже очолювали ("Комп'ютерні інформаційні технології: Матеріали школи-семінару молодих вчених і студентів СІТ'").*

Запропонуйте організувати додаткові секції або залучити більше учасників. Запропонуйте свої послуги для керівництва програмними комітетами інших національних конференцій з інформаційних технологій, програмної інженерії, кібербезпеки, штучного інтелекту в Україні. Слідкуйте за анонсами конференцій на сайтах провідних технічних університетів України (КПІ ім. Ігоря Сікорського, НТУ "Харківський політехнічний інститут", НУ "Львівська політехніка", ДУТ)".

Такий текст охоплює список рекомендованих критеріїв, що відповідають діяльності НПП та дозволяє користувачеві чітко розуміти, які дії потрібно виконати для досягнення кращої результативності.

Ще можна навести приклад критеріїв які сформувала LLM, які відповідають тексту рекомендацій, та фактичну діяльність користувача яка виконувалась ним раніше (ДОДАТОК Ж , таблиця Ж.1). В цій таблиці наведений тип критерію для користувача, варіативність та пояснення (на основі фактичних даних з БД).

Якщо порівнювати з користувачем, який має низьку активність та прогалини у звітності за певні періоди, для нього текст рекомендації виглядає більш “скромно” (за кількістю та складністю рекомендованих критеріїв). Рекомендації формувались на 2024 рік, на основі попередніх активностях, а скріншот щорічних звітів працівника (рисунок 4.28) був сформований пізніше, після того як від вже прозвітував за свою активність в 2024 році. Приклад його персоналізованого тексту рекомендацій:

“*На основі аналізу ваших даних, пропонуємо наступні рекомендації для покращення вашої наукової продуктивності:*

Нарощування публікаційної активності в наукових виданнях, які входять до наукометричних баз даних. Ваша діяльність у 2018 році демонструє значний потенціал (300 балів за публікації), проте в 2023 році цей напрямок не був реалізований. Для відновлення та розвитку цього вектора, варто сфокусуватися на публікаціях у виданнях категорії “Б” або виданнях, що входять до міжнародних наукометричних баз. […]. Успішним прикладом є Користувач 1, який у 2023 році отримав 38.33 балів, публікуючись у виданнях категорії “Б” та міжнародних базах. […]. Зверніть увагу на видання категорії “Б”, які доступні в Україні, наприклад, “Економічний вісник Національного гірничого університету” або “Вісник Тернопільського національного економічного університету”.

Як це реалізувати: для відновлення публікаційної активності, ви можете підготувати статтю на основі ваших досліджень або матеріалів попередніх публікацій. Зверніть увагу на видання категорії Б, які доступні в Україні, наприклад, Вісник Національного технічного університету «ХПІ», Вісник Національного університету «Львівська політехніка» або Науковий вісник НУБіП України. […]. Потенційний приріст балів: 20-40 балів.

Активна участь у науково-дослідних проєктах на національному рівні (держбюджетна НДР). Ваша поточна діяльність не передбачає участі у грантовій активності, що є слабкою стороною, але можна пробувати виконати такі діяльність. Успішний приклад демонструє користувач 11, […].

Як це реалізувати: […]. Потенційний приріст балів: 10-40 балів.

Розширення науково-організаційної активності. Ваша участь у конференціях у 2023 році була успішною, проте відсутня організаційна складова. Для зростання в цьому напрямку, рекомендовано долучитися до організації наукових заходів. Користувач 2, наприклад, отримав 20 балів за рецензування дисертацій, що свідчить про його активну роль у науковій спільноті.

Як це реалізувати: запропонуйте свою кандидатуру до організаційного комітету конференцій, які проводяться на базі вашого університету, або долучитися до організації круглих столів. Це не лише покращить ваш рейтинг, а й сприятиме розширенню наукових зв'язків. Інший варіант — виступити рецензентом дисертацій. Для цього можна звернутися до спеціалізованих вчених рад за вашим профілем. Потенційний приріст балів: 10-30 балів.

Видання навчального посібника, хрестоматії чи словника, затверджених вченою радою університету. Ця діяльність може стати значним внеском у ваш рейтинг, особливо з огляду на те, що ви вже маєте науковий ступінь та вчене звання та подібні користувачі розвивають цю сферу. […]. Зверніться до видавничого центру університету або до видавництв, які спеціалізуються на освітній літературі. […].".

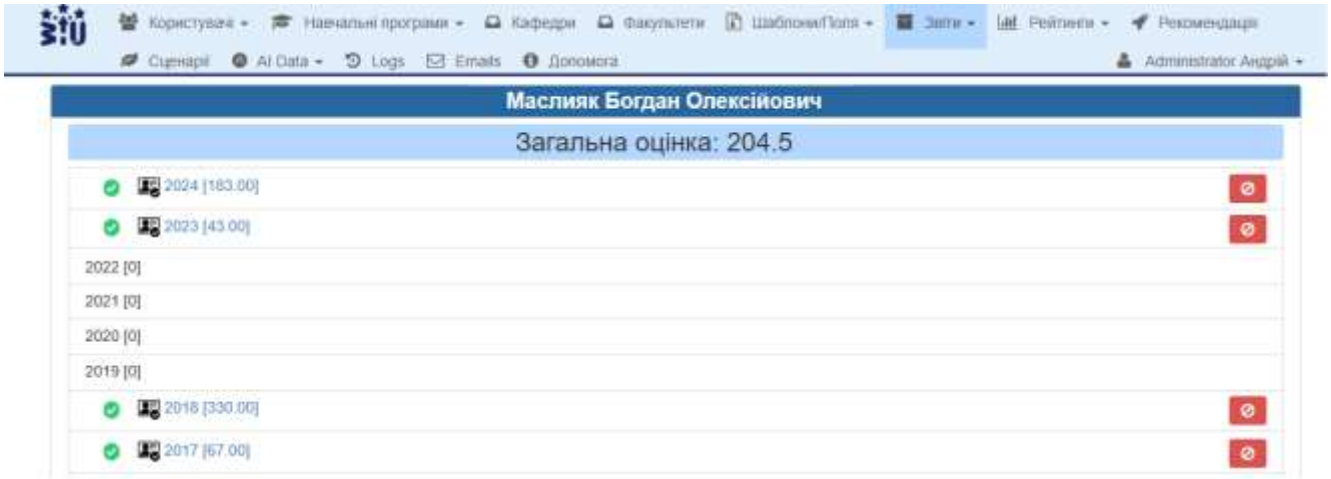


Рис. 4.28. Перегляд наявних звітів користувача в систему рейтингування ЗУНУ

Мовні моделі не можуть надавати деталізоване пояснення чи поради для всіх діяльностей які існують в системі, через те, що існують більш абстракті активності, для яких складно порадити щось конкретне для їх виконання. Для таких активностей, LLM формує загальні поради та базується на прикладах діяльності подібних користувачів. А для працівників з невеликою активністю і наявністю обмеженої кількості звітів, ШІ більше звертає увагу на опис користувача (його посаду, ступінь, спеціальність, кафедру, тривалість роботи), та враховує простіші діяльності для рекомендацій (це прописано в промпті).

Окрім рекомендацій, можна навести оцінку точності автоматизованих процесів, зокрема перевірки відповідності дисциплін до публікацій. Використання LLM для аналізу ключових слів та описів публікацій дозволяє автоматизувати визначення релевантності публікацій до конкретної дисципліни та налаштовувати її результати завдяки промпту. Наприклад, для публікації “SEO-Optimization of Product Content on a Marketplace Platform” та дисципліни “Інженерія кросплатформного ПЗ” система може повернути високий відсоток відповідності (близько, 90%) та надає пояснення, що обидва тексти містять ключові терміни “SEO”, “e-commerce” тощо. Інші приклади публікацій та дисциципліни наведено у таблиці (ДОДАТОК Ж , таблиця Ж.2). Це виконується для автоматизації цього процесу, та використання отриманих результатів під час визначення відповідності НПП ліцензійним вимогам. Цей процес враховує та аналізує семантику сформованих звітів користувачів за допомогою LLM та повертає його статус користувачу, враховуючи попередньо сформовані рекомендації для користувача. Для прикладу, наведемо користувача, який не виконав необхідну кількість діяльності, щоб відповідати вимогам, тобто LLM отримала список діяльностей, за останні 5 років, і визначила невідповідність вимогам, крім того, в цьому прикладі, виконання рекомендацій не призведе до зміни статусу в позитивну сторону:

“*Проведено аналіз вашої професійної діяльності за останні 5 років на відповідність критеріям ліцензійних умов. Наразі ви виконали 3 з 4 необхідних критеріїв. Зокрема, у вас є науковий ступінь, що дозволяє відповідати 4 критеріям. Загалом за останні 5 років ви підготували 1 монографію, що повністю відповідає вимогам щодо наявності виданого підручника, навчального посібника чи монографії. Крім того, ви маєте 7 публікацій у періодичних наукових виданнях, що включені до наукометричних баз Scopus чи Web of Science, що також повністю задовольняє вимоги, які передбачають наявність не менше 5 публікацій. Водночас, необхідно виконати ще два критерії.*

Враховуючи вашу поточну активність та наявні рекомендації системи, для досягнення ліцензійних умов наступного року вам слід зосередитися на таких напрямках. […]. Наприклад, ви можете виконати критерій щодо наявності патенту або свідоцтв про реєстрацію авторського права. Ваша поточна діяльність не охоплює цей напрямок, тому слід запланувати цю активність. Крім того, ви можете розглянути можливість взяти участь у редколегії

наукового видання (рекомендація системи) для виконання вимог щодо науково-організаційної діяльності. Наголошуємо, що без виконання додаткових критеріїв, наступного року ваша діяльність за 2020 рік не буде враховуватися, і ви втратите виконані критерії щодо монографії, публікацій, а також інші здобутки. […].”.

В цьому випадку, користувач отримає статус “Не відповідає ліцензійним вимогам”. В ідеальному сценарії, користувач повинен відповідати вимогам щороку, щоб уникати накопичення значного обсягу не виконаної роботи, яку доцільно реалізовувати поступово, невеликими частками, протягом кожного року.

Отже, результати апробації та оцінки ефективності підтверджують, що розроблена система є дієвим інструментом для підвищення результативності НПП. В ході тестувань, виявлено скорочення часу, необхідного для заповнення звітів, потенційне покращення річних показників при виконанні рекомендацій та розширення функцій системи, шляхом автоматизації частини її внутрішніх процесів.

Для подальшого удосконалення системи пропонується розширити функціонал модуля збору даних, включивши більшу кількість джерел та інших типів даних які використовуються для автоматичного заповнення звітів, також важливо забезпечити сумісність із іншими системами університету (SMART WUNU або відділу кадрів) для створення єдиного інформаційного простору.

Висновки до розділу 4

1. Описано особливості програмної реалізації інтелектуалізованої системи рейтингування, яка ґрунтується на розподіленій архітектурі. Основна частина системи розроблена на фреймворку Laravel і мови прогамування PHP з використанням підходу MVC та бази даних MySQL. Для формування персоналізованих рекомендацій, щоб підвищити ефективність аналізу даних та швидкість обчислень, для ключових модулів застосовано графову базу даних Neo4j та бібліотеку машинного навчання PyTorch. Це дозволило ефективно обробляти запити, будувати зв’язки та використовувати розширений набір інструментів.

2. Здійснено апробацію та оцінку ефективності роботи програмної системи на базі Західноукраїнського національного університету. Аналіз показав, що завдяки впровадженню інструментів часткової автоматизації, час, який необхідний для заповнення звітів, скоротився на 20%. Також встановлено, що серед невеликої тестової вибірки користувачів, рекомендаційний модуль має практичний потенціал для підвищення результативності працівника і відповідно його рейтингового балу. Також наведено приклади варіантів рекомендацій діяльності для НПП та фінального тексту рекомендацій, який містить пояснення та приклади для реалізації діяльності з побудованої моделі знань.

3. Представлено та проаналізовано функціональні можливості та графічний інтерфейс програмної системи, з описом взаємодії та відмінностями між функціоналом різних ролей. Продемонстровано, що інтеграція системи з мовними моделями дозволила автоматизувати частину процесів, таких як заповнення звітів, аналіз семантичної відповідності публікацій до дисциплін, формування персоналізованих рекомендацій щодо річної діяльності та відповідності НПП ліцензійним вимогам.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальне науково-практичне завдання, пов’язане з розробкою інтелектуалізованої програмної системи аналізу та рейтингування діяльності НПП, яка забезпечить комплексний аналіз, прогнозування та формування персоналізованих рекомендацій для покращення результативності науково-педагогічної діяльності працівників. Зазначена інтелектуалізована система є важливим для модернізації механізмів оцінки та стимулювання працівників, враховуючи прямий зв’язок між індивідуальною результативністю НПП та позиціями закладу вищої освіти у міжнародних рейтингах. У рамках дослідження було отримано низку наукових та практичних результатів, що стосуються математичного та програмного забезпечення системи.

1. Проведено комплексний аналіз методів та засобів підвищення результативності діяльності НПП. Виявлено, що більшість існуючих систем рейтингування НПП, особливо в українських ЗВО, є переважно традиційними, та характеризуються низкою суттєвих недоліків, таких як висока трудомісткість, суб’єктивність, обмежені аналітичні можливості та відсутність персоналізованого зворотного зв’язку. На противагу цьому, враховуючи сучасний розвиток цієї сфери, в інтелектуалізовані системи починають інтегрувати технології штучного інтелекту, які використовуються для автоматизації більшості процесів, забезпечення об’єктивності та створення персоналізованих рекомендацій. Запропоновано використовувати гібридний підхід, який поєднує методи графових нейронних мереж, метрики подібності та великі мовні моделі, що дозволить враховувати динаміку, семантику та контекст діяльності НПП для надання обґрунтованих і зрозумілих рекомендацій. Результати аналізу підтвердили актуальність та визначили напрями дисертаційного дослідження.

2. Розроблено модель представлення знань, що базується на знання-орієнтованому підході та реалізована з використанням графової структури даних. На відміну від традиційних реляційних підходів, запропонована модель дозволяє ефективно структурувати та відображати складні, нелінійні взаємозв’язки між

різнорідними сутностями: НПП, критеріями їхньої діяльності, публікаціями, проєктами, дисциплінами та іншими академічними об’єктами. Створення такої моделі стало важливою складовою для інтелектуалізованої системи рейтингування, оскільки забезпечило можливість виконувати контекстуальний аналіз, виявляти приховані закономірності у траєкторіях професійного розвитку та, що найголовніше, стало ключовою передумовою для реалізації наступних етапів.

3. Запропоновано комбінований метод ранжування НПП, що вирішує задачу об’єктивного знаходження релевантних прикладів для наслідування в умовах розрідженості даних. Метод інтегрує комплекс метрик подібності, кожна з яких оцінює різні аспекти діяльності працівника. Зокрема, метрика динамічного часового зміщення аналізує схожість траєкторій результативності в часі, косинусна подібність порівнює співвідношення між різними видами активностей, а модифікований індекс Жаккара враховує змістовну близькість виконуваних робіт. Такий підхід уможливив автоматизоване визначення працівників із максимально наближеними кращими показниками діяльності. Важливим елементом методу є підсилення зв’язків у графовій базі даних між схожими НПП, що дозволило підвищити якість та точність подальшого прогнозування, яке виконується графовою нейронною мережею.

4. Запропоновано та обґрунтовано інтелектуалізований метод формування персоналізованих рекомендацій діяльності НПП, який поєднує прогнозуючу здатність графових нейронних мереж та генеруючі можливості великих мовних моделей, на основі моделі представлення знань, прогнозування показників діяльності з механізмами просторової і часової уваги для прогнозування майбутньої активності НПП та використання великих мовних моделей для генерації текстових персоналізованих рекомендацій з динамічним уточненням структури промптів, що забезпечило скорочення часу їх формування та розширило рекомендації.

5. Спроєктовано модульну, сервісно-орієнтовану архітектуру системи, яка забезпечує гнучкість, масштабованість та ефективну взаємодію між основними компонентами системи та забезпечує реалізацію основного функціоналу.

Використання гібридної архітектури зберігання даних, яка поєднує реляційну базу даних MySQL для основних функцій та графову базу даних Neo4j для процесу формування персоналізованих рекомендацій, дозволило оптимізувати аналіз взаємозв'язків між користувачами та їхньою діяльністю. Основна частина системи розроблена на фреймворку Laravel з MySQL, а для ключових модулів використано Neo4j та бібліотеку PyTorch для ефективності обчислень. Інтеграція з LLM виконана на базі системи рейтингування, що дозволила частково автоматизувати процеси заповнення звітів, визначення статусу працівників щодо здійснення ними освітньої діяльності, семантичного аналізу даних та формування тексту рекомендацій.

6. Апробацію та оцінку ефективності системи проведено на базі Західноукраїнського національного університету. Встановлено, що впровадження засобів автоматизації дозволило скоротити час заповнення звітів на 20 %. Аналіз діяльності тестової вибірки працівників із використанням персоналізованих рекомендацій показав, що вони виконали до 30 % рекомендацій на рівні, що досягає або перевищує прогнозований бал. Застосування рекомендацій сприяло підвищенню їхніх річних показників у середньому на 100 одиниць, а у відсотковому відношенні — в середньому на 27%, в порівнянні з попередніми результатами. Динамічне уточнення структури промптів дозволяє виконати цей процес одним запитом, та в цілому скоротити час, на їх формування, забезпечуючи працівникам можливість безпосередньо взаємодіяти з результатами роботи системи та впливати на них.

Таким чином, у роботі представлено комплексний підхід та реалізовану на його основі програмну систему. Розроблене рішення забезпечує не лише об'єктивне оцінювання та рейтингування діяльності НПП, але й автоматизує ключові процеси, включно з перевіркою відповідності НПП ліцензійним вимогам. Ключовою перевагою системи є здатність генерувати персоналізовані рекомендації для підвищення результативності НПП, що, у свою чергу, позитивно впливає на загальний рейтинг університету.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пукас, А. В., Сімак, А. Ю. (2025). Метод формування прогнозованих показників активності науково-педагогічних працівників для інтелектуалізованої системи аналізу їх діяльності та рейтингування. Наукові праці Вінницького національного технічного університету, (2). Вилучено із https://praci.vntu.edu.ua/index.php/praci/article/view/813.

2. Сімак, А., Юшко, А. (2025). Інтелектуальний модуль збору відкритих даних в системі рейтингування діяльності науково-педагогічного персоналу. Measuring and computing devices in technological processes, (1), 353–361. https://doi.org/10.31891/2219-9365-2025-81-45.

3. Юшко, А. В., Сімак, А. (2024). Інформаційна інтелектуальна система аналізу наукової та науково-педагогічної діяльності академічного колективу. Оптико-електронні інформаційно-енергетичні технології, 47(1), 7–16. https://doi.org/10.31649/1681-7893-2024-47-1-7-16.

4. Мельниченко В.В., Сімак А.Ю., Юшко А.В. (2024). Інтелектуалізована система для підтримки реалізації принципів адаптивності навчання. Матеріали зимової школи-семінару молодих вчених і студентів, “Комп’ютерні інформаційні технології” СІТ’2024 Тернопіль ЗУНУ, с. 85-87.

5. Пукас А.В., Сімак А.Ю., Юшко А. В., Мельниченко В.В. (2024). Модуль збору даних для автоматичного заповнення звіту в інтелектуалізованій системі аналізу діяльності та рейтингування викладачів. Матеріали зимової школи-семінару молодих вчених і студентів, “Комп’ютерні інформаційні технології” СІТ’2024 Тернопіль ЗУНУ, с. 69-71.

6. A. Pukas, A. Simak, A. Yushko, S. Nadvynychnyy, S. Shandruk and O. Vatslavskyi, “Automated Reporting Module in the Academic Staff Performance Appraisal System,” *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)*, Ceske Budejovice, Czech Republic, 2024, pp. 824-827, doi: 10.1109/ACIT62333.2024.10712573.

7. Гуска, Б. Д. Інтелектуалізована система обробки слабкоформалізованої інформації [Текст] / Б. Д. Гуска, А. Ю. Сімак // Комп'ютерні інформаційні технології: матеріали Школи-семінару молодих вчених і студентів СІТ'2023, Тернопіль, 30 листопада 2023 року / відп. за вип. А. В. Пукас. – Тернопіль: ЗУНУ, 2023. – С. 91.

8. Andriy Pukas, Andriy Simak, Tetiana Goncharuk-Cholach, Olena Konoplitska and Vadym Zabchuk "Intelligent Analyzing Module in the Academic Staff Performance Appraisal System", *CEUR Workshop Proceedings*, 2023, 3624, pp. 423–429.

9. Pukas A., Simak A., Shandruk S., Bilovus L., Stepanenko V. and Demianiuk A., "Features of Implementation the Academic Staff Performance Appraisal System," *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*, Ruzomberok, Slovakia, 2022, pp. 568-571, doi: 10.1109/ACIT54803.2022.9913180.

10. A. Pukas, A. Simak, O. Syrnyk, L. Horal, V. Shyjko, and O. Papa, "Software Module for Data Correctness and Completeness Control in the Academic Staff Performance Appraisal System," in Proc. 2019 9th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Czech Republic, 2019, pp. 277–280, doi: 10.1109/ACITT.2019.8779999.

11. Пукас, А. В. Інформаційна система рейтингування діяльності науково-педагогічних працівників закладів вищої освіти [Текст] / Пукас А. В., Сімак А. Ю., Вітенко В. В.// Комп'ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів СІТ'2019 [м. Тернопіль, 29 листопада 2019 р.] / відп. за вип. М. П. Дивак. - Тернопіль: ТНЕУ, 2019. - С. 32-33.

12. Пукас, А. В. Модуль контролю повноти та коректності даних інтелектуалізовананої системи аналізу діяльності та рейтингування викладачів [Текст] / А. В. Пукас, А. Ю. Сімак, О. Й. Сирник // Комп'ютерні науки та інформаційні технології: матеріали семінару CSIT'2018 [м. Тернопіль, 2 червня 2018 р.] / відп. за вип. М. П. Дивак. - Тернопіль: ТНЕУ, 2018. - С. 61-62.

13. Dudarev, I., & Purtov, V. (2024). System of comprehensive evaluation of academic staff activity. Scientific Bulletin of Mukachevo State University. Series “Pedagogy and Psychology”, 10(4), 24-40. doi: 10.52534/msu-pp4.2024.24.

14. Скиба, Ю. О. Науково-педагогічний потенціал університету: сутність та структура [Електронний ресурс] / Юрій Скиба // Міжнародний науковий журнал “Університети і лідерство”. – 2020. – № 2 (10). – С. 13–25. – Режим доступу: https://doi.org/10.31874/2520-6702-2020-10-2-13-25.

15. Yaroshenko, O. (2021). Evaluation of the activity of scientific and pedagogical staff: domestic experience and vectors of development under the university autonomy. International Scientific Journal of Universities and Leadership, (11), 132-143. https://doi.org/10.31874/2520-6702-2021-11-1-132-143.

16. OECD (2023), OECD Digital Education Outlook 2023: Towards an Effective Digital Education Ecosystem, OECD Publishing, Paris, https://doi.org/10.1787/c74f03de-en.

17. Safón, V. “Inter-ranking reputational Ferreira, M., Costas, R., Servedio, V., & Thurner, S. (2023). Scientific mobility, prestige and skill alignment in academic institutions [Electronic resource] / Available date: 12.12.1212. URL: https://doi.org/10.48550/arXiv.2307.06426.

18. effects: an analysis of the Academic Ranking of World Universities (ARWU) and the Times Higher Education World University Rankings (THE) reputational relationship” / Vicente Safón // Scientometrics. – 2019. – Vol. 121, № 1, doi: 10.1007/s11192-019-03214-9.

19. Anowar, F., Helal, M. A., Afroj, S., Sultana, S., Sarker, F., & Mamun, K. A. (2015). A critical review on world university ranking in terms of top four ranking systems. In New trends in networking (pp. 559–566). Springer. https://doi.org/10.1007/978-3-319-06764-3_72.

20. Мельник С. В. Рейтингове оцінювання у вищій освіті: досвід і перспективи. Одеса: Освітній центр, 2022. 156 с.

21. Smith J. Academic Performance Metrics: Global Trends and Challenges. New York: Educational Insights, 2021. 230 p.

22. Banayee, M. R., Faiq, Z., Mohammadi, E. M., & Faizi, F. (2022). *Evaluation of the Effectiveness of the Performance Management System in the Ministry of Higher Education*. *Sprin Journal of Arts, Humanities and Social Sciences*, 1(09), 457–467.

23. Dima, Alina Mihaela, et al. "Performance Evaluation in Higher Education – A Comparative Approach" Proceedings of the International Conference on Business Excellence, vol. 18, no. 1, The Bucharest University of Economic Studies, 2024, pp. 2453-2471. https://doi.org/10.2478/picbe-2024-0207.

24. Gullickson, A.R., & Howard, B.B. (2009). *The Personnel Evaluation Standards: How to assess systems for evaluating educators (2nd ed.)*. Thousand Oaks, CA: Corwin Press.

25. Zhao, Nan, et al. "Construction and Evaluation of KPI Performance Appraisal Model in Personnel Management System of Higher Vocational Colleges and Universities" Applied Mathematics and Nonlinear Sciences, vol. 9, no. 1, Sciendo, 2024, https://doi.org/10.2478/amns-2024-1739.

26. Reheilo, I. (2020). Academic staff research performance evaluation: the foreign experience. Continuing Professional Education: Theory and Practice, (4), 96–102. https://doi.org/10.28925/1609-8595.2020.4.12.

27. Hussain, N. A. S. Performance Evaluation for Educators in Higher Education from Bibliometric Analysis Views [Electronic resource] / Nor Azarif Sofia Hussain, Mohd Effendi, Ewan Mohd Matore, Mohd Izham Mohd Hamzah // International Journal of Learning, Teaching and Educational Research. – 2025. – Vol. 24, № 5. – P. 284 —285. – DOI: https://doi.org/10.26803/ijlter.24.5.15.

28. Шимон О.М., Шиненко М.А., Коваленко В.М. Особливості оцінювання результативності педагогічних досліджень в освітній практиці // Мультимедійні технології в освіті та інших сферах діяльності: науково-практична конференція з міжнародною участю. 10 листопада 2022 р. К.: НАУ, 2023. С.158-160.

29. Masron, T. A. Key Performance Indicators vs Key Intangible Performance Among Academic Staff: a Case Study of a Public University in Malaysia / Tajul Ariffin Masron, Zamri Ahmad, Norizan Baba Rahim // Procedia - Social and Behavioral

Sciences. – 2012. – Vol. 56. – P. 494–503. – DOI: https://doi.org/10.1016/j.sbspro.2012.09.681.

30. Положення про рейтингове оцінювання діяльності науково-педагогічних, педагогічних працівників - УДУНТ: Наказ від 28.09.2022 р. № 61. URL: https://ust.edu.ua/wp-content/uploads/2024/04/rejtyng-_npp_udunt_21_22.pdf.

31. Положення про рейтингове оцінювання діяльності науково-педагогічних, педагогічних працівників - КНУТД: Наказ від 26.04.2024 р. № 137. URL: https://knutd.edu.ua/files/dostupdopi/rating_npp.pdf.

32. Розпорядження №4 від 25 травня 2021 року. URL: https://kdpu.edu.ua/pro-nas/dokumenty-universytetu/rozporiadzhennia/15832-rozporiadzhennia-04-vid-25-travnia-2021-roku.html (дата звернення: 15.07.2025).

33. Multan E, Wójcik-Augustyniak M, Sobotka B, Bis J. Application of Performance and Efficiency Indicators in Measuring the Level of Success of Public Universities in Poland. *Sustainability*. 2023; 15(18):13673. https://doi.org/10.3390/su151813673.

34. DSpace at West Ukrainian National University. [Електронний ресурс]. Режим доступу: http://dspace.wunu.edu.ua (дата звернення: 24.07.2025).

35. Elsevier Developer Portal, Research Products APIs. [Електронний ресурс]. Режим доступу: https://dev.elsevier.com (дата звернення: 24.07.2025).

36. Ali, M. G. (2025). Introducing and Evaluating Key Performance Indicators and Generating Awareness of Competence among Inter-Departments for Quality Performance and Quality Assurance in Large Academic and Research Multidisciplinary Institutions. International Journal of Educational Research Review, 10(1), 20–28.

37. Medne, A. Data-based University Quality Assurance: Whether Ranking Results and Performance Indicators Interrelate? / Aija Medne, Arturs Zeps, Inga Lapiņa // Proceedings of the 6th International Conference on Quality Engineering and Management (ICQEM 2024). – Girona, Spain, 2024. – URL: https://publicacoes.riqual.org/wp-content/uploads/2024/06/icqem_24_557_576.pdf.

38. Про затвердження Положення про рейтингування науково-педагогічних працівників КПІ ім. Ігоря Сікорського та норм бального оцінювання

діяльності науково-педагогічних працівників: Наказ від 25.12.2023 р. № НОН/386/2023. [Електронний ресурс]. – Режим доступу: https://document.kpi.ua/files/2023_НОН-386.pdf.

39. Положення про рейтингове оцінювання роботи науково-педагогічних працівників ЗУНУ, від 2020 р. [Електронний ресурс]. Режим доступу: https://www.dnu.dp.ua/docs/dnu/poloz_reit_npp.doc.

40. Критерії та норми бального оцінювання роботи науково-педагогічних працівників, ЗУНУ, від 2022 р. [Електронний ресурс]. – Режим доступу: https://www.wunu.edu.ua/pdf/pablic_inf/kryterii-ociniuvannia_feb_2022.pdf.

41. Норми часу на виконання основних видів навчальної, методичної, наукової і організаційної роботи науково-педагогічних працівників Національного університету "Чернігівська політехніка", НУ Чернігівська політехніка: Наказ від 30.06.2025 р. № 151/ВС. [Електронний ресурс]. – Режим доступу: https://stu.cn.ua/wp-content/stu-media/normobaza/normdoc/norm-osvitproces/normi-chasu-na-vykonannya-robit-npp.pdf.

42. Tóth, B., Motahari-Nezhad, H., Horseman, N. et al. Ranking resilience: assessing the impact of scientific performance and the expansion of the Times Higher Education Word University Rankings on the position of Czech, Hungarian, Polish, and Slovak universities. Scientometrics 129, 1739–1770 (2024). https://doi.org/10.1007/s11192-023-04920-1.

43. Rochim, A. F. A Citation Data Collector Tool of Author's Profiles in Scopus Based on Web and Application Programming Interface (API) / Adian Fatchur Rochim, et al. // IOP Conference Series: Materials Science and Engineering. – 2021. – Vol. 1077. – P. 012017. – DOI: https://doi.org/10.1088/1757-899X/1077/1/012017.

44. Закон України "Про вищу освіту" від 01.07.2014 № 1556-VII зі змінами [Електронний ресурс]. – Режим доступу: http://zakon3.rada.gov.ua/laws/show/1556-18.

45. Стандарти і рекомендації щодо забезпечення якості в Європейському просторі вищої освіти (ESG) [Електронний ресурс]. Режим доступу:

https://naqa.gov.ua/wp-content/uploads/2019/07/Final-Standards-and-Guidelines_UA201511_press_20151106.pdf.

46. Дробахін, О. О. Автоматизація рейтингового оцінювання діяльності науково-педагогічних працівників / О. О. Дробахін, Н. А. Гук, М. О. Ткаченко // Наукові праці ДонНТУ. Серія: Обчислювальна техніка та автоматизація. – 2021. – Вип. 3 (32). – С. 54–63. – DOI: https://doi.org/10.15421/322107.

47. Зеленський К., Стельмащук Л., Луців І. Рейтингове оцінювання роботи викладача як фактор підвищення його конкурентоспроможності. Галицький економічний вісник. – 2017. № 2 (53). С. 36-48.

48. Cindrakasih, R., Jahara, J., Soegiarto, I., Ma’sum, H., & Asmuri, A. (2024). Strategy for Implementing Artificial Intelligence in the Lecturer Performance Evaluation System in Higher Education. International Journal of Educational Narratives, 2(2), 224–234. https://doi.org/10.70177/ijen.v2i2.822.

49. Dudarev, I., & Purtov, V. (2024). System of comprehensive evaluation of academic staff activity. Scientific Bulletin of Mukachevo State University. Series “Pedagogy and Psychology”, 10(4), 24-40. doi: 10.52534/ msu-pp4.2024.24.

50. Arabi Mianroodi, A. A. Explaining the pros and cons of different sources of faculty evaluation from the viewpoints of medical university academics / Ali Asghar Arabi Mianroodi, Zahra Asgari Baravati, Narges Khanjani // Strides in Development of Medical Education. – 2012. – Vol. 9, № 1. – P. 65–76.

51. Солдатова, М. О. Розподіл ролей користувачів системи рейтингування науково-педагогічних працівників / М. О. Солдатова, А. М. Губський, А. В. Максимюк, Д. О. М’яч // Телекомунікаційні та інформаційні технології. – 2024. – № 1. – С. 106–113.

52. Bryant, Rebecca, Jan Fransen, Pablo de Castro. Brenna Helmstutler, and David Scherer. 2021. Research Information Management in the United States: Part 2—Case Studies. Dublin, OH: OCLC Research. https://doi.org/10.25333/qv1f-9e57.

53. Смірнова, Валерія Андріївна (2022) ”Рейтинг дослідницької діяльності науково-педагогічних працівників закладу вищої освіти як інструмент

забезпечення прозорості результатів досліджень” Відкрите освітнє е-середовище сучасного університету (13). с. 129-140. ISSN 2414-0325.

54. Положення про рейтингове оцінювання професійної діяльності науково-педагогічних працівників, кафедр та факультетів - ДНУ: Наказ від 21.01.2022 р. № 21. URL: https://www.dnu.dp.ua/docs/dnu/polozhennya/Nakaz%20№21_21_01_22_Polozhennya_Rejting_ocinuvannya_prof_diyal_NPP.pdf.

55. Johnson R. (2019). Evaluation Systems in Higher Education: Best Practices. London: Academic Press, 200 p.

56. Smagina, O., Pereyaslavska, S., & Zhykova, V. (2018). Information support system of control and management calculation of scientific, educational and methodological rating of university chair lecturers. Physical and Mathematical Education, 1(15), 311-315.

57. Sockbeson, Caitlin (2018). Performance Appraisal. obo in Psychology. doi: 10.1093/obo/9780199828340-0213.

58. Evaluation and Promotion – TUM “ Evaluation categories and criteria” – 16/07/2025. URL: https://www.tum.de/en/about-tum/careers-and-jobs/faculty-recruiting/evaluation-promotion.

59. Kovári, A. A systematic review of AI-powered collaborative learning in higher education: trends and outcomes from the last decade / Attila Kovári // Social Sciences & Humanities Open. – 2025. – Vol. 11. – P. 101335. – DOI: https://doi.org/10.1016/j.ssaho.2025.101335.

60. Kayyali, Mustafa, The Relationship between Rankings and Academic Quality (May 3, 2023). International Journal of Management, Sciences, Innovation, and Technology IJMSIT, Vol.4, Issue.3, pp.01-11, May (2023), Available at SSRN: https://ssrn.com/abstract=4497493.

61. Yaroshenko, O. (2021). Evaluation of the activity of scientific and pedagogical staff: domestic experience and vectors of development under the university autonomy. International Scientific Journal of Universities and Leadership, (11), 132-143. https://doi.org/10.31874/2520-6702-2021-11-1-132-143.

62. Положення про рейтингове оцінювання діяльності науково-педагогічних працівників, кафедр і факультетів Національного аерокосмічного університету “Харківський авіаційний інститут – ХАІ. URL: https://khai.edu/ua/university/normativna-baza/polozheniya1/polozhennya-pro-poryadok-ocinyuvannya-diyalnosti-poryadok-zamishhennya-posad-zdobuttya-zvan-ta-stupeniv/polozhennya-rejtingove-ocinyuvannya/ (дата звернення: 18.07.2025).

63. Sadiq, B. J. (2024). Assessing Faculty and Staff by using KPI. International Journal of Academic Research in Business and Social Sciences, 14(3), 1230–1237.

64. Sandeepa, A. G. R. Evaluation of Machine Learning Models in Student Academic Performance Prediction [Text] / A. G. R. Sandeepa, Sanka Mohottala // Proceedings of the 5th International Conference on Advanced Research in Computing (ICARC), February 2025. – IEEE, 2025. – P. 1–6. – DOI: https://doi.org/10.1109/ICARC64760.2025.10963104.

65. Chu Hui, Y. AI-Driven Tools in Providing Feedback on Students’ Writing / Chu Hui Youn, Abdul Rahim Hj Salam, Aida A. Rahman // International Journal of Research and Innovation in Social Science. – 2025. – Vol. IX, № IIIS. – P. 58–67. – DOI: https://doi.org/10.47772/IJRISS.2025.903SEDU0006.

66. Zambrano-Romero, W., Rodriguez, C., Pita-Valencia, J., Zambrano-Romero, W. J., & Moran-Tubay, J. M. (2025). Machine Learning in the Teaching Quality of University Teachers: Systematic Review of the Literature 2014–2024. *Information*, *16*(3), 181. https://doi.org/10.3390/info16030181.

67. Рейтингова система ЗУНУ – Головне вікно [Електронний ресурс]. – Режим доступу: https://rating.wunu.edu.ua.

68. Watermark Faculty Success, formerly Digital Measures – Works. URL: https://works.umn.edu/sites/works.umn.edu/files/2021-11/LSBE-Works-Guidelines-08-2021.pdf (accessed on 18 Jul 2025).

69. Faculty Performance System Software - Creatrix Campus. https://www.creatrixcampus.com/faculty-performance (accessed on 18 Jul 2025).

70. University of Pittsburgh Implements Pre-Course Surveys to Innovate Teaching Effectiveness with Explorance Blue. URL: https://explorance.com/case-

study/university-of-pittsburgh-implements-pre-course-surveys-to-innovate-teaching-effectiveness-with-explorance-blue/ (accessed on 18 Jul 2025).

71. Радкевич, О. П. Використання штучного інтелекту для персоналізованого оцінювання професійної діяльності педагогічних працівників / О. П. Радкевич // Професійна освіта в умовах сталого розвитку суспільства: монографія. – 2024. – № 6 (19). – С. 41–45. – ISSN 2786-619X. – ORCID: https://orcid.org/0000-0002-2648-5726.

72. М. Даниленко і І. Колесник, “Методи розробки рекомендаційних систем”, ІТКІ, вип. 52, вип. 3, с. 10–15, Груд 2021.

73. Худік, Б. (2023). Модель представлення даних рекомендаційної системи в сфері освіти на основі нечіткої логіки. Електронне фахове наукове видання “Кібербезпека: освіта, наука, техніка”, 1(21), 260–272. https://doi.org/10.28925/2663-4023.2032.21.260272.

74. Pasichnyk, V. Project of an Educational Content Evaluation Recommender System / Volodymyr Pasichnyk, Nataliia Kunanets, Valentyna Yunchyk, Maria Khomyak, Anatolii Fedonyuk // Proceedings of the 5th International Workshop IT Project Management (ITPM 2024), May 22, 2024, Bratislava, Slovakia. – 2024. – P. 192–203.

75. Худік, Б. О. Застосування рекомендаційної системи в освітній сфері / Б. О. Худік // Зв’язок. – 2023. – № 6. – С. 58-62. – ISSN 2412-9070. – DOI: https://doi.org/10.31673/2412-9070.2023.065868.

76. I.S., S. (2023). PROFESSIONAL DEVELOPMENT OF SCIENTIFIC AND TEACHING EMPLOYEES HIGHER EDUCATION INSTITUTIONS OF UKRAINE: LEGAL PROBLEMS REGULATION. Social Law, (4), 33-44. https://doi.org/ 10.32751/2617-5967-2022-04-06.

77. Joseph Evanick. (2025). The Future of Teaching in Higher Education. The EvoLLLution. URL: https://evolllution.com/the-future-of-teaching-in-higher-education.

78. Bousalem, Z., Qazdar, A., El Guabassi, I., & Haj, A. (2024). A Recommendation System Based on Early Academic Performance Prediction and Student Classification: Utilizing Artificial Intelligence and Mobile-Based

Application. International Journal of Interactive Mobile Technologies (iJIM), 18(15), pp. 169–189. URL: https://doi.org/10.3991/ijim.v18i15.47135.

79. Beskorovainyi, V., Kolesnyk, L., Alokhina, M. і Kosenko V. (2022) “Ввизначення переваг у рекомендаційних системах на основі технології компараторної ідентифікації”, Сучасний стан наукових досліджень та технологій в промисловості, (2 (20), с. 14–21. doi: 10.30837/ITSSI.2022.20.014.

80. Velez-Estevez, A., I. J. Perez, P. Garc’ıa-S’anchez, J. A. Moral-Munoz, and M. J. Cobo. 2023. “New Trends in Bibliometric APIs: A Comparative Analysis.” Information Processing & Management 60(4): 103385. https://doi.org/10.1016/j.ipm.2023.103385.

81. Rudenko, S. Information-Analytical System for Evaluating the Activity of the Scientific-Pedagogical Staff Based on KPIs at Odesa National Maritime University / Sergiy Rudenko, Varvara Piterska, Tetiana Kovtun // Proceedings of the 5th International Workshop IT Project Management (ITPM 2024), May 22, 2024, Bratislava, Slovak Republic. – 2024. – URL: https://ceur-ws.org/Vol-3865/paper1.

82. Raza, S. A Comprehensive Review of Recommender Systems: Transitioning from Theory to Practice [Electronic resource] / Shaina Raza, Mizanur Rahman, Safiullah Kamawal, Armin Toroghi, Ananya Raval, Farshad Navah, Amirmohammad Kazemeini // arXiv preprint arXiv:2407.13699. – 2025. – URL: https://arxiv.org/abs/2407.13699.

83. Algarni, S., & Sheldon, F. (2023). Systematic Review of Recommendation Systems for Course Selection. *Machine Learning and Knowledge Extraction*, *5*(2), 560-596. https://doi.org/10.3390/make5020033.

84. Lampropoulos, G. (2023). Recommender systems in education: A literature review and bibliometric analysis. Advances in Mobile Learning Educational Research, 3(2), 829-850. https://doi.org/10.25082/AMLER.2023.02.011.

85. Peng, Y. A Survey on Modern Recommendation System based on Big Data. arXiv 2022, arXiv:2206.02631.

86. Tran, P. D. T. Personalized learning paths recommendation system with collaborative filtering and content-based approaches / Phong Duong Thanh Tran, Khang Bao Vu, Minh Nhat Doan, Quynh Truc, Quang Viet Dan, gThanh Ho // Science &

Technology Development Journal - Economics - Law and Management. – 2024. – Vol. 8, № 2. – DOI: https://doi.org/10.32508/stdjelm.v8i2.1370.

87. Hikmatyar, M. Book Recommendation System Development Using User-Based Collaborative Filtering / Missi Hikmatyar, Ruuhwan // Journal of Physics: Conference Series. Computer Science. – 2020. – Vol. 1477. – P. 032024. – DOI: https://doi.org/10.1088/1742-6596/1477/3/032024.

88. Meng Liu, Jianjun Li, Ke Liu, Chaoyang Wang, Pan Peng, Guohui Li, Yongjing Cheng, Guohui Jia, and Wei Xie. Graph-ICF: Item-based collaborative filtering based on graph neural network. Knowledge-Based Systems, 251, 2022.

89. (BERT, Pres, MAP) Sangeetha N. A BERT Based Hybrid Recommendation System For Academic Collaboration [Electronic resource] / Sangeetha N, Harish Thangaraj, Varun Vashisht, Eshaan Joshi, Kanishka Verma, Diya Katariya // arXiv preprint arXiv:2502.15223. – 2025. URL: https://arxiv.org/abs/2502.15223.

90. Atalla, S.; Daradkeh, M.; Gawanmeh, A.; Khalil, H.; Mansoor, W.; Miniaoui, S.; Himeur, Y. An Intelligent Recommendation System for Automating Academic Advising Based on Curriculum Analysis and Performance Modeling. Mathematics 2023,11, 1098. https://doi.org/10.3390/math11051098.

91. Hidasi, B., & Karatzoglou, A. (2018). Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM ‘18), pp. 10. URL: https://doi.org/10.1145/3269206.3271761.

92. Shoujin Wang, Qi Zhang, Liang Hu, Xiuzhen Zhang, Yan Wang, and Charu Aggarwal. Sequential/Session-based Recommendations: Challenges, Approaches, Applications and Opportunities. In SIGIR 2022 - Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022.

93. Shoujin Wang, Longbing Cao, Yan Wang, Quan Z. Sheng, Mehmet A. Orgun, and Defu Lian. A Survey on Session-based Recommender Systems. ACM Computing Surveys, 54(7), 2022.

94. Sravya, Rani and D, Dr. Thamaraiselvi and Siva Dhatri, Bhagavathula, Unified Personalized Recommendation System using MongoDB and ALS (April 30, 2025).

95. Bablani, D.; Gupta, R.; Gokhale, T. Hybrid Approach to Music Recommender Systems. Int. J. Res. Appl. Sci. Eng. Technol. 2022.

96. Azzam, M. Hybrid Music Recommendation Approach for Heterogeneous Information Network Using Factorization Machines; Johannes Kepler Universität Linz: Linz, Austria, 2021.

97. Sabiri, B., Khtira, A., El Asri, B., & Rhanoui, M. (2025). Hybrid Quality-Based Recommender Systems: A Systematic Literature Review. *Journal of Imaging*, *11*(1), 12. URL: https://doi.org/10.3390/jimaging11010012.

98. Shi, X., Liu, Q., Bai, Y. et al. RTiSR: a review-driven time interval-aware sequential recommendation method. J Big Data 10, 32 (2023). https://doi.org/10.1186/s40537-023-00707-6.

99. Paytaren, A. V. (2020). Seminar and training programs recommender system for faculty members of higher educational institution. International Journal of Computing Sciences Research, 4(4), 359-369. doi: 10.25147/ijcsr.2017.001.1.43.

100. Amin, S., Uddin, M.I., Alarood, A.A. et al. An adaptable and personalized framework for top-N course recommendations in online learning. Sci Rep 14, 10382 (2024). URL: https://doi.org/10.1038/s41598-024-56497-1.

101. C. Walker et al., “Learning from the Past: using Peer Data to Improve Course Recommendations in Personalized Education,” Proceedings - 2024 IEEE 48th Annual Computers, Software, and Applications Conference, COMPSAC 2024, pp. 127 - 137, Institute of Electrical and Electronics Engineers, Jan 2024. doi: 10.1109/COMPSAC61105.2024.00028.

102. Vineeta Anand and Ashish Kumar Maurya. 2024. A Survey on Recommender Systems Using Graph Neural Network. ACM Trans. Inf. Syst. 43, 1, Article 9 (November 2024), 49 pages. URL: https://doi.org/10.1145/3694784.

103. Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In

Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 726–735.

104. Di Palma, D. Beyond Words: Can ChatGPT Support State-of-the-Art Recommender Systems? / Dario Di Palma, Giovanni Servedio, Vito Walter Anelli, Giovanni Maria Biancofiore, Fedelucio Narducci, Leonarda Carnimeo, Tommaso Di Noia // Proceedings of the International Conference on Information Retrieval (IIR 2024). 2024. P. 13–22. URL: https://ceur-ws.org/Vol-3802/paper6.pdf.

105. Ma, Z., & Wu, J. (2025). Design of an LLM-Driven Personalized Learning Resource Recommendation System: -- A Comparative Study. Frontiers in Computing and Intelligent Systems, 11(3), 91-94. URL: https://doi.org/10.54097/e8et1g27.

106. Забєлін, О., & Кубіцький, С. (2024). РЕКОМЕНДАЦІЇ ЩОДО ВДОСКОНАЛЕННЯ СИСТЕМИ ПІДВИЩЕННЯ КВАЛІФІКАЦІЇ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ. Матеріали конференцій МЦНД, (11.10.2024; Луцьк, Україна), 46–52. Вилучено із https://archives.mcnd.org.ua/index.php/conference-proceeding/article/view/221.

107. Siafis, V., Rangoussi, M., & Psaromiligkos, Y. (2024). Recommender Systems for Teachers: A Systematic Literature Review of Recent (2011–2023) Research. *Education Sciences*, *14*(7), 723. https://doi.org/10.3390/educsci14070723.

108. UNESCO. (2023). UNESCO calls on governments to regulate generative AI in education. United Nations News. Retrieved from https://news.un.org/en/story/2023/05/1137117.

109. Fagbohun, O., Harrison, R. M., Dereventsov, A. (2023). An Empirical Catego-rization of Prompting Techniques for Large Language Models: A Practitioner's Guide. Journal of Artificial Intelligence, Machine Learning and Data Science, 1 (4), 1–11. Url: https://doi.org/10.51219/jaimld/oluwole-fagbohun/15.

110. G. Beri and V. Srivastava, "Advanced Techniques in Prompt Engineering for Large Language Models: A Comprehensive Study," in 2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG), Indore, India, 2024, pp. 1–4, doi: 10.1109/ICTBIG64922.2024.10911672.

111. Vincent-Lancrin, S. and R. van der Vlies (2020), “Trustworthy artificial intelligence (AI) in education: Promises and challenges”, OECD Education Working Papers, No. 218, OECD Publishing, Paris, https://doi.org/10.1787/a6c90fa9-en.

112. Levy, A. A Guide to Similarity Measures [Electronic resource] / Avivit Levy, B. Riva Shalom, Michal Chalamish // arXiv preprint arXiv:2408.07706. – 2024. – URL: https://arxiv.org/abs/2408.07706.

113. Falah Z., Suryawan F. Recommendation System to Propose Final Project Supervisors using Cosine Similarity Matrix // Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika. – 2022. – Vol. 8. – URL: https://doi.org/10.23917/khif.v8i2.16235.

114. Jain G., Mahara T., Sharma S. C. Performance Evaluation of Time-based Recommendation System in Collaborative Filtering Technique // Procedia Computer Science. – 2023. – Vol. 218. – P. 1834–1844. – URL: https://doi.org/10.1016/j.procs.2023.01.161.

115. Van Deventer, H. From Interests to Insights: An LLM Approach to Course Recommendations Using Natural Language Queries [Electronic resource] / Hugh Van Deventer, Mark Mills, August Evrard // arXiv preprint arXiv:2412.19312. – 2024. – URL: https://arxiv.org/abs/2412.19312.

116. Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5), 2023. URL: https://arxiv.org/abs/2203.13366.

117. Molina, M. What is an intelligent system? 2020. DOI: 10.48550/arXiv.2009.09083.

118. Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. 2024. OpenP5: An OpenSource Platform for Developing, Training, and Evaluating LLM-based Recommender Systems. In Proceedings of the 47th International ACM SIGIRConference on Research and Development in Information Retrieval (SIGIR ‘24), July 14–18, 2024, Washington, DC, USA. ACM, New York, NY, USA, 9 pages.https://doi.org/10.1145/3626772.3657883.

119. Zeyu Cui, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. M6-rec: Generative pretrained language models are open-ended recommender systems, 2022. URL: https://arxiv.org/abs/2205.08084.

120. Irani, H. Positional Encoding in Transformer-Based Time Series Models: A Survey [Electronic resource] / Habib Irani, Vangelis Metsis // arXiv preprint arXiv:2502.12370. – 2025. – URL: https://arxiv.org/abs/2502.12370.

121. PyTorch – PyTorch documentation — PyTorch 2.7 documentation [Електронний ресурс] // PyTorch. – Режим доступу: https://docs.pytorch.org/docs/stable/index.html (дата звернення: 22.07.2025).

122. Neo4j – Cypher and Neo4j – Cypher Manual [Електронний ресурс] // Neo4j. – Режим доступу: https://neo4j.com/developer/cypher-basics-i/ (дата звернення: 19.07.2025).

123. Верховна Рада України Online – Про затвердження Ліцензійних умов провадження освітньої діяльності | від 30.12.2015 № 1187 [Електронний ресурс] // Верховна Рада України Online. – Режим доступу: https://zakon.rada.gov.ua/laws/show/1187-2015-п#n1217 (дата звернення: 21.07.2025).

124. OpenAI. [Електронний ресурс]. Режим доступу: https://platform.openai.com/docs/overview (дата звернення: 24.07.2025).

125. OpenRouter – OpenRouter Quickstart Guide | Developer Documentation | OpenRouter | Documentation [Електронний ресурс] // OpenRouter – Режим доступу: https://openrouter.ai/docs/quickstart (дата звернення: 24.07.2025).

126. Walek, B. A text-based recommender system for recommending relevant news articles / Bogdan Walek, Patrik Müller // Expert Systems with Applications. – 2025. – Vol. 266. – P. 125816. – DOI: https://doi.org/10.1016/j.eswa.2024.125816.

127. K. Hooper, “Use OpenAI (GPT-3) to Create an AI Email Parser in JavaScript”, 2023 [Electronic resource] – Url: https://ai.plainenglish.io/using-openai-gpt-3-to-create-an-ai-email-parser-in-javascript-480029ee873f.

128. Radilova M, Kamencay P, Hudec R, Benco M, Radil R. Tool for Parsing Important Data from Web Pages. Applied Sciences. 2022; 12(23):12031. Url: https://doi.org/10.3390/app122312031.

129. Grounding with Google Search | Gemini API [Електронний ресурс] // Google AI for Developers – Режим доступу: https://ai.google.dev/gemini-api/docs/google-search (дата звернення: 29.07.2025).

130. Tomasz Smutek Et Al (2024). A Graph-Based Recommendation System Leveraging Cosine Similarity for Enhanced Marketing Decisions, European Research Studies Journal Volume XXVII Special Issue A, 83-93.

131. Петренко, М. Г. Knowledge-Oriented Tool Complex for Developing Databases of Scientific Publications and Taking into account Semantic Web Technology / М. Г. Петренко, О. В. Палагін, М. О. Бойко, С. М. Матвейшин // Control Systems and Computers. – 2022. – № 3 (299). – С. 11–28. – UDC 004.318. – DOI: https://doi.org/10.15407/csc.2022.03.011.

132. Sofia Eleni Spatharioti, David Rothschild, Daniel G Goldstein, and Jake M Hofman. 2025. Effects of LLM-based Search on Decision Making: Speed, Accuracy, and Overreliance. In Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI ‘25). Association for Computing Machinery, New York, NY, USA, Article 1025, 1–15. https://doi.org/10.1145/3706598.3714082.

133. Vaswani, A. Attention Is All You Need [Electronic resource] / Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin // arXiv preprint arXiv:1706.03762. – 2017. – URL: https://arxiv.org/abs/1706.03762.

ДОДАТОК А

ОСНОВНІ DLL БАЗИ ДАНИХ ІНТЕЛЕКТУАЛІЗОВАНОЇ СИСТЕМИ РЕЙТИНГУВАННЯ

```
---- Table structure for table `answers`
CREATE TABLE `answers` (
  `id` int UNSIGNED NOT NULL,
  `points` decimal(10,2) DEFAULT NULL,
  `text` text  NOT NULL,
  `options` text  NOT NULL,
  `files` varchar(255)  NOT NULL,
  `urls` text  NOT NULL,
  `N` varchar(255)  NOT NULL DEFAULT ‘0’,
  `K` varchar(255)  NOT NULL DEFAULT ‘0’,
  `report_id` int UNSIGNED NOT NULL,
  `field_id` int UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `blocked_by_approver` tinyint(1) NOT NULL DEFAULT ‘0’,
  `publications_ids` text ,
  `keywords` text ,
  `is_auto_filled` tinyint UNSIGNED DEFAULT ‘0’
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `departments`
CREATE TABLE `departments` (
  `id` int UNSIGNED NOT NULL,
  `name` text  NOT NULL,
  `user_id` int UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `faculty_id` int UNSIGNED DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `department_user`
CREATE TABLE `department_user` (
  `id` int UNSIGNED NOT NULL,
  `department_id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;
```

```
---- Table structure for table `disciplines`
CREATE TABLE `disciplines` (
  `id` int UNSIGNED NOT NULL,
  `name` varchar(200) DEFAULT NULL,
  `purpose_studying` text,
  `learning_objectives` text,
  `learning_outcomes` text,
  `academic_program` text,
  `practical_topics` text,
  `scientific_degree` int DEFAULT NULL,
  `is_active` tinyint UNSIGNED DEFAULT '0',
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `AI_academic_program` text,
  `course_number` int DEFAULT NULL,
  `speciality` varchar(3) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `faculties`
CREATE TABLE `faculties` (
  `id` int UNSIGNED NOT NULL,
  `name` varchar(255) NOT NULL,
  `dean` varchar(255) NOT NULL,
  `phone_number` varchar(20) DEFAULT NULL,
  `site` varchar(140) DEFAULT NULL,
  `contact_person` varchar(255) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `email` varchar(70) DEFAULT NULL,
  `abbreviation` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `fields`
CREATE TABLE `fields` (
  `id` int UNSIGNED NOT NULL,
  `text` text  NOT NULL,
  `notes` text  NOT NULL,
  `max_point` text  NOT NULL,
  `priority` int NOT NULL DEFAULT '0',
  `block` int NOT NULL,
  `type` int NOT NULL DEFAULT '1',
  `elements` int NOT NULL DEFAULT '1',
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `coefficient` varchar(20)  DEFAULT '1'
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `historical_change_specialities`
CREATE TABLE `historical_change_specialities` (
  `id` int UNSIGNED NOT NULL,
  `code_2011` varchar(20) DEFAULT NULL,
  `name_2011` varchar(200) DEFAULT NULL,
  `code_2015` varchar(20) DEFAULT NULL,
  `name_2015` varchar(200) DEFAULT NULL,
  `code_2024` varchar(20) DEFAULT NULL,
  `name_2024` varchar(200) DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `options`
CREATE TABLE `options` (
  `id` int UNSIGNED NOT NULL,
  `name` text  NOT NULL,
  `max_point` text  NOT NULL,
  `field_id` int UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `prompts`
CREATE TABLE `prompts` (
  `id` int UNSIGNED NOT NULL,
  `description` text,
  `type` int DEFAULT NULL,
  `prompt_text` text,
  `prompt_params` text,
  `is_system` int DEFAULT ‘0’,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `is_active` int DEFAULT ‘0’,
  `model_group` int DEFAULT ‘0’,
  `special_model` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `publications`
CREATE TABLE `publications` (
  `id` int UNSIGNED NOT NULL,
  `external_id` varchar(50) DEFAULT NULL,
  `number` int DEFAULT NULL,
```

```
  `type` int DEFAULT NULL,
  `category` varchar(255) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  `publisher` varchar(255) DEFAULT NULL,
  `date_publication` datetime DEFAULT NULL,
  `page_numbers` int DEFAULT NULL,
  `page_author_numbers` decimal(10,2) DEFAULT NULL,
  `impact_factor` decimal(10,2) DEFAULT NULL,
  `quartile` int DEFAULT NULL,
  `url` varchar(255) DEFAULT NULL,
  `added_by_user_id` int DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `keywords` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `publication_authors`
CREATE TABLE `publication_authors` (
  `id` int UNSIGNED NOT NULL,
  `publication_id` int UNSIGNED DEFAULT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `fn` varchar(255) DEFAULT NULL,
  `ln` varchar(255) DEFAULT NULL,
  `mn` varchar(255) DEFAULT NULL,
  `page_author_numbers` decimal(10,2) DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `reports`
CREATE TABLE `reports` (
  `id` int UNSIGNED NOT NULL,
  `points` decimal(10,2) DEFAULT NULL,
  `block1` varchar(255)  NOT NULL DEFAULT ‘0’,
  `block2` varchar(255)  NOT NULL DEFAULT ‘0’,
  `block3` varchar(255)  NOT NULL DEFAULT ‘0’,
  `block4` varchar(255)  NOT NULL DEFAULT ‘0’,
  `block5` varchar(255)  NOT NULL DEFAULT ‘0’,
  `block6` varchar(255)  NOT NULL DEFAULT ‘0’,
  `active` tinyint(1) NOT NULL DEFAULT ‘1’,
  `blocked` tinyint NOT NULL DEFAULT ‘0’,
  `template_id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
```

```
  `coefficient` varchar(20)  DEFAULT '1',
  `points_without_coeff` decimal(10,2) DEFAULT NULL,
  `date_first_filling` datetime DEFAULT NULL,
  `date_filling` datetime DEFAULT NULL,
  `date_approve_hod` datetime DEFAULT NULL,
  `approve_status` tinyint(1) NOT NULL DEFAULT '0',
  `count_blocked_answers` int DEFAULT NULL,
  `count_answers` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `templates`
CREATE TABLE `templates` (
  `id` int UNSIGNED NOT NULL,
  `year` int NOT NULL,
  `block1` varchar(255)  NOT NULL,
  `block2` varchar(255)  NOT NULL,
  `block3` varchar(255)  NOT NULL,
  `block4` varchar(255)  NOT NULL,
  `block5` varchar(255)  NOT NULL,
  `block6` varchar(255)  NOT NULL,
  `active` tinyint(1) NOT NULL DEFAULT '1',
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `users`
CREATE TABLE `users` (
  `id` int UNSIGNED NOT NULL,
  `name` text  NOT NULL,
  `email` varchar(255)  NOT NULL,
  `password` varchar(255)  NOT NULL,
  `type` int NOT NULL DEFAULT '3',
  `qualification` int NOT NULL DEFAULT '4',
  `science` int NOT NULL DEFAULT '3',
  `qualification_files` varchar(255)  NOT NULL,
  `science_files` varchar(255)  NOT NULL,
  `active` tinyint(1) NOT NULL DEFAULT '1',
  `remember_token` varchar(100)  DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  `is_approver` tinyint(1) NOT NULL DEFAULT '0',
  `fn` varchar(50)  DEFAULT NULL,
  `ln` varchar(50)  DEFAULT NULL,
  `mn` varchar(50)  DEFAULT NULL,
  `enfn` varchar(50)  DEFAULT NULL,
```

```
  `enln` varchar(50)  DEFAULT NULL,
  `scid` varchar(50)  DEFAULT NULL,
  `wosid` varchar(50)  DEFAULT NULL,
  `gsid` varchar(50)  DEFAULT NULL,
  `orcid` varchar(50)  DEFAULT NULL,
  `dis_science_info1` int DEFAULT NULL,
  `dis_code_info1` varchar(20)  DEFAULT NULL,
  `dis_name_info1` varchar(200)  DEFAULT NULL,
  `dis_protection_date_info1` datetime DEFAULT NULL,
  `dis_science_info2` int DEFAULT NULL,
  `dis_code_info2` varchar(20)  DEFAULT NULL,
  `dis_name_info2` varchar(100)  DEFAULT NULL,
  `dis_protection_date_info2` datetime DEFAULT NULL,
  `is_need_update_contact_info` int DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_publication_discipline_compliances`
CREATE TABLE `user_publication_discipline_compliances` (
  `id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `discipline_id` int UNSIGNED DEFAULT NULL,
  `template_id` int UNSIGNED DEFAULT NULL,
  `year` int DEFAULT NULL,
  `report_id` int UNSIGNED DEFAULT NULL,
  `field_id` int UNSIGNED DEFAULT NULL,
  `answer_id` int UNSIGNED DEFAULT NULL,
  `key_index` int DEFAULT NULL,
  `percentage_article_compliance` decimal(10,2) DEFAULT NULL,
  `is_active` tinyint UNSIGNED DEFAULT '0',
  `AI_model` varchar(100) DEFAULT NULL,
  `prompt_id` int UNSIGNED DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `title` text,
  `keywords` text,
  `explanation` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_recommendation_similar_strategies`
CREATE TABLE `user_recommendation_similar_strategies` (
  `id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `recommend_user_id` int UNSIGNED DEFAULT NULL,
  `year` int DEFAULT NULL,
  `similarity_percentage` decimal(10,2) DEFAULT NULL,
```

```
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_report_accreditations`
CREATE TABLE `user_report_accreditations` (
  `id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `report_id` int UNSIGNED DEFAULT NULL,
  `Pub_Naukovi1` text,
  `Pub_Naukovi_Short1` text,
  `Patent_Avtorsk2` text,
  `Patent_Avtorsk_Short2` text,
  `Book_Monograph3` text,
  `Book_Monograph_Short3` text,
  `Navch_Metodychni4` text,
  `Navch_Metodychni_Short4` text,
  `Disertacia5` text,
  `Disertacia_Short5` text,
  `Keruvannya_Zdobuvach6` text,
  `Keruvannya_Zdobuvach_Short6` text,
  `Spets_Rada7` text,
  `Spets_Rada_Short7` text,
  `Nauk_Kerivnyk8` text,
  `Nauk_Kerivnyk_Short8` text,
  `Ekspert_Rada_MON9` text,
  `Ekspert_Rada_MON_Short9` text,
  `Mizhnar_Proekty10` text,
  `Mizhnar_Proekty_Short10` text,
  `Konstult_Org11` text,
  `Konstult_Org_Short11` text,
  `Aprob_Publik12` text,
  `Aprob_Publik_Short12` text,
  `Inozem_Mova13` text,
  `Inozem_Mova_Short13` text,
  `Keruvannya_Student14` text,
  `Keruvannya_Student_Short14` text,
  `Keruvannya_Shkoliar15` text,
  `Keruvannya_Shkoliar_Short15` text,
  `Boiov_Dii16` text,
  `Boiov_Dii_Short16` text,
  `OON_Operatsii17` text,
  `OON_Operatsii_Short17` text,
  `Viyshk_Navch18` text,
  `Viyshk_Navch_Short18` text,
```

```
  `Prof_Obednannya19` text,
  `Prof_Obednannya_Short19` text,
  `Prakt_Robot20` text,
  `Prakt_Robot_Short20` text,
  `is_active` tinyint(1) DEFAULT NULL,
  `generate_status` int DEFAULT NULL,
  `AI_model` varchar(100) DEFAULT NULL,
  `prompt_id` int UNSIGNED DEFAULT NULL,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `prompt_text` text,
  `total_tokens` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_report_field_recommendations`
CREATE TABLE `user_report_field_recommendations` (
  `id` int UNSIGNED NOT NULL,
  `user_report_recommendation_id` int DEFAULT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `report_id` int UNSIGNED DEFAULT NULL,
  `rec_field_id` int UNSIGNED DEFAULT NULL,
  `potential_score` decimal(10,2) DEFAULT NULL,
  `rec_group` int DEFAULT NULL,
  `variant` int DEFAULT NULL,
  `probability` decimal(10,2) DEFAULT NULL,
  `rec_details` text,
  `is_active` tinyint UNSIGNED DEFAULT ‘0’,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `similar_user_ids` text
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_report_recommendations`
CREATE TABLE `user_report_recommendations` (
  `id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `report_id` int UNSIGNED DEFAULT NULL,
  `prompt_type` int DEFAULT NULL,
  `full_rec_text` text,
  `is_active` tinyint UNSIGNED DEFAULT ‘0’,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `similar_user_ids` text,
  `prompt_id` int UNSIGNED DEFAULT NULL,
  `processing_model` varchar(100) DEFAULT NULL,
```

```
  `user_appraisal` int DEFAULT NULL,
  `user_comments` text,
  `user_report_accreditation_id` int UNSIGNED DEFAULT NULL,
  `accreditation_res_status` int DEFAULT NULL,
  `prompt_text` text,
  `total_tokens` varchar(50) DEFAULT NULL,
  `u_comprehensibility` int DEFAULT NULL,
  `u_usefulness` int DEFAULT NULL,
  `u_realism` int DEFAULT NULL,
  `u_structured` int DEFAULT NULL,
  `u_relevance` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_report_scrapers`
CREATE TABLE `user_report_scrapers` (
  `id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED DEFAULT NULL,
  `report_id` int UNSIGNED DEFAULT NULL,
  `template_id` int UNSIGNED DEFAULT NULL,
  `context_full` text,
  `context_short` text,
  `created_at` datetime DEFAULT NULL,
  `updated_at` datetime DEFAULT NULL,
  `is_generated_short` tinyint(1) DEFAULT ‘1’,
  `con_full_s1` text,
  `con_full_s2` text,
  `con_full_s3` text,
  `con_full_s4` text,
  `con_full_s5` text,
  `con_full_s6` text,
  `article_shStat` text,
  `hInd_shStat` text,
  `articles_sh` text,
  `authors_sh` text,
  `AI_model` varchar(50) DEFAULT NULL,
  `prompt_text` text,
  `total_tokens` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;

---- Table structure for table `user_science_infos`
CREATE TABLE `user_science_infos` (
  `id` int UNSIGNED NOT NULL,
  `user_id` int UNSIGNED NOT NULL,
  `science_info` int NOT NULL,
  `code_info` varchar(20) NOT NULL,
```

```
    `name_info` varchar(200) DEFAULT NULL,
    `protection_date_info` datetime DEFAULT NULL,
    `protection_confirm_file` varchar(255) DEFAULT NULL,
    `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
    `updated_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

ДОДАТОК Б

ЛІСТИНГ КОДУ ОСНОВНИХ МОДУЛІВ ПРОГРАМИ

```
import os
import pickle
import torch
import numpy as np
import pandas as pd
import csv
import torch
import torch.nn.functional as F
import torch.nn as nn
from dtaidistance import dtw
import importlib
required_packages = [
    "torch_geometric"
]
for package in required_packages:
    try:
        importlib.import_module(package)
    except ImportError:
        !pip install {package.replace('_', '-')} --quiet

from torch_geometric.data import HeteroData
from torch_geometric.nn import HGTConv, Linear
from sklearn.preprocessing import MinMaxScaler
from datetime import datetime
from sentence_transformers import SentenceTransformer
from ast import literal_eval
from sklearn.metrics.pairwise import cosine_similarity
from surprise import Dataset, Reader, SVDpp
from surprise.model_selection import train_test_split
from surprise import accuracy
```

Пошук подібних користувачів за діяльністю але з кращими показниками

```
def compute_similarity_index(target_user_id, top_n=5):
    # Prepare data
    pivot_year_score = reports_df.pivot_table(index="user_id", columns="year", values="user_answer_points", aggfunc="sum").fillna(0)
    u1_year_vec_all = pivot_year_score.loc[target_user_id].values.reshape(1, -1)
    years_u1 = set(reports_df[reports_df["user_id"] == target_user_id]["year"])
    pivot_user_crit = reports_df.pivot_table(index="user_id", columns="criterion_id", values="user_answer_points", aggfunc="sum").fillna(0)
    u1_vector = pivot_user_crit.loc[target_user_id].values.reshape(1, -1)
    u1_criteria_set = set((row["year"], row["criterion_id"]) for _, row in reports_df[reports_df["user_id"] == target_user_id].iterrows())
    u1_current_score = reports_df[(reports_df["user_id"] == target_user_id) & (reports_df["year"] == reports_df["year"].max())]["user_answer_points"].sum()
    df_curr_year = reports_df[reports_df["year"] == reports_df["year"].max()].merge(
        criterion_info_df[["criterion_id", "criterion_section"]],
```

```
        on="criterion_id", how="left"
    )
        pivot_section_score = df_curr_year.pivot_table(index="user_id", columns="criterion_section", values="user_answer_points", aggfunc="sum").fillna(0)
    growth_rate = (
        reports_df.groupby("user_id")["user_answer_points"]
        .agg(lambda x: (x.iloc[-1] / x.iloc[0] - 1) if len(x) > 1 else np.nan)
        .to_dict()
    )
    results_raw = []
    for other_id in pivot_user_crit.index:
        if other_id == target_user_id:
            continue

        # Cosine
        u2_vector = pivot_user_crit.loc[other_id].values.reshape(1, -1)
        cosine_sim_crit = cosine_similarity(u1_vector, u2_vector)[0][0]

        # Jaccard
        u2_criteria_set = set((row["year"], row["criterion_id"]) for _, row in reports_df[reports_df["user_id"] == other_id].iterrows())
        inter = len(u1_criteria_set & u2_criteria_set)
        union = len(u1_criteria_set | u2_criteria_set)
        jaccard = inter / union if union else 0

        # Динаміка активності Y()
        gr1 = growth_rate.get(target_user_id, np.nan)
        gr2 = growth_rate.get(other_id, np.nan)
        epsilon = 1e-6
        if not (np.isnan(gr1) or np.isnan(gr2)):
            denominator = max(abs(gr1), abs(gr2)) + epsilon
            growth_sim = max(0, 1 - abs(gr1 - gr2) / denominator)
        else:
            growth_sim = 0

        # Фактор переваги F()
        sa_u1 = reports_df[reports_df["user_id"] == target_user_id]["user_answer_points"].mean()
        sa_u2 = reports_df[reports_df["user_id"] == other_id]["user_answer_points"].mean()
        factor_preference = min(1, sa_u2 / (sa_u1 + 0.01)) if sa_u1 > 0 else 0

        # DTWS
        total_score_u1 = reports_df[reports_df["user_id"] == target_user_id].groupby("year")["user_answer_points"].sum().sort_index().values
        total_score_u2 = reports_df[reports_df["user_id"] == other_id].groupby("year")["user_answer_points"].sum().sort_index().values
        if len(total_score_u1) > 0 and len(total_score_u2) > 0:
            dtws_distance = dtw.distance(total_score_u1, total_score_u2)
        else:
            dtws_distance = float('inf')

        # DTWG
        if other_id in pivot_section_score.index:
```

```
            sga_u1 = pivot_section_score.loc[target_user_id].values
            sga_u2 = pivot_section_score.loc[other_id].values
            if len(sga_u1) > 0 and len(sga_u2) > 0:
                dtwg_distance = dtw.distance(sga_u1, sga_u2)
            else:
                dtwg_distance = float('inf')
        else:
            dtwg_distance = float('inf')

            # сумарна перевага P()
        common_criteria = set(reports_df[reports_df["user_id"] == target_user_id]["criterion_id"]) & \
                set(reports_df[reports_df["user_id"] == other_id]["criterion_id"])
        scores_u1 = reports_df[(reports_df["user_id"] == target_user_id) &
                        (reports_df["criterion_id"].isin(common_criteria))]
        scores_u2 = reports_df[(reports_df["user_id"] == other_id) &
                        (reports_df["criterion_id"].isin(common_criteria))]
        p_value = scores_u2.set_index("criterion_id")["user_answer_points"] - \
                scores_u1.set_index("criterion_id")["user_answer_points"]
        p_sum = p_value[p_value > 0].sum()

        # Build result
            results_raw.append({"user_id": other_id, "cosine_crit": round(cosine_sim_crit, 4), "jaccard":
round(jaccard, 4), growth": round(-1 * growth_sim, 4), factor_preference": round(factor_preference, 4),
"dtws": round(dtws_distance, 4), "dtwg": round(dtwg_distance, 4), "preference_p": round(p_sum, 4)})

    # Normalize
    distances = np.array([r["dtws"] for r in results_raw]).reshape(-1, 1)
    scaler = MinMaxScaler()
    normed_dtws = scaler.fit_transform(distances).flatten()
    for i, r in enumerate(results_raw):
        r["sim_dtws"] = round(1 - normed_dtws[i], 4)
    distances = np.array([r["dtwg"] for r in results_raw]).reshape(-1, 1)
    scaler = MinMaxScaler()
    normed_dtwg = scaler.fit_transform(distances).flatten()
    for i, r in enumerate(results_raw):
        r["sim_dtwg"] = round(1 - normed_dtwg[i], 4)

    # Main simi index
    for r in results_raw:
        r["score"] = r["factor_preference"] * (round(0.2 * r["sim_dtws"] + 0.2 * r["cosine_crit"] + 0.2 *
r["sim_dtwg"] +  0.2 * r["jaccard"] + 0.1 * r["preference_p"]  + 0.1 * r["growth"], 4))
    return sorted(results_raw, key=lambda x: x["score"], reverse=True)[:top_n]
```

Загальні функції, ініціалізація даних, розрахунок семантичної подібності

```
def temporal_encoding(year, dim=16, base=10000):
    position = torch.tensor(year - MAIN_YEAR_FROM, dtype=torch.float32)
    div_term = torch.exp(torch.arange(0., dim, 2) * -(np.log(base) / dim))
    pe = torch.zeros(dim)
    pe[0::2] = torch.sin(position * div_term)
    pe[1::2] = torch.cos(position * div_term)
    return pe
```

```
def get_semantic_similarity(crit_vec1, crit_vec2):
    crit_vec1 = np.array(crit_vec1) if isinstance(crit_vec1, list) else crit_vec1
    crit_vec2 = np.array(crit_vec2) if isinstance(crit_vec2, list) else crit_vec2
    return np.dot(crit_vec1, crit_vec2) / (np.linalg.norm(crit_vec1) * np.linalg.norm(crit_vec2) + 1e-8)

def load_data():
    global df_performance, df_criteria_info, df_similarity, df_departments, df_user_department_raw, df_availability

    df_reports = pd.read_csv(MAIN_REPORTS_FILE, header=None, names=["user_id", "year", "criterion_id", "points"])
    df_reports['user_id'] = df_reports['user_id'].astype(int)
    df_reports['criterion_id'] = df_reports['criterion_id'].astype(int)
    df_criteria_info_local = pd.read_csv(MAIN_CRITERION_INFO_FILE, header=None, names=["criterion_id", "description", "category_id", "criterion_points", "scoreType", "minScore"])
    df_criteria_info_local['criterion_id'] = df_criteria_info_local['criterion_id'].astype(int)

    df_availability = pd.read_csv(MAIN_CRITERION_AVAILABILITY_FILE, header=None, names=["criterion_id", "year"])
    df_availability['criterion_id'] = df_availability['criterion_id'].astype(int)
    df_user_department_raw = pd.read_csv(MAIN_USER_TO_DEPARTMENT_FILE, header=None, names=["user_id", "department_id", "year"])
    df_user_department_raw['user_id'] = df_user_department_raw['user_id'].astype(int)
    df_user_department_raw['department_id'] = df_user_department_raw['department_id'].astype(int)
    df_department_info_raw = pd.read_csv(MAIN_DEPARTMENT_INFO_FILE, header=None, names=["department_id", "department_name", "department_type"])
    df_department_info_raw['department_id'] = df_department_info_raw['department_id'].astype(int)
    df_simi = pd.read_csv(MAIN_USER_SIMILARITY_USERS_FILE, header=None, names=["user_id", "similar_user_id", "simi", "year"])
    df_simi['user_id'] = df_simi['user_id'].astype(int)
    df_simi['similar_user_id'] = df_simi['similar_user_id'].astype(int)
    df_perf = df_reports.copy()
    df_perf.rename(columns={'points': 'score'}, inplace=True)
    df_perf = pd.merge(df_perf, df_criteria_info_local[['criterion_id', 'description', 'category_id', 'criterion_points']], on='criterion_id', how='left')
    df_criteria_info_local.rename(columns={'category_id': 'category', 'criterion_points': 'max_points'}, inplace=True)

    scaler = MinMaxScaler()
    df_perf['score_scaled'] = df_perf.groupby('criterion_id')['score'].transform(lambda x: scaler.fit_transform(x.values.reshape(-1, 1)).flatten())

    bert_model = SentenceTransformer("paraphrase-multilingual-mpnet-base-v2")
    df_criteria_info_local["SBERT_embedding"] = bert_model.encode(df_criteria_info_local["description"].tolist(), convert_to_tensor=False).tolist()

    df_dep_local = df_department_info_raw.copy()
    df_dep_local.rename(columns={'department_type': 'department_type'}, inplace=True)
    df_perf = pd.merge(df_perf, df_user_department_raw[['user_id', 'department_id', 'year']], on=['user_id', 'year'], how='left')
    df_crit_local = df_criteria_info_local.copy()
```

```
    df_performance, df_criteria_info, df_similarity, df_departments = df_perf, df_crit_local, df_simi, df_dep_local
    return df_perf, df_crit_local, df_simi, df_dep_local, df_user_department_raw, df_availability
```

Персональні рекомендації, побудова фіч

```
def get_node_features(df_perf, df_crit, df_dep, year, user_id_map, criterion_id_map, department_id_map):
    user_yearly_stats = df_perf[df_perf['year'] == year].groupby('user_id').agg(num_criteria=('criterion_id', 'nunique'), total_score=('score_scaled', 'sum'), avg_score=('score_scaled', 'mean')).reset_index()
    all_users_in_year = pd.DataFrame({'user_id': user_id_map.keys()})
    user_features = pd.merge(all_users_in_year, user_yearly_stats, on='user_id', how='left').fillna(0)
    temporal_embed = temporal_encoding(year, dim=temporal_embed_dim)
    temporal_embed_full = temporal_embed.repeat(user_features.shape[0], 1)
    x_user = torch.tensor(user_features[['num_criteria', 'total_score', 'avg_score']].values, dtype=torch.float32)
    x_user = torch.cat([x_user, temporal_embed_full], dim=1)

    criterion_categories_onehot = pd.get_dummies(df_crit['category'], prefix='category')
    ordered_criteria = pd.DataFrame({'criterion_id': list(criterion_id_map.keys())})
    ordered_criteria = pd.merge(ordered_criteria, df_crit, on='criterion_id', how='left')
    sbert_embeddings_array = np.array(ordered_criteria['SBERT_embedding'].tolist(), dtype=np.float32)
    x_criterion_components = [
        ordered_criteria[['max_points']].values.astype(np.float32),
        sbert_embeddings_array,
        pd.get_dummies(ordered_criteria['category'], prefix='category').values.astype(np.float32)
    ]
    x_criterion = torch.tensor(np.concatenate(x_criterion_components, axis=1), dtype=torch.float32)
    ordered_departments = pd.DataFrame({'department_id': list(department_id_map.keys())})
    ordered_departments = pd.merge(ordered_departments, df_dep, on='department_id', how='left')
    department_types_onehot = pd.get_dummies(ordered_departments['department_type'], prefix='dept_type')
    x_department = torch.tensor(department_types_onehot.values, dtype=torch.float32)
    return x_user, x_criterion, x_department
```

Побудова річних графів, з негативними кейсами та врахуванням зворотного зв'язку

```
def create_hetero_graph_for_year(df_perf, df_crit, df_simi, df_dep, df_user_department_raw, df_availability, year, user_id_map, criterion_id_map, department_id_map,
                                 all_performed_criteria_by_user_yearly, current_year_active_users_list_raw,
                                 df_unfulfilled_recommendations_current_year=None):
    data = HeteroData()
    data['user'].x, data['criterion'].x, data['department'].x = \
        get_node_features(df_perf, df_crit, df_dep, year, user_id_map, criterion_id_map, department_id_map)

    year_perf = df_perf[df_perf['year'] == year]
    user_criterion_edges_list = user_criterion_edge_labels = user_criterion_edge_scores = []
    user_criterion_edge_max_scores = []
```

```
    max_points_map = df_crit.set_index('criterion_id')['max_points'].to_dict()
    criterion_original_ids = {i: id for id, i in criterion_id_map.items()}

    for _, row in year_perf.iterrows():
        u_idx = user_id_map[row[‘user_id’]]
        c_idx = criterion_id_map[row[‘criterion_id’]]
        user_criterion_edges_list.append([u_idx, c_idx])
        user_criterion_edge_labels.append(1)
        user_criterion_edge_scores.append(row[‘score_scaled’])

        max_score = max_points_map.get(row['criterion_id'], 1.0)
        user_criterion_edge_max_scores.append(max_score)

    active_criteria_in_year = df_availability[df_availability[‘year’] == year][‘criterion_id’].unique()
    active_criterion_mapped_ids = {criterion_id_map[c_id] for c_id in active_criteria_in_year if c_id in
criterion_id_map}

    performed_criteria_by_user_this_year = {}
    for _, row in year_perf.iterrows():
        u_idx = user_id_map[row[‘user_id’]]
        c_idx = criterion_id_map[row[‘criterion_id’]]
        if u_idx not in performed_criteria_by_user_this_year:
            performed_criteria_by_user_this_year[u_idx] = set()
        performed_criteria_by_user_this_year[u_idx].add(c_idx)

        current_year_active_users_list = [uid for uid in current_year_active_users_list_raw if uid in
user_id_map]
    all_users_in_this_year_mapped = [user_id_map[uid] for uid in current_year_active_users_list]

    for u_idx in all_users_in_this_year_mapped:
        executed_criteria = performed_criteria_by_user_this_year.get(u_idx, set())
        non_executed_available_criteria = list(active_criterion_mapped_ids - executed_criteria)
        if len(non_executed_available_criteria) > 0:
            num_positives_for_user = len(executed_criteria)
            if num_positives_for_user == 0:
                                    num_negatives_to_add = min(len(non_executed_available_criteria),
len(active_criterion_mapped_ids) // 2)
            else:
                num_negatives_to_add = min(len(non_executed_available_criteria), num_positives_for_user
* NUM_RANDOM_NEGATIVES_PER_POSITIVE)
            if num_negatives_to_add > 0:
              random_negatives = random.sample(non_executed_available_criteria, num_negatives_to_add)
                for neg_c_idx in random_negatives:
                    if [u_idx, neg_c_idx] not in user_criterion_edges_list:
                        user_criterion_edges_list.append([u_idx, neg_c_idx])
                        user_criterion_edge_labels.append(0)
                        user_criterion_edge_scores.append(0.0)

                        neg_crit_original_id = criterion_original_ids[neg_c_idx]
                        max_score = max_points_map.get(neg_crit_original_id, 1.0)
                        user_criterion_edge_max_scores.append(max_score)
```

```
    if not df_simi.empty:
                                        performed_criteria_by_user = df_perf[df_perf['year'] == year].groupby('user_id')['criterion_id'].apply(set).to_dict()
        for user_original_id in current_year_active_users_list:
            u_idx = user_id_map[user_original_id]
            current_user_performed = performed_criteria_by_user.get(user_original_id, set())
                                        similar_better_users_for_user_df = df_simi[(df_simi['user_id'] == user_original_id)]['similar_user_id']
            similar_better_users_for_user = similar_better_users_for_user_df.tolist()
            potential_negative_criteria = set()
            for s_user_original_id in similar_better_users_for_user:
                if s_user_original_id in current_year_active_users_list:
                    s_performed_criteria = performed_criteria_by_user.get(s_user_original_id, set())
                    potential_negative_criteria.update(s_performed_criteria - current_user_performed)
                    num_positive_for_user = len([1 for u,c in user_criterion_edges_list if u == u_idx and user_criterion_edge_labels[user_criterion_edges_list.index([u,c])] == 1])
                    num_strategic_negatives_to_add = min(len(potential_negative_criteria), max(5, num_positive_for_user * 2))
            if potential_negative_criteria:
                    sampled_negative_criteria_original_ids = random.sample(list(potential_negative_criteria), num_strategic_negatives_to_add)
                for neg_crit_original_id in sampled_negative_criteria_original_ids:
                    if neg_crit_original_id in criterion_id_map:
                        c_idx = criterion_id_map[neg_crit_original_id]
                        if [u_idx, c_idx] not in user_criterion_edges_list:
                            user_criterion_edges_list.append([u_idx, c_idx])
                            user_criterion_edge_labels.append(0)
                            user_criterion_edge_scores.append(0.0)

                    if df_unfulfilled_recommendations_current_year is not None and not df_unfulfilled_recommendations_current_year.empty:
        current_year_unfulfilled_recs = df_unfulfilled_recommendations_current_year[
            (df_unfulfilled_recommendations_current_year['year'] == year)
        ]
        for _, row in current_year_unfulfilled_recs.iterrows():
            u_original_id = row['user_id']
            crit_original_id = row['criterion_id']
            if u_original_id in user_id_map and crit_original_id in criterion_id_map:
                u_idx = user_id_map[u_original_id]
                c_idx = criterion_id_map[crit_original_id]
                if [u_idx, c_idx] not in user_criterion_edges_list:
                    user_criterion_edges_list.append([u_idx, c_idx])
                    user_criterion_edge_labels.append(0)
                    user_criterion_edge_scores.append(0.0)

            data['user', 'reports', 'criterion'].edge_index = torch.tensor(user_criterion_edges_list, dtype=torch.long).T
            data['user', 'reports', 'criterion'].edge_label = torch.tensor(user_criterion_edge_labels, dtype=torch.float32)
            data['user', 'reports', 'criterion'].edge_score = torch.tensor(user_criterion_edge_scores, dtype=torch.float32)
```

```
    data['user', 'reports', 'criterion'].edge_max_score = torch.tensor(user_criterion_edge_max_scores, dtype=torch.float32)

    data[‘criterion’, ‘reported_by’, ‘user’].edge_index = data[‘user’, ‘reports’, ‘criterion’].edge_index.flip([0])

  similar_edges = []
  for _, row in df_simi.iterrows():
    u1_idx = user_id_map.get(row[‘user_id’])
    u2_idx = user_id_map.get(row[‘similar_user_id’])
    if u1_idx is not None and u2_idx is not None:
      similar_edges.append([u1_idx, u2_idx])
  if similar_edges:
    data[‘user’, ‘similar’, ‘user’].edge_index = torch.tensor(similar_edges, dtype=torch.long).T
    data[‘user’, ‘is_similar’, ‘user’].edge_index = data[‘user’, ‘similar’, ‘user’].edge_index.flip([0])
  else:
    data[‘user’, ‘similar’, ‘user’].edge_index = torch.empty((2, 0), dtype=torch.long)
    data[‘user’, ‘is_similar’, ‘user’].edge_index = torch.empty((2, 0), dtype=torch.long)

  dept_edges = []
  year_user_dept = df_user_department_raw[df_user_department_raw[‘year’] == year]
  for _, row in year_user_dept[[‘user_id’, ‘department_id’]].drop_duplicates().iterrows():
    u_idx = user_id_map.get(row[‘user_id’])
    d_idx = department_id_map.get(row[‘department_id’])
    if u_idx is not None and d_idx is not None:
      dept_edges.append([u_idx, d_idx])
  if dept_edges:
    data[‘user’, ‘belongs_to’, ‘department’].edge_index = torch.tensor(dept_edges, dtype=torch.long).T
    data[‘department’, ‘has’, ‘user’].edge_index = data[‘user’, ‘belongs_to’, ‘department’].edge_index.flip([0])
  else:
    data[‘user’, ‘belongs_to’, ‘department’].edge_index = torch.empty((2, 0), dtype=torch.long)
    data[‘department’, ‘has’, ‘user’].edge_index = torch.empty((2, 0), dtype=torch.long)
  data[‘user’].num_nodes = len(user_id_map)
  data[‘criterion’].num_nodes = len(criterion_id_map)
  data[‘department’].num_nodes = len(department_id_map)
  return data
```

Основний клас для нейронної мережі MultiTaskTemporalAttentionHGT

```
class MultiTaskTemporalAttentionHGT(nn.Module):
  def __init__(self, metadata, hidden_channels, out_channels_clf, out_channels_reg,
        num_hgt_layers, num_heads, dropout, temporal_embed_dim, decay_base, num_years):
    super().__init__()
      self.hidden_channels, self.num_hgt_layers, self.temporal_embed_dim, self.num_years, self.decay_base = hidden_channels, num_hgt_layers, temporal_embed_dim, num_years, decay_base
    self.lin_dict = nn.ModuleDict()
    for node_type in metadata[0]:
      if node_type == ‘user’:
        input_dim = 3 + temporal_embed_dim
      elif node_type == ‘criterion’:
        input_dim = 1 + MAIN_EMBEDDING_LENGTH + df_criteria_info[‘category’].nunique()
      elif node_type == ‘department’:
```

```
            input_dim = df_departments['department_type'].nunique()
        else:
            raise ValueError(f"Unknown node type: {node_type}")
        self.lin_dict[node_type] = Linear(input_dim, hidden_channels)
    self.convs = nn.ModuleList()
    for _ in range(num_hgt_layers):
        self.convs.append(HGTConv(hidden_channels, hidden_channels, metadata, num_heads))
        self.temporal_attention = nn.MultiheadAttention(embed_dim=hidden_channels, num_heads=1,
batch_first=True)
    self.attention_weights_linear = nn.Linear(hidden_channels, 1)
    def _make_mlp(self, input_dim, hidden_dim, dropout):
        return nn.Sequential(nn.Linear(input_dim, hidden_dim), nn.ReLU(), nn.Dropout(dropout),
            nn.Linear(hidden_dim, hidden_dim // 2), nn.ReLU(), nn.Dropout(dropout),
            nn.Linear(hidden_dim // 2, 1))
    self.classifier = self._make_mlp(hidden_channels * 2, hidden_channels, dropout)
    self.regressor = self._make_mlp(hidden_channels * 2, hidden_channels, dropout)
    self.dropout = nn.Dropout(dropout)

  def forward(self, x_dict, edge_index_dict):
    x_dict = {node_type: self.lin_dict[node_type](x).relu_() for node_type, x in x_dict.items()}
    for conv in self.convs:
        x_dict = conv(x_dict, edge_index_dict)
    x_dict = {node_type: x.relu_() for node_type, x in x_dict.items()}
    return x_dict

  def aggregate_temporal(self, annual_user_embeddings, annual_criterion_embeddings):
    num_users = annual_user_embeddings[0]['user'].shape[0] if 0 in annual_user_embeddings else 0
                num_criteria = annual_criterion_embeddings[0]['criterion'].shape[0] if 0 in
annual_criterion_embeddings else 0
    user_embs_sequence = torch.zeros(num_users, self.num_years, self.hidden_channels)
    criterion_embs_sequence = torch.zeros(num_criteria, self.num_years, self.hidden_channels)
    for i in range(self.num_years):
        if i in annual_user_embeddings:
            user_embs_sequence[:, i, :] = annual_user_embeddings[i]['user']
        if i in annual_criterion_embeddings:
            criterion_embs_sequence[:, i, :] = annual_criterion_embeddings[i]['criterion']
    attn_output_user, attn_weights_user = self.temporal_attention(
        user_embs_sequence, user_embs_sequence, user_embs_sequence
    )
    beta_t_user_logits = self.attention_weights_linear(attn_output_user).squeeze(-1)
    beta_t_user = F.softmax(beta_t_user_logits, dim=-1)
    gamma_t_values = torch.zeros(self.num_years)
    if self.num_years > 1:
        delta = self.decay_base ** (1.0 / (self.num_years - 1))
    else:
        delta = 1.0
    for t_idx in range(self.num_years):
        gamma_t_values[t_idx] = delta ** (self.num_years - 1 - t_idx)
    gamma_t_user = gamma_t_values.unsqueeze(0).repeat(num_users, 1).to(beta_t_user.device)
    combined_weights_user = beta_t_user * gamma_t_user
            combined_weights_user = combined_weights_user / combined_weights_user.sum(dim=-1,
keepdim=True)
```

```
        final_user_emb = (combined_weights_user.unsqueeze(-1) * user_embs_sequence).sum(dim=1)
        attn_output_crit, attn_weights_crit = self.temporal_attention(
            criterion_embs_sequence, criterion_embs_sequence, criterion_embs_sequence
        )
        beta_t_crit_logits = self.attention_weights_linear(attn_output_crit).squeeze(-1)
        beta_t_crit = F.softmax(beta_t_crit_logits, dim=-1)
        gamma_t_crit = gamma_t_values.unsqueeze(0).repeat(num_criteria, 1).to(beta_t_crit.device)
        combined_weights_crit = beta_t_crit * gamma_t_crit
        combined_weights_crit = combined_weights_crit / combined_weights_crit.sum(dim=-1, keepdim=True)
        final_criterion_emb = (combined_weights_crit.unsqueeze(-1) * criterion_embs_sequence).sum(dim=1)
        return final_user_emb, final_criterion_emb

    def predict(self, user_emb, criterion_emb, edge_label_index):
        user_nodes = edge_label_index[0]
        criterion_nodes = edge_label_index[1]
        user_feats = user_emb[user_nodes]
        criterion_feats = criterion_emb[criterion_nodes]
        combined_features = torch.cat([user_feats, criterion_feats], dim=1)
        logits_clf = self.classifier(combined_features).squeeze(-1)
        probs_clf = torch.sigmoid(logits_clf)
        preds_reg = self.regressor(combined_features).squeeze(-1)
        return probs_clf, preds_reg, logits_clf

class MultiTaskTemporalAttentionHGT_DynamicInput(MultiTaskTemporalAttentionHGT):
    def __init__(self, metadata, hidden_channels, out_channels_clf, out_channels_reg,
                 num_hgt_layers, num_heads, dropout, temporal_embed_dim, decay_base, num_years,
                 criterion_input_dim, department_input_dim):
        super().__init__(metadata, hidden_channels, out_channels_clf, out_channels_reg,
                         num_hgt_layers, num_heads, dropout, temporal_embed_dim, decay_base, num_years)
        self.lin_dict = nn.ModuleDict()
        for node_type in metadata[0]:
            if node_type == 'user':
                input_dim = 3 + temporal_embed_dim
            elif node_type == 'criterion':
                input_dim = criterion_input_dim
            elif node_type == 'department':
                input_dim = department_input_dim
            else:
                raise ValueError(f"Unknown node type: {node_type}")
            self.lin_dict[node_type] = Linear(input_dim, hidden_channels)

# Тренування моделі

def train_model(model, optimizer, train_data_list, val_data_list, epochs, device):
    criterion_clf = nn.BCEWithLogitsLoss(pos_weight=torch.tensor([pos_weight_multiplier], device=device))
    criterion_reg = nn.MSELoss()
    best_val_loss = float('inf')
    for epoch in range(1, epochs + 1):
        model.train()
        total_loss = total_clf_loss = total_reg_loss = 0
```

```
annual_hgt_user_embeddings = annual_hgt_criterion_embeddings = {}
for year_relative_idx, data in enumerate(train_data_list):
    data = data.to(device)
    x_dict_hgt_output = model(data.x_dict, data.edge_index_dict)
    annual_hgt_user_embeddings[year_relative_idx] = {‘user’: x_dict_hgt_output[‘user’]}
    annual_hgt_criterion_embeddings[year_relative_idx] = {‘criterion’: x_dict_hgt_output[‘criterion’]}

final_user_emb, final_criterion_emb = model.aggregate_temporal(annual_hgt_user_embeddings, annual_hgt_criterion_embeddings)
optimizer.zero_grad()
all_logits_clf = all_true_labels_clf = all_preds_reg = all_true_scores_reg = []

for data in train_data_list:
    data = data.to(device)
    edge_label_index = data[‘user’, ‘reports’, ‘criterion’].edge_index
    true_labels_clf = data[‘user’, ‘reports’, ‘criterion’].edge_label
    true_scores_reg = data[‘user’, ‘reports’, ‘criterion’].edge_score
    probs_clf, preds_reg_batch, logits_clf_batch = model.predict(final_user_emb, final_criterion_emb, edge_label_index)

    all_logits_clf.append(logits_clf_batch)
    all_true_labels_clf.append(true_labels_clf)
    all_preds_reg.append(preds_reg_batch)
    all_true_scores_reg.append(true_scores_reg)
all_logits_clf = torch.cat(all_logits_clf)
all_true_labels_clf = torch.cat(all_true_labels_clf).float()
all_preds_reg = torch.cat(all_preds_reg)
all_true_scores_reg = torch.cat(all_true_scores_reg).float()
clf_loss = criterion_clf(all_logits_clf, all_true_labels_clf)

all_true_labels_clf = torch.cat([data['user', 'reports', 'criterion'].edge_label for data in train_data_list]).float().to(device)
all_true_scores_reg = torch.cat([data['user', 'reports', 'criterion'].edge_score for data in train_data_list]).float().to(device)
all_max_scores_reg = torch.cat([data['user', 'reports', 'criterion'].edge_max_score for data in train_data_list]).float().to(device)
clf_loss = criterion_clf(all_logits_clf, all_true_labels_clf) positive_mask_all = (all_true_labels_clf == 1)

positive_mask_all = (all_true_labels_clf == 1)
if positive_mask_all.sum() > 0:
    normalized_preds_reg = all_preds_reg[positive_mask_all] / (all_max_scores_reg[positive_mask_all] + 1e-6)
    normalized_true_scores_reg = all_true_scores_reg[positive_mask_all] / (all_max_scores_reg[positive_mask_all] + 1e-6)
    reg_loss = criterion_reg(normalized_preds_reg, normalized_true_scores_reg)
else:
    reg_loss = torch.tensor(0.0, device=device)

loss = clf_loss + reg_loss
loss.backward()
```

```
        optimizer.step()
        avg_loss, avg_clf_loss, avg_reg_loss = loss.item(), clf_loss.item(), reg_loss.item()
        model.eval()
        val_loss = 0
        with torch.no_grad():
            val_annual_hgt_user_embeddings = val_annual_hgt_criterion_embeddings = {}
            for year_relative_idx, data in enumerate(val_data_list):
                data = data.to(device)
                x_dict_hgt_output = model(data.x_dict, data.edge_index_dict)
                val_annual_hgt_user_embeddings[year_relative_idx] = {'user': x_dict_hgt_output['user']}
                val_annual_hgt_criterion_embeddings[year_relative_idx] = {'criterion': x_dict_hgt_output['criterion']}
                val_final_user_emb, val_final_criterion_emb = model.aggregate_temporal(val_annual_hgt_user_embeddings, val_annual_hgt_criterion_embeddings)

            all_val_logits_clf = all_val_true_labels_clf = all_val_preds_reg = all_val_true_scores_reg = []
            for data in val_data_list:
                data = data.to(device)
                edge_label_index = data['user', 'reports', 'criterion'].edge_index
                true_labels_clf = data['user', 'reports', 'criterion'].edge_label
                true_scores_reg = data['user', 'reports', 'criterion'].edge_score
                _, preds_reg_batch, logits_clf_batch = model.predict(val_final_user_emb, val_final_criterion_emb, edge_label_index)

                all_val_logits_clf.append(logits_clf_batch)
                all_val_true_labels_clf.append(true_labels_clf)
                all_val_preds_reg.append(preds_reg_batch)
                all_val_true_scores_reg.append(true_scores_reg)
            all_val_logits_clf = torch.cat(all_val_logits_clf)
            all_val_true_labels_clf = torch.cat(all_val_true_labels_clf).float()
            all_val_preds_reg = torch.cat(all_val_preds_reg)
            all_val_true_scores_reg = torch.cat(all_val_true_scores_reg).float()
            clf_loss_val = criterion_clf(all_val_logits_clf, all_val_true_labels_clf)
            positive_mask_val = (all_val_true_labels_clf == 1)
            if positive_mask_val.sum() > 0:
                reg_loss_val = criterion_reg(all_val_preds_reg[positive_mask_val], all_val_true_scores_reg[positive_mask_val])
            else:
                reg_loss_val = torch.tensor(0.0, device=device)

            val_loss = (clf_loss_val + reg_loss_val).item()
                print(f'Epoch: {epoch:03d}, Train Loss: {avg_loss:.4f} (Clf: {avg_clf_loss:.4f}, Reg: {avg_reg_loss:.4f}), Val Loss: {val_loss:.4f}')
            if val_loss < best_val_loss:
                best_val_loss = val_loss
                torch.save(model.state_dict(), 'best_model.pt')
                print(f"  New best model saved with Val Loss: {best_val_loss:.4f}")
    return model, final_user_emb, final_criterion_emb

def get_similar_users_with_better_performance(current_user_id, df_simi, df_perf, predict_year, user_id_map, criterion_id_map):
    relevant_similar_users_df = df_simi[(df_simi['user_id'] == current_user_id)]
```

```
    similar_user_ids = relevant_similar_users_df[‘similar_user_id’].tolist()
    recommended_criteria_from_similar = set()
    for s_user_id in similar_user_ids:
            s_user_perf_current_year = df_perf[(df_perf[‘user_id’] == s_user_id) & (df_perf[‘year’] ==
MAIN_YEAR_TO)]
                        active_criteria_for_predict_year = df_availability[df_availability[‘year’] ==
PREDICT_YEAR][‘criterion_id’].unique()
        for _, row in s_user_perf_current_year.iterrows():
            if row[‘criterion_id’] in active_criteria_for_predict_year:
                recommended_criteria_from_similar.add(row[‘criterion_id’])
    return list(recommended_criteria_from_similar)

def recommend_top_criteria(model, user_id_map, criterion_id_map, df_crit, df_perf_history,
                        predict_year_data, final_user_embeddings, final_criterion_embeddings,
                        df_simi, user_original_ids, criterion_original_ids, df_availability):
    model.eval()
    all_recommendations = {}
                        active_criteria_predict_year_ids = df_availability[df_availability[‘year’] ==
PREDICT_YEAR][‘criterion_id’].tolist()
    active_criterion_indices = [criterion_id_map[c_id] for c_id in active_criteria_predict_year_ids if c_id
in criterion_id_map]
    with torch.no_grad():
        for user_original_id in user_original_ids:
            u_idx = user_id_map[user_original_id]
            user_recs = []
            pred_edge_label_index = torch.stack([
                torch.tensor([u_idx] * len(active_criterion_indices), dtype=torch.long),
                torch.tensor(active_criterion_indices, dtype=torch.long)
            ], dim=0).to(final_user_embeddings.device)
                probs_clf, preds_reg, _ = model.predict(final_user_embeddings, final_criterion_embeddings,
pred_edge_label_index)
            for i, c_idx in enumerate(active_criterion_indices):
                criterion_original_id = criterion_original_ids[c_idx]
                prob = probs_clf[i].item()
                predicted_score_normalized = preds_reg[i].item()
                                                                max_points = df_crit[df_crit['criterion_id']
==                                  criterion_original_id]['max_points'].iloc[0predicted_score_real =
predicted_score_normalized * max_points
                predicted_score = preds_reg[i].item()
                max_points = df_crit[df_crit[‘criterion_id’] == criterion_original_id][‘max_points’].iloc[0]
                reg_points_normalized = predicted_score / max_points if max_points > 0 else 0
                            user_executed_criteria_hist = df_perf_history[(df_perf_history[‘user_id’] ==
user_original_id)][‘criterion_id’].unique()
                max_sim = 0
                if len(user_executed_criteria_hist) > 0:
                    for hist_crit_id in user_executed_criteria_hist:
                        hist_crit_row = df_crit[df_crit[‘criterion_id’] == hist_crit_id]
                        current_crit_row = df_crit[df_crit[‘criterion_id’] == criterion_original_id]
                        if not hist_crit_row.empty and not current_crit_row.empty:
                            hist_crit_sbert = hist_crit_row[‘SBERT_embedding’].iloc[0]
                            current_crit_sbert = current_crit_row[‘SBERT_embedding’].iloc[0]
                            sim = get_semantic_similarity(hist_crit_sbert, current_crit_sbert)
```

```
                    if sim > max_sim:
                        max_sim = sim
            heuristic_boost_score = 0
        similar_better_criteria = get_similar_users_with_better_performance(user_original_id, df_simi, df_perf_history, PREDICT_YEAR, user_id_map, criterion_id_map)
            if criterion_original_id in similar_better_criteria:
                heuristic_boost_score = 1.0
            combined_score = (prob +  reg_points_normalized)/2
                user_recs.append({‘criterion_id’: criterion_original_id, ‘prob’: prob, ‘predicted_score’: predicted_score_real, ‘max_points’: max_points, ‘reg_points_normalized’: reg_points_normalized, ‘semantic_similarity’: max_sim, ‘hb’: heuristic_boost_score, ‘combined_score’: combined_score})
        user_recs_sorted = sorted(user_recs, key=lambda x: x[‘combined_score’], reverse=True)
        all_recommendations[user_original_id] = user_recs_sorted[:TOP_N_RECOMMENDATIONS]
    return all_recommendations

def load_all_past_recommendations(main_year_from, main_year_to, base_file_path, df_criteria_info):
    all_recs = []
    criterion_desc_to_id_map = df_criteria_info.set_index(‘description’)[‘criterion_id’].to_dict()
    max_points_map = df_crit.set_index('criterion_id')['max_points'].to_dict()

    for year in range(main_year_from, main_year_to + 1):
        file_path = f”{base_file_path}_{year+1}.csv”
        try:
          df = pd.read_csv(file_path, names=[“user_id”, “rec_year_value”, “criterion_description”, “rank”, “label”, “score_list_str”], header=0)

            df[‘user_id’] = df[‘user_id’].astype(int)
            # convert desc to numberId
            df[‘criterion_id’] = df[‘criterion_description’].map(criterion_desc_to_id_map)
            if df[‘criterion_id’].isnull().any():
                print(f”Warning: Some criterion descriptions in DB were not found in df_criteria_info and will be mapped to -1.”)
                df[‘criterion_id’] = df[‘criterion_id’].fillna(-1).astype(int)
            else:
                df[‘criterion_id’] = df[‘criterion_id’].astype(int)

            df[‘score_list’] = df[‘score_list_str’].apply(literal_eval)
            df[‘rec_year’] = year
            df_final = df[[‘user_id’, ‘rec_year’, ‘criterion_id’, ‘rank’, ‘label’]].copy()
            df_final.rename(columns={‘label’: ‘score’}, inplace=True)
            all_recs.append(df_final)
        except e:
            print(f”Error: {e}”)
            raise e

    expected_cols = [“user_id”, “rec_year”, “criterion_id”, “rank”, “score”]
    final_df = pd.concat(all_recs) if all_recs else pd.DataFrame(columns=expected_cols)
    for col in [“user_id”, “criterion_id”, “rec_year”]:
        if col in final_df.columns:
            final_df[col] = final_df[col].astype(int)
    for col in [“rank”, “score”]:
        if col in final_df.columns:
```

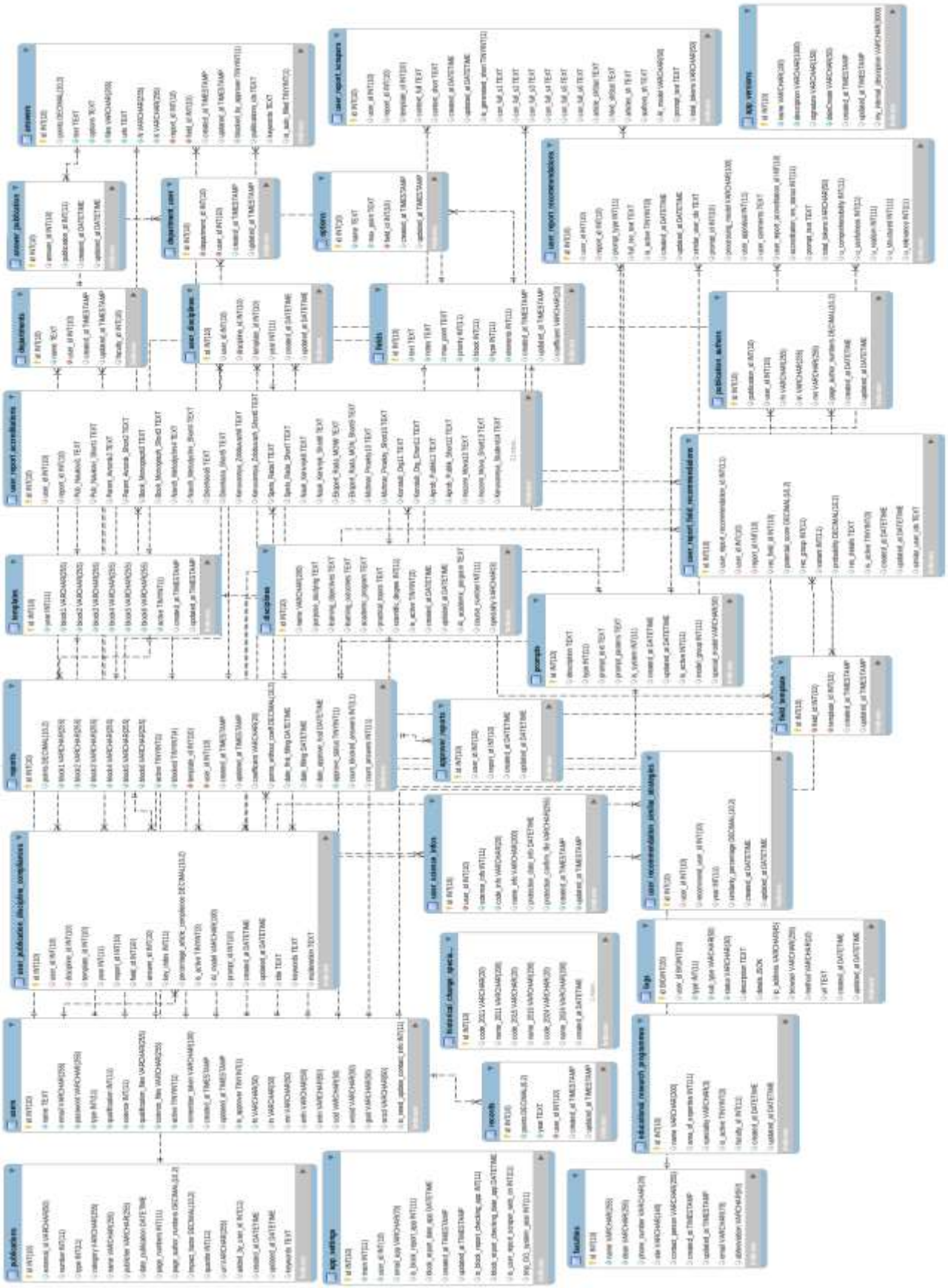
```
        final_df[col] = pd.to_numeric(final_df[col], errors='coerce')
        final_df[col] = final_df[col].fillna(0)
    return final_df

def identify_unfulfilled_recommendations(df_recommends_history, df_performance, years_to_check):
    unfulfilled_recs = []
    for rec_year in years_to_check:
        recs_made_in_rec_year = df_recommends_history[df_recommends_history['rec_year'] == rec_year]
        target_year = rec_year + 1
        perf_in_target_year = df_performance[df_performance['year'] == target_year]
        merged_df = pd.merge(recs_made_in_rec_year, perf_in_target_year, on=['user_id', 'criterion_id'], how='left', indicator=True)
        unfulfilled = merged_df[merged_df['_merge'] == 'left_only']
        if not unfulfilled.empty:
            unfulfilled_recs.append(unfulfilled[['user_id', 'criterion_id']].assign(year=target_year))
    return pd.concat(unfulfilled_recs) if unfulfilled_recs else pd.DataFrame(columns=['user_id', 'criterion_id', 'year'])

# Побудова моделі і визначення персоналізованих рекомендацій

def buildModelWithRecommendations():
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    df_perf, df_crit, df_simi, df_dep, df_user_department_raw_local, df_avail_local = load_data()
    df_recommends_history = load_all_past_recommendations(MAIN_YEAR_FROM, MAIN_YEAR_TO, MAIN_RECOMMENDS_CRITERIONS_TO_FETURE_YEAR_FILE_BASE, df_criteria_info=df_crit)
    all_user_ids = df_perf['user_id'].unique()
    all_criterion_ids = df_crit['criterion_id'].unique()
    all_department_ids = df_dep['department_id'].unique()
    user_id_map = {id: i for i, id in enumerate(all_user_ids)}
    criterion_id_map = {id: i for i, id in enumerate(all_criterion_ids)}
    department_id_map = {id: i for i, id in enumerate(all_department_ids)}
    user_original_ids = {i: id for id, i in user_id_map.items()}
    criterion_original_ids = {i: id for id, i in criterion_id_map.items()}
    train_data_list = val_data_list = []
    train_years = list(range(MAIN_YEAR_FROM, MAIN_YEAR_TO))
    val_years = [MAIN_YEAR_TO]

    all_performed_criteria_by_user_yearly = df_perf.groupby(['year', 'user_id'])['criterion_id'].apply(set).to_dict()
    all_active_users_in_year = df_perf.groupby('year')['user_id'].unique().apply(list).to_dict()
    years_to_check_for_unfulfilled = list(range(MAIN_YEAR_FROM, MAIN_YEAR_TO + 1))
    df_unfulfilled_recs_all_years = identify_unfulfilled_recommendations(df_recommends_history, df_perf, years_to_check_for_unfulfilled)

    print("creating training graphs...")
    for year in train_years:
        print(f"  Year: {year}")
        unfulfilled_for_this_year = df_unfulfilled_recs_all_years[df_unfulfilled_recs_all_years['year'] == year]
        data = create_hetero_graph_for_year(df_perf, df_crit, df_simi, df_dep, df_user_department_raw_local, df_avail_local, year, user_id_map, criterion_id_map,
```

```
department_id_map, all_performed_criteria_by_user_yearly, all_active_users_in_year.get(year, []),
df_unfulfilled_recommendations_current_year=unfulfilled_for_this_year)
        train_data_list.append(data)

    if train_data_list:
        metadata = train_data_list[0].metadata()
    else:
        raise ValueError("No training data to retrieve graph metadata.")

    print("creating validation graphs...")
    for year in val_years:
        print(f"  Year: {year}")
        unfulfilled_for_this_year = df_unfulfilled_recs_all_years[df_unfulfilled_recs_all_years['year'] ==
year]
        data = create_hetero_graph_for_year(df_perf, df_crit, df_simi, df_dep,
df_user_department_raw_local, df_avail_local, year, user_id_map, criterion_id_map,
department_id_map, all_performed_criteria_by_user_yearly, all_active_users_in_year.get(year, []),
df_unfulfilled_recommendations_current_year=unfulfilled_for_this_year)
        val_data_list.append(data)

    criterion_input_dim = 1 + MAIN_EMBEDDING_LENGTH + df_crit['category'].nunique()
    department_input_dim = df_dep['department_type'].nunique()
    num_years_for_model_training = len(train_years) + len(val_years)

    # Init model
    model = MultiTaskTemporalAttentionHGT_DynamicInput(metadata,
hidden_channels=embedding_dim, out_channels_clf=1, out_channels_reg=1,
num_hgt_layers=num_hgt_layers, num_heads=num_heads, dropout=dropout_rate,
temporal_embed_dim=temporal_embed_dim, decay_base=decay_base,
num_years=num_years_for_model_training, criterion_input_dim=criterion_input_dim,
department_input_dim=department_input_dim).to(device)
    optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)
    model_path = 'best_model.pt'
    final_user_emb_pred = final_criterion_emb_pred = None

    if os.path.exists(model_path):
        print(f"Loading pre-trained model from {model_path}...")
        model.load_state_dict(torch.load(model_path))
        model.eval()
        # calculate embeddings for prediction after loading the model
        predict_annual_hgt_user_embeddings_reloaded = {}
        predict_annual_hgt_criterion_embeddings_reloaded = {}

        all_years_for_embedding = list(range(MAIN_YEAR_FROM, MAIN_YEAR_TO + 1))
        with torch.no_grad():
            for year_idx, year in enumerate(all_years_for_embedding):
                data_for_emb = create_hetero_graph_for_year(
                    df_perf, df_crit, df_simi, df_dep, df_user_department_raw_local, df_avail_local, year,
                    user_id_map, criterion_id_map, department_id_map,
                    all_performed_criteria_by_user_yearly, all_active_users_in_year.get(year, [])
                ).to(device)
```

```
                hgt_x_dict = {node_type: model.lin_dict[node_type](x).relu_() for node_type, x in data_for_emb.x_dict.items()}
            for conv in model.convs:
                hgt_x_dict = conv(hgt_x_dict, data_for_emb.edge_index_dict)
            hgt_x_dict = {node_type: x.relu_() for node_type, x in hgt_x_dict.items()}
            predict_annual_hgt_user_embeddings_reloaded[year_idx] = {'user': hgt_x_dict['user']}
                        predict_annual_hgt_criterion_embeddings_reloaded[year_idx] = {'criterion': hgt_x_dict['criterion']}

        final_user_emb_pred, final_criterion_emb_pred = model.aggregate_temporal(
                                                    predict_annual_hgt_user_embeddings_reloaded, predict_annual_hgt_criterion_embeddings_reloaded
        )
        print("Model loaded. Skipping training.")

    else:
        print("No pre-trained model found. Starting model training...")
            model, final_user_emb_pred, final_criterion_emb_pred = train_model(model, optimizer, train_data_list, val_data_list, epochs, device)
        print("training completed.")
        model.load_state_dict(torch.load(model_path))
        model.eval()

            return model, user_id_map, criterion_id_map, department_id_map, user_original_ids, criterion_original_ids, \
            df_perf, df_crit, df_simi, df_dep, df_user_department_raw_local, df_avail_local, \
            all_performed_criteria_by_user_yearly, all_active_users_in_year, metadata, \
            criterion_input_dim, department_input_dim, final_user_emb_pred, final_criterion_emb_pred
```

ДОДАТОК В

ER-ДІАГРАМА БАЗИ ДАНИХ



ДОДАТОК Г
СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Пукас А. В., Сімак А. Ю. (2025). Метод формування прогнозованих показників активності науково-педагогічних працівників для інтелектуалізованої системи аналізу їх діяльності та рейтингування. Наукові праці Вінницького національного технічного університету, (2). Вилучено із https://praci.vntu.edu.ua/index.php/praci/article/view/813.

2. Сімак А., Юшко, А. (2025). Інтелектуальний модуль збору відкритих даних в системі рейтингування діяльності науково-педагогічного персоналу. Measuring and computing devices in technological processes, (1), 353–361. https://doi.org/10.31891/2219-9365-2025-81-45.

3. Юшко А. В., Сімак, А. Ю. (2024). Інформаційна інтелектуальна система аналізу наукової та науково-педагогічної діяльності академічного колективу. Оптико-електронні інформаційно-енергетичні технології, 47(1), 7–16. https://doi.org/10.31649/1681-7893-2024-47-1-7-16.

4. Pukas A., Simak A., Yushko A., Nadvynychnyy S., Shandruk S. and Vatslavskyi O., “Automated Reporting Module in the Academic Staff Performance Appraisal System,” *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)*, Ceske Budejovice, Czech Republic, 2024, pp. 824-827, doi: 10.1109/ACIT62333.2024.10712573.

5. Pukas Andriy, Simak Andriy, Goncharuk-Cholach Tetiana, Konoplitska Olena and Zabchuk Vadym “Intelligent Analyzing Module in the Academic Staff Performance Appraisal System”, *CEUR Workshop Proceedings*, 2023, 3624, pp. 423–429.

6. Pukas A., Simak A., Shandruk S., Bilovus L., Stepanenko V. and Demianiuk A., “Features of Implementation the Academic Staff Performance Appraisal System,” *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*, Ruzomberok, Slovakia, 2022, pp. 568-571, doi: 10.1109/ACIT54803.2022.9913180.

7. Мельниченко В.В., Сімак А.Ю., Юшко А.В. (2024). Інтелектуалізована система для підтримки реалізації принципів адаптивності навчання. Матеріали зимової школи-семінару молодих вчених і студентів, “Комп’ютерні інформаційні технології” СІТ’2024 Тернопіль ЗУНУ, с. 85-87.

8. Пукас А.В., Сімак А.Ю., Юшко А. В., Мельниченко В.В. (2024). Модуль збору даних для автоматичного заповнення звіту в інтелектуалізованій системі аналізу діяльності та рейтингування викладачів. Матеріали зимової школи-семінару молодих вчених і студентів, “Комп’ютерні інформаційні технології” СІТ’2024 Тернопіль ЗУНУ, с. 69-71.

9. Гуска Б. Д. Інтелектуалізована система обробки слабкоформалізованої інформації [Текст] / Гуска Б. Д., Сімак А. Ю. // Комп’ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів СІТ’2023, Тернопіль, 30 листопада 2023 року / відп. за вип. Пукас А. В. – Тернопіль: ЗУНУ, 2023. – С. 91.

10. Pukas A., Simak A., Syrnyk O., Horal L., Shyjko V. and Papa O., “Software Module for Data Correctness and Completeness Control in the Academic Staff Performance Appraisal System,” in Proc. 2019 9th Int. Conf. Adv. Comput. Inf. Technol. (ACIT), Czech Republic, 2019, pp. 277–280, doi: 10.1109/ACITT.2019.8779999.

11. Пукас, А. В. Інформаційна система рейтингування діяльності науково-педагогічних працівників закладів вищої освіти [Текст] / Пукас А. В., Сімак А. Ю., Вітенко В. В.// Комп’ютерні інформаційні технології: матеріали школи-семінару молодих вчених і студентів СІТ’2019 [м. Тернопіль, 29 листопада 2019 р.] / відп. за вип. М. П. Дивак. - Тернопіль: ТНЕУ, 2019. - С. 32-33.

12. Пукас, А. В. Модуль контролю повноти та коректності даних інтелектуалізовананої системи аналізу діяльності та рейтингування викладачів [Текст] / А. В. Пукас, А. Ю. Сімак, О. Й. Сирник // Комп’ютерні науки та інформаційні технології: матеріали семінару CSIT’2018 [м. Тернопіль, 2 червня 2018 р.] / відп. за вип. М. П. Дивак. - Тернопіль: ТНЕУ, 2018. - С. 61-62.

ДОДАТОК Д
АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДІСЛІДЖЕННЯ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАУКОВО-ДОСЛІДНА ЧАСТИНА

46009, Україна, м. Тернопіль, вул. Львівська, буд. 5А, тел. (0352) 51-75-52

№ 21/ 82 - 2025 «20» 08 2025 року

ДОВІДКА
про участь у виконанні науково-дослідної роботи

Видана **СІМАКУ Андрію Юрійовичу** про те, що у 2024 році на посаді молодшого наукового співробітника він брав участь у підготовці молодіжного наукового дослідження, яке виконує Західноукраїнський національний університет за кошти державного бюджету України на тему «Математичне та програмне забезпечення прототипу біогазової установки з підвищеною ефективністю функціонування» (державний реєстраційний номер 0124U000076).

Наказ ЗУНУ від 23 жовтня 2024 року №383.

Заступник начальника
науково-дослідної частини Юрій МОСКАЛЬ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ НАУКОВО-ДОСЛІДНА ЧАСТИНА 33680120

ЗАТВЕРДЖЕНО
Проректор з науково-педагогічної
роботи Західноукраїнського
національного університету
Віктор ОСТРОВЕРХОВ
_____ 2025 р.

АКТ
про впровадження результатів дисертаційної роботи
СІМАКА Андрія Юрійовича
"Інтелектуалізована програмна система аналізу та рейтингування діяльності науково-педагогічних працівників"

Результати дисертаційної роботи аспіранта кафедри комп'ютерних наук Західноукраїнського національного університету СІМАКА Андрія Юрійовича використано для підвищення результативності діяльності науково-педагогічних працівників (НПП), зокрема:

– модель представлення знань про діяльність НПП на основі знання-орієнтованого підходу з використанням графової моделі, що дало змогу реалізувати процедури генерації персоналізованих рекомендацій для підвищення результативності наукової-педагогічної діяльності;

– комбінований метод ранжування НПП на основі комплексу метрик подібності, який враховує їх динаміку активності, семантичну подібність та кількісні характеристики результатів діяльності, що уможливило знаходження користувачів зі схожими, але кращими показниками діяльності;

– інтелектуалізований метод формування персоналізованих рекомендацій діяльності НПП, який ґрунтується на прогнозуванні показників діяльності, моделі представлення знань, використанні великих мовних моделей для генерації текстових персоналізованих рекомендацій з динамічним уточненням структури промптів, що забезпечує скорочення часу їх формування;

– програмне середовище інтелектуалізованої системи рейтингування діяльності НПП, яке побудоване на базі системи рейтингування ЗУНУ, що дозволяє інтегруватись з великими мовними моделями, формувати персоналізовані рекомендації із вбудованими механізмами зворотного зв'язку, що забезпечило підвищення ефективності формування персоналізованих рекомендацій та перевірку відповідності НПП акредитаційним вимогам.

Декан факультету комп'ютерних інформаційних технологій, к.т.н., доцент — Ігор ЯКИМЕНКО

Доцент кафедри комп'ютерних наук, к.т.н., доцент — Ірина СТАСІВ

WorldERP , LLC

Україна, 46002, Тернопільська обл., м. Тернопіль, вул. Збаразька 7, поверх 4

12 серпня 2025 р.

АКТ
про впровадження результатів дисертаційної роботи

Сімака Андрія Юрійовича
“ Інтелектуалізована програмна система аналізу та рейтингування діяльності науково-педагогічних працівників”

Даний акт складений про те, що при реалізації багатьох програмних проєктів у українському представництві компанії «WorldERP, LLC» використано результати дисертаційної роботи аспіранта кафедри комп’ютерних наук Західноукраїнського національного університету Сімака Андрія Юрійовича.

Автором запропоновано:

– підхід до проектування програмних систем, що передбачає інтеграцію інструментів штучного інтелекту безпосередньо в бізнес-логіку системи;

– мікросервісну архітектуру програмного забезпечення для збору даних та автоматизації частини бізнес-процесів користувачів, яка взаємодіє з інструментами штучного інтелекту, що дозволяє розробляти програмні системи та середовища для управління складними бізнес-процесами.

Компанією були впроваджені елементи запропонованої архітектури у процес розробки програмних продуктів і обробки даних, що дозволило підвищити ефективність розробки та супроводу програмного забезпечення.

Директор українського представництва
компанії «WorldERP, LLC» Андрій СТЕПАНЕНКО

ЗАТВЕРДЖЕНО
Проректор з науково-педагогічної роботи Західноукраїнського національного університету
к.е.н., доцент Віктор ОСТРОВЕРХОВ

"1" вересня 2025 р.

АКТ
про впровадження в освітній процес Західноукраїнського національного університету результатів дисертаційної роботи
СІМАКА Андрія Юрійовича
"Інтелектуалізована програмна система аналізу та рейтингування діяльності науково-педагогічних працівників"

Даний акт складений про те, що результати дисертаційної роботи СІМАКА Андрія Юрійовича на тему: «Інтелектуалізована програмна система аналізу та рейтингування діяльності науково-педагогічних працівників» використані в освітньому процесі факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету для студентів спеціальностей «Інженерія програмного забезпечення» та «Інформаційні системи і технології».

При викладанні дисциплін «Основи інформаційних технологій», «Інтелектуальні інформаційні системи», «Архітектура та проєктування програмного забезпечення» розглядаються метод ранжування науково-педагогічних працівників на основі комплексу метрик подібності, який враховує динаміку їх активності, семантичну подібність та кількісні характеристики результатів діяльності, що уможливило знаходження користувачів зі схожими, але кращими показниками діяльності, та метод формування персоналізованих рекомендацій з використанням графової нейронної мережі та великих мовних моделей, а також вдосконалена архітектура інтелектуалізованої програмної системи.

Декан факультету комп'ютерних інформаційних технологій, к.т.н., доцент — Ігор ЯКИМЕНКО

Доцент кафедри комп'ютерних наук, к.т.н., доцент — Ірина СПІВАК

ДОДАТОК Е
МЕТОДИ ПОКРАЩЕННЯ РЕЗУЛЬТАТИВНОСТІ ДІЯЛЬНОСТІ НПП ТА ПОРІВНЯННЯ ПІДХОДІВ У РЕКОМЕНДАЦІЙНИХ СИСТЕМАХ

Таблиця Е.1

Методи покращення результативності діяльності НПП

Метод	Опис	Переваги та вплив на результативність НПП	Виклики та ресурсні вимоги
Профе-сійний розвиток	ІКТ, міжнародні проєкти, стажування	Нові навички, розширення світогляду, мотивація, пок-ращення окремих компе-тенцій і якості діяльності	Фінансові, часові інвестиції, адаптація НПП, тренери, доступ до ресурсів, подолання особистих бар’єрів
Оптимі-зація робочого наванта-ження	Управління часом, гнучкі моделі навантаження	Зменшення стресу, фокус на ефективність, справедливий розподіл обов’язків, підвищення індивідуальної продуктивності	Зміна підходів, перегляд політики навантаження. Підтримка керівництва, реалізація індивідуальних консультацій
Мотива-ція та стимулю-вання	Матеріальні та не матеріальні фактори (кар’єра, управлінські посади)	Задоволеність, залученість, лояльність НПП, стимулювання до якісної роботи, під-вищення активності роботи та продуктивності команди	Фінансові інвестиції, ігнорування не матеріальних факторів, необхідінсть у прозорих системах, корпоративна культура

Продовження таблиці Е.1

Рейтин-гові системи	Бальні системи оцінки діяльності, KPI, групи активності	Прозорість/диферен-ціація оцінювання, стимул розвиватись, підвищення активності за показниками	Кількісні метрики, ризик зниження якості аспектів які не враховуються, збір/обробка даних, розробка критеріїв, контроль достовірності
Рекомен-даційні системи	Різнопланові рекомендацій ні системи для НПП	Персоналізовані реко-мендації, оптимізація ресурсів, корисні дані, співпраця, підтримка прийняття рішень	Обмежена кількість систем для НПП, конфіденцій-ність даних, можлива упередженість, складність впровадження, розробка/ придбання систем
Інстру-менти ШІ	Автоматизація завдань, персоналізація навчання, доступ до інструментів	Економія часу НПП, ефективність навчання, доступ до ресурсів, зворотний зв'язок, підви-щення ефективності рутинних завдань	Конфіденційність даних, упередженість, високі фінансові інвестції, розробка етичних політик, вивчення інструментів

Таблиця Е.2

Підсумкова таблиця порівняння підходів у рекомендаційних системах

Підхід	Контекст діяль-ності	Дина-міка	Пер-сона-лізація	Інтер-прита-ція ре-зультату	Склад-ність реалізації	Обсяг даних	Гнуч-кість до змін даних

Продовження таблиці Е.2

Метрики подібності	Не враховує	Залежить від реалізації	Частково	Проста	Низька	Малий	Середня (не в реальному часі)
Семантичні (USE, SBERT, Content/ Knowledge-Based)	Враховує (через атрибути /описи критеріїв)	Зазвичай не враховують	Висока	Зрозуміла, за схожістю атрибутів	Середня (необхід-ні моделі)	Малий	Висока (при наявності хоча б якоїсь активності)
Item-Item Collaborative Filtering (CF, UBCF, IBCF)	Частково (історія дій, або залежностей)	Не враховує	Обмежена (“холодний старт”)	Проста	Низька	Середній	Низька
MF (SVD++, ALS, з BPR)	Абстрактна	Обмежена	Висока	Слабка	Середня (навчання моделі)	Середній	Середня (перенавчання моделі)
Гібридні (наприклад: CF & CB, та інші)	Враховує	Залежить від реалізації	Висока	Залежить від реалізації	Висока (поєднання кількох моделей)	Середній	Обмежена

Продовження таблиці Е.2

Секвенцій-ноорієнтова-ний (Sequence-aware, Rule-based, RNN/ LSTM/ Transfor-mers)	Частково (через вхідні послі-довності)	Врахо-вує (послі-довнос-ті)	Висока	Складна	Висока (вимагає наявність послідов-ностей, без розрід-женості)	Вели-кий	Середня (пере-навчан-ня моделі)
Машине навчання ML/DL	Враховує	Врахо-вує	Висока	Середня (“чор-ний ящик”)	Висока (навчання моделі, обчис-лювальні ресурси)	Вели-кий	Висока (навчан-ня, пере-навчан-ня)
Графовий (GNN, Node2Vec, GraphSAGE, GAT)	Враховує	Може врахо-вувати	Висока	Середня (без прямої семан-тики, зв’язки)	Висока (потрібні певні біб-ліотеки, навчання моделі)	Може бути серед-нім	Висока (пере-навчан-ня; нові вузлів, зв’язки)

ДОДАТОК Ж
РЕКОМЕНДОВАНІ КРИТЕРІЇ, ЯКІ ФОРМУЄ LLM ТА РЕЗУЛЬТАТ ЇЇ РОБОТИ

Таблиця Ж.1

Рекомендації на основі історичних даних із системи рейтингування

ID	Гру-па/ Ва-ріант	По-тен-ційний бал	Тип	Опис від LLM	Пояснення (відносно фактичної діяльності)
317	1/1	75	Нове	Опублікувати статтю у нау-ковому виданні Scopus/Web of Science 4 квартилю	Нова активність, користувач публікує статті, але інших типів або видів
316	1/2	125	Нове	Опублікувати статтю у нау-ковому виданні Scopus/Web of Science 3 квартилю	Нова активність, користувач публікує статті, але інших типів або видів
265	1/3	20	Від-нов-лення	Одноосібні статті у віт-чизняних фахових наукових виданнях категорії “Б”	Користувач публікував статті під данний критерій кілька років назад
305	2/1	50	Нове	Видати монографію	Враховуючи публікаційну активність та діяльність подібних користувачів, можна рекомендувати

Продовження таблиці Ж.1

265	2/2	15	Від-нов-лення	Одноосібні статті у віт-чизняних фахових наукових виданнях категорії “Б”	Користувач публікував статті під данний критерій кілька років назад
180	2/3	20	Від-нов-лення	Статті у співавторстві у міжнародних наукових виданнях, які входять до інших наукометричних баз та статті у вітчизняних виданнях категорії “Б”	Користувач публікував статті під данний критерій кілька років назад
320	3/1	50	Нове	Видати підручник (у співавторстві чи одноосібно)	Наявна достатня кількість публікацій, рекомендовано видати одноосібну роботу
332	4/1	20	Від-нов-лення	Подача заявки на міжнародні гранти (подача заявок до міжнародних фондів)	Минулого року було подано заявку на міжнародний грант “MineCloud”
353	5/1	20	Пок-ра-щення	Досягнути максимального балу в участі у програм-ному комітеті конференцій	Є можливітьс покращити участь у прог. комітеті, та набрати максимальний бал
345	6/1	10	Новий	Стати членом редакційної колегії наукового видання категорії “Б”	Раніше виконувались обов’язки члена ред. колегії інших типів журналу

Продовження таблиці Ж.1

354	7/1	50	Від-нов-лення	Пройти міжнародне стажу-вання/підвищення кваліфі-кації тривалістю до 3 місяців	Три роки назад виконувалось стажування термінов до 3міс. Рекомен-дуємо поновити діяльність
358	8/1	5	Пок-ра-щення	Збільшення участі у міжнародних та національних конференціях	Раніше данний критерій виконувався в повному обся-зі, проте попередні роки ні
294	9/2	5	Від-нов-лення	Організація або проведення секції на наукових заходах на базі університету	Приймав участь та проведені симпозіумів, ркуглих столів, позаминулого року

Таблиця Ж.2

Результат визначення відповідності публікацій до дисципліни з допомогою LLM

Назва/ Ключові слова	Предмет	Від-повід-ність	Пояснення
“SEO-Optimization of Product Content on a Marketplace Platform”. Text analysis, Pandemics, Software algorithms, Search engines, Software, Filling, Internet, SEO, record similarity, keywords, e-	Інженерія кросплат-форме-ного ПЗ	90%	*“Назва та ключові слова статті безпосередньо згадують SEO-оптимізацію, пошукові системи та ключові слова, які є основними темами навчальної програми в розділі ‘Інструменти Інтернет-маркетингу: seo-оптимізація’“*
	Управлін-ня соціаль-ними ризиками	15%	*“Тематика публікації зосереджена на SEO-оптимізації, програмному забезпеченні та електронній комерції, що має низький зв’язок”*
	Публічне управлінн я в сфері націона-	15%	*“Тема публікації та ключові слова не мають прямого відношення до дисципліни ‘Публічне управління в сфері національної безпеки’, оскільки сфокусовані на технічних аспектах SEO та електронної комерції”*

Продовження таблиці Ж.2

ommerce, Levenstein distance	льної безпеки		
	Системний аналіз	55%	*“Ключові слова, такі як ‘Software algorithms’, Text analysis’, ‘keywords’, які є дотичними до тем ‘моделювання комп’ютеризованих інформаційних систем’ та ‘системний аналіз процесів одержання інформації’. Проте основна тематика SEO-оптимізації не є центральною”*
“Systemic Risks in the Global Financial System and their Influence on Economic Stability”. Systemic risks, globalization, globalfinancial system, economic stability, macroeconomicimbalances, risk monitoring, financial stresses, financial sector	Інженерія кросплатформеного ПЗ	20%	*“Публікація не згадуються у навчальній програмі дисципліни і не мають чіткого зв’язку з інженерією кросплатформного програмного забезпечення, IT-бізнесом чи інтернет-маркетингом”*
	Управління соціальними ризиками	85%	*“Назва та ключові слова, такі як ‘Systemic risks’, ‘global financial system’ і ‘economic stability’, безпосередньо пов’язані з ‘ризиками економічного характеру’, що відповідає змісту дисципліни”*
	Публічне управління в сфері національної безпеки	90%	*“Навчальна програма дисципліни безпосередньо розглядає ‘економічну безпеку’ та ‘загрози національній безпеці в економічній сфері’. Ключові слова, такі як ‘економічна стабільність’ та ‘системні ризики’, є основними поняттями дисципліни”*
	Системний аналіз	15%	*“Тематика публікації стосується ‘Systemic risks’ та ‘financial system’, що дуже поверхнево перетинається з поняттям ‘системний’ з дисципліни. Основний зміст публікації повністю виходить за рамки навчальної програми”*
“Дослідження ефективності паралельної обчислювальної схеми ідентифікації інтервальних дискретних моделей на основі ройового	Інженерія кросплатформеного ПЗ	40%	*“Публікація обговорює ‘паралельні обчислювальні схеми’ та ‘ройовий інтелект’, що пов’язано з розробкою ПЗ та алгоритмами, які є базовими для кросплатформного ПЗ, навіть якщо вони не згадуються в цій конкретній навчальній програмі, яка зосереджена на управлінні та маркетингу”*
	Управління соціаль-	15%	*“Тематика публікації стосується паралельних обчислень та моделювання, що не має очевидного зв’язку з дисципліною”*

Продовження таблиці Ж.2

ройового інтелекту”. Алгоритм штучної бджоли-ної колонії, параметрична ідентифікація, екологічне моде-лювання, парале-льні обчислюва-льні схеми, аналіз інтервальних даних	ними ризиками		
	Публічне управління в сфері націона-льної безпеки	20%	*“Відповідність низька, ключове слово ‘екологічне моделювання’ безпосередньо корелює з темою ‘екологічної безпеки’, що згаданою в навчальній програмі, хоча інші слова є технічними”*
	Систем-ний аналіз	60%	*“Публікація пов’язана з дисципліною завдяки ключовим словам ‘паралельні обчислювальні схеми’, ‘параметрична ідентифікація’ та ‘аналіз інтервальних даних’, що відповідають темам ‘методологічні аспекти моделювання’ та ‘методи системного аналізу’“*