

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

ЮШКО АНДРІЙ ВАСИЛЬОВИЧ

УДК 004.4:004.8:005.8:001.895

ДИСЕРТАЦІЯ


**МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПЛАНУВАННЯ
НАУКОВОЇ 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.В. Юшко

Науковий керівник: Дивак Микола Петрович, доктор технічних наук, професор

Тернопіль – 2025

АНОТАЦІЯ

Юшко А. В. Математичне та програмне забезпечення для планування наукової 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.

Дисертаційна робота присвячена комплексному розв'язанню актуального науково-технічного завдання створення математичного та програмного забезпечення для планування наукової діяльності науково-педагогічних працівників. У сучасних умовах розвитку науки важливою тенденцією є перехід до системи управління, що ґрунтується на даних, де формування стратегічних рішень залежить від можливості своєчасно отримувати повні, достовірні та структуровані відомості про результати наукової роботи. Зростання обсягів інформації, поява нових типів наукових даних, активне використання цифрових ідентифікаторів та міжнародних аналітичних платформ зумовлюють підвищені вимоги до інформаційних систем університетів. У свою чергу, підвищення відповідальності закладів вищої освіти за якість наукової діяльності та необхідність обґрунтованого планування наукового навантаження науково-педагогічних працівників формують потребу у забезпеченні ефективних інструментів для прогнозування дослідницької активності.

Актуальність дослідження обумовлена низкою факторів, притаманних сучасному науковому середовищу. Передусім, значна частина університетів працює в умовах фрагментованості наукових даних, що зберігаються у вигляді розрізнених локальних баз, зовнішніх репозитаріїв, персональних профілів дослідників та ресурсів міжнародних наукометричних систем. Відсутність уніфікованих форматів даних ускладнює їх інтеграцію та підвищує ризики дублювання. До того ж рівень наповненості та якість наукових профілів НПП залишаються нерівномірними: значна кількість працівників не мають актуального ORCID-профілю, не пов'язані з глобальними ідентифікаторами

авторства, не представлені у Scopus чи OpenAlex, що істотно впливає на формування достовірних вибірок для аналітики. Сучасні CRIS- і RIMS-системи, попри високий рівень розвитку та широке застосування, здебільшого орієнтовані на ретроспективний аналіз наукової діяльності та не пропонують математично обґрунтованих механізмів прогнозування, особливо в умовах неповноти вихідних даних. Водночас наявні регламенти МОН України щодо планування наукової діяльності вимагають від університетів підвищення точності й обґрунтованості прогнозних показників, що додатково актуалізує це дослідження.

У роботі обґрунтовано, що традиційні програмні засоби, які використовуються у ЗВО України, орієнтовані переважно на збирання статистичної інформації для формування звітності, тоді як сучасні потреби передбачають застосування систем, здатних забезпечувати автоматизований збір даних із зовнішніх платформ, інтелектуальну їх фільтрацію, формування семантично узгоджених профілів дослідників та побудову персоналізованих прогнозів наукового навантаження. У цьому контексті виникає науково-технічна проблема створення математичної моделі, здатної працювати з неповними, зашумленими та нерівномірними даними, а також програмної системи, що підтримує автоматизовані процеси збору, обробки й інтерпретації наукових даних.

У першому розділі дисертації здійснено комплексний аналіз сучасних інформаційних систем управління науковою діяльністю, включно з національними, інституційними та міжнародними CRIS-платформами. Показано, що навіть найрозвиненіші системи, такі як Elsevier Pure, Converis, Symplectic Elements та VIVO, залишаються інструментами переважно документування та моніторингу дослідницької діяльності, не забезпечуючи функцій достовірного прогнозування. Значну увагу приділено аналізу якості та структури наукових даних, характерних для українських університетів, а також нормативним вимогам, що регламентують облік і планування наукової роботи НПП. У рамках теоретичного аналізу обґрунтовано вибір інтервального аналізу

як методологічної основи для прогнозування, показано його переваги над статистичними та навчальними методами в умовах обмежених вибірок та високого ступеня невизначеності. Окреслено недоліки класичних моделей, що вимагають повноти даних, і показано, що застосування інтервальних моделей дозволяє уникнути помилкового відчуття точності, властивого детермінованим підходам, та формувати прогнози у вигляді діапазонів можливих значень.

Другий розділ дисертації присвячено розробці комплексної інформаційної технології збору наукових даних, у рамках якої запропоновано багаторівневу модель наукової діяльності. Ця модель охоплює різні види наукової активності, включаючи публікаційну, проектну, грантову, дисертаційну, експертну, патентну та інші. Запропонована модель дає змогу структурувати інформацію з різних джерел і забезпечувати її семантичну єдність. У роботі детально описано механізми автоматизованого отримання наукових даних з міжнародних ресурсів Scopus, Crossref, OpenAlex та інституційних репозитаріїв за протоколом OAI-PMH. Особливу увагу приділено реалізації алгоритмів веб-парсингу, що дозволяють отримувати дані з джерел, які не мають формалізованих форматів експорту. Описано розроблений ETL-процес, який включає автоматизоване вилучення, очищення, фільтрацію, нормалізацію, дедублікацію та інтеграцію даних у єдину інформаційну структуру. Також у розділі наведено інтелектуальні методи фільтрації нерелевантних публікацій, що використовують методи обробки природної мови та векторні подання текстів, забезпечуючи високу точність формування наукових профілів НПП.

У третьому розділі представлено математичне забезпечення прогнозування наукової діяльності. У рамках моделі розроблено ієрархію показників наукової активності, визначено інтервальні оцінки вихідних параметрів і сформовано узагальнену інтервальну модель прогнозування сумарного наукового навантаження. У роботі доведено, що застосування інтервальних оцінок дозволяє коректно працювати в умовах неповних або зашумлених даних, забезпечуючи стійкість моделі. Розглянуто властивості інтервальних систем лінійних рівнянь, питання їх розв'язності та умови

існування еліпсоїдних оцінок параметрів моделей. Значне місце у розділі приділено геометричній інтерпретації прогнозованої моделі, де еліпсоїди в просторі параметрів відображають множину всіх можливих варіантів вагових коефіцієнтів, що узгоджуються з історичними даними. Такий підхід забезпечує наочність інтерпретації та дозволяє формувати прогноз як діапазон можливих значень із гарантованими межами, що є важливим для реальних управлінських рішень. У роботі показано, що інтервальний підхід враховує природну варіативність наукової діяльності, дозволяє адаптувати модель до індивідуальних особливостей профілю науковця та забезпечує високий рівень стійкості прогнозів.

Четвертий розділ дисертації присвячено розробці програмної системи, що реалізує математичні та інформаційні підходи, описані в попередніх розділах. Програмне забезпечення побудовано на основі сервісно-орієнтованої архітектури з використанням Node.js, GraphQL, MongoDB та Redis. Розроблена система включає модулі збору даних, фільтрації нерелевантних публікацій, формування профілів науковців, обчислення прогнозів та формування планів наукової діяльності. У роботі детально описано логіку інтеграції модулів, особливості оптимізації продуктивності, механізми логування, відмовостійкість та масштабування. Значну увагу приділено реалізації користувацького інтерфейсу, що забезпечує наочний перегляд наукових профілів, формування прогнозів, аналіз структури наукової активності та генерацію планових документів. Система пройшла всебічне модульне, інтеграційне та навантажувальне тестування, що підтвердило її коректність, ефективність і здатність працювати з великими обсягами даних.

У роботі наведено результати апробації розробленої системи в умовах діяльності реального наукового підрозділу закладу вищої освіти. Апробація підтвердила високу точність інтеграції даних, ефективність автоматизованого формування профілів НПП, можливість коректного прогнозування навіть за умов неповних даних та значне зменшення рутинного навантаження на адміністративний персонал.

Практичне значення дисертаційної роботи полягає у створенні повнофункціональної інтегрованої системи, що забезпечує автоматизований збір, обробку, аналіз та прогнозування наукової діяльності. Система може бути використана у закладах вищої освіти для формування звітів, планування наукового навантаження, аналізу ефективності наукової роботи, підтримки управлінських рішень та розвитку інституційної інфраструктури науки. Розроблені математичні та програмні рішення сприяють підвищенню якості прийняття рішень, забезпечують прозорість наукової діяльності та створюють передумови для ефективного функціонування аналітичних систем нового покоління.

Ключові слова: інформаційна технологія, архітектура програмного забезпечення, математична модель, прогнозування наукової діяльності, інтервальний аналіз даних, інтелектуальна інформаційна система, ранжування публікацій, науковий профіль дослідника, великі мовні моделі, автоматизований збір даних, планування наукового навантаження, модель представлення знань.

ПЕРЕЛІК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Наукові праці, в яких опубліковано основні наукові результати дисертації:

1. Юшко А. В., Сімак А. В. 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: розроблено архітектуру та програмну реалізацію системи аналізу наукової й науково-педагогічної діяльності академічного персоналу).

DOI: <https://doi.org/10.31649/1681-7893-2024-47-1-7-16>

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. (0,7 д.а. / 0,35 д.а.; особистий внесок: розроблено алгоритми збору відкритих даних та модуль інтеграції із зовнішніми джерелами в системі рейтингування НПП).

DOI: <https://doi.org/10.31891/2219-9365-2025-81-45>

3. Дивак М., Юшко А. Математична модель прогнозування показників наукової діяльності науково-педагогічних працівників з використанням інтервального аналізу даних. *Herald of Khmelnytskyi National University. Technical Sciences*. 2025. Т. 353(3.2). С. 375–386. (1,0 д.а. / 0,5 д.а.; особистий внесок: побудовано математичну модель прогнозування наукової активності НПП на основі інтервального аналізу та сформовано експериментальні сценарії прогнозування).

DOI: <https://doi.org/10.31891/2307-5732-2025-353-53>

4. Дивак М., Манжула В., Мельник А., Юшко А. Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. *Measuring and computing devices in technological processes*. 2024. №1. С. 125–139. (1,1 д.а. / 0,3 д.а.; особистий внесок: розроблено архітектурні компоненти хмарного сервісу для моделювання інтервальних даних та реалізовано функціональні модулі їх обробки).

DOI: <https://doi.org/10.31891/2219-9365-2024-77-15>

5. Yushko A., Dyvak M., Melnyk A., Trufanova Y., Kobytzia V., Martsenyuk Y. AI-Based Vector Filtering of Irrelevant Academic Publications Using Research Interest Modeling. *2025 15th International Conference on Advanced Computer Information Technologies (ACIT)*. 2025. P. 779–784. (0,5 д.а. / 0,25 д.а.; особистий внесок: розроблено метод векторної фільтрації нерелевантних публікацій та реалізовано програмний прототип моделювання наукових інтересів).

DOI: <https://doi.org/10.1109/acit65614.2025.11185809>

6. Yushko A., Dyvak M., Koruts U., Taraj M., Hrabar E., Sprinsyan V. The Information-Intellectual System for Evaluating Scientific and Scientific-Pedagogical Activities of the Academic Community. *2024 14th International Conference on ACIT*. 2024. Vol. 2136. P. 828–833. (0,5 д.а. / 0,2 д.а.; особистий внесок: розроблено

концептуальну модель оцінювання наукової діяльності НПП та реалізовано ключові компоненти програмної системи).

DOI: <https://doi.org/10.1109/acit62333.2024.10712561>

Наукові праці, які засвідчують апробацію матеріалів дисертації

7. Pukas A., Simak A., Yushko A., Nadvynychnyy S., Shandruk S., Vatslavskiy O. Automated Reporting Module in the Academic Staff Performance Appraisal System. *2024 14th International Conference on ACIT*. 2024. Vol. 2021. P. 824–827. (0,4 д.а. / 0,1 д.а.; особистий внесок: реалізовано алгоритми автоматизованого формування звітів та створено модуль експорту аналітичних даних).

DOI: <https://doi.org/10.1109/acit62333.2024.10712573>

8. Desyatnyuk O., Dyvak M., Yushko A., Koruts U., Nadvynychnyy S., Ostroverkhov V. The Method of Forecasting a Researcher's Project and Publication Activity Indicators Based on Interval Data Analysis. *2025 15th International Conference on ACIT*. 2025. P. 701–707. (0,45 д.а. / 0,15 д.а.; особистий внесок: сформовано підхід до прогнозування проектної та публікаційної активності науковця в системі інтервального аналізу та забезпечено інтеграцію моделей у програмну систему).

DOI: <https://doi.org/10.1109/acit65614.2025.11185612>

9. Dyvak M., Yushko A., Savenko O. та ін. An intelligent information system for generating a scientist's scientometrics using content analysis methods. *AISTDS'2024*, Київ, 1 October 2024. (0,3 д.а. / 0,1 д.а.; особистий внесок: розроблено модуль аналізу змісту публікацій і формування базових наукометричних показників).

URL: https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

10. Dyvak M., Yushko A., Savenko O. та ін. Web-oriented application for student attendance accounting with automatic parsing of class schedules. *YAISD'2025*, Ternopil, 8 September 2025. (0,3 д.а. / 0,1 д.а.; особистий внесок: реалізовано модуль автоматичного парсингу розкладу та обробки даних відвідуваності).

URL: <https://ceur-ws.org/Vol-3974/paper17.pdf>

11. Yushko A., Shevchuk R., Łopaciński K., Leszczynska M., Yashchuk O., Yurchyshyn T. Shielding Web Application against Cyber-Attacks using SIEM. *2023 13th International Conference on ACIT*. 2023. P. 393–396. (0,4 д.а. / 0,1 д.а.; особистий внесок: розроблено модуль виявлення аномалій у веб-трафіку та інтеграцію з SIEM-системою).

DOI: <https://doi.org/10.1109/acit58437.2023.10275630>

12. Shevchuk R., Mykytiuk V., Miramontes D. A., Yushko A., Yurchyshyn T., Mokhun S. Software Service for Studying Words of the Ukrainian Language. *2021 11th International Conference on ACIT*. 2021. P. 648–651. (0,35 д.а. / 0,1 д.а.; особистий внесок: створено модуль статистичного аналізу мовних даних та оптимізовано алгоритми пошуку й класифікації).

DOI: <https://doi.org/10.1109/acit52158.2021.9548448>

13. Pukas A., Melnyk A., Voytyuk I., Yushko A., Romanyuk M., Honchar L. Transactional Business Application Based on Microservice Architecture. *2021 11th International Conference on ACIT*. 2021. P. 564–567. (0,35 д.а. / 0,1 д.а.; особистий внесок: розроблено окремі мікросервіси бізнес-логіки та забезпечено їх інтеграцію в загальну систему).

DOI: <https://doi.org/10.1109/acit52158.2021.9548385>

ANNOTATION

Yushko A. V. Mathematical model and software tools for planning the research activities of academic staff. – Qualifying scientific work on the rights of a manuscript. Dissertation for the degree of Doctor of Philosophy in Specialty 121 «Software Engineering». – West Ukrainian National University, Ternopil, 2025.

The dissertation is devoted to the comprehensive solution of the urgent scientific and technical task of creating mathematical and software support for the automated collection, integration, analysis, and forecasting of the scientific activity of scientific and pedagogical staff. In modern conditions of scientific development, an important trend is the transition to a data-driven management system, where the formation of strategic decisions depends on the ability to receive complete, reliable, and structured information about the results of scientific work in a timely manner. The growth of information volumes, the emergence of new types of scientific data, and the active use of digital identifiers and international analytical platforms impose increased requirements on university information systems. In turn, increasing the responsibility of higher education institutions for the quality of scientific activity and the need for substantiated planning of the scientific workload of scientific and pedagogical staff create a need for effective tools for forecasting research activity.

The relevance of the study is conditioned by a number of factors inherent in the modern scientific environment. Above all, a significant proportion of universities operate in conditions of fragmented scientific data stored in scattered local databases, external repositories, personal researcher profiles, and resources of international scientometric systems. The lack of unified data formats complicates their integration and increases the risks of duplication. Furthermore, the level of completeness and quality of academic staff profiles remains uneven: a significant number of employees do not have an up-to-date ORCID profile, are not linked to global authorship identifiers, and are not represented in Scopus or OpenAlex, which significantly affects the formation of reliable samples for analytics. Modern CRIS and RIMS systems, despite their high level of development and wide application, are mostly focused on retrospective analysis of scientific activity and do not offer mathematically grounded

forecasting mechanisms, especially under conditions of incomplete source data. At the same time, existing regulations of the Ministry of Education and Science of Ukraine regarding the planning of scientific activity require universities to increase the accuracy and validity of forecast indicators, which further actualizes this study.

The work substantiates that traditional software tools used in Higher Education Institutions (HEIs) of Ukraine are primarily oriented towards collecting statistical information for reporting, whereas modern needs require the application of systems capable of ensuring automated data collection from external platforms, their intelligent filtering, the formation of semantically consistent researcher profiles, and the construction of personalized forecasts of scientific workload. In this context, a scientific and technical problem arises regarding the creation of a mathematical model capable of working with incomplete, noisy, and uneven data, as well as a software system that supports automated processes of collection, processing, and interpretation of scientific data.

In the first chapter of the dissertation, a comprehensive analysis of modern information systems for managing scientific activity is carried out, including national, institutional, and international CRIS platforms. It is shown that even the most developed systems, such as Elsevier Pure, Converis, Symplectic Elements, and VIVO, remain tools primarily for documenting and monitoring research activity, not providing reliable forecasting functions. Significant attention is paid to the analysis of the quality and structure of scientific data characteristic of Ukrainian universities, as well as the regulatory requirements governing the accounting and planning of the scientific work of academic staff. Within the framework of the theoretical analysis, the choice of interval analysis as a methodological basis for forecasting is substantiated, and its advantages over statistical and learning methods under conditions of limited samples and a high degree of uncertainty are shown. The shortcomings of classical models requiring data completeness are outlined, and it is demonstrated that the application of interval models allows avoiding the false sense of accuracy inherent in deterministic approaches and forming forecasts in the form of ranges of possible values.

The second chapter of the dissertation is devoted to the development of a comprehensive information technology for scientific data collection, within the framework of which a multi-level model of scientific activity is proposed. This model covers various types of scientific activity, including publication, project, grant, dissertation, expert, patent, and others. The proposed model allows structuring information from various sources and ensuring its semantic unity. The work details the mechanisms for automated acquisition of scientific data from international resources Scopus, Crossref, OpenAlex, and institutional repositories via the OAI-PMH protocol. Particular attention is paid to the implementation of web-parsing algorithms that allow obtaining data from sources that do not have formalized export formats. The developed ETL process is described, which includes automated extraction, cleaning, filtering, normalization, deduplication, and integration of data into a single information structure. The chapter also presents intelligent methods for filtering irrelevant publications, utilizing natural language processing methods and vector text representations, ensuring high accuracy in the formation of academic staff scientific profiles.

The third chapter presents the mathematical support for forecasting scientific activity. Within the framework of the model, a hierarchy of scientific activity indicators is developed, interval estimates of input parameters are defined, and a generalized interval model for forecasting the total scientific workload is formed. The work proves that the application of interval estimates allows for correct operation under conditions of incomplete or noisy data, ensuring model stability. The properties of interval systems of linear equations, questions of their solvability, and conditions for the existence of ellipsoidal estimates of model parameters are considered. Significant space in the chapter is devoted to the geometric interpretation of the predictive model, where ellipsoids in the parameter space reflect the set of all possible variants of weight coefficients consistent with historical data. This approach ensures the clarity of interpretation and allows forming a forecast as a range of possible values with guaranteed bounds, which is important for real management decisions. The work shows that the interval approach takes into account the natural variability of scientific

activity, allows adapting the model to the individual characteristics of the scientist's profile, and ensures a high level of forecast stability.

The fourth chapter of the dissertation is devoted to the development of a software system implementing the mathematical and information approaches described in the previous chapters. The software is built on the basis of a service-oriented architecture using Node.js, GraphQL, MongoDB, and Redis. The developed system includes modules for data collection, filtering irrelevant publications, forming scientist profiles, calculating forecasts, and generating scientific activity plans. The work details the integration logic of the modules, performance optimization features, logging mechanisms, fault tolerance, and scaling. Significant attention is paid to the implementation of the user interface, which ensures visual viewing of scientific profiles, forecast formation, analysis of the structure of scientific activity, and generation of planning documents. The system underwent comprehensive unit, integration, and load testing, which confirmed its correctness, efficiency, and ability to work with large volumes of data.

The work presents the results of the approbation of the developed system in the conditions of the activity of a real scientific unit of a higher education institution. The approbation confirmed the high accuracy of data integration, the effectiveness of the automated formation of academic staff profiles, the possibility of correct forecasting even under conditions of incomplete data, and a significant reduction in the routine workload on administrative personnel.

The practical value of the dissertation lies in the creation of a fully functional integrated system that ensures automated collection, processing, analysis, and forecasting of scientific activity. The system can be used in higher education institutions for generating reports, planning scientific workload, analyzing the efficiency of scientific work, supporting management decisions, and developing the institutional infrastructure of science. The developed mathematical and software solutions contribute to improving the quality of decision-making, ensure transparency of scientific activity, and create prerequisites for the effective functioning of new-generation analytical systems.

Keywords: information technology, software architecture, mathematical model, forecasting of scientific activity, interval data analysis, intelligent information system, publication ranking, researcher's scientific profile, Large Language Models, automated data collection, scientific workload planning, knowledge representation model.

LIST OF PUBLISHED PAPERS BY THE TOPIC OF THESIS

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

1. Yushko A. V., Simak A. V. Information-intellectual system for the analysis of scientific and scientific-pedagogical activity of the academic staff. *Optic-electronic information-energy technologies*. 2024. Issue 47(1). P. 7–16. (0.9 p.s. / 0.3 p.s.; personal contribution: developed the architecture and software implementation of the system for analyzing the scientific and scientific-pedagogical activity of academic personnel). DOI: <https://doi.org/10.31649/1681-7893-2024-47-1-7-16>
2. Simak A., Yushko A. Intelligent module for collecting open data in the rating system of scientific and pedagogical staff activity. *Measuring and computing devices in technological processes*. 2025. No. 1. P. 353–361. (0.7 p.s. / 0.35 p.s.; personal contribution: developed algorithms for open data collection and a module for integration with external sources in the academic staff rating system). DOI: <https://doi.org/10.31891/2219-9365-2025-81-45>
3. Dyvak M., Yushko A. Mathematical model for forecasting indicators of scientific activity of scientific and pedagogical staff using interval data analysis. *Herald of Khmelnytskyi National University. Technical Sciences*. 2025. Vol. 353(3.2). P. 375–386. (1.0 p.s. / 0.5 p.s.; personal contribution: built a mathematical model for forecasting the scientific activity of academic staff based on interval analysis and formed experimental forecasting scenarios). DOI: <https://doi.org/10.31891/2307-5732-2025-353-53>
4. Dyvak M., Manzhula V., Melnyk A., Yushko A. Software architecture for mathematical modeling based on interval data analysis using cloud technologies. *Measuring and computing devices in technological processes*. 2024. No. 1. P. 125–139. (1.1 p.s. / 0.3 p.s.; personal contribution: developed architectural components of

the cloud service for modeling interval data and implemented functional modules for their processing). DOI: <https://doi.org/10.31891/2219-9365-2024-77-15>

5. Yushko A., Dyvak M., Melnyk A., Trufanova Y., Kobytzia V., Martsenyuk Y. AI-Based Vector Filtering of Irrelevant Academic Publications Using Research Interest Modeling. *2025 15th International Conference on Advanced Computer Information Technologies (ACIT)*. 2025. P. 779–784. (0.5 p.s. / 0.25 p.s.; personal contribution: developed a method for vector filtering of irrelevant publications and implemented a software prototype for modeling research interests). DOI: <https://doi.org/10.1109/acit65614.2025.11185809>

6. Yushko A., Dyvak M., Koruts U., Taraj M., Hrabar E., Sprinsyan V. The Information-Intellectual System for Evaluating Scientific and Scientific-Pedagogical Activities of the Academic Community. *2024 14th International Conference on ACIT*. 2024. Vol. 2136. P. 828–833. (0.5 p.s. / 0.2 p.s.; personal contribution: developed a conceptual model for evaluating the scientific activity of academic staff and implemented key components of the software system). DOI: <https://doi.org/10.1109/acit62333.2024.10712561>

Scientific works certifying the approval of the dissertation materials:

7. Pukas A., Simak A., Yushko A., Nadvynychnyy S., Shandruk S., Vatslavskiy O. Automated Reporting Module in the Academic Staff Performance Appraisal System. *2024 14th International Conference on ACIT*. 2024. Vol. 2021. P. 824–827. (0.4 p.s. / 0.1 p.s.; personal contribution: implemented algorithms for automated report generation and created an analytical data export module). DOI: <https://doi.org/10.1109/acit62333.2024.10712573>

8. Desyatnyuk O., Dyvak M., Yushko A., Koruts U., Nadvynychnyy S., Ostroverkhov V. The Method of Forecasting a Researcher's Project and Publication Activity Indicators Based on Interval Data Analysis. *2025 15th International Conference on ACIT*. 2025. P. 701–707. (0.45 p.s. / 0.15 p.s.; personal contribution: formed an approach to forecasting the project and publication activity of a researcher in the interval analysis system and ensured the integration of models into the software system). DOI: <https://doi.org/10.1109/acit65614.2025.11185612>

9. Dyvak M., Yushko A., Savenko O. et al. An intelligent information system for generating a scientist's scientometrics using content analysis methods. *AISTDS'2024*, Kyiv, 1 October 2024. (0.3 p.s. / 0.1 p.s.; personal contribution: developed a module for analyzing the content of publications and generating basic scientometric indicators). URL: https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

10. Dyvak M., Yushko A., Savenko O. et al. Web-oriented application for student attendance accounting with automatic parsing of class schedules. *YAISD'2025*, Ternopil, 8 September 2025. (0.3 p.s. / 0.1 p.s.; personal contribution: implemented a module for automatic schedule parsing and attendance data processing). URL: <https://ceur-ws.org/Vol-3974/paper17.pdf>

11. Yushko A., Shevchuk R., Łopaciński K., Leszczynska M., Yashchyk O., Yurchyshyn T. Shielding Web Application against Cyber-Attacks using SIEM. *2023 13th International Conference on ACIT*. 2023. P. 393–396. (0.4 p.s. / 0.1 p.s.; personal contribution: developed a module for detecting anomalies in web traffic and integration with the SIEM system). DOI: <https://doi.org/10.1109/acit58437.2023.10275630>

12. Shevchuk R., Mykytiuk V., Miramontes D. A., Yushko A., Yurchyshyn T., Mokhun S. Software Service for Studying Words of the Ukrainian Language. *2021 11th International Conference on ACIT*. 2021. P. 648–651. (0.35 p.s. / 0.1 p.s.; personal contribution: created a module for statistical analysis of language data and optimized search and classification algorithms). DOI: <https://doi.org/10.1109/acit52158.2021.9548448>

13. Pukas A., Melnyk A., Voytyuk I., Yushko A., Romanyuk M., Honchar L. Transactional Business Application Based on Microservice Architecture. *2021 11th International Conference on ACIT*. 2021. P. 564–567. (0.35 p.s. / 0.1 p.s.; personal contribution: developed individual business logic microservices and ensured their integration into the overall system). DOI: <https://doi.org/10.1109/acit52158.2021.9548385>

ЗМІСТ

АНОТАЦІЯ	2
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	19
ВСТУП.....	20
РОЗДІЛ 1 АНАЛІЗ СИСТЕМ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПЛАНУВАННЯ НАУКОВОЇ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ.....	30
1.1. Проблематика планування та обліку наукової діяльності в закладах вищої освіти ..	31
1.2. Аналіз існуючих систем для планування наукової діяльності у ЗВО	34
1.3. Аналіз методів прогнозування показників наукової діяльності	45
1.3.1. Статистичні методи.....	47
1.3.2. Методи машинного навчання.....	49
1.3.3. Інтервальні методи прогнозування	57
1.4. Постановка завдань дисертаційного дослідження	57
Висновки до розділу 1	61
РОЗДІЛ 2 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗБОРУ ДАНИХ ПРО НАУКОВУ ДІЯЛЬНІСТЬ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ.....	63
2.1. Стандарти та принципи опису наукових даних в інформаційному забезпеченні збору наукової діяльності	64
2.2. Інформаційна технологія збору наукової діяльності НПП.....	70
2.3. Модуль семантичного впорядкування зібраної інформації про наукову діяльність ..	73
Висновки до розділу 2	81
РОЗДІЛ 3 МЕТОД ПРОГНОЗУВАННЯ ПОКАЗНИКІВ НАУКОВОЇ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ НА ОСНОВІ АНАЛІЗУ ІНТЕРВАЛЬНИХ ДАНИХ.....	83
3.1. Ієрархія показників наукової діяльності.....	84
3.2. Метод прогнозування показників наукової діяльності на основі аналізу інтервальних даних	87
3.2.1. Формування «портрету» працівника щодо його наукової діяльності у вигляді інтервальної системи алгебричних рівнянь	88

3.2.2. Формування еліпсоїдних оцінок вектора вагових коефіцієнтів «портрету» працівника щодо його наукової діяльності	94
3.2.3. Формування інтервальних оцінок для векторів вагових коефіцієнтів «портрету» працівника щодо його наукової діяльності	96
3.3. Математична модель планування та прогнозування видів наукової діяльності та обсягів часу за кожної з них	99
Висновки до розділу 3	107
РОЗДІЛ 4 ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПЛАНУВАННЯ НАУКОВОЇ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНОЇ ПРАЦІВНИКІВ	108
4.1. Архітектура програмної системи для планування наукової діяльності науково-педагогічних працівників.....	109
4.2. Підсистема аналізу та зберігання даних	124
4.3. Особливості програмної реалізації системи для збору та аналізу наукової та науково-педагогічної діяльності академічного колективу	126
4.4. Тестування програмного середовища	133
4.5. Організація графічного інтерфейсу програмного середовища системи для збору та аналізу науково та науково-педагогічної діяльності академічного колективу.....	144
Висновки до розділу 4	153
ВИСНОВКИ.....	155
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	158
ДОДАТОК А ПОРІВНЯЛЬНА ТАБЛИЦЯ МЕТОДІВ ПРОГНОЗУВАННЯ НАУКОВОЇ ДІЯЛЬНОСТІ	176
ДОДАТОК Б ЛІСТИГ КОДУ ОСНОВНИХ МОДУЛІВ ПРОГРАМИ.....	178
ДОДАТОК В СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ.....	208
ДОДАТОК Г АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО ДОСЛІДЖЕННЯ.....	210

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API	Application Programming Interface
CRIS	Current Research Information System
RIMS	Research Information Management System
ORCID	Open Researcher and Contributor ID
DOI	Digital Object Identifier
CERIF	Common European Research Information Format
NLP	Natural Language Processing
ML	Machine Learning
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GBM	Gradient Boosting Machine
RBF	Radial Basis Function
SVR	Support Vector Regression
KPI	Key Performance Indicators
ЗВО	Заклад вищої освіти
НПП	Науково-педагогічний працівник
НДР	Науково-дослідна робота
ПЗ	Програмне забезпечення
ІТ	Інформаційні технології
ІАС	Інформаційно-аналітична система
ЕП	Еліпсоїдна оцінка (в контексті інтервального аналізу)
МН	Машинне навчання
ІА	Інтервальний аналіз
REST	Representational State Transfer
JSON	JavaScript Object Notation
МА	Moving Average
RMSE	Root Mean Square Error

ВСТУП

Актуальність теми дослідження. У системі вищої освіти наукова діяльність науково-педагогічних працівників є важливим елементом формування іміджу закладу, забезпечення його рейтингових позицій та виконання державних нормативів. Особливої актуальності наукова діяльність у ЗВО набуває після прийняття закону №9600 «Про внесення змін до деяких законів України щодо підтримки наукової роботи в закладах вищої освіти» від 06.06.2024 року. Разом з тим, облік і планування цієї діяльності дедалі більше ґрунтується на цифрових платформах, базах даних і аналітичних системах, здатних збирати, структурувати й інтерпретувати великі обсяги наукометричної інформації. Зростаючий обсяг публікацій, патентів та проєктної діяльності, а також розгалуженість джерел (Scopus, Web of Science, Crossref, ORCID, Укрпатент, УКРНТИ), відсутність уніфікованих форматів ідентифікації авторів та дублювання даних істотно ускладнюють аналіз і потребують впровадження комп'ютеризованих рішень, а останнім часом і засобів інтелектуалізації.

В університетів виникає необхідність переходу від фрагментованих практик обліку до створення єдиного інформаційного середовища, що забезпечує автоматизований збір, нормалізацію, верифікацію та семантичну узгодженість наукових даних. Водночас відповідно до реалізації положень вище згаданого закону «Про внесення змін до деяких законів України щодо підтримки наукової роботи в закладах вищої освіти» від 06.06.2024 року, необхідним є впровадження сучасних цифрових та інтелектуальних інструментів для оцінювання та планування показників наукової діяльності НПП. Для таких цілей вже існують вітчизняні та зарубіжні комерційні CRIS-платформи переважно орієнтовані на ретроспективний облік результатів наукової діяльності. Серед відомих програмних систем варто виділити Elsevier Pure, Converis, Symplectic Elements та VIVO. Архітектурно такі системи побудовані навколо модулів збору, зберігання та аналітики даних, проте в них відсутній окремий компонент, призначений для прогнозування показників наукової діяльності на основі

індивідуального профілю науковця та обмежених вибірок історичних даних. Відсутність цього елемента архітектури програмних систем обмежує функціональні можливості існуючих CRIS-рішень. Іншою проблемою зазначених систем є складність інтеграції у їхні існуючі архітектури модулів для інтелектуалізації процесів збору, інтеграції, аналізу та прогнозування наукових показників. Зокрема зазначені архітектури ускладнюють можливість інтеграції модулів великих мовних моделей.

Разом з тим у працях таких дослідників, як J. A. V. Tohalino, A. Ma та B. Wang, пропонується ряд сучасних методів аналізу й прогнозування показників наукової діяльності, а також наведено окремі елементи архітектурних рішень щодо інтеграції можливостей великих мовних моделей у відповідні інформаційно-аналітичні системи [148, 149, 150]. Однак ситуація ускладнюється неповнотою, шумами та структурною неоднорідністю наукових даних, що знижує ефективність традиційних статистичних методів або моделей машинного навчання. Окремою проблемою є обмежені вибірки даних про наукову діяльність НПП, оскільки зазвичай до уваги приймаються виключно річні звіти. За таких умов актуальною стає розробка нових математичних підходів, здатних враховувати невизначеність та неповноту інформації в умовах обмеженої вибірки історичних даних, а також реалізація нових архітектурних програмних рішень, які поєднують та інтелектуалізують процеси збору, інтеграції, аналізу, фільтрації нерелевантних відомостей, формування структурованих профілів дослідників та прогнозування показників наукової діяльності з урахуванням неточності даних, яку можна представити у вигляді числових інтервалів показників.

Таким чином, розробка нових архітектурних рішень, інформаційної технології та математичних методів і моделей прогнозування показників наукової діяльності НПП в умовах обмеженої вибірки історичних даних та з врахуванням індивідуального профілю науковця є актуальним науково-технічним завданням, яке відповідає сучасним потребам закладів вищої освіти та

сприяє підвищенню ефективності управлінських процесів у сфері наукової діяльності ЗВО.

Мета і завдання дослідження. Метою дисертаційної роботи є комплексна інтелектуалізація процесів збору, інтеграції, аналізу та прогнозування показників наукової діяльності НПП у спосіб реалізації нових архітектурних рішень, інформаційної технології, математичних методів і моделей прогнозування в умовах обмеженої вибірки історичних даних та з врахуванням індивідуального профілю науковця.

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

1. провести аналіз методів та засобів збору, інтеграції, аналізу та прогнозування показників наукової діяльності НПП;
2. розробити ієрархічну систему показників наукової активності;
3. розробити інформаційну технологію збору, інтеграції, аналізу та прогнозування наукових показників, яка забезпечує інтелектуалізоване об'єднання різнорідних джерел інформації у єдине уніфіковане середовище.
4. розробити математичну модель прогнозування сумарного наукового навантаження науково-педагогічних працівників із використанням інтервального аналізу даних.
5. розробити нові архітектурні рішення для інтелектуалізації процесів збору, аналізу та прогнозування показників наукової діяльності, а також реалізувати прикладне програмне забезпечення у вигляді інтелектуалізованої системи для планування наукової діяльності науково-педагогічних працівників;
6. провести апробацію розробленої системи на прикладі ЗВО.

Об'єкт дослідження. Процеси збору, інтеграції, аналізу та прогнозування наукової діяльності науково-педагогічних працівників.

Предмет дослідження. Математичне забезпечення, архітектурні рішення та інтелектуалізована програмна системи планування наукової діяльності науково-педагогічних працівників.

Методи дослідження. Для розробки методу прогнозування показників наукової діяльності та інтервальної математичної моделі планування та прогнозування видів наукової діяльності і обсягів часу за кожної з них, використано методи аналізу інтервальних даних, методи ідентифікації та методи еліпсоїдного оцінювання. Для розробки інформаційної технології збору, інтеграції, аналізу та прогнозування наукових показників НПП використано системний підхід, методи контент аналізу, методи аналізу семантичної подібності та методи класифікації та кластеризації. Для розробки нових архітектурних рішень інтелектуалізованої програмної системи збору, інтеграції, аналізу та прогнозування наукової діяльності, використано методи архітектурного проектування ПЗ, об'єктно орієнтований підхід, а також середовища моделювання та проектування програмних систем, що забезпечують формалізацію компонентної структури, взаємодії сервісів і реалізацію принципів масштабованості та модульності.

Наукова новизна отриманих результатів. У межах дисертаційної роботи вперше:

- запропоновано та обґрунтовано інформаційну технологію збору, інтеграції, аналізу та прогнозування наукових показників НПП, яка забезпечує інтелектуалізоване об'єднання різномірних джерел інформації (міжнародні наукометричні бази, інституційні репозитарії, реєстри дисертацій, грантові портали) у єдине уніфіковане середовище. На відміну від існуючих CRIS- та RIMS-рішень, запропонована технологія містить програмні процедури інтелектуалізованого пошуку та інтеграції даних, а також нормалізацію записів, усунення дублювання, верифікацію авторських профілів на веб ресурсах, персоналізацію та прив'язку інформації з різних джерел до профілю науковця, що в цілому інтелектуалізує процес формування даних про наукову діяльність НПП, підвищує достовірність отриманих даних та слугує основою для прогнозування показників наукової діяльності в залежності від портрету науковця в умовах обмеженої вибірки історичних даних;

- запропоновано та обґрунтовано метод прогнозування показників наукової діяльності науково-педагогічних працівників на основі аналізу інтервальних даних, який на відміну від існуючих ґрунтується на еліпсоїдних оцінках вектора вагових коефіцієнтів для кожної групи показників, що характеризують «портрет» науковця, що у сукупності дає можливість отримати інтервальні оцінки запланованих показників наукової діяльності науково-педагогічного працівника побудованих на мінімальній вибірці даних;

- запропоновано та обґрунтовано інтервальну математичну модель планування та прогнозування видів наукової діяльності та обсягів часу за кожної з них, які на відміну від існуючих дають можливість отримати інтервальні оцінки прогнозованих показників наукової діяльності науково-педагогічного працівника і такі, що ґрунтуються на його науковому портреті, що у сукупності дає можливість запланувати наукову роботу, виходячи із можливостей та попередніх досягнень науково-педагогічного працівника, а також отримати гарантований реалістичний план діяльності на прогнозований рік;

набули подальшого розвитку:

- архітектурні рішення для розробки інтелектуалізованої програмної системи збору, інтеграції, аналізу та прогнозування наукової діяльності, які на відміну від існуючих, об'єднують механізми машинного навчання, обробки природної мови та мультиджерельного збору даних, модулі контент аналізу та прогнозування показників наукової діяльності, що у сукупності забезпечує комплексну інтелектуалізацію усіх зазначених процесів і прогнозування в умовах обмеженої вибірки історичних даних та з врахуванням індивідуального профілю науковця.

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

[18] запропоновано та реалізовано інтелектуальний модуль автоматизованого збору відкритих наукових даних, який поєднує веб-парсинг,

взаємодію з API зовнішніх наукометричних ресурсів та подальшу обробку даних засобами великих мовних моделей;

[36] запропоновано та обґрунтовано метод прогнозування показників наукової та проєктної діяльності науково-педагогічних працівників на основі інтервального аналізу, який формує «профіль дослідника» у вигляді еліпсоїдних оцінок вектора вагових коефіцієнтів.

[11] запропоновано математичну модель прогнозування наукової діяльності науково-педагогічних працівників на основі інтервального аналізу, яка використовує ієрархічну систему показників та дозволяє визначати допустимі межі майбутнього наукового навантаження;

[20] розроблено архітектуру інформаційної інтелектуальної системи для автоматизованого збору, інтеграції та аналізу наукової діяльності науково-педагогічних працівників, яка включає модулі імпорту даних з наукометричних джерел (Scopus, Web of Science, Crossref), інтелектуальний модуль збору веб-даних та механізми семантичної обробки інформації;

[95] запропоновано та обґрунтовано сервісно-орієнтовану архітектуру програмного забезпечення для підтримки математичного моделювання на основі аналізу інтервальних даних у хмарному середовищі, яка використовує RESTful API, моделі розподілених обчислень та хмарні інструменти Google Cloud для оптимізації обчислювальної складності;

[139] запропоновано підхід до автоматизованого формування профілю науковця на основі контент-аналізу публікацій із використанням великих мовних моделей та векторних представлень текстів;

[141] запропоновано та реалізовано математичну та алгоритмічну модель ранжування нерелевантних наукових публікацій на основі векторного моделювання наукових інтересів дослідника;

Апробація результатів дисертації. Основні положення і результати дисертаційної роботи презентовано на 4 конференціях, зокрема:

- 11th International Conference on Advanced Computer Information, ACIT2021, Deggendorf, Germany, 15-17 Sept, 2021;

- 13th International Conference on Advanced Computer Information Technologies ACIT2023, Wroclaw, Poland, 21-23 Sept., 2023;
- 14th International Conference on Advanced Computer Information Technologies, Ceske Budejovice, Czech Republic, 19-21 Sept., 2024.
- 15th International Conference on Advanced Computer Information Technologies, Sibenik, Croatia, 17-19 Sept., 2025.

Публікації. За результатами дисертаційного дослідження опубліковано 13 наукових праць (Додаток В) загальним обсягом 108 сторінок, зокрема 4 статті у фахових наукових видання категорії Б [1][2][3][4], 9 публікацій у матеріалах конференцій [5-13], які входять до міжнародної наукометричної бази Scopus.

Структура та обсяг роботи. Дисертаційна робота складається із вступу, чотирьох розділів, висновків, списку використаних джерел із 150 найменувань та 4 додатків. Загальний обсяг роботи складає 212 сторінку друкарського тексту, з них 136 сторінок основного тексту. Робота містить 58 рисунків і 4 таблиці.

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

Дисертаційна робота виконувалася в рамках пріоритетного напрямку «Інформаційні та комунікаційні технології» відповідно до Закону України «Про пріоритетні напрями розвитку науки і техніки» та пов'язана з науковими дослідженнями, які виконувалися за держбюджетною та госпдоговірною тематикою кафедри комп'ютерних наук Західноукраїнського національного університету. Основні результати дисертаційного дослідження отримано в межах виконання таких тем:

- держбюджетного дослідження на тему: «Математичне та програмне забезпечення для ідентифікації та моніторингу особливо небезпечних джерел забруднення ґрунту та ґрунтових вод» (державний реєстраційний номер 0120U102040), у якому автором набуті навички формалізації, структурування та аналізу великих масивів екологічних даних, що стали фундаментом для розроблення універсальних алгоритмів вилучення, нормалізації та попередньої обробки даних у рамках дисертаційної роботи. Отримані підходи до побудови математичних моделей та методів моніторингу комплексних систем були

адаптовані для створення модулів збору та структуризації інформації про наукову діяльність;

- госпдоговірної роботи «Оптимізація веб-порталу “Газпостач”» (державний реєстраційний номер 0121U110387), у межах якої автором удосконалено механізми оптимізації вебсервісів, потоків даних та роботи серверної частини. Набутий досвід у створенні сервісно-орієнтованих інтерфейсів та оптимізації інформаційних систем став основою для побудови програмної архітектури системи збору та аналізу наукових даних, розробленої в дисертації;

- науково-професійних послуг за темою «Комплексна розробка та SEO-налаштування публічного сайту ПрАТ “Тернопільміськгаз” з модулями особистого кабінету та подачі показників непобутовими споживачами» (2023 р.), у рамках яких автором створено низку клієнтських та серверних модулів, реалізовано механізми автентифікації, обробки показників та інтерактивної взаємодії користувачів із веб-системою. Здобутий практичний досвід використано у дисертації під час розроблення модулів персоналізації, управління доступом та побудови інтерфейсів системи наукової аналітики;

- держбюджетного дослідження на тему: «Математичне та комп’ютерне моделювання об’єктів з розподіленими параметрами на основі поєднання онтологічного та інтервального аналізу» (державний реєстраційний номер 0122U001497), у якому автором поглиблено методи онтологічного та інтервального аналізу, що стали основою для створення інтервальних моделей прогнозування наукової активності та побудови узагальненої моделі наукових об’єктів. Розроблені підходи забезпечили можливість формалізації різнотипних даних та їх інтеграції в єдине аналітичне середовище;

- науково-професійних послуг за темою «Послуги з розробки модуля відслідковування заявок користувачів та розробку технічної документації» (2024 р.), у межах яких автором розроблено інструменти логування, трасування та управління заявками користувачів. Отримані результати адаптовано до потреб дисертаційної роботи для побудови модулів управління подіями наукової

діяльності, контролю актуальності записів та формування технічної документації програмного забезпечення.

Усі вищезазначені роботи виконувалися за безпосередньої участі автора, а їх результати забезпечили формування теоретичних, методичних і прикладних засад дисертаційного дослідження та створення комплексної інформаційної системи збору, аналізу та прогнозування наукової діяльності науково-педагогічних працівників.

Практичне значення отриманих результатів полягає у розробці інтегрованої інтелектуальної системи для автоматизованого збору, обробки та прогнозування наукової діяльності науково-педагогічних працівників. Розроблені інформаційні технології забезпечують мультиджерельну інтеграцію даних, їх нормалізацію, фільтрацію нерелевантних публікацій та формування дослідницьких профілів на основі методів обробки природної мови. Інтервальна модель прогнозування дозволяє отримувати стійкі оцінки майбутнього наукового навантаження в умовах неповних та неоднорідних даних. Реалізоване програмне забезпечення може бути використане у закладах вищої освіти для автоматизації підготовки звітів, формування індивідуальних планів, аналітичної підтримки управлінських рішень та підвищення ефективності наукової діяльності академічних колективів.

Теоретичні та прикладні результати дисертаційної роботи використано:

- в НДЧ Західноукраїнського національного університету при виконанні держбюджетних та госпдоговірних досліджень, де елементи розробленого програмного забезпечення та методичні підходи використано для оптимізації архітектури інформаційних систем, організації процесів збору даних і підвищення ефективності моделювання складних процесів, зокрема у темах:

- «Оптимізація веб-порталу "Газпостач"» (державний реєстраційний номер 0121U110387), у межах якої застосовано розроблені автором методи оптимізації веб-сервісів та структуризації даних (акт про використання результатів дисертаційної роботи від 17 вересня 2025 р.);

- «Методи, засоби та інформаційна технологія ідентифікації джерел забруднень приземистого шару атмосфери на основі їх математичних моделей» (державний реєстраційний номер 0120U100880), у якій застосовано алгоритми структуризації та опрацювання великих масивів даних, що використовуються у створеній автором системі;

- «Математичне та програмне забезпечення для ідентифікації та моніторингу особливо небезпечних джерел забруднення ґрунту та ґрунтових вод» (державний реєстраційний номер 0120U102040), де практично апробовано методи формалізації, нормалізації та інтеграції гетерогенних даних, реалізовані у дисертаційній роботі;

- «Математичне та комп'ютерне моделювання об'єктів з розподіленими параметрами на основі поєднання онтологічного та інтервального аналізу» (державний реєстраційний номер 0122U001497), у межах якої використано розроблені автором інтервальні моделі та онтологічні структури для організації та аналізу інформації;

- виконання науково-професійних послуг із розроблення модулів відслідковування заявок користувачів та створення технічної документації (2024 р.), де апробовано механізми логування, управління подіями та документування, які інтегровано у створену інформаційну систему.

- в освітньому процесі Західноукраїнського національного університету, зокрема при викладанні дисципліни «Інтелектуальні інформаційні системи», де результати дисертаційного дослідження використано для демонстрації застосування інтервальних методів, онтологічних моделей, алгоритмів прогнозування та архітектурних принципів побудови сучасних інформаційних систем (акт про впровадження в освітній процес від 10 вересня 2025 р.)

У додатку Г до дисертації подано акти впровадження результатів дисертаційного дослідження.

РОЗДІЛ 1

АНАЛІЗ СИСТЕМ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ПЛАНУВАННЯ НАУКОВОЇ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

У першому розділі дисертаційного дослідження здійснено аналіз сучасного стану систем для планування та обліку наукової діяльності науково-педагогічних працівників. Розглянуто проблематику організації наукової діяльності в закладах вищої освіти України, визначено ключові недоліки наявних практик, зокрема фрагментарність нормативно-правового забезпечення, формалізм у плануванні та відсутність єдиних механізмів інтеграції даних [2, 12, 93].

Окрему увагу приділено вітчизняним інформаційним рішенням, серед яких центральне місце займає Єдина інформаційна система «Наука», що забезпечує облік і моніторинг науково-дослідних робіт у закладах вищої освіти та інтеграцію даних на державному рівні. Показано, що, попри значний крок уперед у напрямі централізації даних, ця система має обмежений функціонал щодо відображення індивідуальної наукової активності та підтримки процесів прогнозування.

Проаналізовано зарубіжний досвід впровадження CRIS- та RIMS-систем, зокрема таких як Elsevier Pure, Converis, Symplectic Elements, VIVO та DSpace-CRIS. Визначено, що ці рішення відзначаються високим рівнем інтеграції з міжнародними наукометричними базами, автоматизацією збору даних, наявністю аналітичних інструментів і розвинутим функціоналом для стратегічного планування. Водночас акцент зроблено на тому, що навіть у цих системах функції прогнозування наукової активності залишаються недостатньо розвинутими.

На основі проведеного аналізу сформульовано постановку задачі, суть якої полягає у потребі розробки інтелектуальних інформаційних систем, здатних забезпечувати комплексне прогнозування наукової діяльності науково-педагогічних працівників на рівні закладу вищої освіти. Обґрунтовано

необхідність розробки нових математичних і програмних засобів, які поєднують функції збору, інтеграції та аналізу даних із можливістю формування прогнозів, що стане основою для подальшого дослідження. Результати цього розділу опубліковано автором у таких працях: [20, 139].

1.1. Проблематика планування та обліку наукової діяльності в закладах вищої освіти

Організація наукової діяльності науково-педагогічних працівників у закладах вищої освіти є складним процесом, що поєднує елементи стратегічного планування, управління ресурсами та систематичного моніторингу результатів. В умовах інтенсивного розвитку науки та зростання вимог до ефективності досліджень питання планування та обліку наукової діяльності набуває особливої актуальності. Від того, наскільки якісно побудована система організації наукової роботи, залежить не лише індивідуальна продуктивність дослідників, але й конкурентоспроможність університету в цілому.

Проблематика планування наукової діяльності в Україні визначається кількома ключовими чинниками. По-перше, нормативно-правове поле вимагає від університетів забезпечення прозорого й системного обліку наукової роботи. Закон України «Про вищу освіту» [15] та Закон України «Про наукову і науково-технічну діяльність» [16] встановлюють необхідність ведення звітності щодо результатів досліджень, грантової активності, публікацій, патентів та інших форм наукового доробку. Однак механізми реалізації цих вимог залишаються здебільшого фрагментарними, що створює розрив між законодавчими приписами і реальними практиками університетів. Значна частина облікової документації досі формується у вигляді традиційних паперових звітів або електронних таблиць, що не забезпечує інтегрованості й не дозволяє ефективно здійснювати аналітику.

По-друге, у більшості закладів вищої освіти планування наукової діяльності носить формальний характер і часто зводиться до щорічного подання

індивідуальних планів роботи науково-педагогічних працівників. Такі плани здебільшого виконують функцію адміністративного звіту, а не інструмента реального управління науковою діяльністю. У результаті виникає ситуація, коли планування не стимулює розвиток дослідницького потенціалу, не враховує індивідуальну динаміку наукової продуктивності та не забезпечує узгодження між стратегіями університету й інтересами конкретного науковця.

По-третє, важливим викликом є проблема даних. Облік наукових результатів ускладнюється неоднорідністю джерел інформації: публікації відображаються в різних наукометричних базах (Scopus, Web of Science, Google Scholar), проектна діяльність часто фіксується у відомчих системах, а участь у конференціях, редакційних колегіях чи експертних радах узагалі не має уніфікованої форми представлення. Відсутність централізованої бази даних призводить до дублювання інформації, втрати важливих показників і зниження достовірності аналітичних висновків.

Ще однією суттєвою проблемою є складність прогнозування наукового навантаження. Традиційні системи планування орієнтовані переважно на фіксацію фактів, тоді як сучасні умови потребують застосування методів аналітики та прогнозування, що дозволяють передбачати тенденції розвитку дослідницької активності. Без таких інструментів університети стикаються з труднощами у плануванні ресурсів, визначенні пріоритетних напрямів розвитку, а також у формуванні персоналізованих рекомендацій для науковців.

Не менш важливим аспектом є питання інтеграції планування наукової діяльності з іншими підсистемами університетського управління. У багатьох випадках наукова активність розглядається окремо від освітньої чи адміністративної роботи, що знижує ефективність стратегічного менеджменту. Відсутність єдиного інформаційного простору унеможливорює побудову повноцінної картини діяльності кожного науково-педагогічного працівника, ускладнює оцінку його внеску та перешкоджає обґрунтованому розподілу ресурсів.

Таким чином, проблематика планування й обліку наукової діяльності в закладах вищої освіти зводиться до поєднання кількох факторів: недосконалості нормативної та інституційної бази, формалізму існуючих практик планування, розрізненості та неоднорідності даних, відсутності інструментів прогнозування й слабкої інтегрованості з іншими управлінськими процесами. Подолання цих обмежень можливе лише шляхом упровадження сучасних інформаційних систем, здатних не лише автоматизувати облік, але й створити підґрунтя для стратегічного розвитку наукового потенціалу університетів.

Разом із цим слід відзначити, що ефективне планування наукової діяльності неможливе без урахування прогнозування як його невід'ємної складової. Традиційні підходи зосереджуються переважно на ретроспективному обліку результатів чи формальному визначенні завдань на майбутній рік, проте сучасні умови вимагають переходу до передбачення тенденцій розвитку наукової активності.

Застосування методів прогнозування дозволяє оцінювати потенційну динаміку наукової продуктивності окремих дослідників та колективів, виявляти закономірності у зростанні чи зниженні результативності, визначати ймовірні ризики та можливості. Прогнозування також дає змогу враховувати різні сценарії розвитку науки, що особливо важливо для університетів в умовах конкуренції за ресурси та міжнародного визнання.

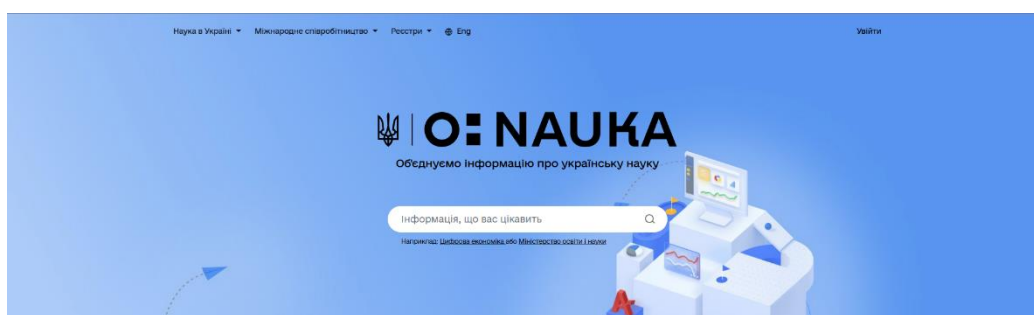
Ключовим інструментом у цьому процесі виступають ключові показники ефективності (KPI) [103, 145], які забезпечують кількісне відображення досягнень і перспектив. До таких показників належать кількість публікацій у виданнях, індексованих у міжнародних базах даних, рівень цитованості, участь у грантових проєктах, одержання патентів, участь у міжнародних конференціях і колабораціях. Використання KPI дає змогу не лише здійснювати об'єктивну оцінку результатів, але й прогнозувати їхню майбутню динаміку на основі аналізу історичних даних.

Особливої уваги потребує залучення сучасних методів аналізу даних, включаючи бібліометричні підходи, наукометричні індикатори та алгоритми

машинного навчання, здатні виявляти приховані закономірності в масивах інформації. Поєднання таких методів із функціоналом інформаційних систем управління наукою відкриває нові можливості для формування персоналізованих траєкторій розвитку науковців, прогнозування наукового навантаження та побудови більш гнучких стратегій підвищення ефективності дослідницької діяльності.

1.2. Аналіз існуючих систем для планування наукової діяльності у ЗВО

Система організації та планування наукової діяльності в Україні базується на поєднанні державних ініціатив, відомчих інформаційних ресурсів і внутрішніх механізмів університетів. На сьогоднішній день немає єдиного уніфікованого підходу до автоматизації цього процесу, проте простежується тенденція до поступового впровадження електронних засобів, здатних забезпечити інтеграцію даних і формування комплексної картини наукової активності. Одним із ключових елементів державної інфраструктури у сфері управління науковою діяльністю є Єдина інформаційна система «Наука», Міністерства освіти і науки України (Рис. 1.1).



[Установам та ЗВО](#) [Профілі](#)

Процедури



Рисунок 1.1 – Головна сторінка єдиної інформаційної системи «Наука»

Система створена з метою організації обліку, планування та моніторингу науково-дослідних робіт, які виконуються у закладах вищої освіти, а також для

інтеграції даних про наукову діяльність на національному рівні. Вона використовується для збору пропозицій щодо нових наукових тем, забезпечує конкурсну процедуру відбору проєктів, сприяє формуванню державного замовлення у сфері науки та надає консолідовану інформацію про результати діяльності університетів. Портал «НАУКА» виконує функції інтегратора інформації: він містить відомості про державні програми, пріоритетні напрями розвитку, наукові установи та їхні здобутки, формуючи інформаційне середовище для стратегічного планування. Таким чином, Єдина інформаційна система «Наука» забезпечує централізоване управління даними і прозорість у сфері науки, проте її функціонал переважно орієнтований на управління науково-дослідними роботами як формалізованими проєктами і поки що не охоплює повною мірою індивідуальну наукову активність викладачів.

На інституційному рівні більшість університетів застосовують власні підходи до планування й обліку наукової роботи. Традиційно цей процес реалізується через індивідуальні плани та звіти викладачів, які щорічно затверджуються кафедрами й адміністрацією. У сучасних умовах дедалі частіше запроваджуються інформаційні системи, інтегровані в університетські автоматизовані системи управління («АСУ Деканат», «АСУ ВНЗ»), що містять спеціалізовані модулі для обліку наукової діяльності. Ці модулі дозволяють вести статистику публікацій, участі в конференціях, грантовій та проєктній роботі, а також формувати узагальнені звіти для потреб ректорату та кафедр.

Переважає більшість українських закладів вищої освіти для зберігання та обліку результатів наукової діяльності використовують інституційні репозитарії на платформі DSpace, що стало де-факто стандартом у національному освітньо-науковому середовищі. В окремих університетах ця інфраструктура поступово набуває ознак систем, наближених до світових стандартів CRIS (Current Research Information Systems). Так, у Київському національному університеті імені Тараса Шевченка функціонує репозитарій на базі DSpace-CRIS (див. рис. 1.2), який забезпечує розширене управління метаданими та підтримку дослідницьких профілів [51, 66, 67].

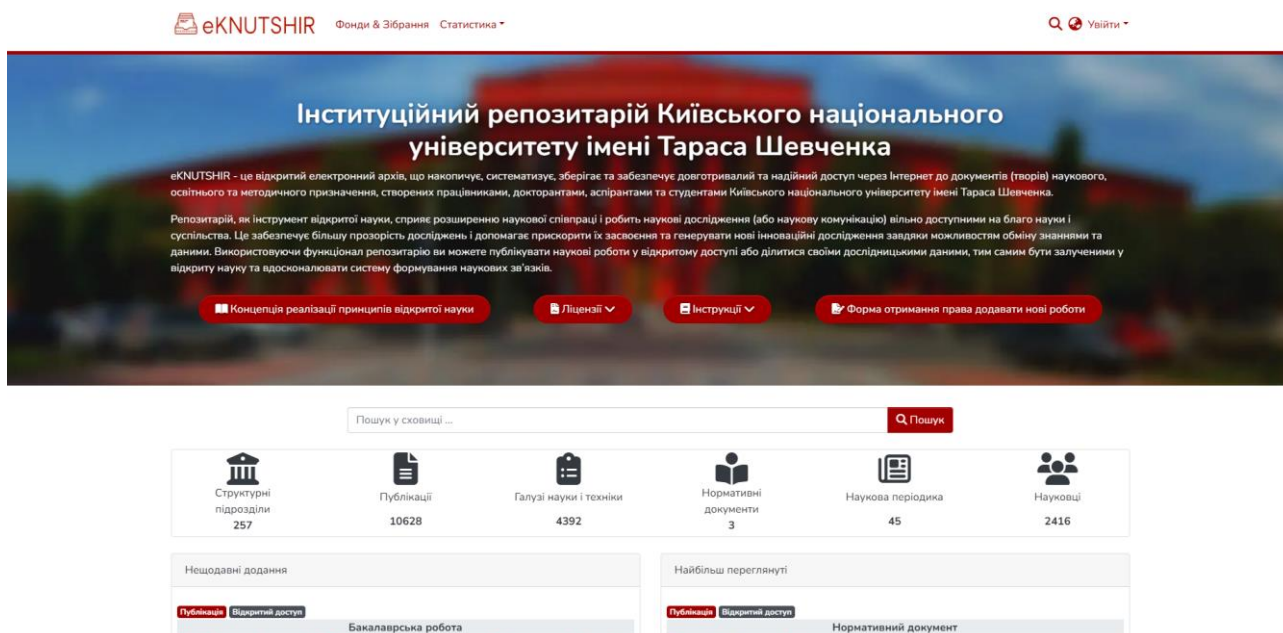


Рисунок 1.2 – Головна сторінка інституційного репозитарію Київського національного університету імені Тараса Шевченка

У Сумському державному університеті реалізовано політику відкритої науки, де CRIS-системи визначаються як стратегічний напрям розвитку, а ключовим елементом є репозитарій eSSUIR на основі DSpace (див. рис. 1.3).

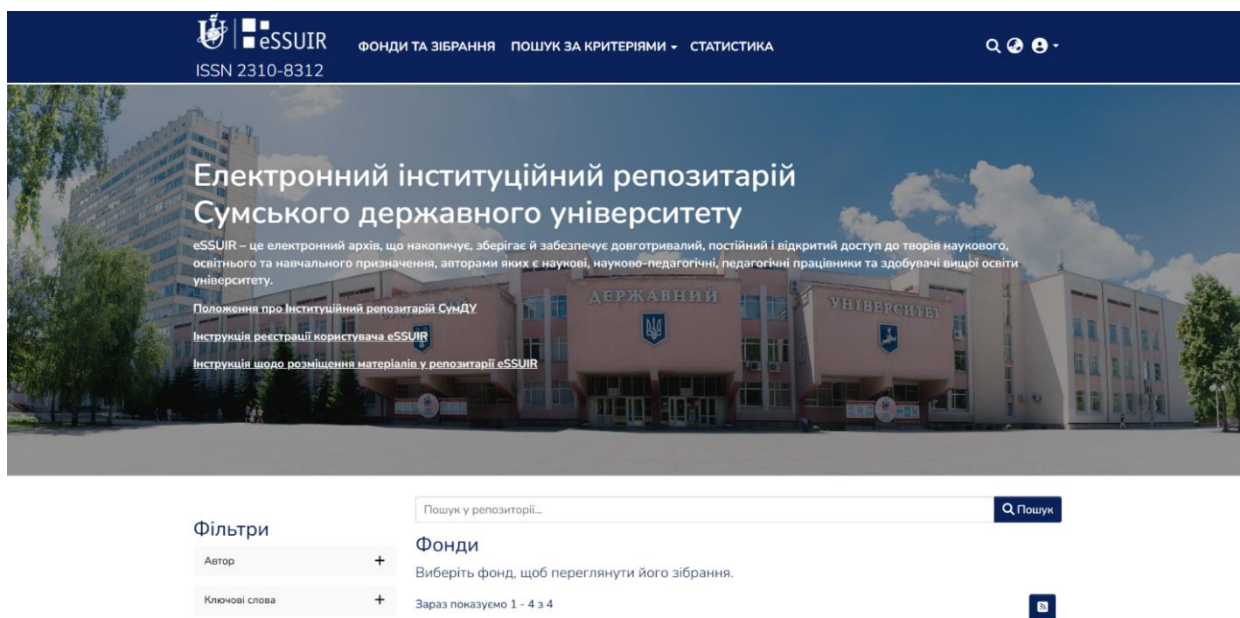


Рисунок 1.3 – Головна сторінка електронного інституційного репозитарію Сумського державного університету

У Національному університеті «Львівська політехніка» діє репозитарій DSpace (див. рис. 1.4), водночас університет долучається до проєктів забезпечення інтероперабельності з CRIS (зокрема, Open4UA), що свідчить про орієнтацію на інтеграцію у сучасні дослідницькі інформаційні екосистеми.

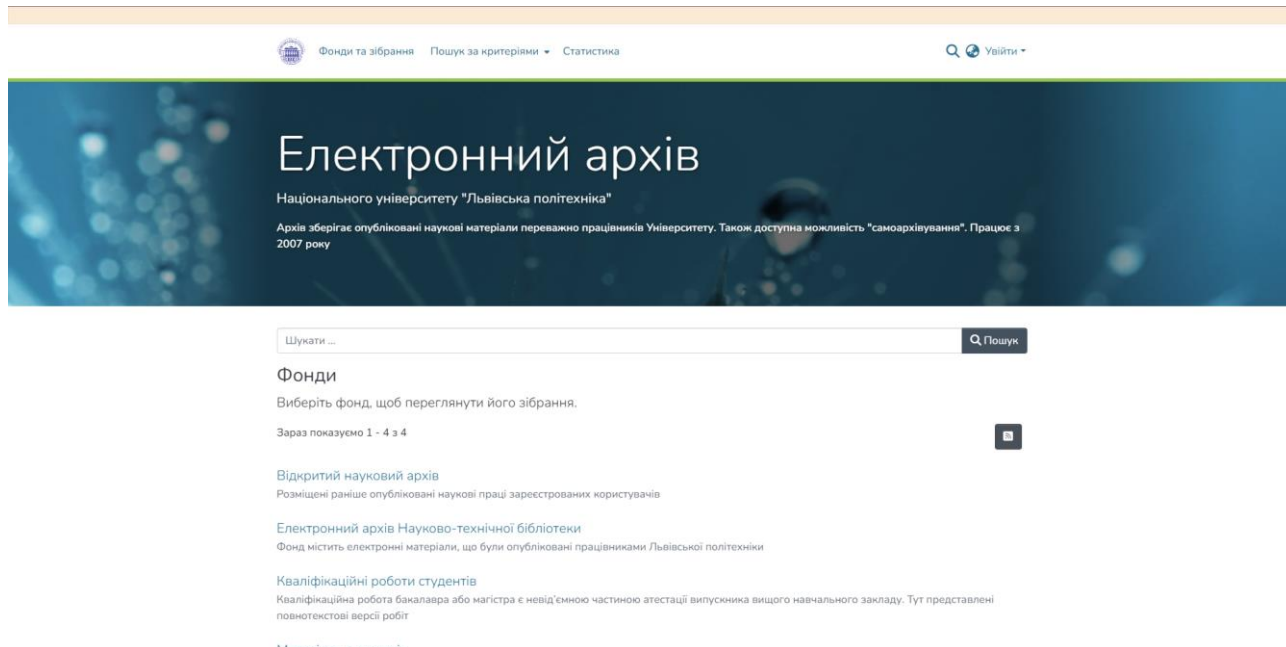


Рисунок 1.4 – Головна сторінка електронного архіву Національного університету «Львівська політехніка»

Водночас більшість українських університетів усе ще перебувають на етапі переходу від паперових або змішаних форм обліку до комплексних електронних систем. Проблемою залишається відсутність єдиних стандартів обміну даними, що ускладнює інтеграцію університетських платформ із міжнародними наукометричними базами та державними реєстрами. Наявна розрізненість підходів призводить до дублювання інформації, зростання навантаження на науково-педагогічних працівників та обмеженої можливості здійснювати стратегічну аналітику.

Отже, системи планування наукової діяльності в Україні характеризуються неоднорідністю та різним рівнем технологічної зрілості. На державному рівні функціонують базові інструменти для обліку науково-дослідних робіт і формування політики, тоді як на рівні університетів поступово впроваджуються власні інформаційні рішення, спрямовані на автоматизацію обліку й підвищення

прозорості. Однак більшість цих систем поки що орієнтовані радше на фіксацію результатів, ніж на прогнозування чи стратегічне планування, що залишає значний простір для їхнього подальшого розвитку [13, 145].

У міжнародній практиці організація планування та обліку наукової діяльності науково-педагогічних працівників характеризується значно вищим рівнем цифровізації та стандартизації. Університети та дослідницькі центри активно впроваджують інформаційні системи, що дозволяють не лише здійснювати облік публікацій, проєктів і грантів, але й інтегрувати дані з різних джерел, формувати профілі науковців, аналізувати їхню продуктивність та будувати прогнози щодо майбутньої дослідницької активності. Такі системи, відомі як CRIS (Current Research Information Systems) або RIMS (Research Information Management Systems), стали фактично стандартом у сфері управління науковою діяльністю.

Однією з найбільш поширених комерційних платформ є Elsevier Pure[33], яка використовується провідними університетами світу, зокрема в Європі та Північній Америці (Рис. 1.5.).

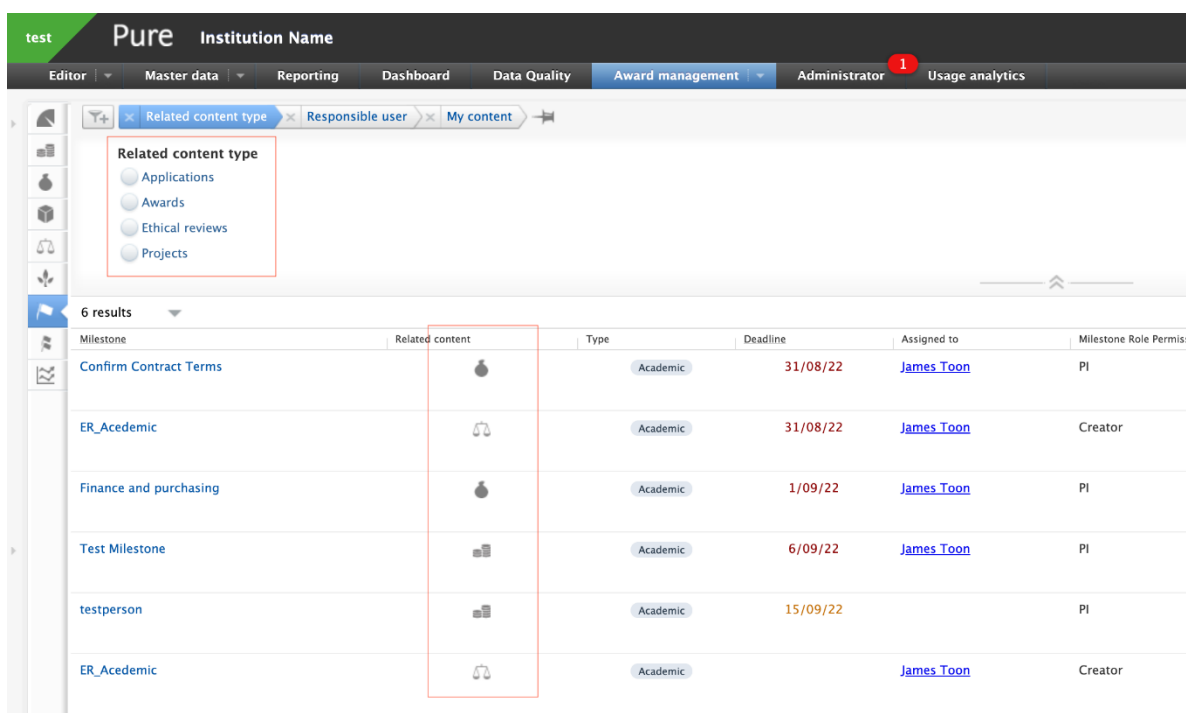


Рисунок 1.5 – Головна сторінка CRIS Elsevier Pure

Система забезпечує комплексний збір та інтеграцію даних про публікації, фінансовані проєкти, патенти, нагороди та інші результати наукової роботи. Pure дозволяє створювати профілі дослідників, автоматично пов'язувати їх із відповідними результатами, формувати інституційні звіти та здійснювати бібліометричний аналіз. Особливою перевагою цієї системи є тісна інтеграція з наукометричною базою Scopus, що забезпечує високий рівень автоматизації збору даних.

Іншим поширеним рішенням є Converis, що належить компанії Clarivate Analytics (Рис.1.6).

The screenshot shows the Converis web interface for a user named John Northrop. The interface includes a navigation menu on the left with options like Dashboard, Scholarship, Grants and Projects, Research Infrastructures, Intellectual Property, Activities, People, Organizations, Classifications, Notifications, and Statistics. The main content area displays the user's profile, including a photo, name, location, and email. Below the profile, there are three orange boxes showing statistics: 14 Publications, 217 Times Cited, and 4 h-Index. The interface also features sections for 'Things to do' (1 Application for completion) and 'Recently edited' (listing a research output and a CV activity).

Рисунок 1.6 – Вебінтерфейс CRIS Converis

На відміну від Pure, ця система робить акцент на управлінні життєвим циклом наукових проєктів, включаючи їх планування, подання заявок на фінансування, моніторинг виконання та звітування. Converis активно використовується університетами, які прагнуть інтегрувати наукову діяльність із фінансовими та адміністративними процесами. Завдяки підтримці стандарту CERIF (Common European Research Information Format) ця система здатна

забезпечувати сумісність з іншими міжнародними інформаційними платформами.

Важливе місце займає також Symplectic Elements — система, орієнтована на управління профілями науковців і збір інформації з численних зовнішніх джерел. Її особливістю є гнучкість інтеграції: система підтримує автоматичний імпорт даних із різних наукометричних баз, репозитаріїв та фінансових систем університету (Рис. 1.7.). Elements дозволяє не лише фіксувати досягнення, а й відображати взаємозв'язки між дослідниками, формуючи цілісну картину наукових колаборацій.

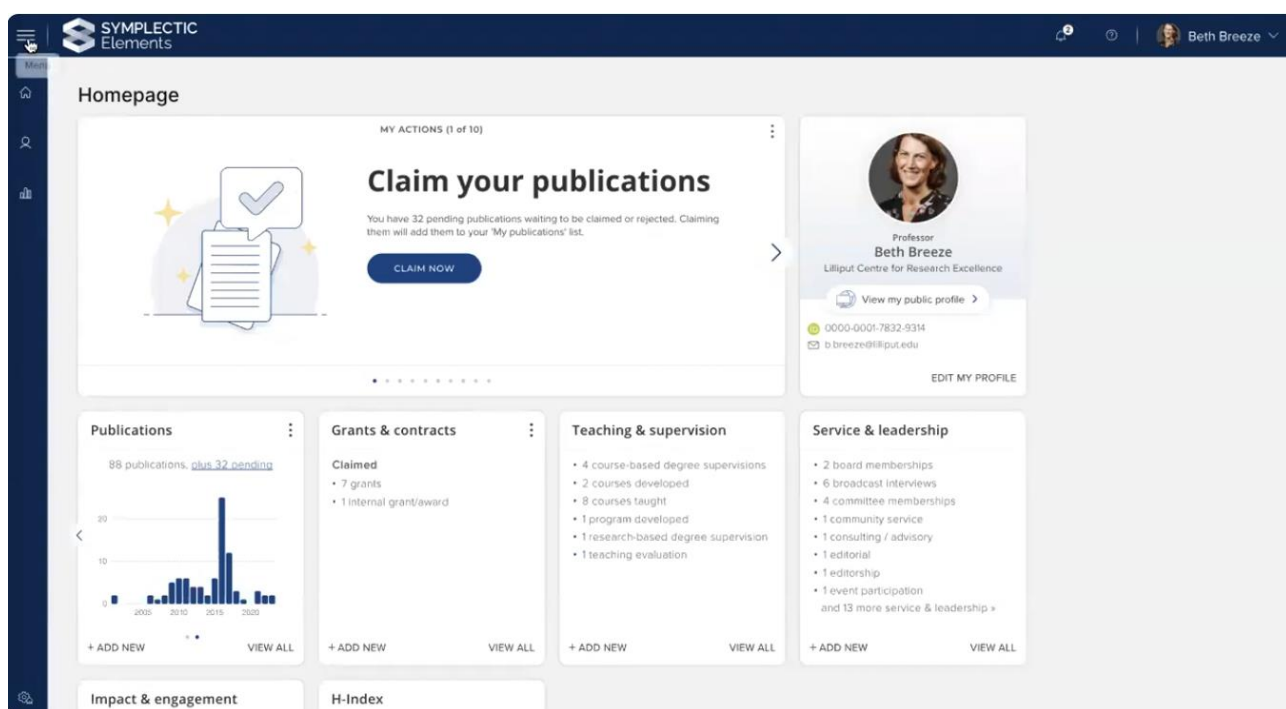


Рисунок 1.7 – Вебінтерфейс CRIS Symplectic Elements

Серед відкритих і безкоштовних рішень найбільш відомим є VIVO (Рис. 1.8.), розроблений у межах проєкту з відкритим вихідним кодом [26, 114]. Ця система орієнтована на представлення та візуалізацію інформації про науковців, їхні публікації, проєкти й партнерства. VIVO широко застосовується в університетах США та Європи завдяки своїй здатності створювати розширені

профілі дослідників і надавати інструменти для пошуку експертів у певних галузях.

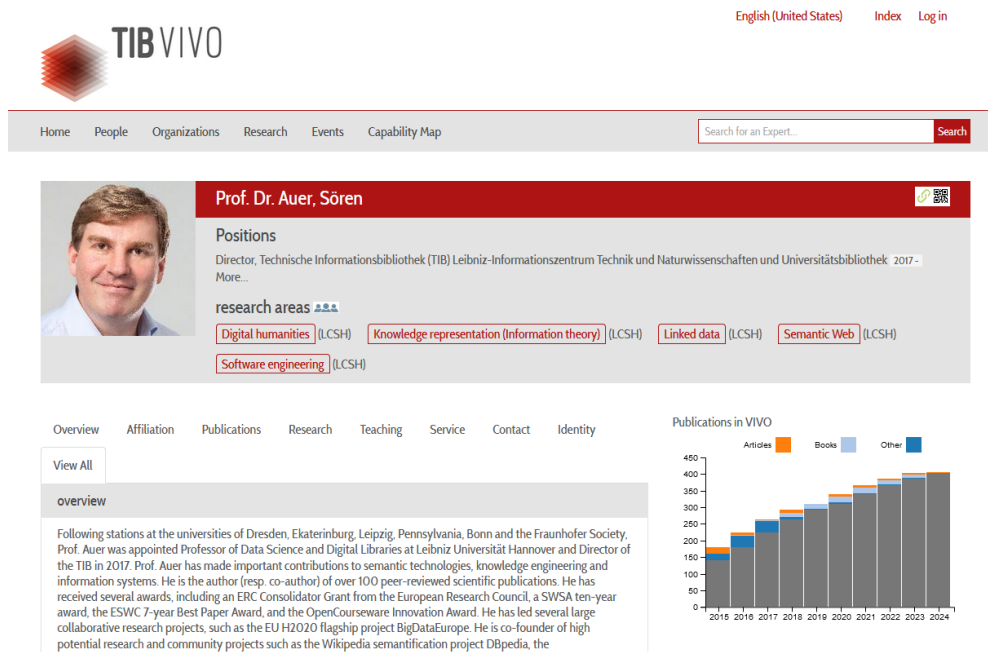


Рисунок 1.8 – Вебінтерфейс CRIS VIVO

Іншим відкритим рішенням є DSpace-CRIS (Рис. 1.9.), яке поєднує можливості інституційного репозитарію з функціоналом CRIS-системи, що робить його привабливим для університетів, орієнтованих на збереження та поширення результатів досліджень у відкритому доступі.

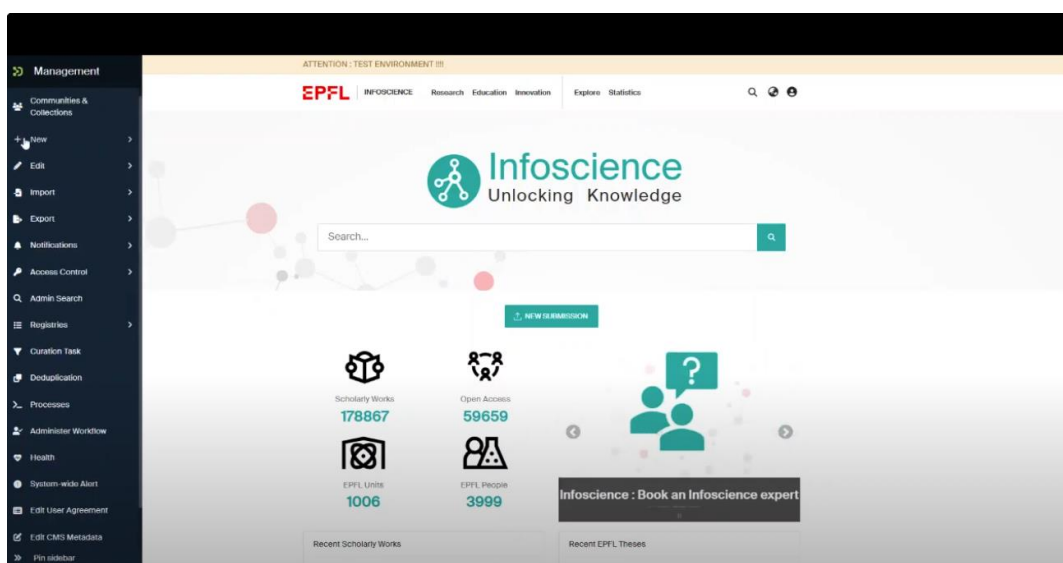


Рисунок 1.9 – Веб інтерфейс DSpace-CRIS

Варто зазначити, що зарубіжні системи характеризуються високим рівнем інтеграції з національними та міжнародними інформаційними інфраструктурами науки. Наприклад, у багатьох європейських країнах впроваджено національні CRIS-платформи, які акумулюють дані з університетів і наукових установ у єдиному інформаційному просторі [65, 117]. Це дозволяє державним органам здійснювати моніторинг ефективності науки на рівні країни, а університетам — інтегрувати свої локальні системи в ширший контекст європейської наукової політики.

Для забезпечення об'єктивного аналізу та порівняння інформаційних систем, що застосовуються у сфері планування наукової діяльності, необхідно визначити комплекс критеріїв, які відображають їхню результативність і функціональність. Такий підхід дозволяє уникнути фрагментарності у висновках, створити єдине підґрунтя для оцінювання різних рішень та окреслити перспективи їх удосконалення.

Насамперед значущим є рівень інтеграції системи з міжнародними наукометричними та бібліометричними базами даних. Чим повніше і точніше відбувається автоматичне оновлення інформації про публікаційну активність, цитованість чи участь у наукових проектах, тим достовірнішими є результати аналізу й менше навантаження припадає на користувачів. Водночас важливою характеристикою виступає можливість формування індивідуальних профілів науково-педагогічних працівників, які надають цілісне уявлення про їхні досягнення й дозволяють простежувати динаміку розвитку наукової продуктивності.

Оцінюючи системи, необхідно також враховувати наявність аналітичних інструментів і здатність до прогнозування на основі історичних даних. Саме цей аспект відрізняє сучасні рішення від традиційних, орієнтованих переважно на звітність. Можливість аналізувати тенденції, передбачати рівень наукової активності та пропонувати персоналізовані траєкторії розвитку є критично важливою умовою для підвищення ефективності управління наукою.

Не менш значущим є питання інтегрованості таких систем у ширший управлінський контекст закладу вищої освіти. Їх ефективність суттєво зростає, коли вони взаємодіють із кадровими, освітніми чи фінансовими модулями, забезпечуючи комплексне представлення діяльності університету. Важливими є й характеристики гнучкості та масштабованості, адже система повинна адаптуватися до потреб різних установ і підтримувати розширення функціоналу відповідно до нових завдань.

Окремої уваги заслуговує дотримання принципів відкритості та інтероперабельності. Використання міжнародних стандартів обміну даними, таких як CERIF чи ORCID, створює можливість для взаємодії з глобальними інформаційними інфраструктурами науки та сприяє включенню університетів у міжнародні дослідницькі мережі.

У підсумку можна констатувати, що застосування визначених критеріїв дозволяє сформулювати системний підхід до аналізу інформаційних систем для планування наукової діяльності. Це створює основу для побудови порівняльної оцінки, що відображатиме не лише функціональні можливості, але й наявні обмеження кожного рішення, відкриваючи шлях до їх подальшого вдосконалення (Таблиця 1.1).

Таблиця 1.1

Порівняльна таблиця систем планування наукової діяльності

Система	Збір даних з різних джерел	Інтеграція з науковими базами	Профілі НП	Аналітика і прогнозування	Інтеграція з управлінськими процесами	Гнучкість і масштабованість	Відкритість та інтероперабельність
Єдина інформаційна система «Наука» (Україна)	частково автоматизований, значна частина даних вводиться вручну	часткова інтеграція (ORCID, частково Scopus)	є, але функціонал обмежений	орієнтація на облік, прогнозування відсутнє	обмежена інтеграція з модулями ЗВО	середній рівень гнучкості	часткова підтримка ORCID, обмежена інтероперабельність

Продовження таблиці 1.1

Elsevier Pure	автоматизований збір із зовнішніх баз та внутрішніх систем університету	повна інтеграція (Scopus, WoS, Crossref)	розвинуті персональні профілі з повною історією	є розширена аналітика, прогнозування обмежене	тісна інтеграція з HR, фінансовими та освітніми системами	висока масштабованість, використовується глобально	повна підтримка CERIF, ORCID, DOI
Converis (Clarivate)	автоматизований збір з WoS, Crossref, внутрішніх джерел	повна інтеграція з WoS, Crossref, DOI	розвинуті профілі з гнучкою структурою	є аналітика, прогнозування не реалізоване	сильна інтеграція з адміністративними процесами	висока масштабованість	відповідність CERIF, ORCID, DOI
Symplectic Elements	потужне агрегування з WoS, PubMed, Scopus, Crossref, інституційних систем	інтеграція з WoS, PubMed, Scopus, Crossref	розвинуті профілі (публікації, проекти, гранти)	має аналітичні модулі, прогнозування обмежене	інтеграція з управлінськими модулями університетів	висока гнучкість	підтримка відкритих стандартів (ORCID, DOI)
VIVO	залежить від налаштування, можливе агрегування з багатьох відкритих джерел	інтеграція через відкриті API (налаштовується установою)	розвинуті дослідницькі профілі	орієнтована на візуалізацію зв'язків, прогнозування відсутнє	інтеграція можлива, але залежить від налаштування	висока завдяки open-source	відкрита архітектура, підтримка CERIF, ORCID
DSpace-CRIS	агрегація можлива, але обмежена, часто потребує ручного втручання	інтеграція часткова, базові конектори	профілі реалізуються, але без глибокої аналітики	базова аналітика, прогнозування відсутнє	інтеграція з інституційними репозитаріями	середня масштабованість	відкрита архітектура, підтримка OAI-PMH, ORCID

Аналіз українських та зарубіжних систем для планування наукової діяльності показав, що, незважаючи на різний рівень технологічної зрілості та

інтегрованості, вони мають спільні суттєві обмеження. Передусім жодна з розглянутих платформ не забезпечує повноцінного використання історичних даних для прогнозування майбутньої наукової активності. Крім того, значною проблемою є вибірка даних: у багатьох випадках попередні результати не були накопичені в цифрових форматах, а молоді науково-педагогічні працівники взагалі не мають достатнього обсягу історичних даних для формування репрезентативних прогнозів. Водночас навіть у найбільш функціональних рішеннях обмеженим залишається механізм збору даних із різних джерел: або він потребує суттєвого ручного втручання, або не охоплює всі релевантні інформаційні потоки.

Оскільки планування наукової діяльності в закладах вищої освіти здебільшого здійснюється у річному розрізі, відсутність комплексного аналізу попередніх даних та неповнота вибірок істотно знижують ефективність прогнозних оцінок. Це підкреслює потребу у створенні нових математичних і програмних засобів, здатних подолати окреслені обмеження та забезпечити якісно новий рівень планування і прогнозування наукової діяльності.

1.3. Аналіз методів прогнозування показників наукової діяльності

У контексті автоматизації збору та аналізу наукової діяльності академічного колективу особливе значення набуває задача прогнозування ключових показників наукової діяльності. Ефективні підходи до прогнозування дозволяють не лише ретроспективно оцінити результати роботи дослідників, а й формувати обґрунтовані плани наукової діяльності на майбутні періоди. Це, своєю чергою, сприяє раціональному розподілу наукового навантаження, прийняттю управлінських рішень та узгодженню індивідуальних цілей із загальними стратегічними пріоритетами закладу вищої освіти.

Сучасні методи прогнозування показників наукової діяльності можна умовно класифікувати на 3 основні групи:

1) Класичні статистичні методи, до яких відносяться лінійна регресія, екстраполяція часових рядів, метод ковзних середніх, експоненціальне згладжування. Їх перевагами є простота реалізації, доступність математичного апарату та зрозумілість інтерпретації. Проте ці методи є чутливими до припущень про лінійність трендів і часто вимагають повного та репрезентативного масиву історичних даних. За наявності шуму, різких змін динаміки або зовнішніх факторів, які складно кількісно оцінити, точність прогнозів істотно знижується.

2) Методи машинного та глибинного навчання, до яких належать лінійна регресія, дерева рішень, випадкові ліси, градієнтний бустинг, опорні вектори (SVM), а також штучні нейронні мережі, забезпечують високу точність прогнозування за умови наявності достатнього обсягу якісно підготовлених даних. Ці методи здатні моделювати як лінійні, так і складні нелінійні залежності, виявляти приховані закономірності та враховувати вплив багатьох факторів одночасно. На відміну від класичних статистичних підходів, алгоритми машинного навчання мають здатність до самоадаптації — вони навчаються на історичних даних, постійно уточнюючи власні параметри. Глибинні нейронні мережі, зокрема рекурентні (RNN), довготривалої пам'яті (LSTM) та згорткові (CNN), відкривають можливість моделювання часових залежностей і тенденцій розвитку наукової діяльності. Водночас ці методи характеризуються високою обчислювальною складністю, потребують ретельної валідації та оптимізації гіперпараметрів, а результати їхньої роботи нерідко складно інтерпретувати через ефект так званої «чорної скриньки».

3) Інтервальні методи, які базуються на поданні вихідної інформації у вигляді інтервалів, є ефективними у випадках, коли точні значення показників є невідомими або варіюються в певних межах. Перевагою цього підходу є його здатність працювати з нечіткими або неповними даними, а також формувати прогнози у вигляді діапазонів, що враховують невизначеність і знижують ризик помилкових оцінок. Це особливо важливо в контексті планування наукової

діяльності, коли кількісні характеристики змінюються залежно від організаційних, кадрових або зовнішніх умов.

Подальший аналіз кожної з вказаних груп методів дозволить оцінити їхню ефективність, обмеження та доцільність застосування в рамках запропонованої системи підтримки прийняття рішень щодо планування наукової діяльності.

1.3.1. Статистичні методи

Статистичні методи прогнозування є одними з найдавніших і водночас найбільш формалізованих інструментів для оцінювання майбутніх значень показників на основі історичних даних. У контексті прогнозування наукової діяльності вони застосовуються для екстраполяції динаміки обсягу публікацій, участі у наукових заходах, отримання грантів та інших ключових параметрів, що характеризують активність дослідників.

Лінійна регресія

Лінійна регресія є одним із базових методів статистичного прогнозування, який дозволяє моделювати залежність однієї змінної (відгуку) від однієї або кількох незалежних змінних (предикторів) шляхом побудови лінійної функції. У загальному вигляді для однієї незалежної змінної модель має вигляд:

$$y = \beta_0 + \beta_1 x_1 + \varepsilon \quad (1.1)$$

де y – прогнозоване значення залежної змінної, x – незалежна змінна (наприклад, рік), β_0, β_1 – параметри моделі, а ε – випадкова похибка. У загальному випадку з кількома предикторами формується модель множинної лінійної регресії:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon \quad (1.2)$$

де x_1, x_2, \dots, x_n – значення незалежних змінних, що можуть описувати різні характеристики наукової діяльності (кількість публікацій, участь у конференціях, кількість проєктів тощо).

Коефіцієнти моделі обчислюються методом найменших квадратів — шляхом мінімізації суми квадратів відхилень між реальними й модельованими значеннями:

$$\min_{\beta_0, \beta_1, \dots, \beta_n} \sum_{i=1}^m \left(y_i - \left(\beta_0 + \sum_{j=1}^n \beta_j x_{ij} \right) \right)^2 \quad (1.3)$$

У контексті прогнозування наукової діяльності лінійну регресію доцільно застосовувати для оцінювання обсягу сумарного наукового навантаження в межах установленої нормативної величини — від 480 до 800 годин.

Метод ковзних середніх

Метод ковзних середніх (англ. *moving average*) належить до класичних методів згладжування часових рядів і використовується для виявлення основної тенденції шляхом усунення короткострокових випадкових коливань. Суть методу полягає в заміні кожного значення ряду середнім арифметичним кількох попередніх значень, об'єднаних у фіксоване вікно розміром k . Останнє усереднене значення може розглядатися як прогнозне для наступного періоду.

Математично просте ковзне середнє обчислюється за формулою:

$$\bar{y}_t = \frac{1}{k} \sum_{i=k+1}^t y_i \quad (1.4)$$

де \bar{y}_t — згладжене значення у момент часу t , k — довжина вікна усереднення, y_i — фактичні значення за попередні k періодів.

Метод екстраполяції

Метод екстраполяції належить до класичних статистичних підходів і передбачає продовження виявленої тенденції на підставі попередніх значень часових рядів. На відміну від регресійного моделювання, яке будує функціональну залежність між змінними, екстраполяція часто виконується на рівні ряду без побудови явної моделі. Її головне припущення — сталість динаміки зміни показника в майбутньому періоді [24].

Найпростішою формою є лінійна екстраполяція на основі різниці між останніми спостереженнями:

$$\hat{y}_{t+1} = y_t + (y_t - y_{t-1}) = 2y_t - y_{t-1} \quad (1.5)$$

де y_t – останнє фактичне значення, y_{t-1} – передостаннє, а \hat{y}_{t+1} – прогноз на наступний період.

Більш узагальнено можна використовувати середню швидкість приросту:

$$\Delta_y = \frac{y_t - y_{t-k}}{k}, \quad \hat{y}_{t+1} = y_t + \Delta_y \quad (1.6)$$

Даний підхід дає змогу екстраполювати тренд, зберігаючи його середню інтенсивність [77, 136].

Узагальнюючи, можна зробити висновок, що всі три статистичні методи можуть бути використані для прогнозування сумарного наукового навантаження, однак їхня придатність залежить від характеру вхідних даних, наявності трендів, рівня флуктуацій і цілей прогнозування. Проте, для забезпечення високої достовірності прогнозу необхідно застосовувати значні вибірки даних, що не реально реалізувати у розрізі значної кількості років. До тогож такі вибірки на тривалий період будуть неоднорідним, що знижує адекватність та точність прогнозу [85, 108, 115].

У практичних системах підтримки прийняття рішень доцільно використовувати статистичні методи як базовий рівень оцінки, результати якого можуть бути уточнені або скориговані за допомогою складніших підходів, зокрема машинного навчання або інтервального аналізу, що будуть розглянуті у наступних підрозділах.

1.3.2. Методи машинного навчання

Методи машинного навчання (МН) представляють собою клас алгоритмів, що дозволяють моделювати складні залежності між змінними на основі

емпіричних даних без необхідності формулювання явних аналітичних залежностей. У контексті прогнозування наукової активності машинне навчання відкриває нові можливості для врахування великої кількості факторів, виявлення прихованих закономірностей, обробки нечітких та неповних даних, а також адаптації до зміни динаміки показників у часі.

На відміну від класичних статистичних підходів, моделі МН здатні враховувати як лінійні, так і нелінійні взаємозв'язки, взаємодії між змінними, а також автоматично виявляти ознаки, що мають найбільший вплив на результат.

Лінійна регресія з регуляризацією (Ridge, Lasso)

Лінійна регресія є одним із найбільш базових методів у машинному навчанні, що широко використовується для задач прогнозування. Однак, у випадку роботи з невеликим обсягом навчальних даних, високою мультиколінеарністю ознак або великою кількістю нерелевантних змінних, класична реалізація цього методу може демонструвати нестабільність та схильність до перенавчання. Для подолання цих проблем застосовуються регуляризовані модифікації лінійної регресії, серед яких найбільш поширеними є Ridge-регресія (L2-регуляризація) та Lasso-регресія (L1-регуляризація).

Регуляризація полягає у введенні додаткового штрафу до функції втрат, що дозволяє обмежити величини коефіцієнтів моделі та, відповідно, підвищити її узагальнюючу здатність. Формально, у випадку Ridge-регресії мінімізується функція:

$$J(\beta) = \sum_{i=1}^m (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^n \beta_j^2 \quad (1.7)$$

де $\lambda \geq 0$ – параметр регуляризації, що визначає силу штрафу, β_j – вагові коефіцієнти моделі. У випадку Lasso-регресії другий доданок змінюється на $\lambda \sum_{j=1}^n |\beta_j|$, що сприяє нульовим оцінкам частини коефіцієнтів і, таким чином, виконує автоматичний вибір ознак.

Ансамблеві методи: дерева рішень, Random Forest

Методи, засновані на деревоподібних структурах, зокрема дерева рішень (Decision Trees), є інтуїтивно зрозумілими та широко використовуваними у задачах регресії та класифікації. У своїй основі дерево рішень буде послідовність бінарних розгалужень, кожне з яких відповідає перевірці певної умови щодо значення однієї з ознак. Таким чином, простір ознак рекурсивно розбивається на підмножини, в межах яких цільова змінна є відносно однорідною. Це дає змогу моделювати нелінійні залежності та взаємодії між ознаками без необхідності їх явного задання.

Попри високу інтерпретованість, окреме дерево рішень схильне до перенавчання, особливо в умовах невеликих вибірок. Для подолання цієї проблеми використовуються ансамблеві методи, зокрема випадкові ліси (Random Forest) [49, 97, 105, 122]. Цей підхід полягає у побудові великої кількості дерев рішень, кожне з яких навчається на випадковій підмножині даних та ознак. Підсумковий прогноз формується як середнє значення прогнозів усіх дерев, що значно зменшує варіативність результатів і підвищує стійкість моделі.

Математично модель випадкового лісу визначається як ансамбль з T незалежно побудованих дерев рішень $\{h_t(x)\}_t^T$, кожне з яких прогнозує значення цільової змінної y для заданого вектора ознак $x \in \mathbb{R}^n$. Кінцевий прогноз визначається як середнє арифметичне прогнозів усіх дерев:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(x) \quad (1.8)$$

де $h_t(x)$ – вихід t -го дерева для прикладу x , а $\hat{y}(x)$ – фінальний прогноз моделі. Кожне дерево формує розбиття простору ознак на області з локальною апроксимацією функції $y(x)$, що забезпечує гнучке моделювання складних функціональних залежностей.

Радіально-базисні функції (RBF)

Мережі на основі радіально-базисних функцій (Radial Basis Function Networks, RBF) становлять клас нейронних мереж, які особливо ефективні у

задачах апроксимації функцій та регресії. Вони поєднують у собі властивості локальної чутливості до вхідних змін і здатності моделювати складні нелінійні залежності [143].

Модель RBF складається з трьох шарів: вхідного, прихованого та вихідного. Кожен нейрон прихованого шару реалізує **радіально-базисну функцію**, зазвичай гаусову, що оцінює відстань між вхідним вектором $x \in \mathbb{R}^n$ і центром $c_j \in \mathbb{R}^n$:

$$\varphi_j(x) = \exp\left(-\frac{\|x-c_j\|^2}{2\sigma_j^2}\right) \quad (1.9)$$

де c_j – ширина функції (розмір області впливу), а $\varphi_j(x)$ – вихід j -го базисного нейрона.

Мережа формує прогноз як лінійну комбінацію виходів RBF-функцій:

$$\hat{y}(x) = \sum_{j=1}^M \omega_j \cdot \varphi_j(x) \quad (1.10)$$

де $\omega_j \in \mathbb{R}$ – вагові коефіцієнти, а M – кількість базисних функцій. Навчання моделі зводиться до визначення параметрів c_j, σ_j і ω_j на основі тренувального набору.

Градiєнтний бустинг

Градiєнтний бустинг (Gradient Boosting Machine, GBM) належить до класу ансамблевих методів машинного навчання, що базуються на принципі послідовного побудування слабких моделей (зазвичай дерев рішень) із подальшим поетапним удосконаленням прогнозу за рахунок мінімізації похибки попередніх ітерацій [107, 127]. Ідея методу полягає в тому, щоб об'єднати велику кількість простих моделей, кожна з яких по черзі компенсує помилки попередніх, унаслідок чого формується потужна композиція, здатна відтворювати складні нелінійні залежності.

З формальної точки зору, нехай маємо набір навчальних даних

$$D = \{(x_i, y_i)\}_{i=1}^n, \quad (1.11)$$

x_i – вектор ознак (наприклад, кількість публікацій, участь у конференціях, обсяг проєктної діяльності тощо), а y_i – цільова змінна, що описує сумарне наукове навантаження.

Мета методу — знайти функцію $F(x)$, яка мінімізує задану функцію втрат та наближується у вигляді послідовної суми слабких моделей (базових предикторів):

$$F_M(x) = F_0(x) + \sum_{m=1}^M \gamma_m h_m(x), \quad (1.12)$$

де $F_0(x)$ – початкова модель (наприклад, середнє значення цільової змінної), $h_m(x)$ – слабка модель, побудована на m -й ітерації (зазвичай дерево рішень), а γ_m – коефіцієнт, який визначає вагу внеску поточного предиктора.

На кожному кроці алгоритм обчислює градієнт функції втрат за попереднім прогнозом і навчає нове дерево $h_m(x)$, яке апроксимує цей градієнт, тобто напрямок, у якому слід покращити прогноз. Таким чином, кожна наступна модель «виправляє» помилки попередніх.

У задачах прогнозування показників наукової діяльності це означає, що перше дерево може оцінювати загальний тренд за кількістю публікацій, друге — коригувати прогноз з урахуванням грантової активності, третє — уточнювати вплив участі у конференціях, тощо. У результаті формується узгоджена модель, здатна враховувати складні взаємозв'язки між численними показниками.

Удосконалені варіації градієнтного бустингу — такі як XGBoost (Extreme Gradient Boosting), LightGBM або CatBoost — використовують регуляризацію, стохастичний відбір ознак і оптимізовані алгоритми обчислення градієнтів, що дозволяє підвищити швидкодію та зменшити ризик перенавчання.

Завдяки цьому такі моделі добре масштабуються та забезпечують високу точність прогнозів навіть за складних, нелінійних залежностей між параметрами.

Метод опорних векторів (Support Vector Regression, SVR)

Метод опорних векторів належить до класу алгоритмів машинного навчання, що базуються на побудові оптимальної гіперплощини, яка описує залежність між вхідними ознаками та цільовим показником. У випадку регресійного аналізу цей підхід відомий як Support Vector Regression (SVR) [70, 84, 127]. Його основна ідея полягає в пошуку функції $f(x)$, яка з певною точністю апроксимує цільові значення y_i , забезпечуючи при цьому максимальну «плоскість» гіперплощини. Це означає, що алгоритм прагне мінімізувати складність моделі, допускаючи незначні похибки в межах допустимого інтервалу ε .

Формально задача оптимізації в SVR визначається як мінімізація виразу:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*), \quad (1.13)$$

за умов:

$$\begin{cases} y_i - (w^T \phi(x_i) + b) \leq \varepsilon + \xi_i \\ (w^T \phi(x_i) + b) - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (1.14)$$

де w — вектор вагових коефіцієнтів, b — зміщення, C — параметр, що контролює компроміс між точністю апроксимації та складністю моделі, а ξ_i, ξ_i^* — змінні, які визначають порушення меж ε -інсенситивної зони.

Розв'язок задачі визначається лише для частини навчальних прикладів, які отримали ненульові значення множників Лагранжа. Ці приклади формують так звані опорні вектори, що безпосередньо впливають на положення гіперплощини. Завдяки цьому SVR відзначається високою стійкістю до шумів і викидів у даних,

оскільки більшість спостережень, що лежать усередині ε -інсенситивної зони, не впливають на навчання моделі.

Для моделювання нелінійних залежностей використовується механізм **ядерних функцій** $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, який неявно відображає початковий простір ознак у простір вищої розмірності. Вибір типу ядра визначає характер моделі: лінійне ядро доцільно застосовувати для задач із монотонним трендом, тоді як радіально-базисна функція (RBF) ефективна для апроксимації складних нелінійних залежностей, але менш придатна до екстраполяції за межами навчального діапазону [74].

У контексті прогнозування показників наукової діяльності метод опорних векторів дозволяє відтворювати як лінійні, так і слабко нелінійні залежності між показниками публікаційної, проєктної та експертної активності науковців. Для невеликих вибірок цей метод є особливо ефективним завдяки регуляризаційним властивостям, що запобігають перенавчанню та забезпечують стабільність прогнозу [128].

Рекурентні нейронні мережі (Recurrent Neural Networks, RNN)

Рекурентні нейронні мережі належать до класу глибинних моделей, призначених для обробки послідовних даних, у яких поточний стан залежить від попередніх спостережень. На відміну від класичних багатошарових нейронних мереж, що розглядають усі вхідні ознаки незалежно, RNN використовують механізм зворотних зв'язків, який забезпечує пам'ять про попередні елементи послідовності. Це робить їх особливо ефективними для аналізу часових рядів, де важливим є врахування динаміки змін у часі [80, 138].

Основна ідея RNN полягає у рекурентному оновленні прихованого стану мережі, який на кожному кроці зберігає інформацію про попередні вхідні значення. Для кожного моменту часу t вхідний вектор x_t разом із попереднім прихованим станом h_{t-1} використовується для обчислення нового стану h_t :

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \quad (1.15)$$

де W_{xh} – матриця ваг для вхідного шару, W_{hh} – матриця рекурентних ваг, b_h – вектор зсуву, а f – нелінійна функція активації.

Вихід моделі на кроці t обчислюються як:

$$y_t = g(W_{hy}h_y + b_y), \quad (1.16)$$

де W_{hy} – матриця ваг між прихованим і вихідним шарами, а g – функція активації виходу.

Такий підхід дає змогу моделі накопичувати інформацію про попередні стани та враховувати контекст при формуванні прогнозу. Однак традиційні RNN мають обмеження, пов'язані з проблемою зникання або вибуху градієнтів під час навчання на довгих послідовностях. Через це їхня здатність утримувати довготривалу залежність між спостереженнями є обмеженою.

Для подолання цієї проблеми розроблено модифіковані архітектури рекурентних мереж — LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit), які завдяки спеціальним коміркам пам'яті та механізмам керування потоками інформації (вхідні, вихідні та забувальні ворота) здатні ефективно зберігати інформацію на тривалих часових інтервалах [146].

У контексті прогнозування наукової діяльності рекурентні нейронні мережі дозволяють аналізувати історичну динаміку показників — кількість публікацій, участь у конференціях, виконання проєктів, експертну активність — і будувати моделі, які враховують як короткострокові коливання, так і довготривалі тенденції. Застосування RNN забезпечує можливість прогнозування сумарного наукового навантаження на основі попередніх років, відображаючи закономірності розвитку наукової активності у часі [80].

Значною проблемою використання методів машинного навчання складність формування вибірки даних: у багатьох випадках попередні результати не були накопичені в цифрових форматах, а молоді науково-педагогічні працівники, або ті що працюють короткий термін в університеті

взагалі не мають достатнього обсягу історичних даних для формування репрезентативних прогнозів.

1.3.3. Інтервальні методи прогнозування

Інтервальні методи прогнозування є відносно новим і перспективним підходом до задач аналізу наукової діяльності в умовах обмеженості або невизначеності вихідних даних. Їх ключова особливість полягає в тому, що вхідні значення параметрів подаються не як точкові оцінки, а у вигляді інтервалів, які відображають можливі коливання, похибки вимірювання або неповноту інформації.

На відміну від класичних статистичних та машинних методів, що потребують повних і детермінованих даних, інтервальний підхід дозволяє будувати прогнози навіть у випадках, коли історична інформація є неповною або неточною. Це особливо актуально для задач прогнозування наукової активності, де значна частина параметрів (наприклад, участь у проєктах, виконання грантів, отримання патентів) може бути представлена з часовими затримками, в неповному обсязі або в нечіткій формі.

Застосування інтервального підходу дозволяє не лише отримати орієнтовне значення майбутнього наукового навантаження, а й визначити діапазон можливих варіацій, що є важливим для планування діяльності в умовах невизначеності. Прогноз у цьому випадку формується у вигляді інтервалу, що охоплює найгірший та найкращий можливий сценарій, з урахуванням усіх невизначеностей у вихідних параметрах [28, 69, 147].

У рамках даного дослідження інтервальні методи розглядаються як один із ключових підходів до вирішення задачі прогнозування сумарного наукового навантаження. Їх докладний опис, математичне обґрунтування, приклади реалізації та оцінка ефективності будуть наведені у розділі 2, який повністю присвячено побудові інтервальної моделі прогнозування та аналізу її переваг у порівнянні з іншими підходами.

1.4. Постановка завдань дисертаційного дослідження

Проведений аналіз існуючих систем для планування та обліку наукової діяльності науково-педагогічних працівників засвідчив наявність низки проблем, що обмежують їхню ефективність і знижують практичну користь для розвитку наукового потенціалу закладів вищої освіти. З одного боку, вітчизняні рішення орієнтовані переважно на централізований збір звітних даних і виконання адміністративних процедур, що дозволяє державним органам здійснювати контроль та формувати політику у сфері науки. З іншого боку, зарубіжні системи, хоча й значно більш розвинуті, мають акцент на облікових, аналітичних та інтеграційних функціях і також лише частково охоплюють завдання прогнозування наукової діяльності.

Серед виявлених проблем варто виокремити такі: відсутність уніфікованих стандартів інтеграції даних у межах національних наукових систем; недостатній рівень автоматизації збору інформації з міжнародних наукометричних баз; обмежені можливості для формування персоналізованих профілів дослідників та врахування динаміки їхньої продуктивності; переважна орієнтація на фіксацію результатів замість підтримки процесів планування та управління науковим розвитком.

Найбільш критичною проблемою, що простежується як у вітчизняному, так і в зарубіжному досвіді, є відсутність функціоналу для прогнозування наукової діяльності науково-педагогічних працівників. Існуючі системи здебільшого спрямовані на ретроспективний облік і статистичний аналіз уже отриманих результатів, тоді як завдання передбачення майбутнього наукового навантаження, оцінки потенціалу дослідника та моделювання сценаріїв його діяльності залишаються поза їхніми можливостями. Це створює суттєвий розрив між наявними інформаційними інструментами та потребами управління наукою в умовах зростаючої конкуренції й необхідності стратегічного планування.

Проведений аналіз сучасних методів прогнозування показників наукової діяльності показав, що їх ефективність суттєво залежить від обсягу та структури вихідних даних. Статистичні методи (лінійна регресія, метод ковзного

середнього, експоненційне згладжування) забезпечують задовільні результати лише за наявності стабільних часових рядів. Методи машинного навчання (Random Forest, Gradient Boosting, Support Vector Regression) дають змогу досягати високої точності, проте вимагають значного обсягу історичних даних і суттєвих обчислювальних ресурсів. Моделі глибокого навчання, зокрема рекурентні нейронні мережі типу LSTM, здатні враховувати часові залежності, але їх застосування також обмежується невеликим обсягом вибірки, характерним для даних про індивідуальну наукову активність.

З огляду на це, найбільш доцільним підходом для вирішення завдання прогнозування наукової діяльності є інтервальний підхід, який забезпечує стійкість до неповноти, шумів і невизначеності даних. На відміну від класичних моделей, інтервальні методи дозволяють формувати прогноз у вигляді діапазону можливих значень показників, що дає змогу враховувати варіації у темпах наукової активності та підвищує достовірність результатів. Такий підхід особливо ефективний за умов обмеженого набору історичних даних, що притаманно локальним академічним системам моніторингу. У зв'язку з цим у дисертаційному дослідженні ставляться такі основні завдання: розробка математичних та програмних засобів, які забезпечують прогнозування наукової активності науково-педагогічних працівників; створення методів інтеграції й аналізу даних з різних джерел наукометричної інформації; побудова моделей, що дозволять не лише фіксувати результати наукової діяльності, а й визначати її перспективні напрями з урахуванням індивідуальних особливостей дослідників. Реалізація цих завдань має на меті подолати обмеження існуючих систем та закласти підґрунтя для створення інтелектуалізованих інформаційних технологій у сфері управління науковою діяльністю.

Таким чином на підставі проведеного аналізу можемо констатувати, що метою дисертаційної роботи є розробка математичного та програмного забезпечення для системи планування наукової діяльності науково-педагогічних працівників з урахуванням індивідуального профілю дослідника та обмеженого обсягу історичних даних на основі методів інтервального аналізу даних.

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

- 1) Провести аналіз сучасних систем для планування наукової діяльності;
- 2) провести аналіз методів та засобів збору наукової діяльності;
- 3) провести аналіз методів прогнозування показників наукової діяльності в умовах обмеженої вибірок історичних даних;
- 4) побудувати ієрархічну систему показників наукової активності;
- 5) розробити математичну модель прогнозування сумарного наукового навантаження науково-педагогічних працівників із використанням інтервального аналізу даних;
- б) створити архітектуру та реалізувати прикладне програмне забезпечення для планування наукової діяльності науково-педагогічних працівників;
- 7) провести апробацію розробленої системи у реальних умовах діяльності закладу вищої освіти

Висновки до розділу 1

1. Проаналізовано сучасні підходи до організації планування та обліку наукової діяльності науково-педагогічних працівників у вітчизняних і зарубіжних закладах вищої освіти. Встановлено, що українська практика ґрунтується переважно на централізованому зборі звітних даних та адміністративному контролі. Єдина інформаційна система «Наука» і локальні університетські модулі виконують важливу роль у формуванні національного інформаційного простору, проте їхній функціонал зосереджений здебільшого на фіксації результатів і не охоплює завдань прогнозування чи персоналізованого планування наукового розвитку.

2. Проведено огляд і порівняльний аналіз зарубіжного досвіду управління науковою діяльністю. Установлено, що CRIS- та RIMS-платформи (Elsevier Pure, Converis, Symplectic Elements, VIVO, DSpace-CRIS) забезпечують високий рівень цифрової зрілості, інтеграцію з міжнародними базами даних і формування розширених профілів дослідників. Такі системи поєднують збір, зберігання та аналітику даних, однак у більшості випадків залишаються орієнтованими на ретроспективний облік і лише частково реалізують функції прогнозування.

3. Виявлено основні обмеження існуючих систем планування наукової діяльності. До ключових проблем належать неповнота вибірок, відсутність єдиних ідентифікаторів авторів, дублювання записів, обмежений доступ до наукометричних баз і низький рівень інтеграції між окремими управлінськими підсистемами університетів. У результаті більшість наявних рішень не забезпечують комплексного підходу до планування та прогнозування наукової активності.

4. На основі проведеного аналізу здійснено систематизацію результатів і визначено ключові напрями подальшого розвитку систем управління науковою діяльністю. Встановлено, що ефективне планування та прогнозування можливе лише за умови комплексного поєднання автоматизованого збору, багаторівневої

обробки та аналітичного опрацювання даних. До пріоритетних напрямів віднесено інтеграцію даних з різнорідних джерел у єдине інформаційне середовище, удосконалення моделей оцінювання наукової активності з урахуванням якісних показників і контексту дослідницької роботи, а також розробку програмних інструментів, що забезпечують персоналізоване прогнозування наукового навантаження на основі інтелектуальних методів аналізу даних.

РОЗДІЛ 2

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ЗБОРУ ДАНИХ ПРО НАУКОВУ ДІЯЛЬНІСТЬ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ

Ефективне управління науковою діяльністю в закладах вищої освіти ґрунтується на наявності достовірної, повної та структурованої інформації про результати науково-дослідної роботи науково-педагогічних працівників. Саме якість вихідних даних визначає можливість формування об'єктивних аналітичних показників, забезпечує коректність оцінювання наукової продуктивності та є ключовою передумовою побудови прогнозних моделей наукової активності. Актуальність, узгодженість і семантична єдність відомостей безпосередньо впливають на точність подальших аналітичних висновків і управлінських рішень.

Аналіз існуючих інформаційних систем, проведений у попередньому розділі, засвідчив, що більшість з них зорієнтовані переважно на фіксацію вже отриманих результатів наукової діяльності. При цьому питання комплексного збору, стандартизації, верифікації, семантичного узгодження та автоматизованої інтеграції даних залишаються значною мірою невирішеними. Як наслідок, інформація про публікаційну, проєктну, грантову, патентну чи експертну активність дослідників нерідко є фрагментарною, дубльованою або поданою у несумісних структурних форматах, що суттєво ускладнює її об'єднання в єдину інформаційну інфраструктуру університету.

Подолання цих проблем вимагає переходу від традиційних, переважно ручних або частково автоматизованих форм обліку до впровадження цілісного інформаційного забезпечення, яке поєднує стандартизовані моделі представлення наукових даних, автоматизовані методи їх отримання та багаторівневі механізми інтеграції з внутрішніми й зовнішніми джерелами. Важливу роль у цьому процесі відіграють міжнародні стандарти метаданих і відкриті протоколи обміну інформацією (CERIF, Dublin Core, DataCite, OAI-

PMH), а також системи унікальної ідентифікації (DOI, ORCID, ROR), які забезпечують міжсистемну взаємодію та підвищують достовірність і точність відтворення наукової інформації.

Разом із тим ефективна система збору наукових даних повинна поєднувати різні технологічні підходи, зокрема отримання даних через офіційні API міжнародних платформ, harvesting за протоколом OAI-PMH для інституційних репозитаріїв та методи веб-парсингу для джерел, що не підтримують стандартизованих форматів метаданих. Комплексне застосування таких методів дає змогу формувати повний, узгоджений та актуальний масив даних про всі види наукової активності.

У цьому розділі розглянуто засади побудови інформаційного забезпечення процесів збору, структуризації та інтеграції наукових даних. Послідовний аналіз охоплює міжнародні стандарти і концептуальні принципи, що визначають вимоги до структури й семантичної узгодженості інформації; модель інформаційного забезпечення, що описує логіку побудови комплексної інфраструктури; а також методи отримання наукових даних, включаючи інтеграцію через програмні інтерфейси, автоматизовані протоколи harvesting та веб-парсинг. Завершальна частина розділу присвячена архітектурі та ETL-процесу, який забезпечує поетапне вилучення, очищення, нормалізацію, семантичне узгодження та консолідацію різномірних даних у єдиному структурованому сховищі, що слугує основою для подальших аналітичних і прогностичних досліджень.

2.1. Стандарти та принципи опису наукових даних в інформаційному забезпеченні збору наукової діяльності

Ефективне інформаційне забезпечення процесів збору результатів наукової діяльності ґрунтується на принципах структурованості, достовірності та семантичної узгодженості даних, які становлять основу для формування аналітичних і прогностичних моделей у системах управління наукою. У

сучасних умовах значних обсягів інформації, що генерується науковцями, а також швидких темпів розвитку цифрових технологій саме стандартизація метаданих стає ключовим механізмом, який забезпечує сумісність інформаційних ресурсів, можливість їх інтеграції та повторного використання. Застосування усталених міжнародних стандартів гарантує, що дані, отримані з різних джерел, можуть бути інтерпретовані ідентичним чином незалежно від програмного середовища чи країни походження.

Наукові результати характеризуються значним різноманіттям об'єктів опису, серед яких публікації, проекти, патенти, гранти, звіти про участь у наукових подіях, а також показники експертної, рецензентської та організаційної активності. Кожен із цих видів діяльності має власну логіку формування метаданих. Наприклад, наукові публікації містять інформацію про авторів, джерело опублікування, DOI, а також про зміст, тематику, ключові слова, індекси цитування. У свою чергу, проекти визначаються бюджетом, термінами виконання, складом виконавців і джерелами фінансування. Така різноманітність ускладнює побудову універсальних моделей даних, а відтак потребує використання стандартизованих формалізованих схем, які б узгоджували структуру інформації та забезпечували її коректне відтворення в автоматизованих системах [59, 61].

Одним із базових стандартів загального призначення є Dublin Core [106], який містить мінімально необхідний набір елементів метаданих для опису цифрових ресурсів, включно з науковими. Його структура, подана на рисунку 2.1, включає такі ключові елементи, як назва, автор, тема, опис, мова, дата, видавець та ідентифікатор. Простота Dublin Core стала підґрунтям його широкої підтримки в інституційних репозитаріях, однак обмеженість кількості елементів створює труднощі у випадках, коли виникає потреба в детальному описі об'єктів наукової діяльності, зокрема багатокомпонентних дослідницьких проектів або пов'язаних між собою наборів результатів.

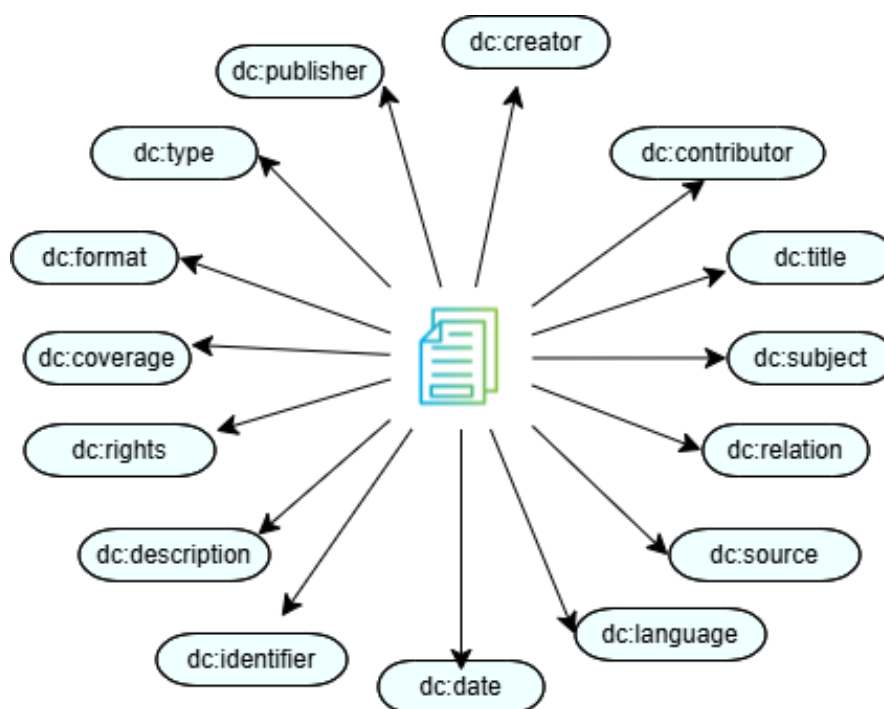


Рисунок 2.1 – Представлення Dublin Core стандарту

Для вирішення подібних завдань використовується розширеніший та функціонально складніший стандарт CERIF (Common European Research Information Format) [27]. Його структура, відображена на рисунку 2.2, базується на реляційній моделі, яка охоплює численні сутності: особи, організації, проекти, результати досліджень, гранти, інфраструктуру та інші елементи наукової екосистеми. CERIF дозволяє не лише описати окремий об'єкт, але й зафіксувати множинні взаємозв'язки між цими об'єктами, що є ключовим для побудови комплексних наукових інформаційних систем. Саме тому CERIF визнано європейським стандартом для систем CRIS (Current Research Information Systems), які широко використовуються у провідних університетах та дослідницьких установах.

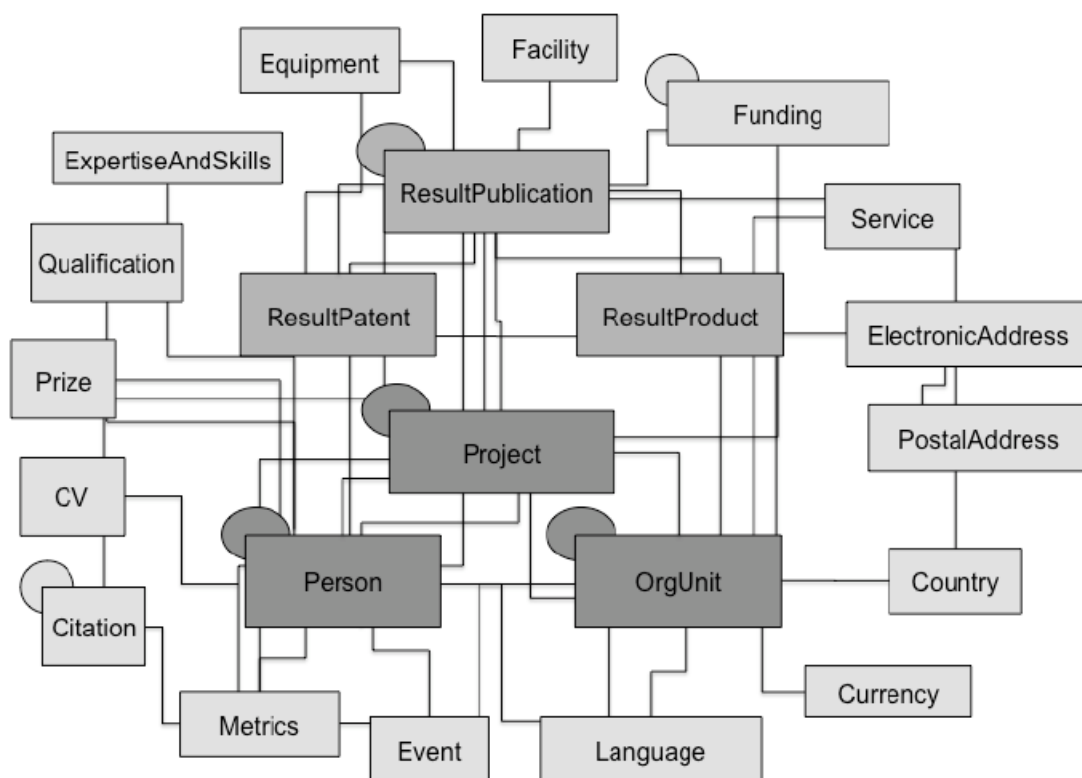


Рисунок 2.2 – Структура стандарту CERIF в абстрактному вигляді та зв'язками

У контексті інтеграції даних з різних інформаційних платформ особливого значення набуває протокол OAI-PMH, який забезпечує автоматизоване збирання структурованих метаданих із репозитаріїв та інших наукових сховищ [54, 66, 87]. На рисунку 2.3 подано загальну архітектуру цього протоколу, який дозволяє регулярно обмінюватися метаданими між різними системами та відтворювати єдине інформаційне середовище. Застосування OAI-PMH є критичним для побудови національних і міжнародних агрегаторів наукових даних, таких як OpenAIRE або CORE, оскільки він забезпечує сумісність між різними реалізаціями репозитаріїв [65, 109].

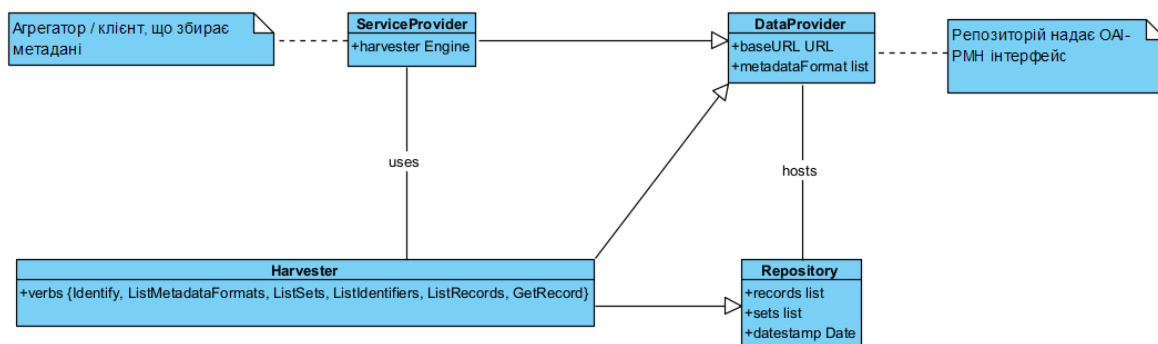


Рисунок 2.3 – Архітектура обміну метаданими за протоколом OAI-PMH

Для опису об'єктів, пов'язаних з дослідницькими даними, наборів експериментальних чи емпіричних результатів, широко застосовується DataCite Metadata Schema [50], секторальну структуру якої подано на рисунку 2.4. DataCite підтримує деталізований опис таких об'єктів, включно з інформацією про фондівську підтримку, типи ідентифікаторів пов'язаних об'єктів, типи внеску авторів та структуру даних. Цей стандарт відіграє ключову роль у формуванні інфраструктури FAIR-даних (Findable, Accessible, Interoperable, Reusable), яка є основою сучасної відкритої науки..

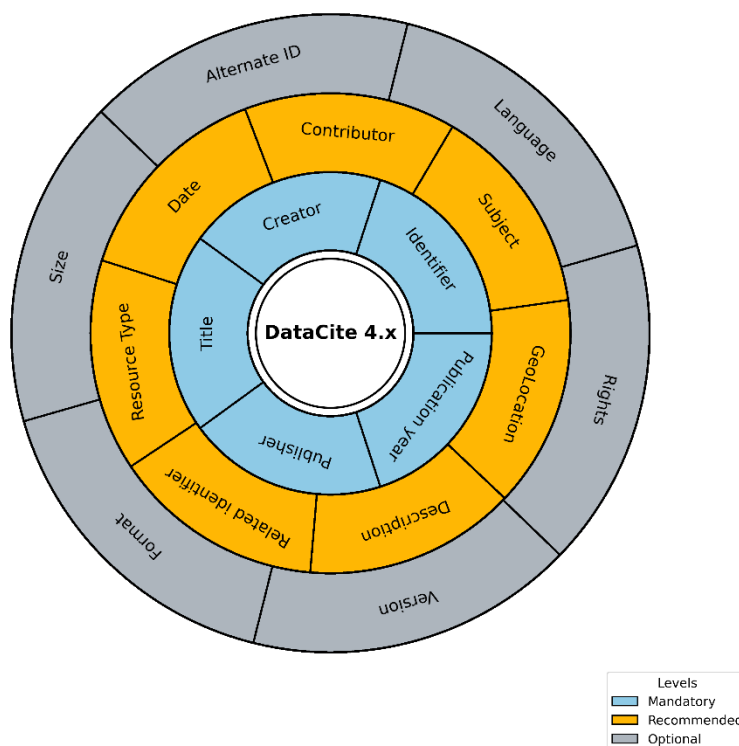


Рисунок 2.4 – Секторальна діаграма структури схеми метаданих DataCite

Поряд із наведеними стандартами значну роль відіграють і додаткові метадані-схеми та моделі опису, серед яких Crossref Metadata Schema [29], що забезпечує реєстрацію бібліографічних записів публікацій у міжнародному реєстрі DOI; JATS (Journal Article Tag Suite), що застосовується в автоматизованих видавничих процесах та дозволяє зберігати структуру повного тексту статті; Schema.org, який забезпечує семантичну розмітку наукових даних у веб-просторі; та OpenAIRE Guidelines, обов'язкові для більшості європейських репозитаріїв. Використання цих інструментів розширює можливості інтеграції міжсистемних ресурсів та підвищує якість представлення наукових результатів.

Не менш вагому роль відіграють системи унікальної ідентифікації, такі як DOI, ORCID, ROR які забезпечують однозначність визначення об'єктів та суб'єктів наукової діяльності. Наявність універсальних ідентифікаторів усуває проблему дублювання даних, полегшує збирання інформації з різних платформ та підвищує точність аналітичних моделей. У сучасних наукових екосистемах саме зв'язність даних, забезпечена унікальними ідентифікаторами, дозволяє автоматично формувати профілі дослідників і відтворювати історію їхньої наукової активності [71, 82, 123].

Попри широке поширення стандартів і протоколів на міжнародному рівні, їхнє впровадження в українських закладах вищої освіти відбувається нерівномірно. У багатьох установах наукові дані зберігаються у формі ізольованих документів, електронних таблиць або неструктурованих форматів, що ускладнює автоматизацію процесів збору та інтеграції. Відсутність єдиної політики стандартизації та обмежене використання унікальних ідентифікаторів створюють суттєві перешкоди для побудови комплексних систем наукової аналітики.

З огляду на зазначене, стандарти та принципи опису наукових даних виступають базовим елементом сучасного інформаційного забезпечення збору наукової діяльності. Саме вони формують фундамент для подальшого впровадження автоматизованих методів збору даних, інтеграції з міжнародними

системами та розроблення інтелектуальних інструментів аналізу наукової діяльності, які розглядатимуться у наступних підрозділах.

2.2. Інформаційна технологія збору наукової діяльності НПП

Запропонована інформаційна технологія збору наукових даних спрямована на забезпечення комплексного, системного та автоматизованого опрацювання інформації про результати наукової діяльності науково-педагогічних працівників. Її побудова ґрунтується на потребі усунути фрагментарність, неузгодженість та надмірну залежність від ручного введення даних, які характерні для традиційних форм звітності у закладах вищої освіти. Технологія створює умови для безперервного оновлення інформаційних ресурсів, підтримуючи актуальність і достовірність відомостей, необхідних як для внутрішнього управління, так і для аналітичних, оціночних та прогностичних моделей.

Першим етапом є вилучення метаданих із внутрішніх та зовнішніх джерел, що різняться структурою, доступністю та рівнем формалізації. Використання комбінованого підходу до збору дозволяє охопити широкий спектр інформаційних ресурсів, включно з міжнародними бібліографічними базами, національними реєстрами, інституційними репозитаріями та веб-ресурсами наукових проєктів [98, 110, 114, 130]. API-інтерфейси дають можливість отримувати високоструктуровані дані з Crossref, DataCite, OpenAlex та інших систем. Протокол OAI-PMH забезпечує harvesting уніфікованих наборів метаданих з університетських репозитаріїв. Веб-парсинг використовується як додатковий інструмент для вилучення інформації з джерел, що не підтримують формальних стандартів обміну [30, 118]. Така комбінація методів дозволяє уникнути втрати даних, охопити максимально можливий масив інформації та забезпечити регулярність її оновлення. Технічним засобом реалізації цього етапу є багатокomпонентний модуль збору даних, що координує взаємодію з API,

генерує та опрацьовує запити OAI-PMH, а також виконує оброблення HTML-структур.

Другим етапом є узгодження та нормалізація отриманих відомостей. На цьому рівні відбувається системне усунення типових невідповідностей, таких як дублювання записів, різночитання імен авторів, нерелевантні або неповні атрибути та відхилення від стандартних структур даних. Нормалізація враховує специфіку кожного джерела, оскільки навіть у межах однієї платформи можуть існувати відмінності у форматі публікацій, метаданих чи способах зазначення співавторів. Технологія використовує методи семантичного зіставлення, алгоритми текстового узгодження та ідентифікацію об'єктів за DOI, ORCID, ROR та іншими унікальними ідентифікаторами. Даний етап забезпечує перехід від «сирих» зібраних даних до уніфікованих структур, які відповідають вимогам форматів CERIF та DataCite. Технічним засобом є модуль нормалізації та семантичного узгодження, що виконує автоматизовану перевірку записів, визначає повтори, коригує атрибути та формує узгоджену структуру метаданих.

Після нормалізації відбувається сегментація даних відповідно до видів наукової діяльності. Оскільки метадані, що описують публікації, проекти, патенти, гранти чи конференційні виступи, мають принципові відмінності, їхня одночасна обробка уніфікованим способом є неможливою. Для забезпечення коректної подальшої інтеграції система групує об'єкти за їхньою природою, відокремлюючи публікаційну активність від проектною чи експертною. На цьому етапі застосовуються алгоритми автоматичної класифікації, структурного аналізу записів та контекстного визначення типу документа. Завдяки цьому забезпечується предметна впорядкованість інформації, що сприяє точності подальшого аналізу та формування зв'язків між об'єктами різних категорій.

Інтеграція наукових даних є наступним ключовим етапом технології. Вона полягає у встановленні логічних і структурних зв'язків між усіма зібраними та узгодженими об'єктами, що дозволяє формувати багатовимірні моделі діяльності окремих науковців і наукових підрозділів. Інтеграція охоплює поєднання інформації про публікації з даними про авторів, зв'язування грантів і

проектів з учасниками, зіставлення патентів з організаціями-замовниками, визначення спільних наукових напрямів тощо. У процесі інтеграції використовуються методи встановлення відповідностей, аналіз зв'язності, зіставлення за ідентифікаторами та контекстуальний аналіз метаданих. Технічний модуль інтеграції формує єдину структуру, у якій відображено всі взаємозв'язки між об'єктами, що дозволяє відтворити повну картину наукової активності та забезпечує можливість виконання складних аналітичних операцій.

Завершальним етапом технології є формування структурованих профілів науково-педагогічних працівників. У рамках цього етапу інтегровані дані перетворюються у завершену модель наукової діяльності кожного дослідника, яка включає публікації, проекти, участь у грантах, патентну активність, експертні функції та інші результати. Профіль містить як перелік наукових об'єктів, так і встановлені між ними зв'язки, що дозволяє відображати як кількісні, так і структурні характеристики наукової роботи. Формовані профілі забезпечують можливість проведення оцінювання, моніторингу, порівняльного аналізу та слугують вихідною платформою для прогнозування наукового навантаження, розглянутого у наступному розділі. Результатом етапу є інтегрований структурований опис наукової діяльності НПП, готовий до подальшого використання у системах аналітики та управління.

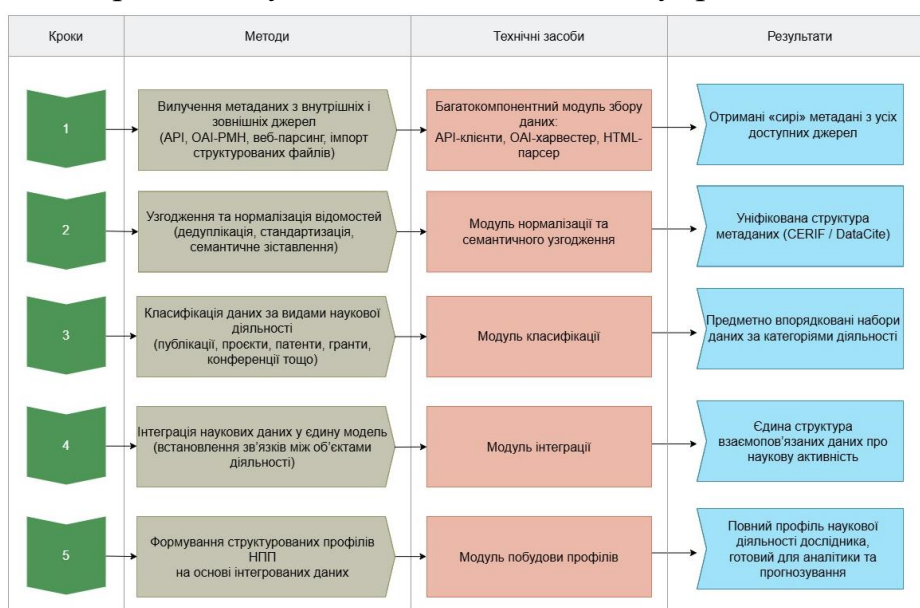


Рисунок 2.5 – Загальна схема процесу збору, інтеграції та структуризації даних про наукову діяльність НПП

Узагальнюючи наведене, запропонована інформаційна технологія охоплює повний цикл оброблення інформації: від її вилучення до інтегрованого представлення результатів. Вона забезпечує системну автоматизацію, опирається на міжнародні стандарти метаданих та поєднує методи збору, узгодження, сегментації й інтеграції в єдину цілісну модель. На відміну від існуючих систем, що орієнтовані переважно на фіксацію окремих результатів, дана технологія створює основу для побудови інтелектуалізованої інфраструктури підтримки прийняття рішень у сфері наукової діяльності університету.

2.3. Модуль семантичного впорядкування зібраної інформації про наукову діяльність

Комплексний збір даних про наукову діяльність охоплює публікації, проекти, гранти, патенти, дисертації, експертну та організаційну активність, а також різні форми участі у наукових подіях. Проте навіть після нормалізації та інтеграції ці дані можуть мати різний ступінь тематичної близькості до реального наукового профілю дослідника. Оскільки автоматичний збір інформації з широкого кола джерел неминуче включає записи суміжного, частково релевантного або формально пов'язаного характеру, постає завдання не жорсткого відсіювання, а семантичного ранжування всіх об'єктів наукової активності. Такий підхід дозволяє зберігати повноту даних, водночас забезпечуючи їх змістовне впорядкування відповідно до ключових напрямів роботи дослідника.

Модуль ранжування базується на використанні сучасних методів векторного подання текстів, що дозволяють формувати уніфіковане представлення змісту об'єктів незалежно від їхнього типу. Для цього застосовуються великі мовні моделі, здатні однаково якісно обробляти короткі анотації публікацій, описи наукових проектів, тематичні частини дисертацій, формулювання завдань грантів чи текстові звіти про наукову участь. Усі ці

об'єкти приводяться до векторної форми, після чого порівнюються з векторною репрезентацією профілю дослідника, сформованою на основі його попереднього доробку та домінантних наукових інтересів [21, 129, 131].

Подальша обробка зібраних матеріалів здійснюється за допомогою векторних ембеддингів, сформованих великими мовними моделями (LLM). Використання таких моделей дає змогу аналізувати зміст об'єктів наукової діяльності на рівні семантичної близькості, враховуючи контекст, латентні теми, синонімічні конструкції та понятійні зв'язки [53, 92]. Завдяки цьому модуль ранжування може не лише визначати формальну відповідність ключовим словам, а й оцінювати глибинну тематичну спорідненість об'єкта з науковим профілем дослідника. На основі сформованих ембеддингів реалізується алгоритм семантичного впорядкування, що включає такі основні етапи:

Етап 1. Збирання даних про всі об'єкти наукової діяльності.

На першому етапі здійснюється автоматизоване збирання даних про всі види наукової активності дослідника, що включають не лише публікації, але й наукові проєкти, гранти, патенти, дисертаційні дослідження, участь у конференціях, експертну та організаційну діяльність. Джерелами даних виступають міжнародні та національні наукометричні бази (Scopus, Web of Science, Crossref, OpenAlex), репозитарії університетів, реєстри наукових проєктів і грантів, офіційні каталоги патентів, а також внутрішні інформаційні системи закладу.

Для кожного типу об'єкта система вилучає найбільш репрезентативні метадані. Зокрема:

- для публікацій — назва, авторський склад, анотація, джерело опублікування, ключові слова, дата виходу;
- для наукових проєктів і грантів — опис мети та завдань, інформація про фінансування, виконавців, терміни реалізації;
- для патентів — формула винаходу, сфера застосування, опис технічного рішення, юридичні атрибути;

- для дисертаційних робіт — тема, анотація, науковий напрям, ключові терміни;
- для конференційної активності — назви доповідей, тематичні секції, тези виступів;
- для експертної діяльності — зміст експертних висновків, характеристика наукових задач, контекст оцінювання.

Отримані метадані подаються у стандартизованому структурованому вигляді (наприклад, JSON або XML), після чого зберігаються у відповідному сховищі даних для подальшого опрацювання в наступних етапах модуля семантичного впорядкування.

Етап 2. Формування векторних представлень (ембеддингів) об'єктів наукової діяльності.

Формування ембеддингів полягає в тому, щоб за допомогою великої мовної моделі (LLM) перетворити текстовий зміст об'єктів наукової діяльності у вектори високої розмірності, які відображають їхню семантичну структуру. Такий підхід дає можливість порівнювати між собою публікації, грантові заявки, описи наукових проєктів, патентні формулювання, дисертаційні фрагменти та інші текстові матеріали не лише за ключовими словами, а на рівні латентної семантичної близькості [21, 91, 135, 142].

Для кожного об'єкта o попередньо виділяються змістові фрагменти, які найбільш точно характеризують його науковий зміст. Це можуть бути анотації, постановки завдань, ключові слова, короткі описи проєкту, формула винаходу, тези виступу чи інші текстові елементи, залежно від типу об'єкта. У випадку великих документів (наприклад, повних статей, грантових заявок чи розділів дисертацій) текст розбивають на декілька блоків $Block_i(o)$, кожен із яких подається до мовної моделі як окреме вхідне повідомлення.

Мовна модель обчислює D-вимірний вектор-ембеддинг:

$$\varphi(Block_i(o))$$

для кожного такого блоку. Кінцеве векторне представлення об'єкта

$$\varphi(o)$$

отримують шляхом агрегації всіх ембеддингів. Найчастіше використовується середнє або зважене середнє:

$$\varphi(o) = \frac{1}{m} \sum_{i=1}^m \varphi(\text{Block}_i(o)) \quad (2.1)$$

де m — кількість блоків.

За потреби можуть застосовуватися вагові коефіцієнти ω_i , якщо окремі частини документа є більш інформативними (наприклад, анотація, формулювання мети проєкту або патентна формула):

$$\varphi(o) = \frac{\sum_{i=1}^m \omega_i \varphi(\text{Block}_i(o))}{\sum_{i=1}^m \omega_i} \quad (2.2)$$

Такий підхід дозволяє враховувати різний змістовий внесок окремих частин одного об'єкта.

Етап 3. Формування векторів інтересів дослідника

У контексті використання векторних ембеддингів великої мовної моделі вектор інтересів $\varphi(I)$ представляє собою узагальнений змістовий профіль дослідника, отриманий на основі всіх доступних текстових джерел, що відображають його попередню наукову діяльність. Цей вектор використовується не для фільтрації, а для семантичного порівняння та ранжування нових об'єктів наукової активності відповідно до їхньої тематичної близькості до пріоритетних напрямів дослідника.

Метою цього етапу є формування інтегрованого вектора $\varphi(I)$, який би узгоджено репрезентував основні наукові інтереси, напрями досліджень, ключові теми та домінантні концепти у всій попередній діяльності науковця. Для цього виконують збір текстів, що найкраще відображають його науковий профіль. До таких джерел можуть належати:

- анотації попередніх статей;
- фрагменти дисертаційних робіт, розділи звітів, презентаційні матеріали;
- описи завершених або поточних наукових проєктів та грантів;
- патенти;

- ключові слова й термінологічні списки, що характеризують наукові пріоритети.

Для кожного текстового джерела d_k застосовуються стандартні процедури лінгвістичної обробки — очищення від технічних даних, лематизація та видалення стоп-слів [72, 73, 116]. Основна вимога полягає в тому, щоб зберегти ті елементи змісту, які найбільш точно відображають науковий доробок дослідника та сфери його зацікавлень.

Подібно до етапу формування ембеддингів об'єктів, кожен документ d_k , що описує інтереси дослідника, подається до мовної моделі. Якщо документ має значний обсяг, його розділяють на блоки і модель обчислює окремий ембеддинг для кожного блоку:

$$\varphi(\text{Block}_i(d_k)) \in R^D \quad (2.3)$$

Припустимо, документ d_k розділено на m_k блоків. Тоді інтегрований вектор цього документа отримують шляхом усереднення (або зваженої суми) усіх ембеддингів:

$$\varphi(d_k) = \frac{1}{m_k} \sum_{i=1}^{m_k} \varphi(\text{Block}_i(d_k)) \quad (2.4)$$

Як і в попередньому етапі, доцільно застосовувати вагові коефіцієнти, якщо окремі частини документа є більш інформативними чи краще характеризують профіль дослідника.

Найпростішим підходом до формування узагальненого вектора інтересів є рівномірне усереднення всіх векторів джерел:

$$\varphi(I) = \frac{1}{n} \sum_{k=1}^n \varphi(d_k) \quad (2.5)$$

де n — загальна кількість документів (або набору блоків), що визначають тематичний профіль дослідника. Після обчислення цієї суми отримують вектор $\varphi(I) \in R^D$ тієї самої розмірності D , що й ембеддинги публікацій.

Альтернативно, у разі різномірного набору джерел (наприклад, якщо одні документи охоплюють пріоритетну сферу, а інші — периферійну), можна застосувати зважену схему:

$$\varphi(I) = \frac{\sum_{k=1}^n \omega_k \varphi(d_k)}{\sum_{k=1}^n \omega_k} \quad (2.6)$$

де ω_k – коефіцієнт важливості документа d_k . Такий підхід дає змогу підкреслити основний векторний напрям у просторі ознак, оминаючи «шум» від другорядних тем.

Етап 4. Обчислення релевантності

Обчислення релевантності об'єктів наукової діяльності полягає у порівнянні їхніх векторів $\varphi(o)$, сформованих на попередньому етапі, з інтегрованим вектором наукових інтересів дослідника $\varphi(I)$.

Завдяки використанню векторних ембеддингів великої мовної моделі, які відображають семантичний зміст текстів, процес оцінювання переходить від поверхневого порівняння ключових слів до багатовимірної інтерпретації тематичної близькості.

Найпоширенішою метрикою для оцінювання схожості між двома ембеддингами є косинусна подібність [100, 119]. Вона визначається як:

$$rel(o) = \frac{\varphi(I) \cdot \varphi(o)}{\|\varphi(I)\| \|\varphi(o)\|} \quad (2.7)$$

де $\varphi(I) \cdot \varphi(o)$ – скалярний добуток векторів, а $\|\varphi(I)\| \|\varphi(o)\|$ – їхні евклідові норми. Якщо обидва вектори нормовано до одиничної довжини, значення $rel(o)$ лежить у межах від 0 до 1. Показник, близький до 1, свідчить про високу тематичну подібність, а близький до 0 — вказує на віддаленість або зовсім відмінну сферу змісту.

Етап 5. Класифікація та збереження результатів

На завершальному етапі виконується ухвалення остаточного рішення щодо того, наскільки отримані об'єкти наукової діяльності відповідають змістовному профілю дослідника, а також здійснюється збереження результатів для подальшої роботи системи. На цьому етапі застосовуються розраховані значення релевантності $\varphi(o)$, отримані шляхом порівняння векторів $\varphi(I)$ та $\varphi(o)$. Якщо

показник перевищує задане порогове значення τ , об'єкт вважається таким, що тематично відповідає науковим інтересам дослідника та включається до колекції відібраних результатів. Якщо ж $rel(o) < \tau$ об'єкт відноситься до категорії малорелевантних або другорядних і може бути переміщений до менш пріоритетного сховища або архівований.

Цей крок можна також розглядати як дві взаємопов'язані процедури:

1) маркування об'єктів: кожному об'єкту присвоюється мітка «релевантний» або «нерелевантний». У базі даних зберігається не лише статус, а й значення $rel(o)$, що дозволяє деталізувати рівень відповідності та забезпечує можливість подальшого аудиту прийнятих рішень.

2) Управління сховищем наукової активності: релевантні об'єкти можуть зберігатися в спеціалізованому каталозі, де вони впорядковуються за релевантністю, датою або тематичними напрямками. За необхідності може бути застосований розширений поділ на категорії — наприклад, якщо $rel(o) \geq 0.9$, об'єкт зараховується до групи «високої релевантності». Натомість об'єкти з низькими значеннями подібності можуть переноситися до архіву для можливого подальшого перегляду.

Прикладом може бути система з порогом $\tau = 0.75$. Якщо об'єкт отримав $rel(o) \geq 0.82$, алгоритм однозначно класифікує його як тематично релевантний. У протилежній ситуації, коли значення нижче порогового, результат маркується як «нерелевантний» і залишається в базі даних для можливого повторного аналізу. Такий механізм дозволяє відстежити кількість відхилених результатів і уточнювати τ залежно від специфіки дослідження.

Додатково передбачається можливість використання зворотного зв'язку від користувачів. Якщо дослідник виявляє серед «нерелевантних» об'єктів такі, що насправді відповідають його тематиці, він може позначити їх як пропущені релевантні (false negatives). У такому разі система може бути переналаштована для підвищення показника Recall — шляхом зниження τ або модифікації формули обчислення $rel(o)$. Аналогічно, у випадку появи false positives (коли

об'єкт позначено як релевантний, хоча він не відповідає інтересам дослідника) рекомендується збільшити τ або додати додаткові критерії для класифікації.

Загалом, класифікація та збереження результатів становлять завершальний етап алгоритму, надаючи досліднику впорядкований набір об'єктів, структурованих за ступенем близькості до його змістового профілю. При цьому зберігаються всі проміжні дані, що дозволяє аналізувати рішення, удосконалювати модель та швидко реагувати на зміни у наукових пріоритетах [18, 46, 113, 140].

Усі вище наведені етапи також можна представити у вигляді блок-схеми алгоритму(Рис.2.6.) для полегшення розуміння механізмів оцінки релевантності.

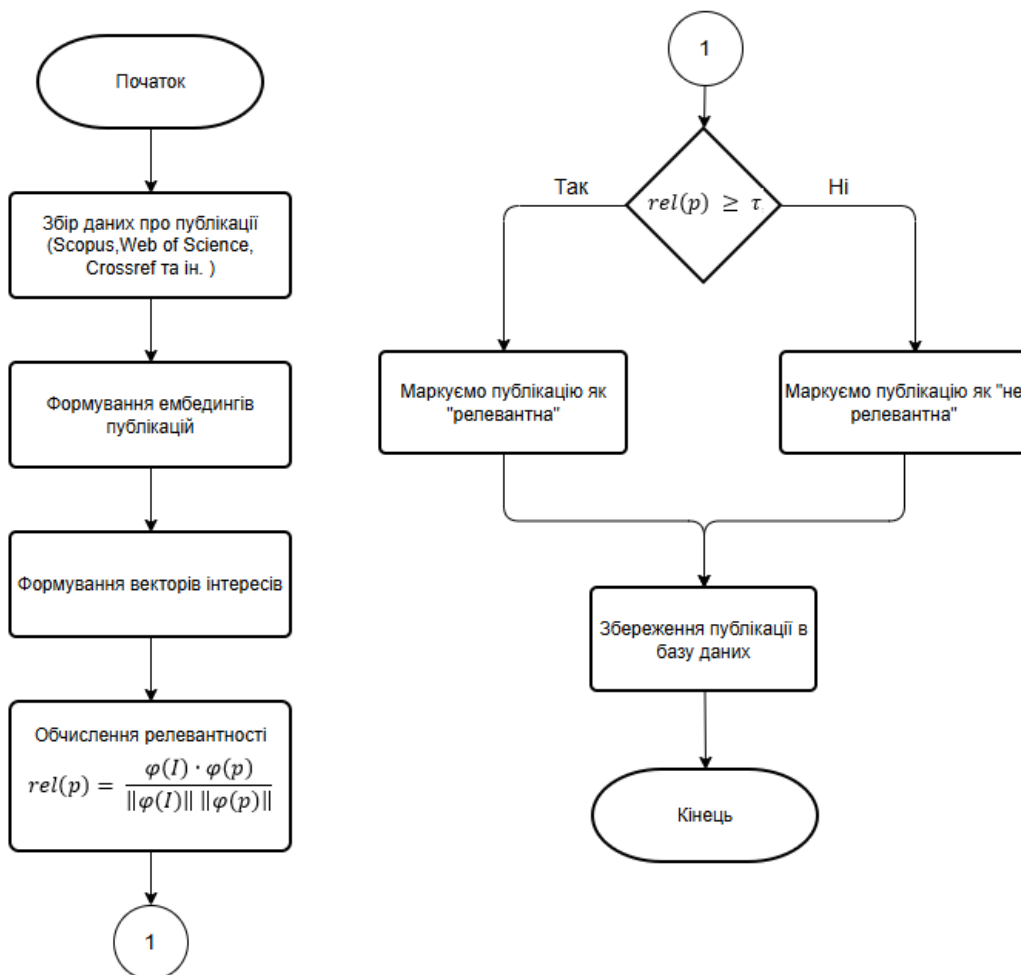


Рисунок 2.6 – Блок-схема алгоритму семантичного впорядкування зібраної інформації про наукову діяльність

Висновки до розділу 2

1. У розділі обґрунтовано необхідність створення цілісного інформаційного забезпечення для збору, узгодження та інтеграції даних про наукову діяльність НПП, оскільки наявні системи переважно орієнтовані на фіксацію окремих результатів і не забезпечують повноти, структурної узгодженості та міжсистемної сумісності наукової інформації. Доведено, що якість вихідних даних визначає достовірність аналітичних показників та ефективність управлінських і прогностичних рішень.

2. Проаналізовано міжнародні стандарти опису наукових даних (Dublin Core, CERIF, DataCite, Crossref, JATS, Schema.org), а також системи унікальної ідентифікації (DOI, ORCID, ROR), які забезпечують семантичну узгодженість та інтероперабельність інформації між інституційними, національними та глобальними науковими інфраструктурами. Показано, що застосування цих стандартів формує основу для методологічно правильного та уніфікованого представлення різномірних об'єктів наукової діяльності.

3. Встановлено, що інтеграція даних на основі поєднання API-інтерфейсів, harvesting за протоколом OAI-PMH та веб-парсингу є необхідною умовою охоплення повного спектра джерел наукової інформації. Такий багатоканальний підхід дозволяє формувати актуальний і узгоджений масив даних навіть за відсутності уніфікованих форматів у зовнішніх системах або при різній доступності метаданих.

4. Запропоновано та детально описано інформаційну технологію збору даних про наукову діяльність НПП, яка охоплює етапи вилучення, нормалізації, сегментації, інтеграції та формування структурованих профілів дослідників. Показано, що застосування алгоритмів зіставлення, семантичного узгодження та ідентифікації за міжнародними унікальними ідентифікаторами забезпечує перехід від фрагментарних даних до цілісної моделі наукової активності.

5. Розроблено модуль семантичного впорядкування на основі векторних ембеддингів великих мовних моделей, який здійснює змістовне ранжування

об'єктів наукової діяльності за ступенем їх відповідності тематичному профілю дослідника. Доведено, що відмова від жорсткої фільтрації на користь ранжування дозволяє зберегти повноту наукових даних, забезпечуючи їх впорядкування за змістовими характеристиками та створюючи підґрунтя для подальших аналітичних і прогностичних процедур.

РОЗДІЛ 3

МЕТОД ПРОГНОЗУВАННЯ ПОКАЗНИКІВ НАУКОВОЇ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНИХ ПРАЦІВНИКІВ НА ОСНОВІ АНАЛІЗУ ІНТЕРВАЛЬНИХ ДАНИХ

У першому розділі показано, що прогнозування показників наукової діяльності науково-педагогічних працівників є важливою задачею в силу внесених змін в «Закон про вищу освіту» та в «Закон про наукову та науково-технічну діяльність». Зокрема, при формуванні складу науково-педагогічного персоналу в поточному навчальному році, необхідно спрогнозувати на наступний рік показники наукової діяльності для кожного працівника цієї категорії і на цій підставі поставити завдання на виконання запланованих годин по науковій роботі, кількість яких не може бути меншою як 30% від загальної кількості годин, які виконує працівник протягом навчального року. Методи прогнозування із використанням математичних моделей розглянуто в першому розділі. При цьому обґрунтовано, що в умовах обмеженої за потужністю вибірки даних, доцільно використовувати методи аналізу інтервальних даних.

Варто зазначити, що набір «історичних» даних про діяльність кожного працівника, які можливо використати для моделі прогнозування, є обмежений кількістю років його звітності про наукову діяльність, які зазвичай розміщено в рейтинговій системі науково-педагогічних працівників, а актуальних даних і того менше. Тобто вибірка даних зазвичай складається не більше як з 10 наборів даних. За цих умов запропоновано усі показники наукової діяльності агрегувати у підгрупи, щоб визначити часові витрати для агрегованих показників, а потім на цій підставі визначити часові витрати для показників в межах кожної групи. Такий підхід дає можливість у разі зменшити вибірку даних, необхідних для ідентифікації моделі прогнозування наукової діяльності.

Далі у розділі розглянуто методи аналізу інтервальних даних з метою отримання еліпсоїдних оцінок допустимих значень показників наукової діяльності з подальшим перетворення еліпсоїдних оцінок в інтервальну форму.

На завершення розділу розглянуто приклади застосування запропонованого методу та математичної моделі для прогнозування показників наукової діяльності науково-педагогічних працівників.

Результати цього розділу опубліковано у працях [11, 36].

3.1. Ієрархія показників наукової діяльності

Однією з ключових передумов для побудови математичної моделі прогнозування наукової діяльності є формалізація системи показників, які її характеризують. Така формалізація передбачає структурування основних видів наукової активності науково-педагогічних працівників у вигляді ієрархії, що ґрунтується на чинних нормативно-правових актах, зокрема Законах України «Про освіту» [17], «Про вищу освіту» [15], «Про наукову і науково-технічну діяльність» [16], а також на типових положеннях закладів вищої освіти щодо планування та обліку навчальної й наукової роботи. Узагальнена класифікація видів наукової діяльності, запропонована в межах цієї роботи, охоплює найпоширеніші напрями активності, які формалізовані в індивідуальних планах роботи працівників і підлягають обліку та нормативному оцінюванню з боку ЗВО.

Узагальнена класифікація, що пропонується, базується на аналізі нормативних документів, внутрішніх положень закладів вищої освіти, типових форм індивідуального плану роботи та облікових таблиць виконання [1, 3, 14]. На основі цього аналізу виокремлено п'ять функціональних груп, що репрезентують ключові напрями наукової активності. Кожна з цих груп у моделі представлена окремою змінною, яка описує обсяг часу, витраченого на відповідний тип діяльності, що дозволяє знизити розмірність вхідного простору

та забезпечити кращу ідентифікацію моделі при обмеженій кількості спостережень.

Першу групу становить підготовка наукових кадрів вищої кваліфікації, якій у подальшому відповідає змінна x_{j1} . Вона охоплює наукове консультування й керівництво здобувачами ступеня доктора філософії та доктора наук, участь у разових спеціалізованих вчених радах, підготовка відгуків як офіційного опонента та рецензента, супровід процедур захисту, підготовку рецензій.

Друга група – проєктно-патентна діяльність, що позначається змінною x_{j2} . До цієї групи включено підготовку проєктних заявок, виконання дослідницьких робіт в межах міжнародних і національних грантів, участь у комерціалізації наукових результатів, підготовка та оформлення охоронних документів (патентів, авторських свідоцтв), а також впровадження результатів досліджень в практику.

Третьою функціональною групою є експертна, рецензійна та редакторська діяльність – змінна x_{j3} . До неї відноситься участь у діяльності експертних рад і груп при державних органах (зокрема МОН, НАЗЯВО), рецензування навчальних посібників, наукових публікацій, монографій, членство в редколегіях журналів, виконання функцій рецензента у наукових виданнях, участь у наукових комітетах конференцій.

Четверту групу становить публікаційна активність, яку в моделі представляє змінна x_{j4} . Вона охоплює підготовку та публікацію статей у фахових і міжнародних виданнях, публікацію монографій та розділів монографій, опрацювання випусків журналів, публікацію тез доповідей на конференціях, виконання функцій редактора або відповідального секретаря. Особлива увага приділяється публікаціям у виданнях, що індексуються в міжнародних наукометричних базах (Scopus, Web of Science), з урахуванням їх квартилю або індексу впливу.

Завершує структуру організація наукових заходів – змінна x_{j5} . Сюди належать підготовка й проведення наукових конференцій, семінарів, круглих

столів, участь у роботі організаційних комітетів, наукових гуртків, супровід студентських досліджень, підготовка учасників олімпіад і конкурсів.

Виходячи з вище описаної структури повний вектор наукової діяльності науково-педагогічного працівника у j -тому році по певних категоріях можна подати в такому вигляді:

$$x_j = \begin{bmatrix} x_{j1} \\ x_{j2} \\ x_{j3} \\ x_{j4} \\ x_{j5} \end{bmatrix}$$

де x_{ji} – кількість годин, витрачених на i -ту групу показників у j -тому році.

Для аналізу показників на більш нижчих рівнях ієрархії доцільно агреговані показники x_{ji} ще більше деталізувати до рівня вхідних змінних. Тоді можна припустити, що група i містить m_i підпоказників. Після цього елементарну змінну, що відображає обсяг часу на k -тий показник групи i у j році, можна позначити як x_{jik} , де:

$j \in \{1, \dots, N\}$ – номер року спостереження,

$i \in \{1, \dots, 5\}$ – номер групи,

$k \in \{1, \dots, m_i\}$ – номер підпоказника в межах групи.

Після цього загальний вектор наукової діяльності у j -тому році, який охоплює усі показники, матиме вигляд:

$$x_j = \left(x_{j11}, x_{j11}, \dots, x_{j1m_1}, x_{j21}, \dots, x_{j5m_5} \right)^T \in R^n, n = \sum_{i=1}^5 m_i$$

Тоді агрегованні змінні x_{ji} можна обчислити як суму:

$$x_{ji} = \sum_{k=1}^{m_i} x_{jik}$$

На рисунку 3.1 наведено графічне представлення ієрархічної структури з відображенням взаємозв'язків між функціональними групами показників і відповідними видами наукової діяльності.

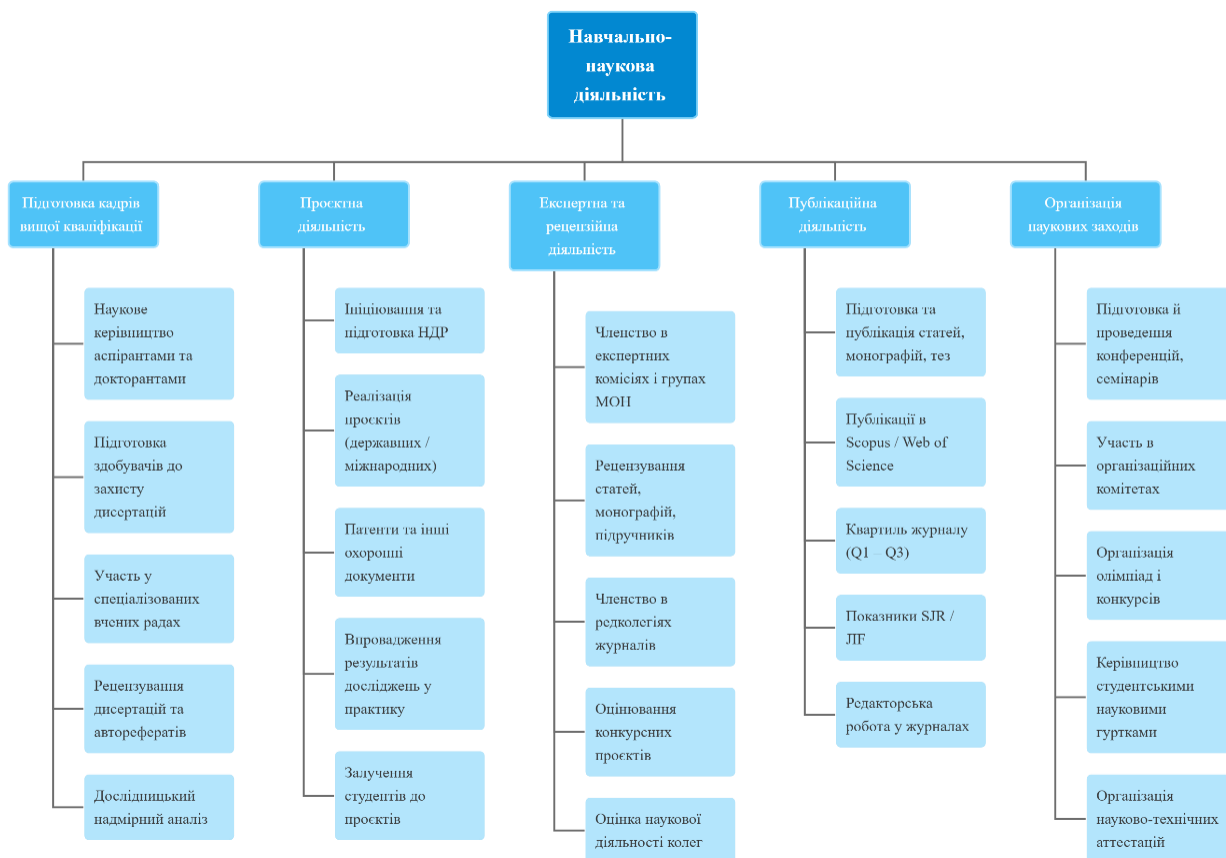


Рисунок 3.1 – Ієрархічна структура показників наукової діяльності

Не порушуючи загальності, в подальшому розгляді методу планування та прогнозування наукової діяльності будемо брати до уваги тільки сумарні показники по групах.

3.2. Метод прогнозування показників наукової діяльності на основі аналізу інтервальних даних

Розглянемо випадок оцінювання результатів наукової діяльності для показників, що характеризують кожен групу. По аналогії із розглянутою системою агрегованих показників в подальшому можемо застосувати такий самий підхід для показників в межах групи.

3.2.1. Формування «портрету» працівника щодо його наукової діяльності у вигляді інтервальної системи алгебричних рівнянь

Для визначення «портрету» працівника щодо його наукової діяльності, сумарну кількість годин, виконану науково-педагогічним працівником у j -тому році, подаємо у вигляді такого алгебричного рівняння:

$$y_j = w_1 \cdot x_{j1} + w_2 \cdot x_{j2} + \dots + w_i \cdot x_{ji} + \dots + w_n \cdot x_{jn}, = \bar{w}^T \cdot \bar{x}_j, \quad j = 1, \dots, N, \quad (3.1)$$

де y_j – сумарна кількість годин наукової роботи за усіма її групами показників, виконана працівником у j -тому році;

$\bar{x}_j = (x_{j1}, \dots, x_{jn})^T$ – вектор показників наукової діяльності, компоненти якого дорівнюють кількості виконаних годин для кожної групи показників в j -тому році;
 $\bar{w} = (w_1, \dots, w_n)^T$ – вектор вагових коефіцієнтів для кожної групи показників, що характеризуватиме «портрет» науковця.

Для виконання умов контракту науковець повинен виконати заплановані години по науковій роботі, кількість яких не може бути меншою як 30% від загальної кількості годин, які виконує працівник протягом навчального року та не може бути більшою як 60% від загальної кількості годин, які виконує працівник протягом навчального року (оскільки до 30% робочого часу науково-педагогічного працівника має бути обсяг навчальної роботи).

Відповідно, для визначення «портрету» працівника щодо його наукової діяльності, інтервальні значення сумарної кількості годин наукової роботи $[y_j^-, y_j^+]$ можуть бути одержані спираючись на ці вимоги.

Тоді, беручи до уваги вираз 3.1 та задаючи для кожного з попередніх років умови виконання годин по науковій роботі в межах заданих інтервалів $[y_j^-, y_j^+]$, отримаємо умови для обчислення вагових коефіцієнтів для кожної групи показників, що характеризують «портрет» науковця:

$$\begin{cases} y_1^- \leq w_1 \cdot x_{11} + w_2 \cdot x_{12} + \dots + w_i \cdot x_{1i} + \dots + w_n \cdot x_{1n} \leq y_1^+ \\ \vdots \\ y_j^- \leq w_1 \cdot x_{j1} + w_2 \cdot x_{j2} + \dots + w_i \cdot x_{ji} + \dots + w_n \cdot x_{jn} \leq y_j^+ \\ \vdots \\ y_N^- \leq w_1 \cdot x_{N1} + w_2 \cdot x_{N2} + \dots + w_i \cdot x_{Ni} + \dots + w_n \cdot x_{Nn} \leq y_N^+ \end{cases} \quad (3.2)$$

Серед важливих властивостей, отриманої інтервальної системи лінійних алгебричних рівнянь (ІСЛАР), варто зазначити, що вона є системою N лінійних нерівностей відносно n невідомих вагових коефіцієнтів $\bar{w} = (w_1, \dots, w_n)^T$.

Перепишемо систему (3.2) в матричному вигляді:

$$\bar{Y}^- \leq X \cdot \bar{w} \leq \bar{Y}^+, \quad (3.3)$$

де $\bar{Y}^- = \{y_j^-, j=1, \dots, N\}$, $\bar{Y}^+ = \{y_j^+, j=1, \dots, N\}$ – вектори, складені із верхніх та нижніх меж інтервалів $[y_j^-, y_j^+]$, відповідно (варто зазначити, що у нашому випадку ці інтервали будуть однаковими, оскільки визначені нормативними документами); $X = \{x_{ji}, i=1, \dots, n; j=1, \dots, N\}$ – матриця значень показників наукової діяльності, компоненти якої дорівнюють кількості виконаних годин для кожної групи показників в кожному році.

Система (3.3) може не мати жодного розв'язку, тобто бути несумісною або мати багато розв'язків [4, 6, 8].

Нехай система (3.3) є сумісною. Позначимо через Ω множину (область) її розв'язків, тобто

$$\Omega = \{\bar{w} \in R^n \mid \bar{Y}^- \leq X \cdot \bar{w} \leq \bar{Y}^+\}. \quad (3.4)$$

Щодо отриманої системи інтервальних рівнянь, то її розв'язок геометрично є опуклий многогранник в просторі розв'язків. Зокрема, якщо $N < n$, то система недовизначена її потрібно доповнити обмеженнями типу $w_i^- \leq w_i \leq w_i^+$. Вибір цих обмежень залежить від суб'єктивних міркувань працівника, але може і ґрунтуватися на міркуваннях особи, яка є керівником працівника [39, 78].

Якщо ж маємо умову $N=n$, то ІСЛАР є насиченою і геометрично її розв'язок є паралелограмом [5, 9].

У випадку $N > n$, - ІСЛАР (3.3) за допомогою процедур, описаних у працях [6, 9] можна звести до насиченої [39, 40, 41, 42, 43, 44, 78].

Таким чином випадок ІСЛАР (3.3) за умови $N = n$ є дуже важливим, оскільки дає можливість отримати в аналітичному вигляді еліпсоїдні оцінки вектора вагових коефіцієнтів [7, 38, 45]. Тому розглянемо цей випадок детальніше.

Для цього випадку X – матриця значень (годин) показників наукової діяльності є квадратною (N стрічок на n стовпців).

Якщо визначник матриці не дорівнює нулю, то можна отримати матрицю X^{-1} , обернену до X і, відповідно, розв’язок такої системи лінійних алгебричних рівнянь:

$$X^{-1} \cdot \vec{w}_s = \vec{Y}_s \quad (3.5)$$

де \vec{Y}_s – вектор, який складено із межових значень інтервалів $[y_j^-, y_j^+]$, наприклад, він може мати такий вигляд $\vec{Y}_s = (y_1^-, y_2^+, y_3^+, \dots, y_N^-)$

Запишемо розв’язок цієї системи у такому вигляді:

$$\vec{w}_s = X^{-1} \cdot \vec{Y}_s. \quad (3.6)$$

Вектор \vec{w}_s є однією із вершин, загальна кількість яких $R = 2^n$ паралелотопа Ω , яку утворено перетином відповідних площин, заданих нерівностями (інтервальними рівняннями) системи (3.2). Центр паралелотопа обчислюємо за формулою:

$$\vec{\bar{w}} = \frac{1}{2^n} \sum_{s=1}^{2^n} \vec{w}_s = X^{-1} \cdot \vec{\bar{Y}}, \quad (3.7)$$

де $\vec{\bar{Y}}$ – вектор є середнім арифметичним усіх векторів, складених із межових значень річної норми годин, витрачених на наукову діяльність:

$$\vec{\bar{Y}} = \frac{1}{2^n} \cdot \left(\sum_{s=1}^{2^n} \vec{Y}_s \right) = (\bar{y}_1, \dots, \bar{y}_N)^T. \quad (3.8)$$

В подальшому, кожен частину отриманих властивостей розв’язків ІСЛАР (3.2), їх оцінок будемо ілюструвати прикладами.

Приклад 1. Розглянемо приклад формування розв’язків ІСЛАР у вигляді паралелотопа Ω . Обмежимося випадком, коли кількість видів наукової

діяльності становить $N=n=2$.

У якості вихідних даних використано показники наукової активності одного з науковців, отримані на основі рейтингової системи ЗУНУ. Для 2023 року зафіксовано дві основні складові наукової діяльності: публікаційна та проєктно-патентна активність, яким відповідають значення $x_1 = 114$ год. та $x_2 = 152$ год. відповідно. У 2024 році значення цих самих показників становили $x_1 = 200$ год. і $x_2 = 80$ год.

Для побудови моделі сформуємо інтервальну систему, яка відображає залежність обсягу виконаного наукового навантаження від вагових коефіцієнтів видів діяльності.

Система має такий вигляд:

$$\begin{cases} y_1 = w_1 \cdot 114 + w_2 \cdot 152 \\ y_2 = w_1 \cdot 200 + w_2 \cdot 80 \end{cases}$$

де y_1 та y_2 – обсяги виконаного наукового навантаження у відповідні роки.

Задаймо необхідні умови, за яких сумарний обсяг виконаного наукового навантаження протягом 2023 та 2024 років повинен належати інтервалу $[480, 800]$, тобто $y^- = 480$, $y^+ = 800$.

На основі емпіричних даних щодо кількості виконаних годин у 2023 та 2024 роках інтервальну систему можна переписати в матричному вигляді (3.3), де матриця X набуває вигляду:

$$X = \begin{bmatrix} 114 & 152 \\ 200 & 80 \end{bmatrix},$$

а вектори меж інтервалів визначаються як:

$$\bar{Y}^- = \begin{bmatrix} 480 \\ 480 \end{bmatrix}, \quad \bar{Y}^+ = \begin{bmatrix} 800 \\ 800 \end{bmatrix},$$

У цьому випадку кожен рядок матриці X відповідає кількості годин, витрачених на відповідні види наукової діяльності у певному році, а вектор вагових коефіцієнтів $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ визначає відносну значущість кожного виду діяльності. Інтервальна система (3.3) в такому разі описує множину всіх

допустимих комбінацій ваг, які забезпечують виконання заданого обсягу наукового навантаження в обох часових періодах.

Для прикладу, розглянутого вище, множина розв'язків системи (3.3), що відповідає умовам щодо необхідного обсягу наукового навантаження у 2023 та 2024 роках, має вигляд:

$$\Omega = \left\{ \vec{w} \in R^2 \left[\begin{array}{c} 480 \\ 480 \end{array} \right] \leq \left[\begin{array}{cc} 114 & 152 \\ 200 & 80 \end{array} \right] \cdot \vec{w} \leq \left[\begin{array}{c} 800 \\ 800 \end{array} \right] \right\}.$$

Отримана множина Ω є підмножиною евклідового простору R^2 і відображає всі можливі значення вагових коефіцієнтів $\vec{w} = (w_1, w_2)^T$, при яких виконується нормативне навантаження в межах заданих інтервалів. Геометрично така множина є опуклою (паралелограмом), який обмежує допустимі поєднання значущості видів діяльності відповідно до встановлених річних нормативів.

Обчислюємо визначник матриці X :

$$X = \left[\begin{array}{cc} 114 & 152 \\ 200 & 80 \end{array} \right], \det(X) = 114 \cdot 80 - 152 \cdot 200 = -21280 \neq 0,$$

Визначник не дорівнює нулю і це свідчить про те, що система має єдиний розв'язок для кожного вектора навантаження \vec{Y}_s , сформованого як комбінація нижніх та верхніх меж інтервалів $[y_j^-, y_j^+]$. Наприклад, якщо взяти вектор:

$$\vec{Y}_s = \left[\begin{array}{c} 800 \\ 800 \end{array} \right],$$

тобто максимальні допустимі значення навантаження для обох років, то відповідна точкова оцінка вагових коефіцієнтів визначається за формулою (3.6).

Аналогічно, можна побудувати $R = 2^r = 4$ таких векторів \vec{Y}_i , кожен з яких відповідає одній із можливих комбінацій меж інтервалів. Зокрема:

$$\vec{Y}_1 = \left[\begin{array}{c} 480 \\ 480 \end{array} \right], \quad \vec{Y}_2 = \left[\begin{array}{c} 480 \\ 480 \end{array} \right], \quad \vec{Y}_3 = \left[\begin{array}{c} 800 \\ 800 \end{array} \right], \quad \vec{Y}_4 = \left[\begin{array}{c} 800 \\ 800 \end{array} \right],$$

Відповідні значення $\vec{w}_s = X^{-1} \cdot \vec{Y}_s$ формують вершини цього паралелограма у просторі вагових коефіцієнтів.

Центр паралелограма — як середнє арифметичне всіх його вершин — обчислюємо за формулою (2.7) та отримуємо такий результат:

$$\bar{Y} = \frac{1}{4} \begin{bmatrix} 480+480+800+800 \\ 480+800+480+800 \end{bmatrix} = \begin{bmatrix} 640 \\ 640 \end{bmatrix}$$

Отже, підставивши цей середній вектор у рівняння (2.7), отримаємо центр паралелограма у просторі вагових коефіцієнтів:

$$\bar{w} = X^{-1} \cdot \begin{bmatrix} 640 \\ 640 \end{bmatrix} = \frac{1}{-21280} \cdot \begin{bmatrix} 80 & -152 \\ -200 & 114 \end{bmatrix} \cdot \begin{bmatrix} 640 \\ 640 \end{bmatrix} \approx \begin{bmatrix} 2.165 \\ 2.586 \end{bmatrix}$$

Цей вектор може розглядатися як усереднена оцінка значущості кожного виду наукової діяльності, необхідного для досягнення бажаного результату за умов заданої інтервальної невизначеності.

Тепер знайдемо графічний розв'язок системи, що визначає множину допустимих значень вагових коефіцієнтів $\bar{w} = (w_1, w_2)^T$, які задовольняють інтервальні обмеження (3.3).

Як видно на представленому графіку (Рис. 3.2), кожна з паралельних смуг відповідає одному з інтервальних рівнянь, у яких права частина варіюється в межах $[480;800]$. Світло-сірим та середньо-сірим півпрозорим заповненням позначено допустимі області значень, які задовольняють відповідно першу та другу нерівність окремо. Темно-сіра область перетину цих смуг позначена символом Ω і являє собою геометричне відображення множини допустимих розв'язків — паралелограма Ω .

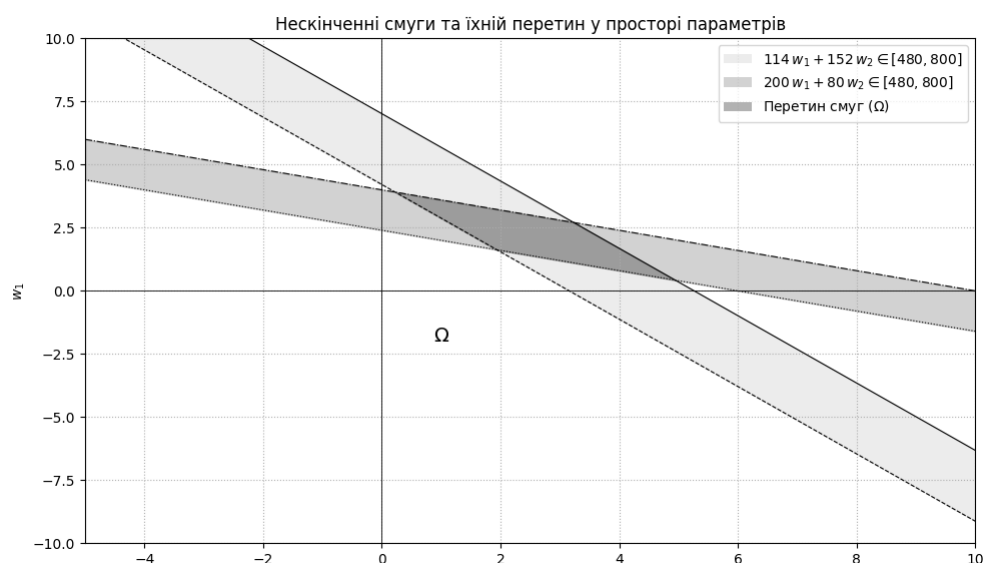


Рисунок. 3.2 – Область допустимих розв'язків системи (3.3), побудована як перетин смуг, що відповідають інтервалам значень для 2023 та 2024 років

З графіка видно, що ця множина є **опуклою, замкненою та обмеженою**, що повністю узгоджується з теоретичними властивостями розв'язків інтервальних систем при невиродженій матриці коефіцієнтів X . Центральна точка області Ω , що відповідає **центру тяжіння паралелограма** та збігається з аналітично обчисленим усередненим значенням вагових коефіцієнтів згідно з формулою (3.7).

3.2.2. Формування еліпсоїдних оцінок вектора вагових коефіцієнтів «портрету» працівника щодо його наукової діяльності

Тепер перейдемо до отримання еліпсоїдних оцінок вектора вагових коефіцієнтів для кожної групи показників, що характеризуватиме «портрет» науковця. Для цього, спираючись на лему 4.1 із праці [4] запишемо загальний вигляд допустимих оцінок вектора вагових коефіцієнтів для кожної групи показників у вигляді еліпсоїда.

$$Q_n^- = \{ \vec{w} \in R^n \mid (\vec{w} - \vec{\bar{w}})^T \cdot X^T \cdot E^{-2} \cdot X \cdot (\vec{w} - \vec{\bar{w}}) \leq 1 \}, \quad (3.9)$$

де

$$\vec{\bar{w}} = \frac{1}{2^n} \sum_{s=1}^{2^n} \vec{w}_s = X^{-1} \cdot \vec{Y} - \quad (3.10)$$

центр ваги еліпсоїда, який співпадає з центром паралелограма Ω .

У виразі (3.9) за символ E позначено діагональну матрицю половин різниці верхньої та нижньої норми наукового навантаження, обчисленої з інтервалу $[y_j^-, y_j^+]$ Отже діагональні елементи матриці E обчислюємо за допомогою такої формули:

$$\tilde{\Delta}_j = 0,5 \cdot (y_j^+ - y_j^-) \quad (j = 1, \dots, N) \quad (3.11)$$

Як бачимо, центри паралелограма (у вигляді виразу (3.7) і еліпсоїда (у вигляді 3.10) співпадають.

Приклад 2. Розглянемо побудову еліпсоїдної оцінки для вагових

коефіцієнтів на даних прикладу 1.

Як зазначалося, у якості вихідних даних використано показники наукової активності одного з науковців, отримані на основі рейтингової системи ЗУНУ за 2023 та 2024 рік. Для цього прикладу отримано область розв'язків вагових коефіцієнтів у вигляді паралелограма

$$\Omega = \left\{ \vec{w} \in R^2 \left| \begin{bmatrix} 480 \\ 480 \end{bmatrix} \leq \begin{bmatrix} 114 & 152 \\ 200 & 80 \end{bmatrix} \cdot \vec{w} \leq \begin{bmatrix} 800 \\ 800 \end{bmatrix} \right. \right\}$$

Центр еліпсоїда співпадає з центром паралелограма, обчислено за формулою:

$$\vec{w} = X^{-1} \cdot \begin{bmatrix} 640 \\ 640 \end{bmatrix} = \frac{1}{-21280} \cdot \begin{bmatrix} 80 & -152 \\ -200 & 114 \end{bmatrix} \cdot \begin{bmatrix} 640 \\ 640 \end{bmatrix} \approx \begin{bmatrix} 2.165 \\ 2.586 \end{bmatrix}$$

Діагональна матриця E^{-2} має такий вигляд

$$E^{-2} = \begin{pmatrix} \frac{1}{\Delta_1^2} & 0 \\ 0 & \frac{1}{\Delta_2^2} \end{pmatrix} = \begin{pmatrix} \frac{1}{25600} & 0 \\ 0 & \frac{1}{25600} \end{pmatrix}$$

Таким чином формула для еліпсоїда в нашому випадку буде мати такий вигляд:

$$Q_n^- = \left\{ \vec{w} \in R^n \mid \left(\vec{w} - \begin{pmatrix} 2.165 \\ 2.586 \end{pmatrix} \right)^T \cdot \begin{pmatrix} 114 & 200 \\ 152 & 80 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{25600} & 0 \\ 0 & \frac{1}{25600} \end{pmatrix} \cdot \begin{pmatrix} 114 & 200 \\ 152 & 80 \end{pmatrix} \cdot \left(\vec{w} - \begin{pmatrix} 2.165 \\ 2.586 \end{pmatrix} \right) \leq 1 \right\}$$

На рисунку 3.3 наведено область допустимих розв'язків Ω ІСЛАР (3.3) та її еліпсоїдна оцінка для векторів вагових коефіцієнтів для кожної групи показників, що характеризуватиме «портрет» науковця.

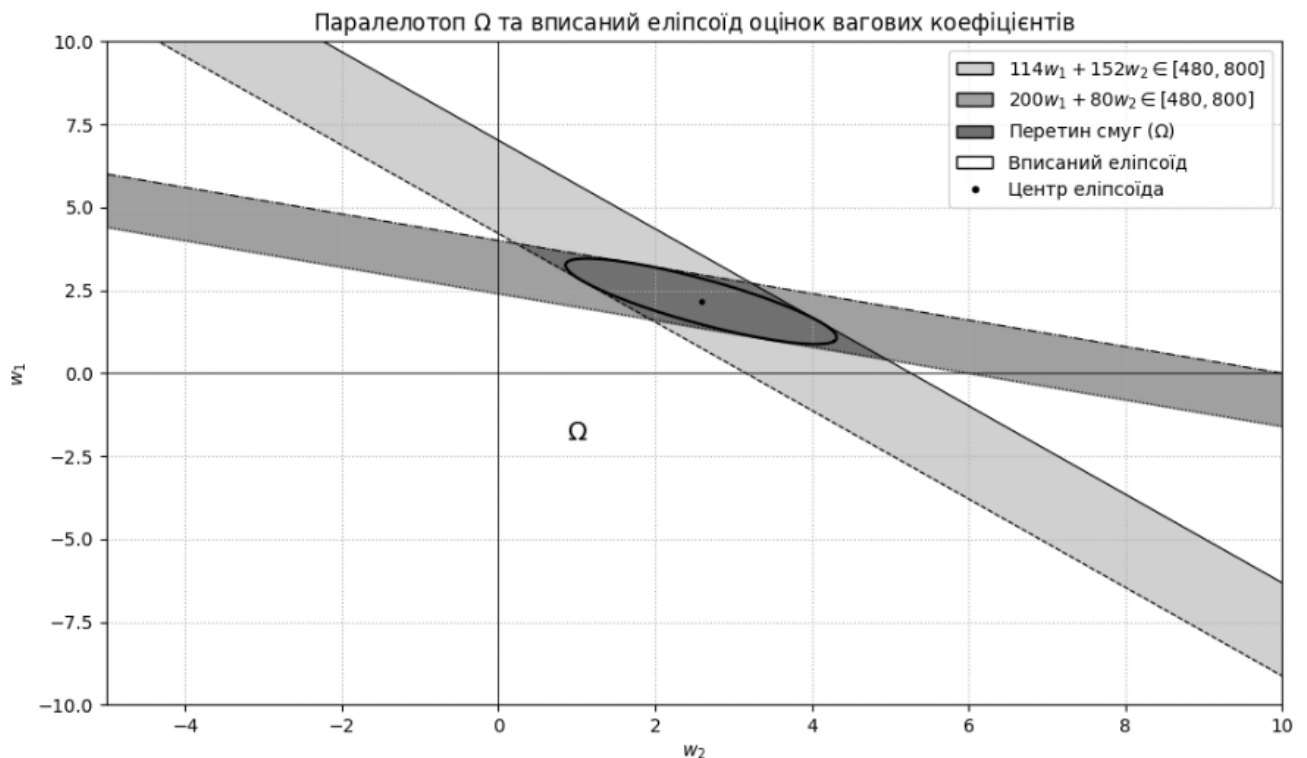


Рисунок 3.3 – Область допустимих розв’язків Ω ІСЛАР (3.3) та її еліпсоїдна оцінка

3.2.3. Формування інтервальних оцінок для векторів вагових коефіцієнтів «портрету» працівника щодо його наукової діяльності

Тепер перейдемо до інтервальних оцінок для векторів вагових коефіцієнтів для кожної групи показників, що характеризуватиме «портрет» науковця. З цією метою задаймо загальний вигляд інтервальної оцінки вагових коефіцієнтів у такому вигляді:

$$-\bar{\sigma} \cdot W \leq \bar{w} - \bar{w} \leq \bar{\sigma} \cdot W, \quad (3.12)$$

де $\bar{\sigma}$ – заданий, чи обчислений вектор впливів вагових коефіцієнтів
 W - невідомий коефіцієнт, що задає ширину інтервалу.

Варто зазначити, що у випадку, якщо припустити, що значення вагових коефіцієнтів для кожної групи показників, обчислене для історичних даних

розподілено за нормальним законом, то умову (3.12) можемо переписати у такому вигляді:

$$-\vec{\sigma} \cdot u(\alpha) \leq \bar{w} - \mu(\bar{w}) \leq \vec{\sigma} \cdot u(\alpha), \quad (3.13)$$

де в цьому випадку

$\vec{\sigma}$ - вектор відомих стандартних відхилень значень вагових коефіцієнтів від сподіваного значення $\mu(\bar{w})$;

$u(\alpha)$ – табличне значення (квантиль) нормованого нормального закону розподілу;

α – довірча імовірність.

Підставимо формулу для інтервалів (3.12) $-\vec{\sigma} \cdot W \leq \bar{w} - \bar{\bar{w}} \leq \vec{\sigma} \cdot W$ замість $\bar{w} - \bar{\bar{w}}$ у формулу для еліпсоїда (3.9), отримаємо:

$$W^2 \cdot ([-\vec{\sigma}; \vec{\sigma}]^T \cdot X^T \cdot E^{-2} \cdot X \cdot [-\vec{\sigma}; \vec{\sigma}]) \leq 1, \quad (3.14)$$

де $[-\vec{\sigma} \cdot W; \vec{\sigma} \cdot W]$ - інтервальна оцінка вагових коефіцієнтів.

Враховуючи симетричність інтервалів (3.12), а також, що центри паралелограма і вписаного в нього еліпсоїда співпадають, вираз (3.14) перепишемо в такому вигляді:

$$W^2 \cdot (\vec{\sigma})^T \cdot X^T \cdot E^{-2} \cdot X \cdot (\vec{\sigma}) \leq 1. \quad (3.15)$$

Тоді, для визначення невідомого коефіцієнта W отримаємо таке рівняння:

$$W^2 \cdot \vec{\sigma}^T \cdot X^T \cdot E^{-2} \cdot X \cdot \vec{\sigma} = 1 \quad (3.16)$$

$$W = \frac{1}{\sqrt{\vec{\sigma}^T \cdot X^T \cdot E^{-2} \cdot X \cdot \vec{\sigma}}} \quad (3.17)$$

У загальному випадку, із рівняння (3.16) отримаємо оцінку коефіцієнта W .

Підставляючи отриману оцінку, замість W у вираз (3.12) отримаємо інтервали для вагових коефіцієнтів для кожної групи показників:

$$-\bar{\sigma} \cdot W + \bar{w} \leq \bar{w} \leq \bar{\sigma} \cdot W + \bar{w} \quad (3.18)$$

Таким чином отримали інтервальні оцінки вагових коефіцієнтів для кожної групи показників, що характеризуватиме «портрет» науковця.

Окреме питання, це вибір вектора впливів вагових коефіцієнтів $\bar{\sigma}$. Як вже зазначалося, якщо є достатня вибірка статистичних даних, то можемо обчислити їх як вектор стандартних відхилень значень вагових коефіцієнтів від сподіваного значення $\mu(\bar{w})$. У випадку, коли це зробити неможливо, пропонується їх обирати на основі такого співвідношення:

$$\bar{\sigma} = \frac{1}{N} \sum_{j=1}^N \bar{x}_j / y_j \quad (3.19).$$

Формула (3.19) дає можливість врахувати вплив кожної групи показників на загальну кількість годин, щодо наукового навантаження виконаного науково-педагогічним працівником.

Приклад 3. Розглянемо приклад отримання інтервальних оцінок коефіцієнтів для кожної групи показників, на даних прикладу 1.

Використовуючи формулу (3.17) та (3.19), отримаємо:

$$\begin{aligned} W &= \frac{1}{\sqrt{\bar{\sigma}^T \cdot X^T \cdot E^{-2} \cdot X \cdot \bar{\sigma}}} = \\ &= \frac{1}{\sqrt{(0,571; 0,429) \cdot \begin{pmatrix} 114 & 200 \\ 152 & 80 \end{pmatrix}^T \cdot \begin{pmatrix} \frac{1}{25600} & 0 \\ 0 & \frac{1}{25600} \end{pmatrix} \cdot \begin{pmatrix} 114 & 200 \\ 152 & 80 \end{pmatrix} \cdot \begin{pmatrix} 0,571 \\ 0,429 \end{pmatrix}}} = 0,827 \end{aligned}$$

$$\text{де } \bar{\sigma} = \frac{1}{N} \sum_{j=1}^N \bar{x}_j / y_j = \begin{pmatrix} 0,571 \\ 0,429 \end{pmatrix}$$

Надалі, при визначенні інтервалу вагових коефіцієнтів, підставляємо цей отриманий результат разом із вектором центрів \bar{w} у вираз (3.18), отримуємо:

$$[\bar{w}] = \begin{pmatrix} [w_1^-; w_1^+] \\ [w_2^-; w_2^+] \end{pmatrix} = \begin{pmatrix} [2.165 - 0.571 \cdot 0.827; 2.165 + 0.571 \cdot 0.827] \\ [2.586 - 0.429 \cdot 0.827; 2.586 + 0.429 \cdot 0.827] \end{pmatrix} = \begin{pmatrix} [1.693; 2.637] \\ [2.232; 2.940] \end{pmatrix}$$

Варто зазначити, що отримані інтервальні оцінки, дають можливість спрогнозувати та запланувати години по науковій роботі, в розрізі показників наукової діяльності для кожної групи показників в j -тому році, кількість яких не може бути меншою як 30% від загальної кількості годин, які виконує працівник протягом навчального року та не може бути більшою як 60% від загальної кількості годин (оскільки до 30% робочого часу науково-педагогічного працівника має бути обсяг навчальної роботи), які виконує працівник протягом навчального року. При цьому буде враховано «портрет» працівника щодо його наукової діяльності у вигляді інтервального вектора $[\bar{w}] = \begin{pmatrix} [w_1^-; w_1^+] \\ [w_2^-; w_2^+] \end{pmatrix}$.

3.3. Математична модель планування та прогнозування видів наукової діяльності та обсягів часу за кожної з них

Тепер, на підставі отриманого «портрету» працівника щодо його наукової діяльності можемо запланувати йому кількість годин в межах кожної групи показників. Зауважимо, що в силу отриманого «портрету» у вигляді інтервального вектора вагових коефіцієнтів для кожної групи показників, заплановані години також матимуть інтервальний вигляд.

Отже, спираючись на математичну модель для визначення «портрету» працівника щодо його наукової діяльності (3.1), сумарну необхідну кількість годин, яку необхідно виконати науково-педагогічним працівником для прогнозованого року, подаємо у вигляді такого інтервального алгебричного рівняння (двохстороння нерівність):

$$y^- \leq [x_1^-; x_1^+] + [x_2^-; x_2^+] + \dots + [x_i^-; x_i^+] + \dots + [x_n^-; x_n^+] \leq y^+ \quad (3.20)$$

Варто зазначити, що умовою (2.20) можемо переписати у такому вигляді:

$$[x_1^-; x_1^+] + [x_2^-; x_2^+] + \dots + [x_i^-; x_i^+] + \dots + [x_n^-; x_n^+] \subseteq [y^-; y^+] \quad (3.21)$$

Позначимо:

$$\begin{aligned} [x_1^-; x_1^+] &= y_p \cdot [w_1^-; w_1^+]; \\ [x_2^-; x_2^+] &= y_p \cdot [w_2^-; w_2^+] \\ &\vdots \\ [x_i^-; x_i^+] &= y_p \cdot [w_i^-; w_i^+]; \\ &\vdots \\ [x_n^-; x_n^+] &= y_p \cdot [w_n^-; w_n^+] \end{aligned} \quad , \quad (3.22)$$

де y_p – деяке невідоме базове значення годин, які необхідно виконати науково-педагогічному працівнику для прогнозованого року.

Підставляючи (3.22) в інтервальне рівняння (3.21), отримаємо інтервальне рівняння для обчислення базове значення годин, які необхідно виконати науково-педагогічному працівнику для прогнозованого року:

$$y^- \leq [w_1^-; w_1^+] \cdot y_p + [w_2^-; w_2^+] \cdot y_p + \dots + [w_i^-; w_i^+] \cdot y_p + \dots + [w_n^-; w_n^+] \cdot y_p \leq y^+ \quad (3.23)$$

Перепишемо умову (3.23) у такому вигляді:

$$[w_1^-; w_1^+] \cdot y_p + [w_2^-; w_2^+] \cdot y_p + \dots + [w_i^-; w_i^+] \cdot y_p + \dots + [w_n^-; w_n^+] \cdot y_p \subseteq [y^-; y^+] \quad (3.24)$$

Тоді з використанням інтервальної арифметики отримаємо:

$$[y_p^-; y_p^+] \cdot ([w_1^-; w_1^+] + [w_2^-; w_2^+] + \dots + [w_i^-; w_i^+] + \dots + [w_n^-; w_n^+]) \subseteq [y^-; y^+] \quad (3.25).$$

Враховуючи, що усі інтервали $[y_p^-; y_p^+], [w_1^-; w_1^+], [w_2^-; w_2^+], \dots, [w_i^-; w_i^+], \dots, [w_n^-; w_n^+]$ мають додатні межі, отримаємо:

$$\begin{aligned} y_p^- \cdot (w_1^- + w_2^- + \dots + w_i^- + \dots + w_n^-) &\geq y^- \\ y_p^+ \cdot (w_1^+ + w_2^+ + \dots + w_i^+ + \dots + w_n^+) &\leq y^+ \end{aligned} \quad (3.26)$$

Звідси

$$\begin{aligned} y_p^- &= \cdot y^- / (w_1^- + w_2^- + \dots + w_i^- + \dots + w_n^-) \\ y_p^+ &= \cdot y^+ / (w_1^+ + w_2^+ + \dots + w_i^+ + \dots + w_n^+) \end{aligned} \quad (3.27)$$

Підставляючи отримане у значення годин, які необхідно виконати науково-педагогічному працівнику для прогнозованого року у виразі (2.22) отримаємо заплановані йому кількість годин в межах кожної групи показників в інтервальному вигляді:

$$\begin{aligned} [x_1^-; x_1^+] &= [y_p^-; y_p^+] \cdot [w_1^-; w_1^+], [x_2^-; x_2^+] = [y_p^-; y_p^+] \cdot [w_2^-; w_2^+], \dots, \\ [x_i^-; x_i^+] &= [y_p^-; y_p^+] \cdot [w_i^-; w_i^+], \dots, [x_n^-; x_n^+] = [y_p^-; y_p^+] \cdot [w_n^-; w_n^+] \end{aligned} \quad (3.24)$$

Тепер для прикладу візьмем інтервальні вагові коефіцієнти, які ми отримали раніше:

$$[w_1^-, w_1^+] = [1.693; 2.637], \quad [w_2^-, w_2^+] = [2.232; 2.940]$$

Обчислимо y_p^- та y_p^+ :

$$\begin{aligned} y_p^- &= \cdot y^- / (w_1^- + w_2^-) = 480 / (1.693 + 2.232) = 121.982 \\ y_p^+ &= \cdot y^+ / (w_1^+ + w_2^+) = 800 / (2.637 + 2.940) = 143.446 \end{aligned}$$

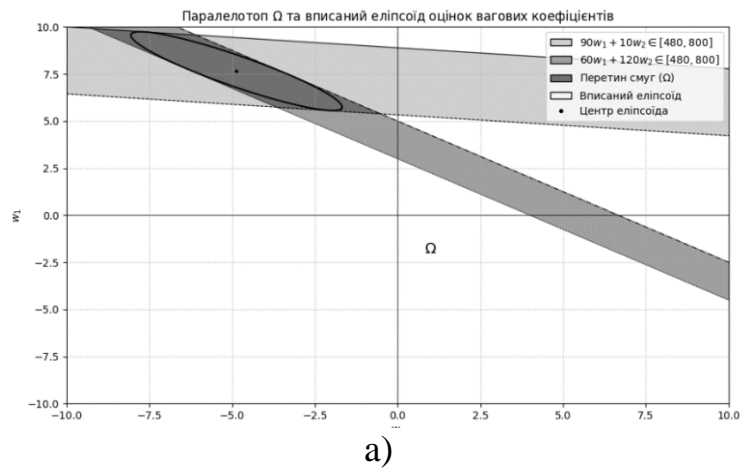
Тоді інтервальний вигляд запланованої кількості годин в межах кожної групи показників буде:

$$[x_1^-; x_1^+] = [y_p^-; y_p^+] \cdot [w_1^-; w_1^+] = [121.982; 143.446] \cdot [1.693; 2.637] = [206.515; 378.267]$$

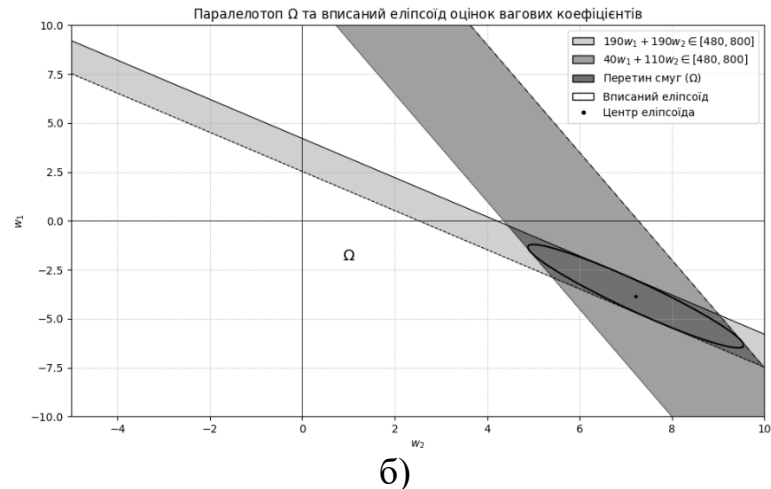
$$[x_2^-; x_2^+] = [y_p^-; y_p^+] \cdot [w_2^-; w_2^+] = [121.982; 143.446] \cdot [2.232; 2.940] = [272.263; 421.731]$$

Розглянемо тепер деякі особливості запропонованого методу та отриманої математичної моделі. Вище отримані результати ґрунтуються на припущенні, що область розв'язків ІСЛАР містить тільки додатні значення вектора вагових коефіцієнтів для кожної групи показників, що характеризує «портрет» науковця.

Разом з тим, можуть бути ситуації, коли деякі коефіцієнти можуть бути від'ємними. Такі випадки показано на рис. 3.4. (а, б).



а)



б)

Рисунок 3.4 – Ілюстрація випадків від'ємності вагових коефіцієнтів для групи показників

Так на рис. 3.4. а) від'ємним є коефіцієнт w_2 , а на рис. 3.4. б) – коефіцієнт w_1 .

В цих випадках означає часткову відсутність закономірності в історичних даних для певних груп показників. Вирішення цієї проблеми полягає у збільшенні вибірки даних, для уточнення «портрета» науковця. Інший підхід полягає в обнуленні від'ємних меж інтервалів (3.25), як це показано на рис.3.5 (а,б,в)

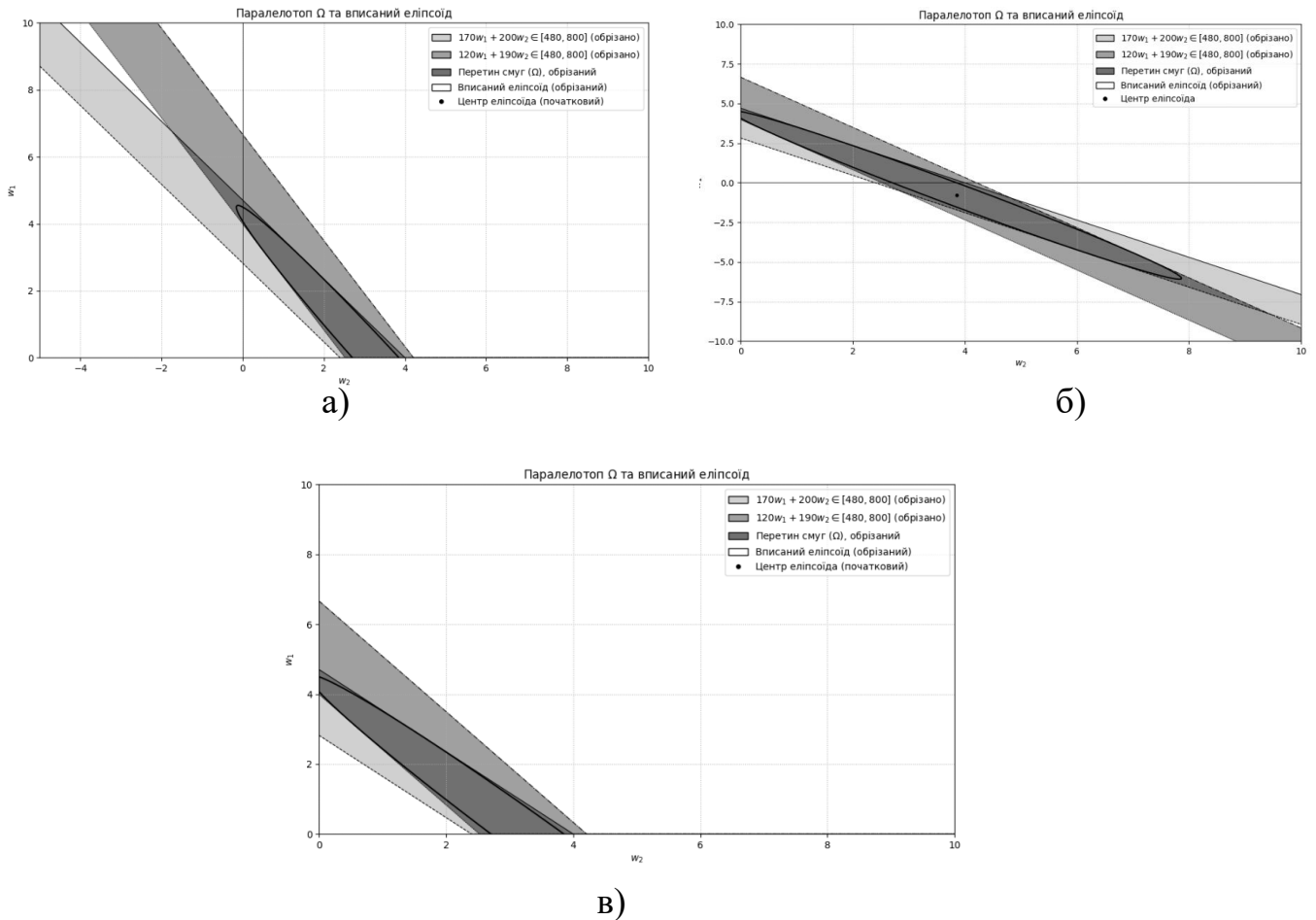


Рисунок 3.5 – Ілюстрація вирішення проблеми випадків від'ємності вагових коефіцієнтів для певної групи показників.

Як бачимо з рис 3.5 а), обнулено нижню межу коефіцієнта. На рис. 3.5 б) – обнулено верхню межу коефіцієнта. На рис. 3.5 в) – обнулено нижню межу коефіцієнта та верхню межу коефіцієнта.

Тепер розглянемо приклад планування науково-педагогічному працівнику наукових показників в межах кожної групи показників в інтервальному вигляді,

спираючись на його «портрет» науковця, який побудуємо на основі історичних даних.

Для ілюстрації роботи запропонованого методу розглянемо випадок, коли кількість функціональних груп показників наукової діяльності дорівнює п'яти. На основі історичних даних одного з науково-педагогічних працівників сформуємо систему алгебраїчних рівнянь для визначення вагових коефіцієнтів, що характеризують його «портрет» науковця:

$$\begin{cases} y_1 = w_1 \cdot 135 + w_2 \cdot 97 + w_3 \cdot 127 + w_4 \cdot 113 + w_5 \cdot 141 \\ y_2 = w_1 \cdot 142 + w_2 \cdot 126 + w_3 \cdot 135 + w_4 \cdot 103 + w_5 \cdot 105 \\ y_3 = w_1 \cdot 129 + w_2 \cdot 106 + w_3 \cdot 106 + w_4 \cdot 128 + w_5 \cdot 144 \\ y_4 = w_1 \cdot 130 + w_2 \cdot 128 + w_3 \cdot 124 + w_4 \cdot 135 + w_5 \cdot 97 \\ y_5 = w_1 \cdot 128 + w_2 \cdot 143 + w_3 \cdot 126 + w_4 \cdot 109 + w_5 \cdot 102 \end{cases}$$

У матричній формі:

$$X = \begin{bmatrix} 135 & 97 & 127 & 113 & 141 \\ 142 & 126 & 135 & 103 & 105 \\ 129 & 106 & 106 & 128 & 144 \\ 130 & 128 & 124 & 135 & 97 \\ 128 & 143 & 126 & 109 & 102 \end{bmatrix},$$

Згідно з нормативними вимогами, інтервальні значення річного наукового навантаження задаються як:

$$\vec{Y}^- = \begin{bmatrix} 480 \\ 480 \\ 480 \\ 480 \\ 480 \end{bmatrix}, \quad \vec{Y}^+ = \begin{bmatrix} 800 \\ 800 \\ 800 \\ 800 \\ 800 \end{bmatrix},$$

Таким чином, область допустимих розв'язків визначається системою:

$$\Omega = \left\{ \vec{w} \in R^2 \left\| \begin{bmatrix} 480 \\ 480 \\ 480 \\ 480 \\ 480 \end{bmatrix} \leq \begin{bmatrix} 135 & 97 & 127 & 113 & 141 \\ 142 & 126 & 135 & 103 & 105 \\ 129 & 106 & 106 & 128 & 144 \\ 130 & 128 & 124 & 135 & 97 \\ 128 & 143 & 126 & 109 & 102 \end{bmatrix} \cdot \vec{w} \leq \begin{bmatrix} 800 \\ 800 \\ 800 \\ 800 \\ 800 \end{bmatrix} \right\}.$$

Оскільки визначник матриці X не дорівнює нулю ($\det(X) = -118273815$), система має єдиний розв'язок для кожного вектора межових значень. Для середнього значення навантаження (640 год) отримуємо:

$$\vec{w} = X^{-1} \cdot \begin{bmatrix} 640 \\ 640 \\ 640 \\ 640 \\ 640 \end{bmatrix} = \begin{bmatrix} -0.06163 & 0.08722 & 0.04628 & -0.01025 & -0.06017 \\ -0.0152 & -0.00805 & 0.00879 & -0.01376 & 0.03041 \\ 0.06365 & -0.04514 & -0.05812 & 0.01526 & 0.02602 \\ 0.00383 & -0.01906 & -0.00223 & 0.02907 & -0.01017 \\ 0.01637 & -0.02204 & 0.00378 & -0.01776 & 0.02141 \end{bmatrix} \cdot \begin{bmatrix} 640 \\ 640 \\ 640 \\ 640 \\ 640 \end{bmatrix} \approx \begin{bmatrix} 0.923 \\ 1.200 \\ 1.070 \\ 0.921 \\ 1.128 \end{bmatrix}$$

Далі, обчислимо коефіцієнт нормування W для переходу від точкових до інтервальних оцінок вагових коефіцієнтів:

$$W = \frac{1}{\sqrt{\vec{\sigma}^T \cdot X^T \cdot E^{-2} \cdot X \cdot \vec{\sigma}}} = \frac{1}{(0.1915; 0.1909; 0.1915; 0.1840; 0.1900) \cdot \begin{pmatrix} 135 & 97 & 127 & 113 & 141 \\ 142 & 126 & 135 & 103 & 105 \\ 129 & 106 & 106 & 128 & 144 \\ 130 & 128 & 124 & 135 & 97 \\ 128 & 143 & 126 & 109 & 102 \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{25600} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{25600} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{25600} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{25600} \end{pmatrix} \cdot \begin{pmatrix} 135 & 97 & 127 & 113 & 141 \\ 142 & 126 & 135 & 103 & 105 \\ 129 & 106 & 106 & 128 & 144 \\ 130 & 128 & 124 & 135 & 97 \\ 128 & 143 & 126 & 109 & 102 \end{pmatrix} \cdot \begin{pmatrix} 0.1915 \\ 0.1909 \\ 0.1915 \\ 0.1840 \\ 0.1900 \end{pmatrix}} = 0.6167$$

Використовуючи це значення, визначаємо інтервальні оцінки вагових коефіцієнтів:

$$[\vec{w}] = \begin{pmatrix} [w_1^-; w_1^+] \\ [w_2^-; w_2^+] \\ [w_3^-; w_3^+] \\ [w_4^-; w_4^+] \\ [w_5^-; w_5^+] \end{pmatrix} = \begin{pmatrix} [0.923 - 0.1915 \cdot 0.6167; 0.923 + 0.1915 \cdot 0.6167] \\ [1.200 - 0.1909 \cdot 0.6167; 1.200 + 0.1909 \cdot 0.6167] \\ [1.070 - 0.1915 \cdot 0.6167; 1.070 + 0.1915 \cdot 0.6167] \\ [0.921 - 0.1840 \cdot 0.6167; 0.921 + 0.1840 \cdot 0.6167] \\ [1.128 - 0.1900 \cdot 0.6167; 1.128 + 0.1900 \cdot 0.6167] \end{pmatrix} = \begin{pmatrix} [0.8049; 1.0411] \\ [1.0823; 1.3177] \\ [0.9519; 1.1881] \\ [0.8075; 1.0345] \\ [1.0108; 1.2452] \end{pmatrix}$$

Далі, нормуємо ці коефіцієнти для прогнозування інтервальних значень річного навантаження:

$$y_p^- = \cdot y^- / (w_1^- + w_2^- + w_3^- + w_4^- + w_5^-) = 480 / (0.8049 + 1.0823 + 0.9519 + 0.8075 + 1.0108) = 103.0618$$

$$y_p^+ = \cdot y^+ / (w_1^+ + w_2^+ + w_3^+ + w_4^+ + w_5^+) = 800 / (1.0411 + 1.3177 + 1.1881 + 1.0345 + 1.2452) = 137.3060$$

Тоді прогнозовані інтервали годин для кожної категорії наукової діяльності:

$$[x_1^-; x_1^+] = [y_p^-; y_p^+] \cdot [w_1^-; w_1^+] = [103.0618; 137.3060] \cdot [0.8049; 1.0411] = [82.9515; 143.0673]$$

$$[x_2^-; x_2^+] = [y_p^-; y_p^+] \cdot [w_2^-; w_2^+] = [103.0618; 137.3060] \cdot [1.0823; 1.3177] = [111.5645; 180.9329]$$

$$[x_3^-; x_3^+] = [y_p^-; y_p^+] \cdot [w_3^-; w_3^+] = [103.0618; 137.3060] \cdot [0.9519; 1.1881] = [98.1130; 163.1092]$$

$$[x_4^-; x_4^+] = [y_p^-; y_p^+] \cdot [w_4^-; w_4^+] = [103.0618; 137.3060] \cdot [0.8075; 1.0345] = [83.2384; 142.9298]$$

$$[x_5^-; x_5^+] = [y_p^-; y_p^+] \cdot [w_5^-; w_5^+] = [103.0618; 137.3060] \cdot [1.0108; 1.2452] = [104.0950; 171.0772]$$

Отримані результати свідчать про можливість планування наукового навантаження в інтервальній формі з урахуванням «портрета» науковця. Це дозволяє керівництву ЗВО розподіляти планові години для кожного працівника, враховуючи його реальну наукову активність та історичні дані.

Висновки до розділу 3

1. Проведено аналіз показників наукової діяльності науково-педагогічних працівників і встановлено, що для побудови «портрета» науковця можна використовувати набір «історичних» даних про діяльність кожного працівника. Також показано, що дані, які можливо використати для моделі прогнозування, є обмежені кількістю років його звітності про наукову діяльність, які зазвичай розміщено в рейтинговій системі науково-педагогічних працівників, а актуальних даних і того менше. Тобто вибірка даних зазвичай складається не більше як з 10 наборів даних. За цих умов запропоновано усі показники наукової діяльності агрегувати у підгрупи, щоб визначити часові витрати для показників агрегованих показників, а потім на цій підставі визначити часові витрати для показників в межах кожної групи. Такий підхід дає можливість у разі зменшити вибірку даних, необхідних для ідентифікації моделі прогнозування наукової діяльності і застосовувати ієрархічну структуру показників [10].

2. Вперше запропоновано та обґрунтовано метод прогнозування показників наукової діяльності науково-педагогічних працівників на основі аналізу інтервальних даних, який на відміну від існуючих ґрунтується на еліпсоїдних оцінках вектора вагових коефіцієнтів для кожної групи показників, що характеризують «портрет» науковця, що у сукупності дає можливість отримати інтервальні оцінки запланованих показників наукової діяльності науково-педагогічного працівника побудованих на мінімальній вибірці даних.

3. Для підтвердження розробленого методу та математичної моделі розглянуто приклад планування науково-педагогічному працівнику наукових показників в межах кожної групи показників в інтервальному вигляді, спираючись на його «портрет» науковця, який побудовано на основі історичних даних.

РОЗДІЛ 4

ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПЛАНУВАННЯ НАУКОВОЇ ДІЯЛЬНОСТІ НАУКОВО-ПЕДАГОГІЧНОЇ ПРАЦІВНИКІВ

У розділі описано основні особливості архітектурного рішення програмної реалізації системи для планування наукової діяльності науково-педагогічних працівників, зокрема з використанням сучасних технологій розробки, таких як Vue.js [99], Node.js, MongoDB та GraphQL, а також багаторівневої архітектури, що забезпечує високу масштабованість, гнучкість та надійність роботи прикладного програмного забезпечення. Система побудована за принципами поділу на клієнтську та серверну частини, що забезпечує незалежність компонентів, спрощує розробку, обслуговування та розширення функціональності.

Для перевірки системи було проведено інтеграційне тестування та тестування продуктивності. В межах інтеграційного тестування здійснено перевірку взаємодії між окремими модулями та компонентами системи, що дозволило підтвердити коректність роботи їх інтерфейсів та забезпечити цілісність програмного рішення. При тестуванні продуктивності були обрані типові сценарії використання системи, такі як авторизація користувачів, збір та імпорт наукових публікацій з зовнішніх баз даних, генерація звітів про наукову діяльність, аналітична обробка інформації для формування профілів науковців, а також виконання запитів через інтерфейс GraphQL.

Крім того, проведено функціональне тестування програмного інтерфейсу за такими категоріями: позитивні тести для типових сценаріїв роботи, негативні тести із валідними та невалідними вхідними даними, деструктивні тести для перевірки стійкості до нестандартних ситуацій, а також спеціальні тести з питань безпеки та авторизації. За результатами тестування було встановлено, що система відповідає заявленим вимогам щодо продуктивності та стабільності, демонструє високу швидкість обробки запитів навіть за умови великої кількості

одночасних звернень, а також ефективно масштабується за необхідності подальшого зростання кількості користувачів та обсягу оброблюваних даних. Це підтверджує готовність системи до впровадження в умовах реальної експлуатації у закладах освіти.

У кінці розділу здійснено опис графічного інтерфейсу програмного середовища. Наведено основні екрани, які ілюструють процес авторизації користувачів, роботу з головною панеллю управління, механізми формування аналітичних звітів, а також перегляд профілів науковців. Особливу увагу приділено інтуїтивній зрозумілості інтерфейсу та ергономічності його елементів, що забезпечує швидке освоєння системи користувачами навіть без спеціальної підготовки.

4.1. Архітектура програмної системи для планування наукової діяльності науково-педагогічних працівників

Ефективне вирішення завдань зі збору, уніфікації, аналізу та прогнозування наукової діяльності потребує такої архітектури програмної системи, яка забезпечує надійну інтеграцію з різнотипними зовнішніми джерелами, масштабовану обробку великих обсягів даних, підтримку інтелектуальних методів аналізу та високопродуктивний доступ користувачів до аналітичних результатів. У зв'язку з цим система побудована як сервісно-орієнтована (SOA) багатокomпонентна інфраструктура, у межах якої кожен функціональний модуль виконує окрему роль, але залишається логічно узгодженим у єдиному інформаційному потоці [10, 47, 95]. На рисунку 4.1 наведено загальну структуру архітектури, що відображає взаємодію між її ключовими секціями.

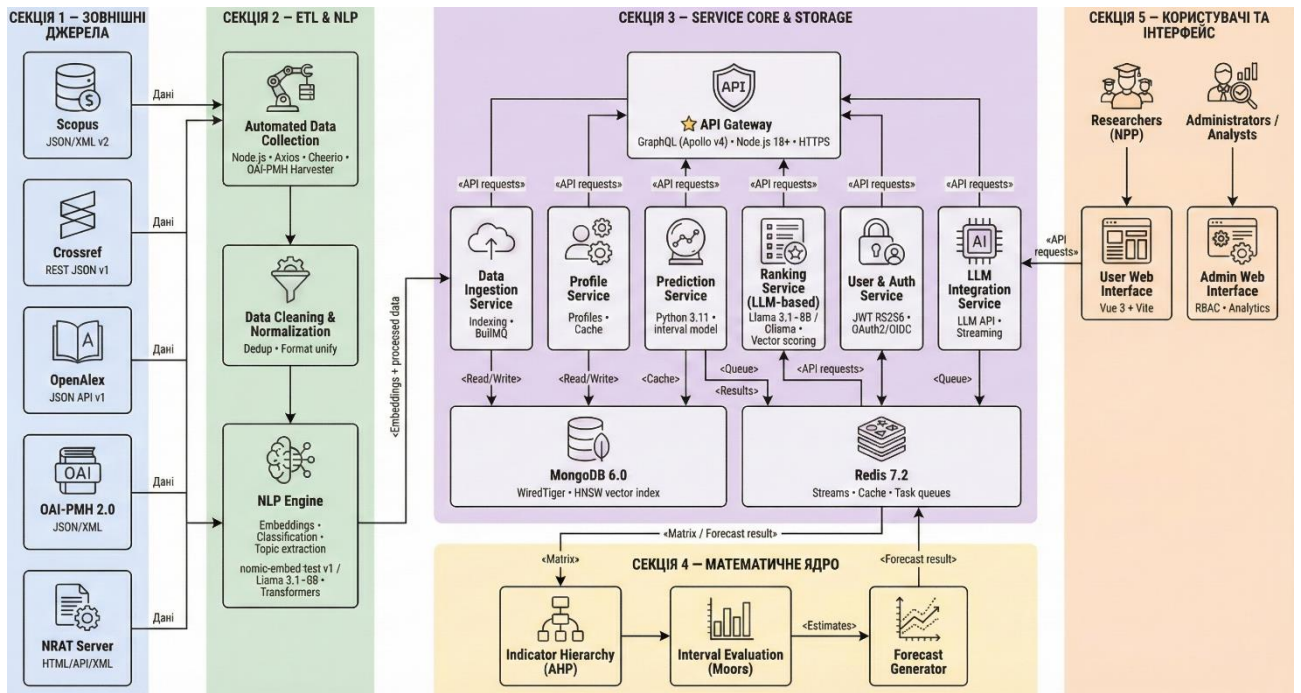


Рисунок 4.1 – Загальна архітектура системи

Першу секцію формують зовнішні наукометричні та бібліографічні джерела, з яких систему постачають структурованими та напівструктурованими даними. До таких джерел належать Scopus, Crossref, OpenAlex, OAI-PMH-сумісні репозитарії та NRAT Server. Вони повертають метадані у форматах JSON, XML або HTML через REST та OAI-PMH API. Завдяки цьому забезпечується широке охоплення наукових результатів різного типу — статей, тез, дисертацій, проєктів та інших форм наукової активності..

Наступний етап реалізовано в межах секції ETL та NLP, де відбуваються збирання, очищення та уніфікація отриманої інформації. Автоматизований модуль збору даних, побудований з використанням Node.js, Axios, Cheerio та OAI-PMH-харвестера, виконує гармонізацію вхідних потоків і виділення релевантних метаданих. Подальше очищення та нормалізація забезпечують дедублікацію, усунення структурних невідповідностей і формальну узгодженість даних у межах внутрішньої моделі системи. Ключовим інтелектуальним компонентом секції є NLP-двигун, який формує семантичні ембединги, здійснює тематичну класифікацію та витягування ключових понять

на основі моделей `nomic-embed-text` і `Llama 3.1-8b` [55, 68, 89, 90]. Цей етап забезпечує переходи від сирих текстових даних до семантично збагачених інформаційних представлень, придатних для подальшого аналітичного опрацювання.

Секція сервісного ядра та сховища даних становить центральну частину архітектури. Координацію взаємодій здійснює `API Gateway`, реалізований на графовому інтерфейсі `GraphQL (Apollo v4)`, що забезпечує маршрутизацію запитів, агрегацію даних та стандартизований доступ до функціональних мікросервісів [32, 56, 132, 134]. До таких мікросервісів належать модуль індексування та завантаження даних, сервіс побудови профілів дослідників, модуль прогнозування на основі інтервальної моделі, сервіс ранжування публікацій із використанням великих мовних моделей, підсистема автентифікації користувачів на базі `JWT` і `OAuth2/OIDC` [31, 60, 112, 120], а також сервіс `LLM-інтеграції` для потокового виконання високорівневих інтелектуальних операцій. Центральним сховищем є `MongoDB 6.0`, що підтримує векторні `HNSW`-індекси для прискореного семантичного пошуку, а `Redis 7.2` забезпечує високу продуктивність завдяки механізмам кешування, стрімів і черг задач [63, 76].

Четверта секція — математичне ядро програмної системи — реалізує алгоритми формування інтегральних наукових показників і прогнозних оцінок. Ієрархічна модель показників ґрунтується на методі аналізу ієрархій (`АНР`), що дозволяє визначати вагомість різних категорій наукової діяльності у загальному обчисленні індивідуального наукового навантаження. Інтервальна модель `Мoorca` забезпечує врахування невизначеності в даних і формує коректні діапазони можливих значень. Генератор прогнозів об'єднує результати математичного моделювання, ембеддинги та профільні метадані, отримані з попередніх секцій, що дозволяє отримувати обґрунтовані оцінки майбутньої динаміки наукової активності[124].

Фінальна секція орієнтована на користувачів і забезпечує доступ до функціональних можливостей системи через веб-інтерфейси. Інтерфейс

дослідника реалізований на основі Vue 3 та Vite, що забезпечує високу реактивність, динамічне оновлення даних та адаптивність до різних пристроїв. Адміністративний інтерфейс підтримує рольову модель доступу (RBAC) і надає розширені можливості для аналізу, керування даними та моніторингу процесів синхронізації. Комунікація між фронтендом і серверною частиною виконується виключно через GraphQL-запити, що зменшує надлишковість переданих даних і забезпечує високу продуктивність взаємодії.

Визначені архітектурні компоненти та їхні взаємозв'язки формують концептуальну основу для переходу до аналізу функціональної поведінки системи. На цьому етапі важливо окреслити, які саме дії користувачі можуть виконувати в межах відповідних модулів та які результати очікуються від їхньої взаємодії з програмним середовищем.

У практиці проектування інформаційних систем для формального представлення функціональних вимог широко застосовується уніфікована мова моделювання (UML) [125]. Одним із її ключових інструментів є діаграми варіантів використання (use case diagrams), що дозволяють структуровано подати основні сценарії роботи системи та визначити коло користувачів, залучених до виконання цих сценаріїв. Таке моделювання сприяє систематизації вимог і забезпечує узгодженість між очікуваною поведінкою системи та її архітектурними рішеннями [62, 64].

Наукові дослідження, зокрема праця Ozkaya M. та Romeo J., підтверджують ефективність UML у моделюванні прикладних задач та проектуванні інформаційних систем, що істотно спрощує етапи кодування та реалізації програмного забезпечення [88, 102].

У контексті розроблюваної системи для збору та аналізу наукової та науково-педагогічної діяльності академічного колективу діаграми варіантів використання дають змогу чітко окреслити ролі користувачів та їхні цілі, визначити межі відповідальності кожного компонента та забезпечити послідовність у формуванні функціональних вимог. Це створює структуровану

основу для подальшого проектування та підвищує зрозумілість роботи системи для різних категорій користувачів [96, 141].

Програмна система для збору та аналізу наукової та науково-педагогічної діяльності академічного колективу містить широкий спектр функціональних можливостей, які визначаються відповідно до ролей користувачів. Основними акторами системи є:

- співробітник НДЧ (науково-дослідної частини);
- співробітник відділу аспірантури;
- викладач;
- завідувач кафедри.

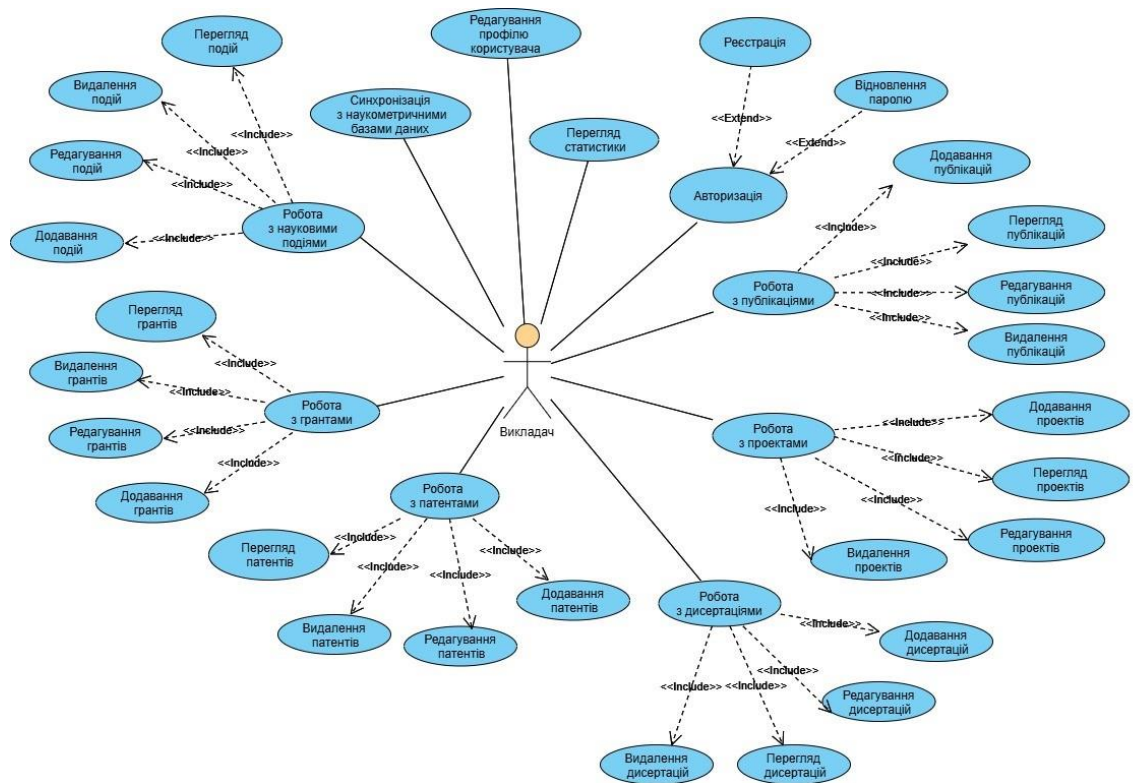


Рисунок 4.2 – Діаграма варіантів використання користувача «Викладач» в системі

На рисунку 4.2. зображено діаграму варіантів використання для ролі «Викладач». Йому надається можливість переглядати, редагувати та додавати інформацію про власні публікації, проекти, дисертації, гранти й патенти. Також

викладач може керувати інформацією про свою участь у наукових заходах та аналізувати статистику власної дослідницької діяльності.

На відміну від викладача, завідувач кафедри має розширені права доступу. Він може переглядати всі наукові дані кафедри та формувати відповідні звіти про її діяльність. Це включає аналіз наукової продуктивності викладачів кафедри, моніторинг участі в дослідницьких проєктах і міжнародних грантах(Рис.4.3).

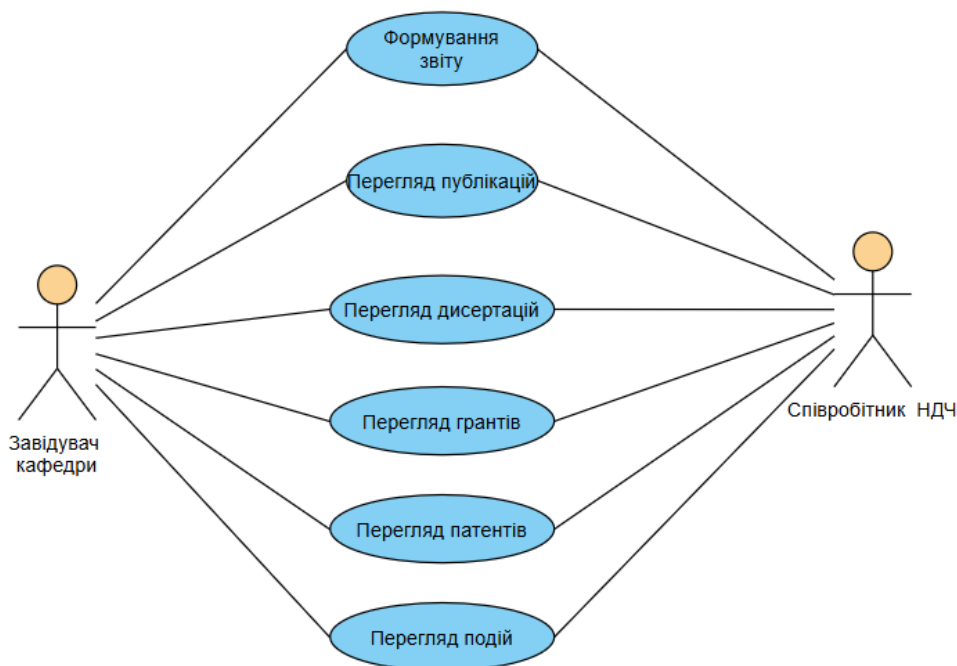


Рисунок 4.3 – Діаграма варіантів використання для ролей «Завідувач кафедри» та «Співробітник НДЧ»

Співробітники НДЧ здійснюють адміністрування наукової діяльності, включаючи створення та аналіз звітів, перегляд наукових публікацій, дисертацій, грантів і патентів, а також управління інформацією про наукові події.

Співробітники відділу аспірантури працюють з аспірантами та їх науковими керівниками, забезпечуючи контроль за навчальним процесом та здійснюючи керування базою даних аспірантів(Рис.4.4).

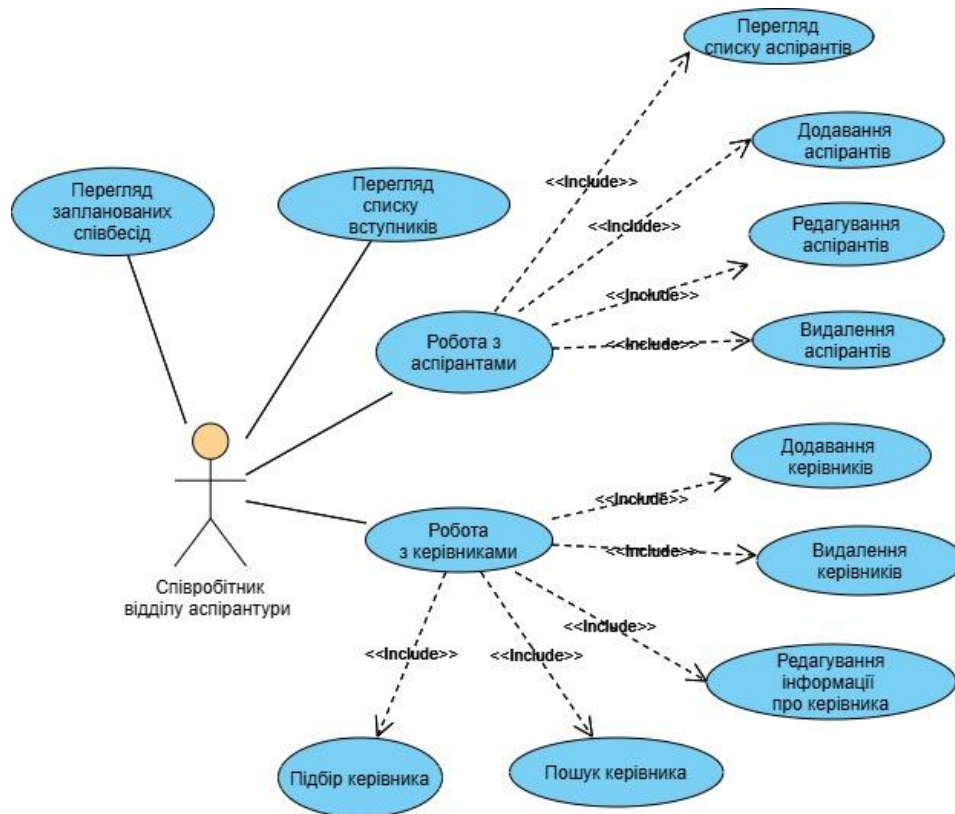


Рисунок 4.4 – Діаграма варіантів використання для ролі «Співробітник відділу аспірантури»

У процесі автоматизації збору та аналізу наукових публікацій для формування профілю дослідника важливо забезпечити ефективний механізм автентифікації користувачів, отримання релевантних даних та їх обробки. Взаємодія між викладачем, клієнтським інтерфейсом, серверною частиною, базами даних та зовнішніми джерелами інформації має бути чітко визначеною та синхронізованою. На рисунку 4.5 представлено діаграму послідовності для процесу взаємодії користувача з системою, починаючи від автентифікації та запиту на синхронізацію публікацій до отримання проранжованих наукових записів з відповідними коефіцієнтами релевантності.

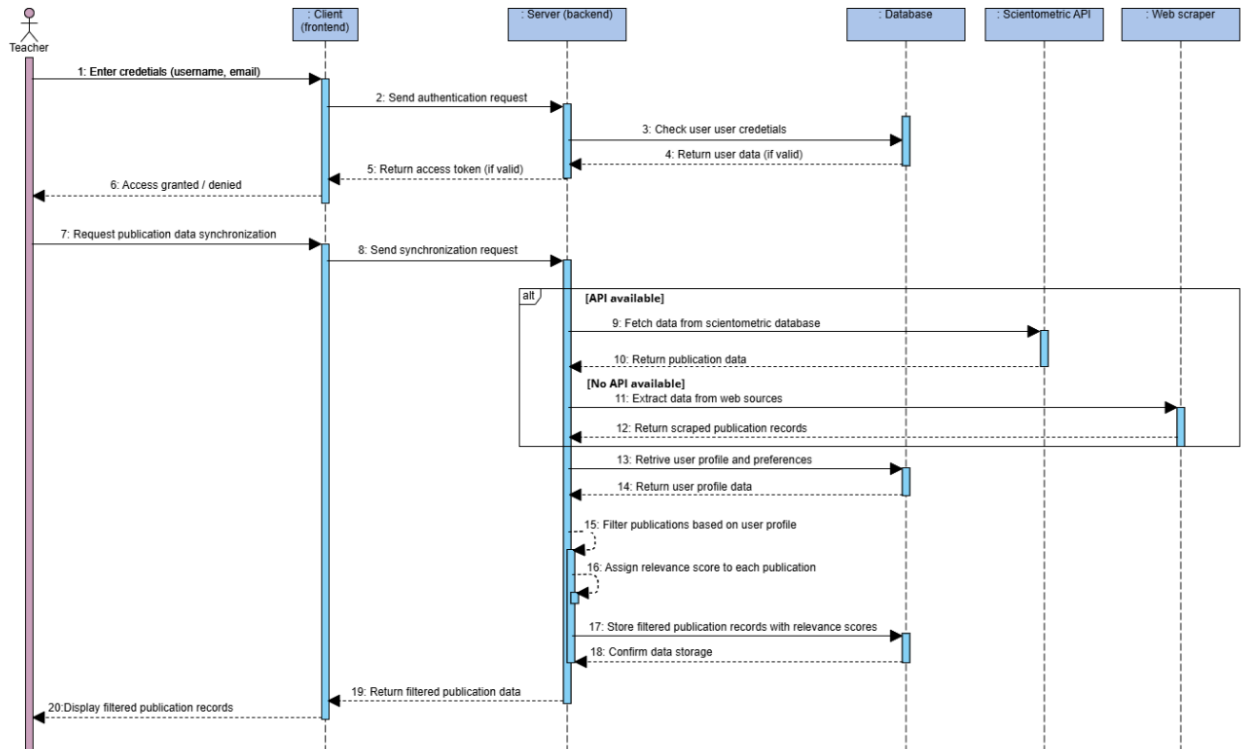


Рисунок 4.5 – Діаграма послідовності для користувача системи з ролю «Викладач»

На рисунку 4.6 представлено діаграму послідовності взаємодії завідувача кафедри або працівника НДЧ з системою, що демонструє ключові етапи роботи з базою даних. Згідно з даною діаграмою, процес починається з авторизації, під час якої система перевіряє облікові дані користувача через звернення до бази даних. Після успішного проходження аутентифікації завідувач або співробітник НДЧ має можливість ініціювати запит на перегляд публікацій, при цьому всі необхідні дані отримуються безпосередньо з бази даних. Формування загального звіту також здійснюється шляхом запиту до бази даних, що дозволяє агрегувати дані про публікаційну активність.

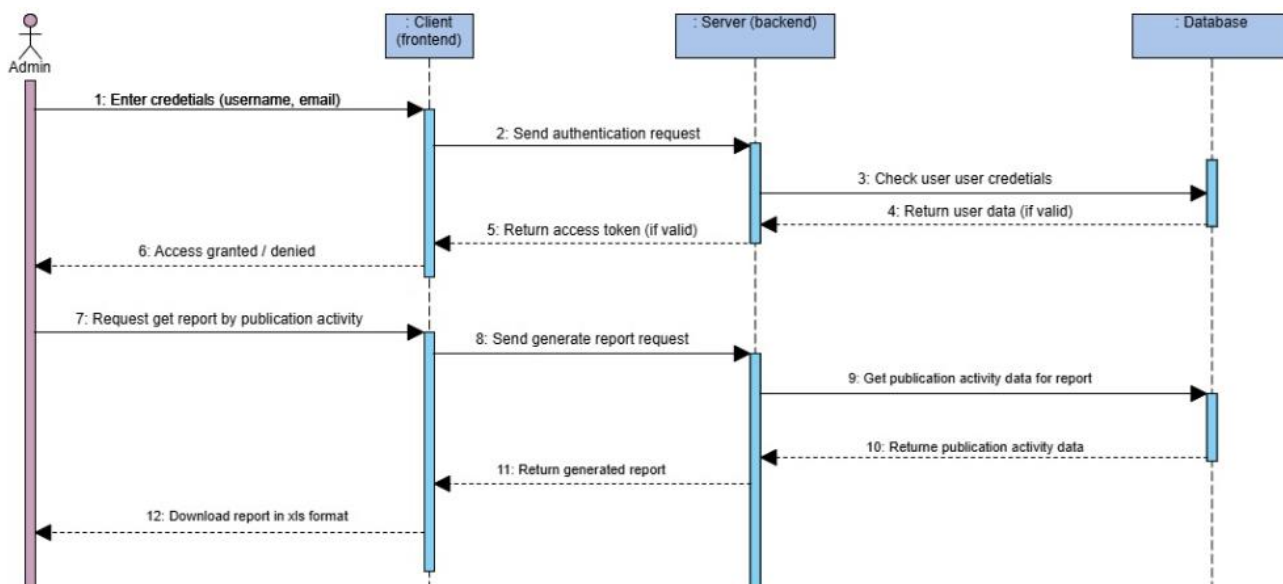


Рисунок 4.6 – Діаграма послідовності для користувача з роллю «Завідувач кафедри» або «Співробітник НДЧ»

На рисунку 4.7 зображено діаграму послідовності взаємодії співробітника відділу аспірантури з системою. Співробітник відділу аспірантури виконує низку операцій, що забезпечують адміністрування даних, пов'язаних з процесом прийому та навчання аспірантів. Зокрема, він має можливість переглядати заплановані співбесіди, що дозволяє організувати та оперативно реагувати на зміни в розкладі. Крім того, співробітник може переглядати список вступників, що подали документи для вступу в аспірантуру, та здійснювати детальний аналіз контактної та освітньої інформації.

Окрім цього, функціональна роль співробітника включає додавання, редагування та видалення даних аспірантів для підтримання актуальності інформації. З цією метою також передбачено управління науковими керівниками: додавання нових, редагування існуючих даних щодо керівників та їхнє призначення на відповідні наукові завдання.

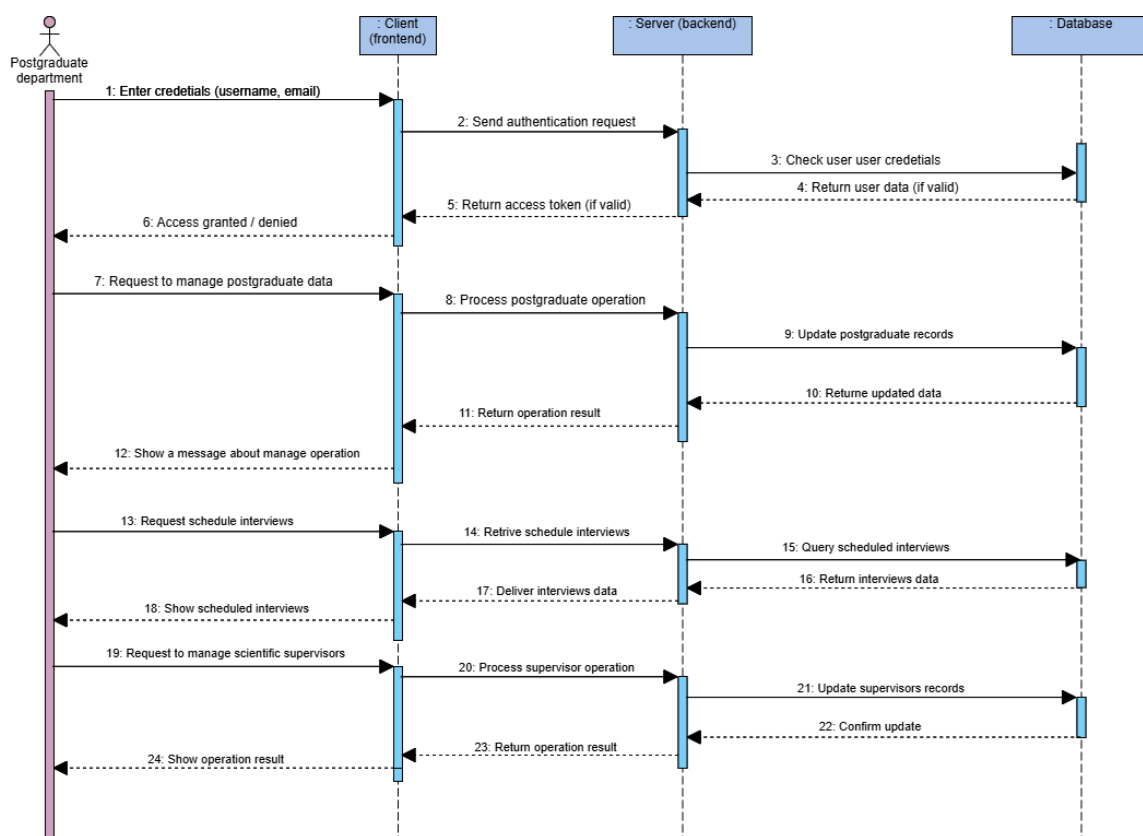


Рисунок 4.7 – Діаграма послідовності для користувача з роллю «Співробітник відділу аспірантури»

Модуль збору інформації про наукову та науково-технічну діяльність академічного колективу є однією з ключових особливостей системи, оскільки забезпечує автоматизований збір, обробку та інтеграцію даних, що дозволяє оперативно відстежувати поточний стан досліджень, проєктів і публікацій для прийняття обґрунтованих управлінських рішень.

На рисунку 4.8 наведено діаграму діяльності процесу збору інформації про наукову та науково-технічну діяльність академічного колективу, що включає кілька послідовних етапів. Спочатку здійснюється вибір цільової веб-сторінки, далі виконується парсинг та визначення структурних елементів, що дає змогу виокремити метадані. Отримані дані підлягають розпізнаванню тексту з метою подальшого аналізу і класифікації, у результаті чого формується структурована інформація про доповіді, проєкти, публікації та патенти. Завершується процес

збереженням оброблених даних у базі та відображенням отриманих результатів для подальшого використання й аналізу.

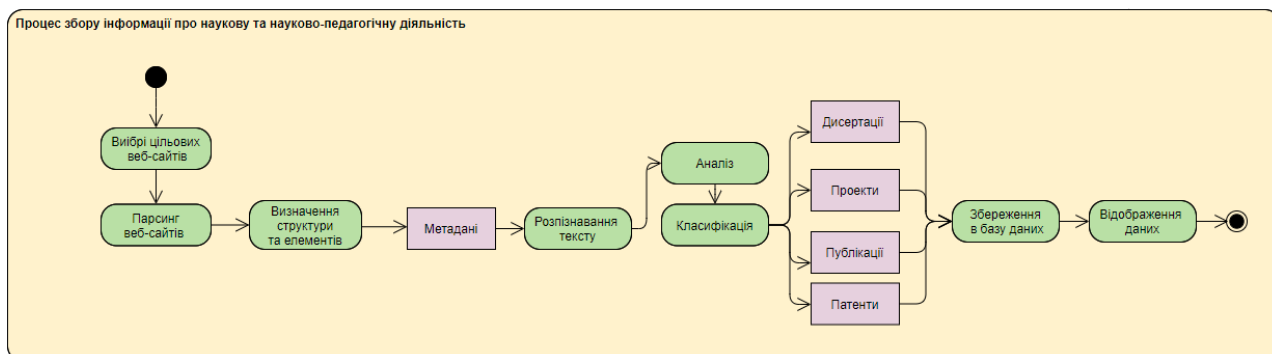


Рисунок 4.8 – Діаграма діяльності процесу збору інформації про наукову та науково-технічну діяльність академічного колективу

Як видно із діаграм, для ефективного збору даних, інтелектуальний модуль спочатку аналізує обрані веб-сайти, використовуючи розроблені алгоритми парсингу[26]. Це дозволяє автоматично ідентифікувати та витягувати необхідні дані, такі як метадані публікацій, інформацію про авторів, проекти та патенти[3]. Використання штучного інтелекту на етапі обробки даних дозволяє не тільки розпізнавати текстові елементи з високою точністю, але й класифікувати зібрані дані, виділяючи найважливішу інформацію. Моделі машинного навчання допомагають визначити контекст зібраних даних, що є важливим для подальшого наукового аналізу та досліджень[129].

Після завершення збору та обробки даних, інформація систематизується та зберігається у базі даних, з якої вона може бути легко доступна через графічний інтерфейс. Цей інтерфейс дозволяє користувачам швидко знаходити необхідну інформацію, використовуючи фільтри за авторами, датами публікації, ключовими словами або іншими параметрами.

На рисунку 4.9 розглянуто мультисервісну архітектуру веб-додатку в вигляді діаграми розгортання, яка розподіляється на три основні сегменти: клієнтський інтерфейс, сервер додатків та база даних. Кожен з цих сегментів виконує визначені функції у межах загальної інфраструктури, забезпечуючи гнучкість, масштабованість та високу доступність сервісу.

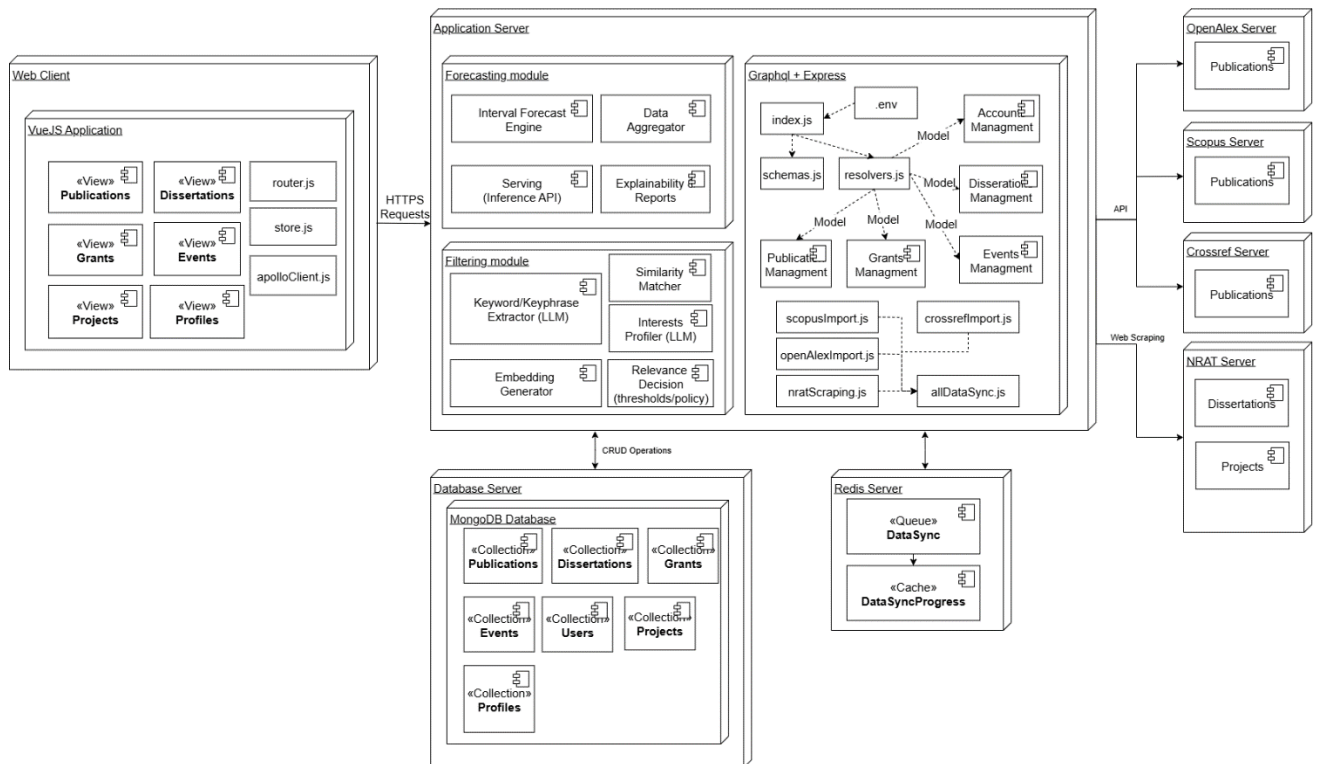


Рисунок 4.9. Діаграма розгортання системи

Клієнтська частина (Рис. 4.10) реалізована на фреймворку Vue.js та включає модульну структуру представлень (Views), кожна з яких призначена для взаємодії з певним сегментом даних: публікації, дисертації, гранти, події та проекти. Клієнтський роутер (router.js) і система управління станом (store.js) відповідають за навігацію та консистентність стану веб-додатку в реальному часі. Apollo Client інтегрований для забезпечення зв'язку з GraphQL API, що впроваджений на сервері додатків.

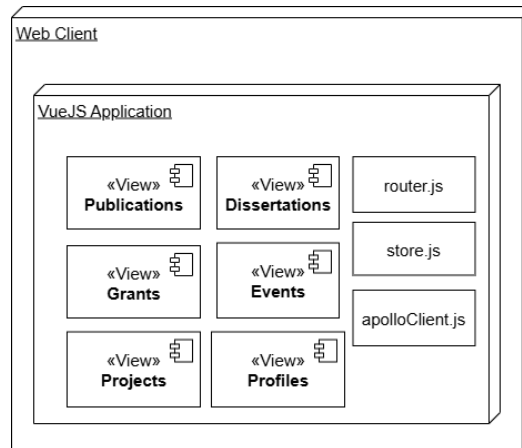


Рисунок 4.10 – Компонент клієнтської частини

Серверна частина (Рис. 4.11) базується на Node.js з використанням фреймворку Express та GraphQL для організації API, яке обслуговує HTTP-запити від клієнтської частини. Конфігурація сервера та його змінні оточення знаходяться у файлі .env, а точка входу в систему визначена в index.js. Моделі даних (Model*) відповідають за визначення структури даних та їх взаємодію з базою даних MongoDB. Управлінські модулі, такі як Account, Dissertation, Grants, та Events Management, відповідають за бізнес-логіку, що пов'язана з кожним із цих елементів. Резольвери GraphQL (resolvers.js) визначають методи для обробки запитів та мутацій.

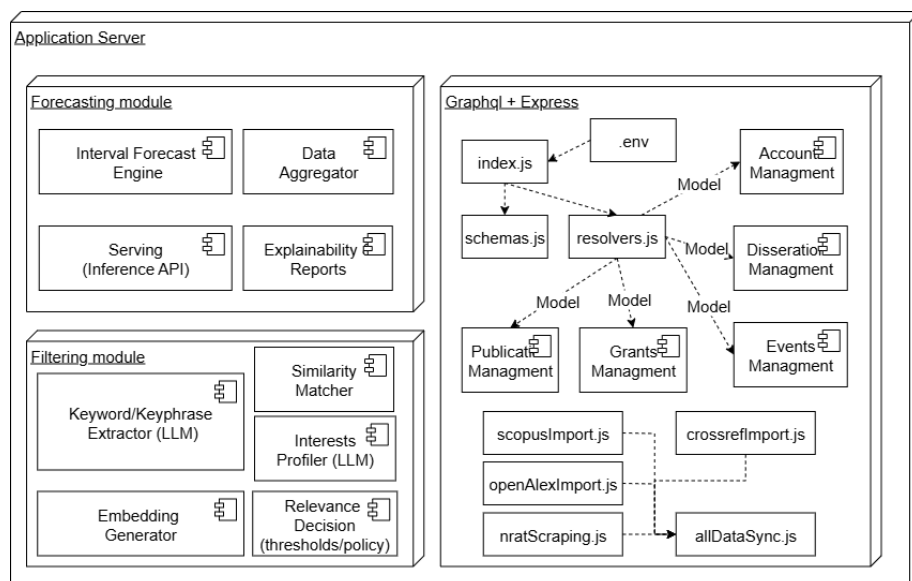


Рисунок 4.11 – Компонент серверної частини веб-додатку

Додатково, інтегровано Redis Server, який використовується як брокер повідомлень для управління чергою задач, зокрема для задач, що стосуються обробки публікацій. Це сприяє розподіленню навантаження та оптимізації продуктивності системи [104].

База даних, реалізована з використанням MongoDB, представляє собою NoSQL-рішення, яке містить колекції для кожного з типів даних (Рис. 4.12), забезпечуючи швидкий і гнучкий доступ до даних. Операції CRUD відіграють ключову роль у маніпуляції даними.

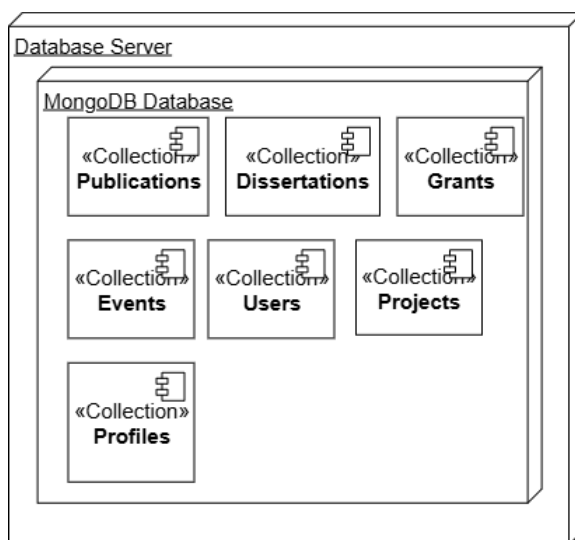


Рисунок 4.12 – Компонент бази даних веб-додатку

Додаток інтегровано з зовнішніми науковими базами даних (Рис. 3.12) через Scopus Server, OpenAlex Server та Crossref Server, що забезпечує імпорт структурованих даних про публікації за допомогою модулів `scopusImport.js`, `crossrefImport.js` та `openAlexImport.js` відповідно. Використання цих сервісів дозволяє отримувати високоякісні метадані у стандартизованих форматах, що суттєво підвищує точність подальшої обробки та аналізу наукової діяльності. На відміну від них, взаємодія з NRAT Server здійснюється шляхом веб-парсингу, оскільки цей ресурс не підтримує API-доступ. Модуль `nratScraping.js` реалізує механізми вилучення та нормалізації інформації про дисертації й наукові проекти, представлені у національних реєстрах, що дозволяє доповнити профіль

науковця даними про результати дисертаційних досліджень та участь у проєктній діяльності.

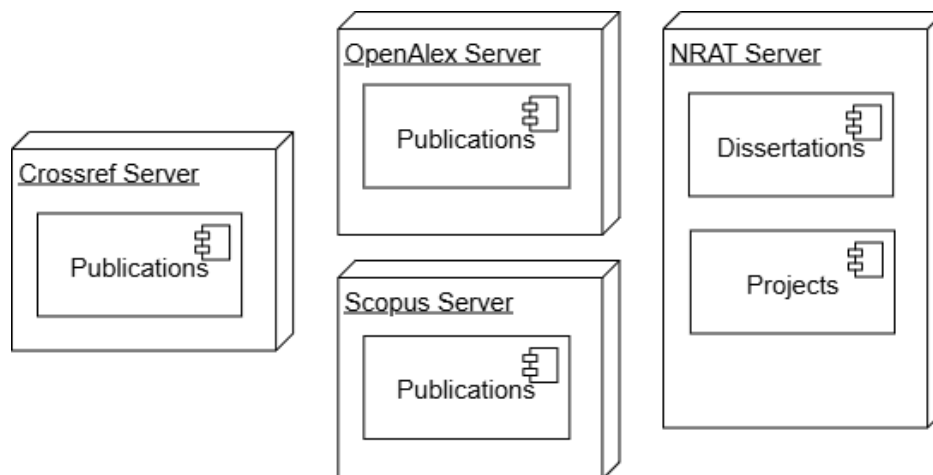


Рисунок 4.13 – Компоненти зовнішніх наукометричних баз даних

Ключовим елементом інфраструктури оброблення даних, зображеної на рисунку 4.14, є Redis Server, який виконує функції високопродуктивного оперативного сховища та механізму керування потоками синхронізації. На відміну від постійної документної бази MongoDB, Redis працює в режимі in-memory, що забезпечує надзвичайно низьку затримку доступу та високу швидкість виконання операцій, критичних для процесів імпорту, оновлення та узгодження наукових даних із зовнішніх джерел. У межах системи реалізовано двокомпонентну модель використання Redis: черга DataSync відповідає за послідовність, розподіл навантаження та керування паралельними задачами інтеграції з таких сервісів, як Scopus, Crossref, OpenAlex та NRAT, тоді як кеш DataSyncProgress забезпечує постійне відстеження стану виконання кожного процесу, фіксує метрики тривалості, кількість оброблених записів та службові повідомлення. Така організація дає змогу підвищити надійність роботи системи, гарантувати коректне завершення тривалих імпортних операцій, уникати блокувань і дублювання даних, а також забезпечувати адміністративним модулям можливість моніторингу й аналізу прогресу синхронізації в режимі реального часу. Включення Redis Server у загальну архітектуру значно підсилює

масштабованість рішення та оптимізує обчислювальні ресурси під час інтенсивних аналітичних і сервісних запитів.

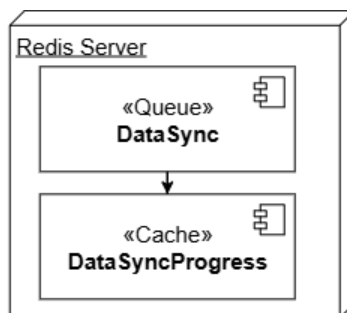


Рисунок 4.14 – Компонент Redis Server із реалізацією черги DataSync та кешу DataSyncProgress

Узагальнюючи, представлена діаграма розгортання відображає складну багаторівневу архітектуру веб-додатку, орієнтовану на забезпечення ефективної взаємодії між клієнтським інтерфейсом, серверною логікою та базою даних, а також зовнішніми даними

4.2. Підсистема аналізу та зберігання даних

Підсистема аналізу та зберігання даних є невід’ємною складовою загальної архітектури системи, що забезпечує збір, обробку та зберігання інформації про публікаційну активність. Основним компонентом цієї підсистеми є база даних MongoDB, яка обрана завдяки своїй високій масштабованості, гнучкості та здатності ефективно працювати з неструктурованими даними [23, 79].

MongoDB реалізує модель зберігання даних у вигляді документів формату JSON, що дозволяє безпосередньо зберігати ієрархічні структури даних [48, 57, 81, 83, 133]. Такий підхід значно спрощує процес інтеграції нових типів інформації та адаптацію системи до змінних вимог дослідження публікаційної активності. Схема бази даних має низку ключових особливостей:

- *гнучкість схеми*: відсутність строго визначеної схеми дозволяє зберігати різномірні дані, що є важливим при роботі з великим обсягом інформації з різних джерел;
- *масштабованість*: розподілена архітектура MongoDB забезпечує горизонтальне масштабування, що дозволяє збільшувати обсяг оброблюваних даних без втрати продуктивності;
- *висока доступність*: механізми реплікації забезпечують збереження даних у випадку збоїв, що гарантує надійність та безперервність роботи системи.

На рисунку 4.15 представлено загальну ER діаграму бази даних, яка використовується в системі збору та аналізу публікаційної активності академічного колективу.

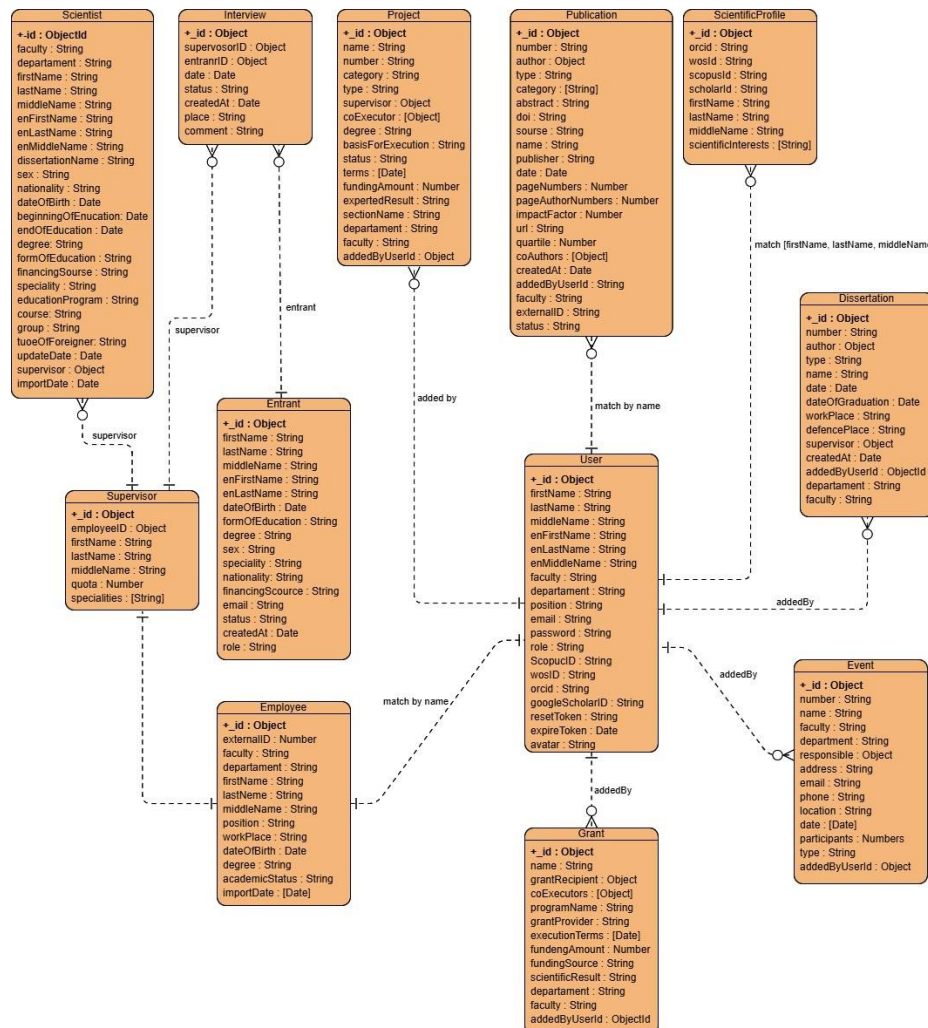


Рисунок 4.15 – ER-діаграма бази даних системи для збору та аналізу публікаційної активності академічного колективу

Розроблена база даних призначена для управління інформацією про наукову діяльність, освітній процес та організаційну взаємодію в академічному середовищі. Вона включає такі ключові сутності, як дисертації, наукові публікації, заходи, гранти, співробітники, студенти, наукові керівники та користувачі.

Кожна сутність містить набір атрибутів, що визначають її унікальні характеристики, та має чіткі взаємозв'язки з іншими відношеннями, які відображають реальні взаємодії між науковцями, студентами, дослідницькими проектами, публікаціями та фінансуванням.

4.3. Особливості програмної реалізації системи для збору та аналізу наукової та науково-педагогічної діяльності академічного колективу

Програмна реалізація системи має багаторівневу архітектуру, яка поділяє додаток на чітко визначені клієнтську та серверну частини, що забезпечує зручність у розробці, підтримці та масштабуванні рішення. Структура серверної частини організована за модульним принципом, що передбачає чітке розмежування на схеми (schemas), резолвери (resolvers) та контролери (controllers). Це дозволяє ефективно керувати бізнес-логікою, забезпечувати гнучкість запитів за допомогою GraphQL [128] та оптимізувати взаємодію з базою даних MongoDB.

У розробці використано сучасні технології: Vue.js для реалізації зручного і реактивного інтерфейсу користувача, Node.js як платформу для серверної частини з підтримкою асинхронних операцій, MongoDB [35] для гнучкого та ефективного зберігання інформації, GraphQL для створення запитів з високою точністю отримуваних даних, а також Redis для ефективного керування чергами та організації асинхронних процесів.

Одним із ключових компонентів системи є механізм авторизації та автентифікації користувачів, який реалізований з використанням JSON Web Tokens (JWT) (Рис. 4.16.). Це дозволяє забезпечити безпечний доступ до системи та захист персональних даних:

```
login: async (_, args) => {
  const { email } = args.input;

  const result = await User.findOne({ email }).lean();
  let entrantLogin = null;

  if (!result) {
    entrantLogin = await Entrant.findOne({ email }).lean();
  }

  if (!result && !entrantLogin) {
    throw new Error("Неправильний пароль [абс] email");
  }

  const isValidPassword = await verifyPassword(result ? result.password : entrantLogin.password, args.input.password);

  if (!isValidPassword) {
    throw new Error("Неправильний пароль [абс] email");
  }

  const { _id,
    password,
    ...other
  } = result ? result : entrantLogin

  const token = result ?
    signToken({ userId: _id, role: result.role, department: result.department }) :
    signToken({ userId: _id, role: entrantLogin.role });

  return {
    user: {
      id: _id,
      ...other
    },
    token,
  };
};
```

Рисунок 4.16 – Лістинг коду автентифікації користувача в системі

Даний метод перевіряє користувача за email та паролем, шукаючи його серед двох категорій користувачів (наприклад, співробітників та вступників). Після успішної перевірки паролю генерується JWT-токен з відповідними правами доступу та додатковими даними користувача, які використовуються для авторизації подальших запитів до системи.

На рисунку 4.17 наведено фрагмент реалізації процесу реєстрації та авторизації користувачів.

```

registration: async (root, { input }) => {
  const { password, email, ...other } = input;
  const userExist = await User.findOne({ email });
  if (!userExist) {
    const hashedPassword = await hashPassword(password);
    const newUser = new User({
      ...other,
      email,
      password: hashedPassword,
      role: "TEACHER"
    });

    return new Promise((resolve, reject) => {
      newUser.save((err) => {
        if (err) reject(err);
        else resolve({
          user: {
            ...other,
            email,
            id: newUser._id,
          },
          token: signToken({ userId: newUser._id, role: newUser.role, department: newUser.department })
        });
      })
    })
  }
  throw new Error("Користувач з такою електронною поштою вже зареєстрований");
},

```

Рисунок 4.17 – Лістинг коду реєстрації та авторизації користувача

Цей метод дозволяє створити нового користувача з роллю викладача (TEACHER), хешує пароль для захисту даних, зберігає користувача у базі даних та генерує JWT-токен для подальшої авторизації.

У системі також реалізовано механізм авторизації на основі бібліотеки graphql-shield, яка є middleware для перевірки прав доступу до різних запитів GraphQL (Рис. 4.18.).

```

import { shield, and, or } from 'graphql-shield';
import { isAuthorized, isAdmin, isEntrant, isTeacher, isHeadOfDepartment, isScientificResearchUnit, isDepartmentOfPostgraduateStudies } from './rules/index.js';

const isCommonRole = or(isTeacher, isHeadOfDepartment, isScientificResearchUnit, isDepartmentOfPostgraduateStudies);

export const permissions = shield({
  Query: {
    getAllUsers: and(isAuthorized, isAdmin),
    getPublications: and(isAuthorized, isCommonRole),
    // інші правила доступу
  },
  Mutation: {
    deleteUser: and(isAuthorized, isAdmin),
    updateUser: and(isAuthorized, or(isTeacher, isHeadOfDepartment, isScientificResearchUnit)),
    // інші правила доступу
  },
});

```

Рисунок 4.18 – Лістинг коду перевірки прав доступу до запитів

Цей механізм гарантує, що кожен запит користувача відповідає його ролі та правам, забезпечуючи додатковий рівень безпеки даних та коректності функціонування системи.

Ще одним важливим модулем системи є модуль синхронізації даних із наукометричними базами даних (наприклад, Scopus, Web of Science, HPAT, Crossref). Даний модуль реалізований із залученням черг на основі Redis, що дозволяє оптимізувати роботу системи, ефективно обробляти складні й довготривалі асинхронні задачі, забезпечуючи надійність та стабільність роботи. На рисунку 4.19 наведено фрагмент коду, що демонструє роботу з чергами для виконання синхронізації.

```
import Queue from "bull";
import { redis } from "../redisConnection.js";
import { Job } from "../db/dbConnector.js";
import { sendEmail } from "../utils/mailer.js";
import { fullSynchronization } from "../jobs/fullSynchronization.js";

export const fullSyncQueue = new Queue("fullSyncQueue", {
  redis: redis.options
});

fullSyncQueue
  .on('active', async (job) => {
    const { userId } = job.data.user;
    await Job.findOneAndUpdate({ userId }, { $set: { userId, status: "INPROGRESS", createdAt: new Date() } }, { upsert: true });
  })
  .on('completed', async (job) => {
    const { userId, email } = job.data.user;
    try {
      const jobbed = await fullSyncQueue.getJob(job.id);
      if (jobbed) {
        await jobbed.remove();
      }
      await Job.findOneAndUpdate({ userId }, { $set: { userId, status: "COMPLETED", createdAt: new Date() } }, { upsert: true });
      await sendEmail("../emails/synchronize-result", email, {});
    } catch (error) {
      await Job.findOneAndUpdate({ userId }, { $set: { userId, status: "COMPLETED_WITH_ERROR", createdAt: new Date() } }, { upsert: true });
    }
  })
  .on('failed', async (job, err) => {
    const { userId } = job.data.user;
    await Job.findOneAndUpdate({ userId }, { $set: { userId, status: "FAILED", createdAt: new Date(), error: err } }, { upsert: true });
  })
  .process(async (job) => {
    try {
      await fullSynchronization(job.data.user, job.data.dateFrom, job.data.dateTo, job.data.isScopus, job.data.isCrossref, job.data.isMRAT);
      return Promise.resolve({ success: true });
    } catch (err) {
      return Promise.reject(err);
    }
  });
```

Рисунок 4.19 – Лістинг коду синхронізації даних з наукометричних баз даних

Фрагмент коду вище демонструє роботу з чергами для синхронізації даних. Спочатку завдання потрапляє у чергу (fullSyncQueue), де його статус змінюється на «INPROGRESS» під час активації. Після успішного завершення синхронізації статус завдання оновлюється на «COMPLETED», а користувачу відправляється

повідомлення на електронну пошту про результати. Якщо під час виконання синхронізації виникає помилка, статус завдання змінюється на «COMPLETED_WITH_ERROR», і відповідна інформація зберігається в базі даних.

На рисунку 4.20 наведено фрагмент коду, який демонструє безпосередню реалізацію функції, яка виконує синхронізацію даних.

```
import { Dissertation, Project, Publication } from '../db/dbConnector.js'
import { scopusImport } from '../helpers/scopusImport.js'
import { crossrefSearch } from '../helpers/crossrefSearch.js'
import { nratParser } from '../helpers/nratParser.js'

export const fullSynchronization = async (user, dateFrom, dateTo, isScopus, isCrossref, isNRAT) => {
  try {
    const { userId } = user;
    if (isScopus) {
      const scopusPublications = await scopusImport(userId, dateFrom, dateTo);
      scopusPublications.forEach(async (publication) => {
        await Publication.findOneAndUpdate({ externalID: publication.externalID, department: publication.department }, { $set: publication }, { upsert: true });
      });
    }
    if (isCrossref) {
      const crossrefSearchPublications = await crossrefSearch(userId, dateFrom, dateTo);
      crossrefSearchPublications.forEach(async (publication) => {
        await Publication.findOneAndUpdate({ name: publication.name, department: publication.department }, { $set: publication }, { upsert: true });
      });
    }
    if (isNRAT) {
      const nratData = await nratParser(user, dateFrom, dateTo)
      nratData.forEach(async (item) => {
        if (item.author) {
          await Dissertation.findOneAndUpdate({ number: item.number, department: item.department }, { $set: item }, { upsert: true });
        } else {
          await Project.findOneAndUpdate({ number: item.number, department: item.department }, { $set: item }, { upsert: true });
        }
      })
    }
  } catch(e) {
    throw new Error(e)
  }
}
```

Рисунок. 4.20 – Лістинг коду завдання, яке запускає процес синхронізації з кожного джерела

Код, представлений на зображенні, описує асинхронну функцію `fullSynchronization`, яка виконує процес синхронізації даних про публікації з кількох зовнішніх та внутрішніх джерел. Ця функція приймає набір параметрів, що містять ідентифікатор користувача (`user`), діапазон дат (`dateFrom`, `dateTo`) і прапорці, що вказують на необхідність обробки джерел Scopus (`isScopus`), Crossref (`isCrossref`) та внутрішнього ресурсу (`isNRAT`). Початково ініціалізується змінна з ідентифікатором користувача (`userId`) для подальшого використання у запитах до зовнішніх служб.

Наступним етапом функція перевіряє прапорець `isScopus` і, у разі його активації, здійснює імпорт публікацій зі служби Scopus за допомогою функції `scopusImport`. Отриманий список публікацій обробляється за допомогою методу

forEach, який здійснює пошук та оновлення записів публікацій у локальній базі даних за допомогою методу findOneAndUpdate. Оновлення записів базується на зовнішньому ідентифікаторі (externalID) та приналежності до відповідного департаменту установи.

Подібний механізм застосовується при роботі з джерелом Crossref. Якщо прапорець isCrossref встановлено у значення true, виконується пошук за допомогою функції crossrefSearch. Результати цієї функції так само проходять цикл обробки з оновленням записів публікацій в базі даних відповідно до отриманих даних, що включають назву публікації та її департамент.

Останнім блоком функції є обробка даних внутрішнього джерела (NRAT), активована прапорцем isNRAT. Тут здійснюється виклик парсера nratParser, що повертає певний набір метаданих. Отримані метадані перебираються методом forEach, при цьому здійснюється умовна перевірка типу елемента: якщо елемент є авторською роботою (item.author), він зберігається або оновлюється у колекції дисертацій (Dissertation), в іншому разі — в колекції проектів (Project). В обох випадках пошук і оновлення здійснюється за ідентифікатором елемента та департаментом.

Усі ці операції обгорнуті у блок try-catch, що дозволяє фіксувати і обробляти помилки, які можуть виникнути під час виконання асинхронних запитів до зовнішніх API або бази даних, генеруючи нові помилки у разі виняткових ситуацій. Такий підхід забезпечує цілісність процесу синхронізації та коректність оновлення інформації у внутрішніх базах даних.

Ще одним важливим компонентом розробленої системи є модуль, що відповідає за формування профілів науковців на основі автоматизованого вилучення та синхронізації інформації про викладачів із зовнішніх та внутрішніх ресурсів. Цей модуль передбачає комплексний аналіз HTML-даних вебсторінок факультетів і кафедр, що дозволяє зібрати актуальні метадані викладачів,

зокрема персональні дані, ідентифікатори наукових профілів (ORCID, Scopus, Scholar тощо), та визначити їхні наукові інтереси(Рис. 4.21).

```

const parseTeachersMetadata = ($, commentValue, sectionClass) => {
  const elements = findElementAfterComment($, commentValue, sectionClass);
  const medatData = [];

  elements.each(, elem) => {
    const [firstName, lastName, middleName] = $(elem).find('h4').text().trim().replace('\u00a0', " ").split(" ");
    const orcidUrl = $(elem).find('a[href*="orcid.org"]').attr('href');
    const wosUrl = $(elem).find('a[href*="wos"]').attr('href');
    const scopusUrl = $(elem).find('a[href*="scopus"]').attr('href');
    const scholarUrl = $(elem).find('a[href*="google"]').attr('href');

    const interestsElement = findElementAfterComment($(elem), 'div_spoiler Коло наукових інтересів', 'div.text_spoiler');
    const scientificInterests = interestsElement.text().split(",").map(key => key.trim());

    medatData.push({
      orcid: orcidUrl ? extractOrcidId(orcidUrl) : null,
      wosId: wosUrl ? extractWosId(wosUrl) : null,
      scopusId: scopusUrl ? parseAuthorID(scopusUrl, 'authorId') : null,
      scholarId: scholarUrl ? parseAuthorID(scholarUrl, 'user') : null,
      firstName,
      lastName,
      middleName,
      scientificInterests,
    });
  });
  return medatData;
}

export const getTeachersMetadata = async () => {
  const facultyData = await parseFacultyData();
  let headOfDepartments = [];
  let teachers = [];

  for (const departmentUrl of facultyData) {
    const $ = await fetchHTML(`${URL_LINK}${departmentUrl}`);
    headOfDepartments = [...headOfDepartments, ...parseTeachersMetadata($, 'faculty_bottom', '#article:not(:nth-child(2)):not(:nth-child(3))');
    teachers = [...teachers, ...parseTeachersMetadata($, 'Професорсько-викладацький колектив', '.mb0')];
  }
  return teachers
};

```

Рисунок 4.21 – Лістинг коду для веб-скрпінгу профілю науковця

Як видно з лістингу коду вище він реалізує дві основні функції: `parseTeachersMetadata` та `getTeachersMetadata`, призначені для збору та обробки метаданих викладачів із веб-сторінок підрозділів навчального закладу. Функція `parseTeachersMetadata` отримує три параметри: об'єкт DOM, значення коментаря (`commentValue`) та CSS-клас секції (`sectionClass`). На початковому етапі за допомогою допоміжної функції знаходяться елементи HTML, які відповідають заданим параметрам.

Для кожного знайденого елемента виконується парсинг персональних даних викладача, зокрема імені, прізвища, по-батькові, ідентифікатора ORCID, посилання на профілі в Scopus, Google Scholar, а також вилучаються список наукових інтересів. Вилучені дані формуються в об'єкт, який потім додається до загального масиву метаданих викладачів.

Друга функція, `getTeachersMetadata`, відповідає за завантаження та подальшу обробку інформації щодо викладачів, організованої за департаментами. Спочатку функція отримує дані про підрозділи (`facultyData`) за допомогою асинхронного запиту. Далі відбувається ітерація по URL кожного департаменту, з яких окремо завантажуються HTML-вміст.

Кожен отриманий HTML-документ обробляється за допомогою попередньо описаної функції `parseTeachersMetadata`, після чого результат об'єднується в загальний масив викладачів. Додатково створюється окремий масив з інформацією про керівників департаментів, який формується аналогічним чином.

4.4. Тестування програмного середовища

Тестування програмного середовища є ключовим етапом життєвого циклу розробки програмного забезпечення, що спрямований на перевірку якості, функціональності та безпеки програм перед їх впровадженням. Цей процес включає різноманітні методики та завдання, необхідні для виявлення дефектів, підтвердження відповідності функціональним вимогам і покращення досвіду користувача. В умовах зростаючої складності програмних систем та важливості дотримання нормативних стандартів ефективно тестування стало вагомим складовою успішної розробки програм, яка впливає як на задоволеність користувачів, так і на продуктивність організацій.

Головними завданнями тестування програмного середовища є виявлення та усунення дефектів, перевірка відповідності програмного забезпечення заданим функціональним специфікаціям, покращення продуктивності й безпеки, а також забезпечення зручності використання. Тестування виступає механізмом забезпечення якості, що дозволяє командам розробників системно оцінювати програмні продукти на відповідність встановленим вимогам. Впроваджуючи суворі та систематичні стратегії тестування, організації можуть зменшити

ризика збоїв програм, скоротити витрати на подальшу підтримку та підвищити загальну надійність продуктів [22, 111].

Оскільки система для збору та аналізу наукової та науково-педагогічної діяльності педагогічного колективу є складним програмним рішенням, яке передбачає значні обсяги даних та численні інтеграції, особливу увагу було приділено наступним видам тестування:

1) *Модульне тестування*: це метод перевірки окремих компонентів програмного забезпечення, таких як функції або методи, з метою підтвердження їхньої коректності та надійності. Цей підхід дозволяє виявляти та виправляти помилки на ранніх етапах розробки, що сприяє підвищенню якості та надійності кінцевого продукту. Автоматизація модульного тестування, зокрема через використання систем автоматизованого генерування тестів, може значно підвищити ефективність процесу тестування та забезпечити більш повне покриття коду [75, 94, 101, 144].

2) *Тестування API*: цей вид тестування дозволяє перевірити взаємодію між окремими програмними компонентами через визначені інтерфейси. Він охоплює оцінку функціональності, стабільності, безпеки та продуктивності API. Використання таких інструментів, як GraphQL Playground, дозволяє ефективно взаємодіяти з API на основі GraphQL, створювати тестові запити, аналізувати відповіді сервера та автоматизувати процес тестування. Крім того, сучасні підходи, які враховують залежності між окремими операціями API, сприяють більш повному покриттю тестування та підвищують загальну якість системи [25, 34, 37, 128].

3) *Тестування продуктивності*: тип тестування програмного забезпечення, метою якого є оцінка швидкодії, масштабованості та стабільності системи в умовах заданого навантаження. Основними завданнями такого тестування є визначення межі працездатності додатку, виявлення вузьких місць та прогнозування поведінки системи при зростанні кількості користувачів або запитів [58]. Зазвичай воно включає такі підтипи як навантажувальне тестування, стрес-тестування, тестування на витривалість, а також тестування

масштабованості. Для його реалізації використовуються спеціалізовані програмні інструменти, що дають можливість моделювати навантаження, збирати метрики та здійснювати глибокий аналіз продуктивності програмного забезпечення [86]. Крім того, автоматизація тестування продуктивності дозволяє значно підвищити якість та надійність програмних продуктів завдяки оперативному виявленню та усуненню проблем, що виникають у процесі роботи системи під навантаженням [126, 137].

Оскільки під час розробки даного програмного забезпечення було реалізовано набір GraphQL резолверів, які взаємодіють з базою даних MongoDB через Mongoose. Щоб переконатися, що кожен резолвер працює коректно та стабільно, було написано модульні тести за допомогою бібліотеки Jest. Ці тести дозволили ізолювати бізнес-логіку резолверів шляхом мокання зовнішніх залежностей, таких як виклики до бази даних (наприклад, `User.findOne`, `User.save`) та утиліт (наприклад, `hashPassword`, `signToken`, `verifyPassword`). Такий підхід дозволив швидко виявити регресії та полегшив локалізацію проблем у логіці резолверів.

На рисунку 4.22 наведено фрагмент коду тесту для мутації `registration`. Цей тест перевіряє, чи коректно працює логіка реєстрації нового користувача в системі з електронною скринькою, яка вже була використана раніше:

- перевіряє відсутність користувача за допомогою `User.findOne`;
- хешує наданий пароль за допомогою `hashPassword`;
- зберігає нового користувача та повертає токен автентифікації, згенерований за допомогою `signToken`.

```

> JS resolvers.test.js > ...
it('should register a new user when email is not in use', async () => {
  const input = {
    email: 'newuser@example.com',
    password: 'secret',
    firstName: 'Alice',
    lastName: 'Smith',
    department: 'Computer Science'
  };

  User.findOne = jest.fn().mockResolvedValue(null);

  const saveMock = jest.fn((callback) => callback(null));
  function DummyUser(data) {
    Object.assign(this, data);
    this.save = saveMock;
  }

  const originalUser = User;
  User.prototype.constructor = DummyUser;

  const result = await resolvers.Mutation.registration(null, { input });

  expect(User.findOne).toHaveBeenCalledWith({ email: input.email });
  expect(saveMock).toHaveBeenCalled();
  expect(result).toHaveProperty('user');
  expect(result).toHaveProperty('token', 'signed-token');

  User.prototype.constructor = originalUser;
});

```

Рисунок 4.22 – Фрагмент тесту для перевірки реєстрації користувача в системі

Основною метою тесту є відтворення сценарію, коли новий користувач з унікальним email ще не існує у базі даних. Для цього метод пошуку користувача (`User.findOne`) замокано таким чином, щоб завжди повертати `null`, що імітує відсутність запису з заданою електронною адресою. Далі, за допомогою `dummy`-реалізації конструктора класу користувача, створюється об'єкт, у якому метод збереження (`save`) також замокано для симуляції успішного виконання операції збереження даних без реального взаємодії з базою.

Після виклику резолвера `registration`, перевіряється коректність виконання алгоритму. Резолвер повинен викликати метод пошуку користувача з переданою email адресою, ініціювати створення нового об'єкта користувача, викликати метод збереження цього об'єкта, а також повернути об'єкт, що містить як дані користувача, так і згенерований токен автентифікації. Згенерований токен у тестовому середовищі має бути константним значенням, яке визначено в мокованій функції генерації токенів.

Для всіх інших резолверів також було розроблено ряд тестів, що охоплюють як позитивні, так і негативні сценарії. У позитивних тестових

випадках перевіряється коректне виконання бізнес-логіки при нормальних умовах (наприклад, повернення коректних даних за запитом або успішне виконання операції), тоді як у негативних тестах симулюються помилкові умови або некоректні вхідні дані, що дозволяє перевірити коректність обробки помилок. Це дає можливість повністю охопити функціональність кожного резолвера та дозволяє підвищити якість коду [103].

На рисунку 4.23 наведено загальний звіт з виконання тестів, згенерованого Jest, який містить результати для всіх тестових випадків.

```

PASS ./resolvers.test.js
Mutation Resolvers
  ✓ registration - Register a new user (10 ms)
  ✓ entrantRegistration - Register a new entrant (10 ms)
  ✓ resetPassword - Request password reset (7 ms)
  ✓ updatePassword - Update users password (7 ms)
  ✓ addPublication - Add a new publication (8 ms)
  ✓ addDissertation - Add a new dissertation (8 ms)
  ✓ addEvent - Add a new event (8 ms)
  ✓ addGrant - Add a new grant (8 ms)
  ✓ addProject - Add a new project (8 ms)
  ✓ addScientist - Add a new scientist (6 ms)
  ✓ addSupervisor - Add a new supervisor (6 ms)
  ✓ updatePublication - Update a publication (7 ms)
  ✓ updateDissertation - Update a dissertation (7 ms)
  ✓ updateEvent - Update an event (7 ms)
  ✓ updateGrant - Update a grant (7 ms)
  ✓ updateProject - Update a project (7 ms)
  ✓ updateScientist - Update scientist data (7 ms)
  ✓ updateSupervisor - Update supervisor data (7 ms)
  ✓ updateInterview - Update an interview with user notification (9 ms)
  ✓ deletePublication - Delete a publication (5 ms)
  ✓ deleteGrant - Delete a grant (5 ms)
  ✓ deleteProject - Delete a project (5 ms)
  ✓ deleteScientist - Delete a scientist (5 ms)
  ✓ deleteSupervisor - Delete a supervisor (5 ms)
  ✓ deleteDissertation - Delete a dissertation (5 ms)
  ✓ login - Authenticate user (10 ms)
  ✓ deleteUser - Delete a user (6 ms)
  ✓ updateUser - Update user data (7 ms)
  ✓ fullSynchronization - Run full synchronization (12 ms)
  ✓ updateUserAvatar - Update user avatar (6 ms)
  ✓ exportReport - Export report (10 ms)
  ✓ importEmployees - Import employees from CSV (15 ms)
  ✓ importScientists - Import scientists from CSV (15 ms)
  ✓ crossrefSearch - Search publications via Crossref (10 ms)
  ✓ requestInterview - Request an interview (8 ms)
  ✓ collectScientificProfiles - Collect scientific profiles for teachers (9 ms)
Query Resolvers
  ✓ getAllUsers - Should return all users (8 ms)
  ✓ getEvents - Fetch events with filters applied (12 ms)
  ✓ getProjects - Fetch projects with pagination and filtering (10 ms)
  ✓ getGrants - Fetch grants with proper sorting (11 ms)
  ✓ getEvent - Fetch a specific event by ID (5 ms)
  ✓ getGrant - Fetch a specific grant by ID (5 ms)
  ✓ getPublications - Fetch publications with pagination (15 ms)
  ✓ getDissertations - Fetch dissertations with pagination (15 ms)
  ✓ getPublicationActivityCount - Get publication count statistics by category and type (10 ms)
  ✓ getPublicationActivity - Aggregate publication activity for authors (20 ms)
  ✓ checkImportStatus - Check import status (4 ms)
  ✓ getScientists - Fetch list of scientists with pagination (10 ms)
  ✓ getSupervisors - Fetch supervisors based on given filters (12 ms)
  ✓ getEntrants - Fetch entrants with pagination (6 ms)
  ✓ searchSupervisor - Search for supervisor by name (8 ms)
  ✓ searchAuthor - Search for author by name (10 ms)
  ✓ getInterviews - Fetch interviews based on user role (14 ms)
  ✓ hasActivity - Check if publication activity exists (3 ms)

Test Suites: 1 passed, 1 total
Tests:       48 passed, 48 total
Snapshots:   0 total
Time:        2.345 s
Ran all test suites.

```

Рисунок 4.23 – Звіт з результатами проведення модульних тестів

Результати які наведені вище, свідчать про стабільну та коректну роботу всіх резолверів розробленої системи. Загалом було виконано 48 тестових сценаріїв, які охоплюють як запити (Query Resolvers), так і мутації (Mutation Resolvers), і всі вони пройшли успішно без жодної помилки. Час виконання тестів склав 2.345 секунд, що демонструє високу ефективність реалізації функціоналу. Цей звіт підтверджує, що кожен компонент системи відповідає вимогам щодо коректності виконання бізнес-логіки, а також гарантує, що як позитивні, так і негативні сценарії були опрацьовані належним чином, що є важливим показником якості та стабільності програмного продукту.

Враховуючи, що створене програмне забезпечення побудоване за сервісно-орієнтованою архітектурою, у межах дослідження було виконано тестування програмного інтерфейсу (API).

Для повноцінного тестування API системи було проведено такі групи тестів у зазначеній послідовності:

- базові позитивні тестові сценарії;
- розширені позитивні тести із залученням додаткових параметрів;
- негативні тести з коректними вхідними даними;
- негативні тести з некоректними вхідними даними;
- деструктивне тестування;
- тестування безпеки та авторизації.

У таблиці 4.1 представлено перелік тестових сценаріїв, які були використані для перевірки роботи програмного інтерфейсу системи для збору та аналізу наукової та науково-педагогічної діяльності академічного колективу.

Таблиця 4.1

Тестові сценарії для перевірки API

Тестові категорії	Тестові сценарії
Базові позитивні тестові сценарії	<ul style="list-style-type: none"> • Успішне отримання списку публікацій автора • Успішне додавання публікації

Продовження таблиці 4.1

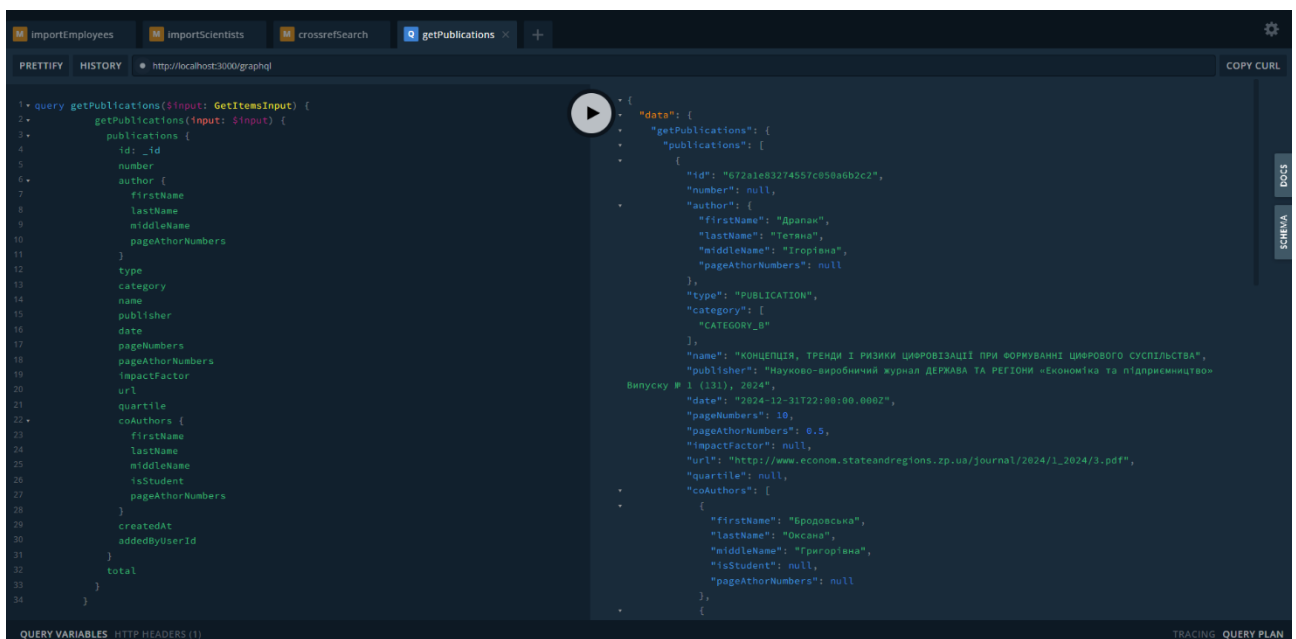
	<ul style="list-style-type: none"> • Успішне оновлення інформації про публікацію • Успішне отримання списку проектів • Успішне додавання нового проекту • Успішне отримання списку грантів • Успішне додавання нового гранту • Успішне отримання списку дисертацій • Успішне додавання нової дисертації • Успішне отримання списку подій • Успішне додавання нової події
Розширені позитивні тести із додатковими параметрами	<ul style="list-style-type: none"> • Отримання списку публікацій з фільтрацією за автором та датою • Отримання списку проектів із сортуванням за періодом
Негативні тести з коректними вхідними даними	<ul style="list-style-type: none"> • Спроба додавання вже існуючого автора до системи • Видалення неіснуючої публікації • Спроба отримання неіснуючого запису за правильним ID
Негативні тести з некоректними вхідними даними	<ul style="list-style-type: none"> • Введення некоректного формату дати • Додавання запису з неповними обов'язковими полями • Введення некоректного ID запису
Деструктивне тестування	<ul style="list-style-type: none"> • Запит з надмірно довгими текстовими значеннями • Спроба впровадження SQL-ін'єкцій та JavaScript-скриптів у поля • Масова генерація запитів для перевірки стабільності

Продовження таблиці 4.1

Тестування безпеки та авторизації	<ul style="list-style-type: none"> • Спроба доступу без авторизації • Спроба доступу з недійсним або простроченим токеном • Перевірка прав доступу на додавання, оновлення та видалення даних за ролями
-----------------------------------	--

Тестування API виконувалося з використанням інструменту GraphQL Playground. Він являє собою спеціалізоване інтерактивне середовище, яке призначене для перевірки працездатності та дослідження API на основі технології GraphQL. Серед його ключових переваг потрібно відзначити інтуїтивно зрозумілий графічний інтерфейс, наявність функції автодоповнення запитів, підсвічування синтаксису, вбудовану документацію зі схемою API та можливість збереження і подальшого повторного використання створених запитів. Це забезпечує зручність і швидкість проведення тестування, ефективно виявлення помилок, перевірку роботи запитів та мутацій, а також оперативний аналіз відповідей сервера.

На рисунку 4.24 продемонстровано результат тестування API з використанням GraphQL Playground.



The screenshot shows the GraphQL Playground interface. On the left, a query is entered: `query getPublications($input: GetItemsInput) { getPublications(input: $input) { publications { id: _id number author { firstName lastName middleName pageAuthorNumbers } type category name publisher date pageNumbers pageAuthorNumbers impactFactor url quartile coAuthors { firstName lastName middleName isStudent pageAuthorNumbers } createdAt addedByUserId } total } }`. On the right, the JSON response is displayed: `{ "data": { "getPublications": { "publications": [{ "id": "672a1e83274557c859a6b2c2", "number": null, "author": { "firstName": "Дранак", "lastName": "Тетана", "middleName": "Тропіана", "pageAuthorNumbers": null }, "type": "PUBLICATION", "category": ["CATEGORY_B"], "name": "КОНЦЕПЦІЯ, ТRENДИ І РИЗИКИ ЦИФРОВІЗАЦІЇ ПРИ ФОРМУВАННІ ЦИФРОВОГО СУСПІЛЬСТВА", "publisher": "Науково-виробничий журнал ДЕРЖАВА ТА РЕГІОНИ «ЕКОНОМІКА ТА ПІДПРИЄМНИЦТВО» Випуск # 1 (133), 2024", "date": "2024-12-31T22:00:00.000Z", "pageNumbers": 10, "pageAuthorNumbers": 6.5, "impactFactor": null, "url": "http://www.econom.stateandregions.zp.ua/journal/2024/1_2024/3.pdf", "quartile": null, "coAuthors": [{ "firstName": "Бродовська", "lastName": "Оксана", "middleName": "Григорівна", "isStudent": null, "pageAuthorNumbers": null }] }] } }`

Рисунок 4.24 – Тестування API з використанням інструменту GraphQL Playground

На наведеному скриншоті зображено процес тестування API системи за допомогою інструменту GraphQL Playground. На прикладі конкретного запиту, який реалізує операцію отримання публікацій (метод `getPublications`), продемонстровано можливості цього інструмента. У лівій частині інтерфейсу представлений сформований GraphQL-запит, що містить визначення полів, які необхідно отримати у відповідь на запит до сервера, такі як ідентифікатор публікації, назва, тип, автори та інші метадані. У правій частині наведено відповідь сервера, яка має структурований JSON-формат та містить дані, відповідні зазначеному запиту.

Наступним видом тестування, яке було проведено, є навантажувальне тестування. Цей вид тестування має критичне значення для оцінки працездатності систем, особливо коли йдеться про великі обсяги даних, як у запропонованому програмному комплексі. Навантажувальне тестування дозволяє визначити, як система реагує на одночасну роботу великої кількості користувачів або процесів, імітуючи реальні умови експлуатації. Це дає змогу виявити потенційні обмеження, вузькі місця в обробці запитів, а також оцінити здатність системи масштабуватися при зростанні навантаження. Завдяки такому підходу можна своєчасно провести оптимізацію продуктивності, підвищити стабільність роботи програмного забезпечення та зменшити ймовірність виникнення непередбачених ситуацій під час інтенсивного навантаження. Для оцінки результатів навантажувального тестування були обрані такі ключові метрики:

- час відповіді:
 - середній;
 - медіана;
 - 90 перцентиль;
 - 95 перцентиль;
 - 99 перцентиль;
- частота помилок;
- пропускна здатність

Після чого було розроблено наступні тестові сценарії:

- логін користувача;
- отримання списку публікацій;
- імпорт даних з наукометричних баз;
- ручне додавання даних в систему;
- формування звіту по науковій діяльності.

В таблиці 4.2 продемонстровано результати тестування продуктивності системи.

Таблиця 4.2

Результати тестування продуктивності системи

Сценарій	К-ть	Середнє(мс)	Медіана (мс)	90 процентиль (мс)	95 процентиль (мс)	99 процентиль (мс)	Частота помилок %	Пропускна здатність 1/сек
Логін користувача	100	4440	3988	7985	8099	8176	0%	5.6
Отримання списку публікацій	100	3200	2850	5843	6541	7094	0%	4.4
Імпорт даних з наукометричних баз	100	1531	1040	5113	5291	5295	0%	2.1
Ручне додавання даних в систему;	100	1788	1652	3530	4001	5496	0%	3.3
Формування звіту по науковій діяльності	100	26743	26849	34593	34707	37626	0%	1.8

За результатами проведеного тестування продуктивності системи було отримано наступні дані. При виконанні 100 запитів для сценарію «Логін користувача» середній час відповіді сервера становив 4440 мс, медіана — 3988

мс. Значення перцентилів (90, 95 і 99), які характеризують стабільність часу відповіді, дорівнюють відповідно 7985 мс, 8099 мс та 8176 мс. Частота помилок для цього сценарію становила 0%, а пропускна здатність системи — 5.6 запитів за секунду. Аналогічно, сценарій «Отримання списку публікацій» показав середній час відповіді 3200 мс з медіаною 2850 мс та 99-перцентилем 7094 мс при відсутності помилок та пропускній здатності 4.4 запитів за секунду. Найшвидше оброблялися запити за сценарієм «Імпорт даних з наукометричних баз», де середнє значення становило 1531 мс, медіана — 1040 мс, 99-перцентиль — 5295 мс, а пропускна здатність досягала 2.1 запитів за секунду. Для сценарію «Ручне додавання даних в систему» середній час відповіді склав 1788 мс із медіаною 1652 мс, а 99-перцентиль дорівнював 4906 мс при пропускній здатності 3.3 запитів за секунду. Найбільш ресурсомістким виявився сценарій «Формування звіту по науковій діяльності», для якого середній час становив 26743 мс, медіана — 26849 мс, 99-перцентиль — 37626 мс, при пропускній здатності лише 1.8 запитів за секунду. Усі сценарії мали нульовий рівень помилок, що свідчить про стабільну роботу системи у процесі тестування.

Тестування продуктивності здійснювалося шляхом використання інструменту Apache JMeter (Рис.4.25), в якому були створені відповідні тестові скрипти.

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Receive...	Send...
User login	100	4440	3988	7985	8099	8176	200	8202	0.00%	5.6/sec	3.24	
Get list of publications	100	3200	2850	5843	6541	7094	49	7383	0.00%	4.4/sec	12.11	
Add user publication	100	1788	1652	3530	4001	5496	23	7815	0.00%	3.3/sec	0.94	
Generate summary repor...	100	26743	26849	34593	34707	37626	8290	40379	0.00%	1.8/sec	58.33	
Synchronize publications	100	1531	1040	5113	5291	5295	24	6069	0.00%	2.1/sec	0.61	
TOTAL	500	7541	2904	26849	31946	34707	23	40379	0.00%	9.0/sec	64.96	

Рисунок 4.25 – Результати виконання тестів в програмі Apache JMeter

4.5. Організація графічного інтерфейсу програмного середовища системи для збору та аналізу науково та науково-педагогічної діяльності академічного колективу

Інтерфейс системи розроблений базуючись на принципах побудови інтуїтивності та простоти використання, що забезпечує зручний доступ користувача до основної функціональності без необхідності в додатковому навчання персоналу.

На рисунку 4.26 можна побачити початковий екран сторінки на якій знаходяться форми авторизації та аутентифікації.

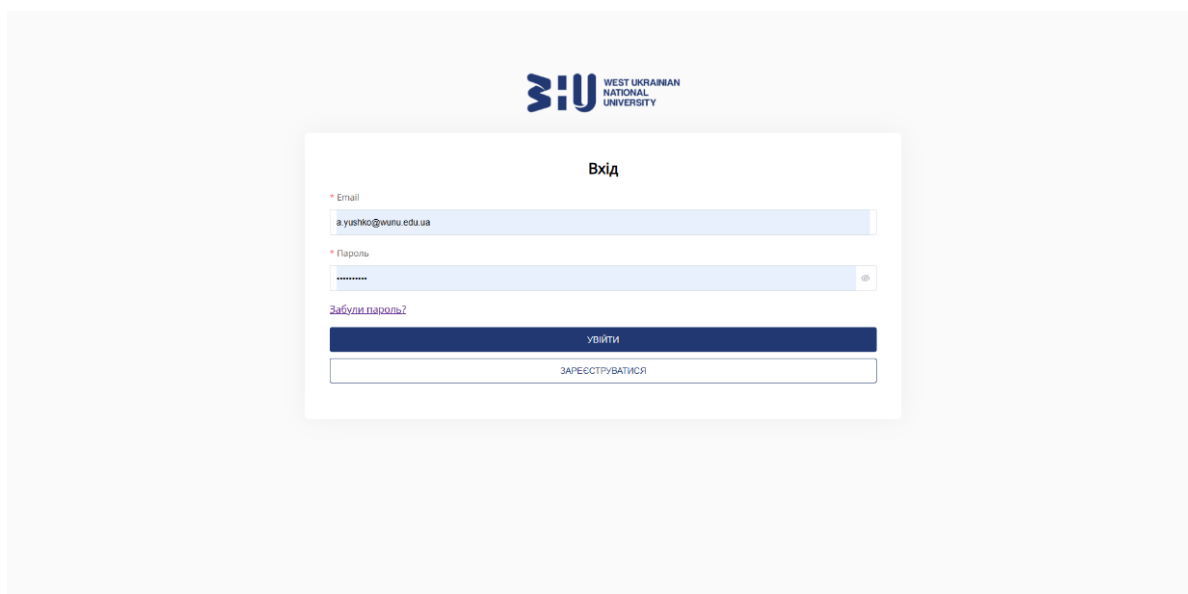


Рисунок 4.26 – Форма авторизації

Як видно з знімку екрана вище форма авторизації є досить простою, оскільки вимагає від користувача введення лише адреси електронної скриньки та паролю, який було придумано під час реєстрації в системі(рис. 4.27).

Рисунок 4.27 – Форма реєстрації в системі

Форма реєстрації передбачає заповнення користувачем бази інформації про себе, такої як, прізвище, ім'я, по батькові, посада, факультет, кафедра та інше. Також при реєстрації працівник повинне вказати свої ідентифікатори в інших наукометричних базах для забезпечення процесу автоматизованого збору інформації. Ще одними із головних полів цієї форми є прізвище та ім'я науковця латиницею, оскільки ці дані будуть необхідні для пошуку публікацій в базі даних Crossref [121].

Після успішної авторизації в системі користувач потрапляє на сторінку «Огляд»(Рис. 4.28) де він може побачити кількісні показники публікаційної активності.

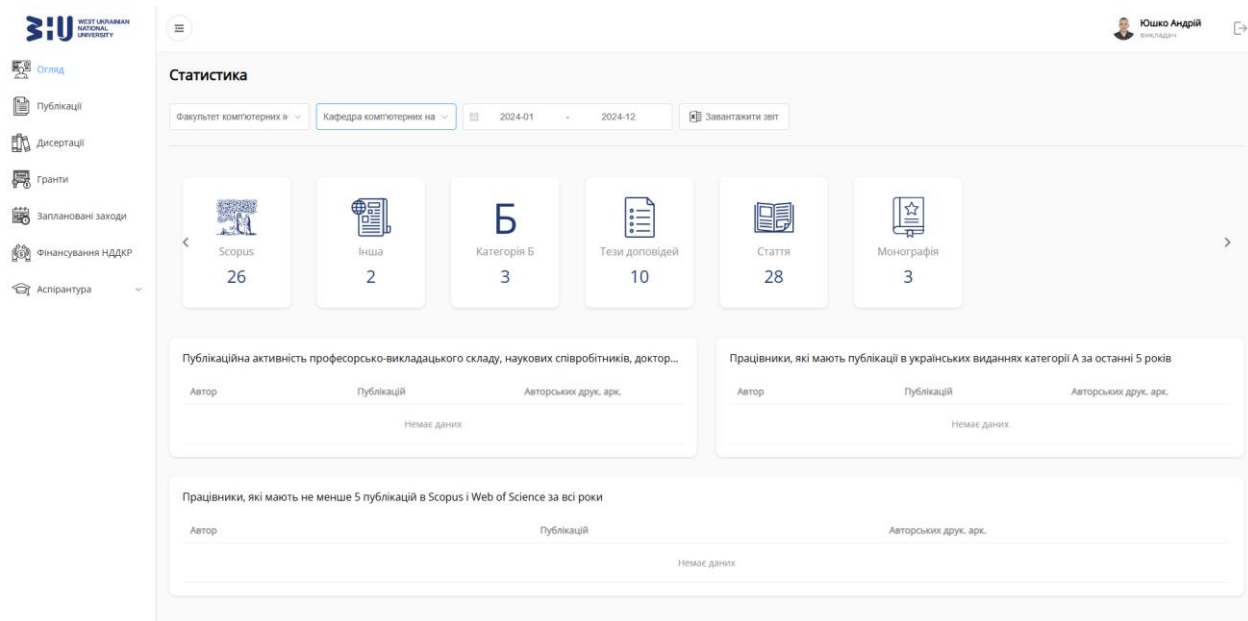


Рисунок 4.28 – Перегляд кількості публікацій по кожній із категорій автора

Як видно з знімку екрана вище користувач має можливість відфільтрувати усі показники по факультету, кафедрі та періоду публікації.

Також присутня можливість сформувати звіт по певному підрозділу натиснувши кнопку «Завантажити звіт». Ця можливість є лише в працівників з відповідними правами доступу (наприклад завідувача кафедри, декана факультету, проректора з наукової роботи).

Якщо користувач потрапив в систему та немає ще доданих даних він бачитиме вікно привітання та кнопку, яка дозволить виконати синхронізацію усієї публікаційної активності з інших наукометричних баз(Рис. 4.29).

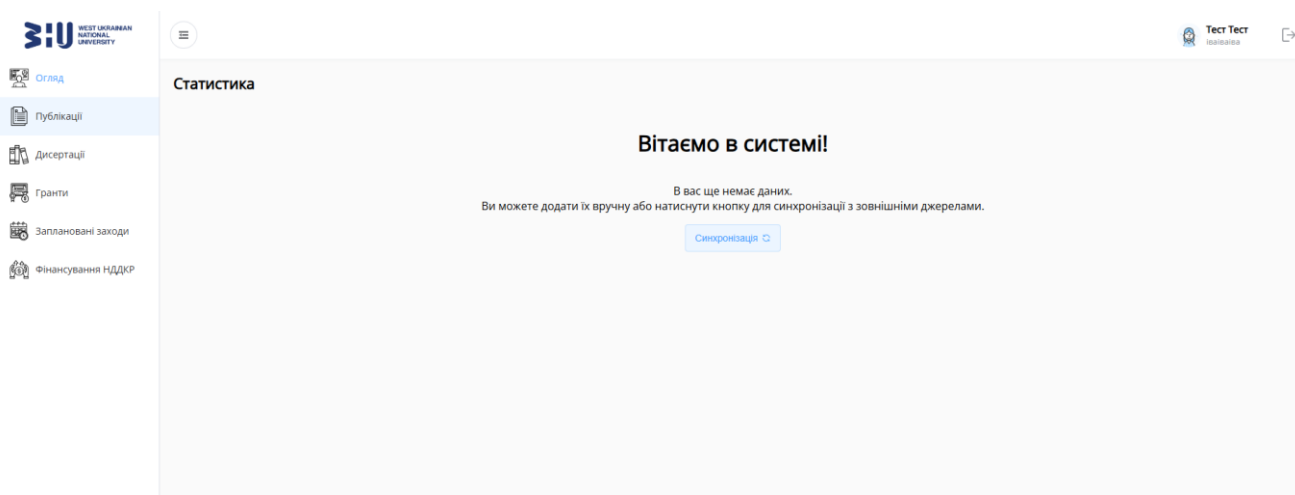


Рисунок 4.29 – Сторінка привітання користувача в системі

Після натискання кнопки «Синхронізація», відкриється вікно з описом баз даних в яких буде здійснюватися пошук інформації(Рис. 4.30.).

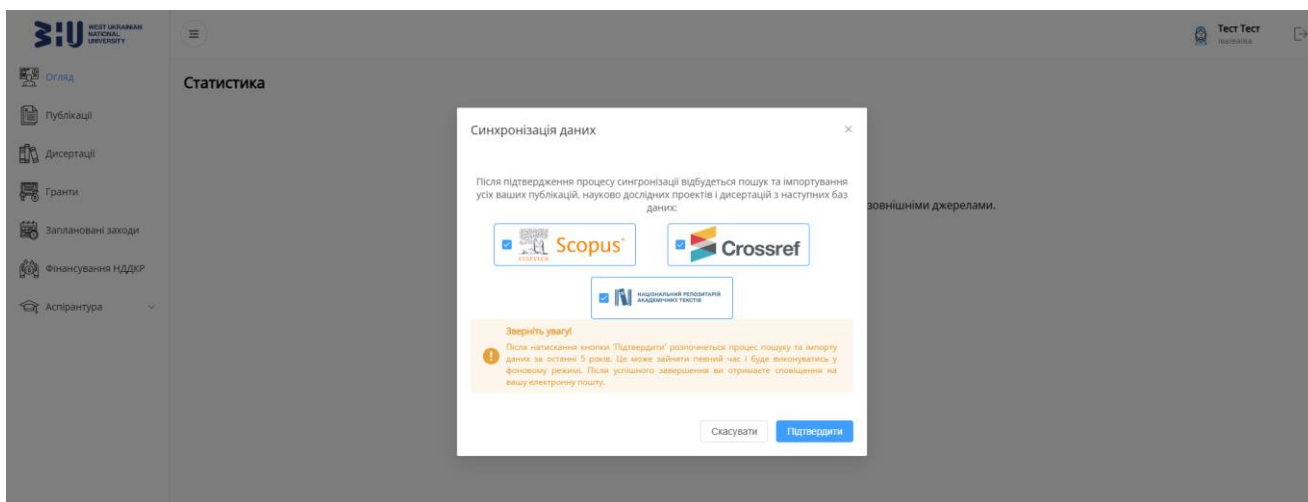


Рисунок 4.30 – Вікно підтвердження синхронізації

Також, як видно з зображення вище користувачу надається коротка довідка про те за який період буде здійснюватися вибірка його даних(по замовчуванню 5 років) та як саме відбуватиметься цей процес.

Після успішної синхронізації користувачу системи приходиться повідомлення на його електронну скриньку, яке інформує, що його публікаційна активність була успішно зібрана.

Перейшовши в розділ «Публікації», користувач зможе переглянути усі публікації, які вдалося знайти системі(Рис. 4.31).

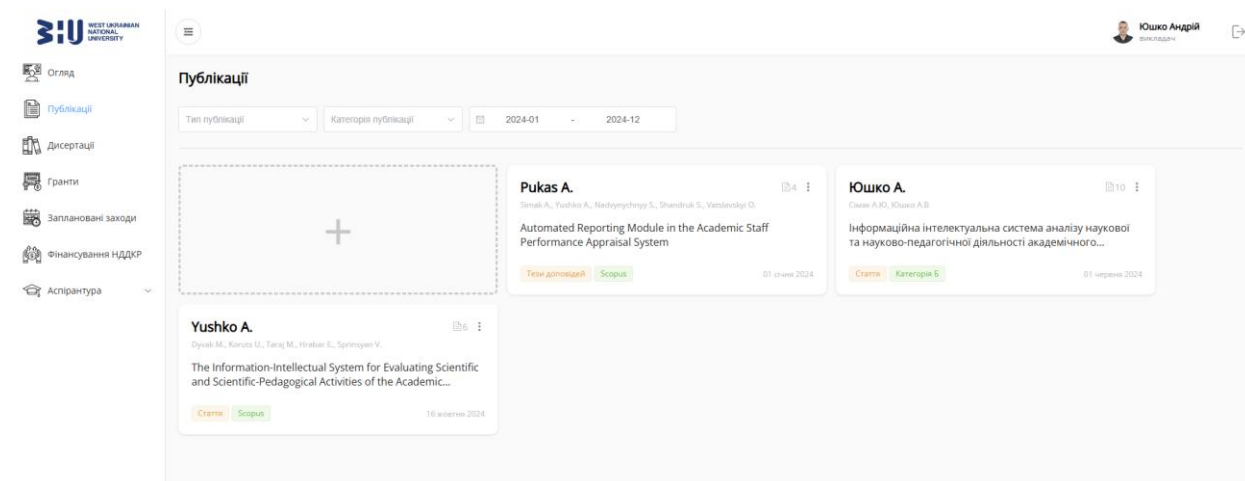


Рисунок 4.31 – Сторінка відображення публікацій

Також наша система передбачає розділ в якому можна переглядати список наданих грантів(Рис. 4.34.).

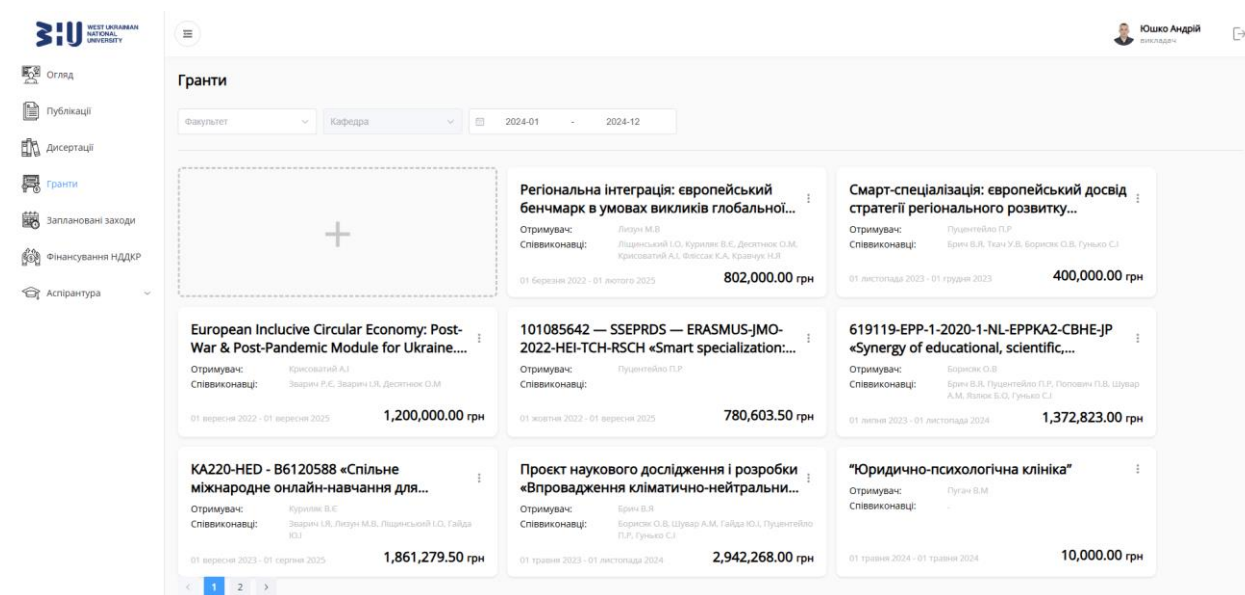


Рисунок 4.34 – Сторінка перегляду грантів

На рисунку 4.35 можна побачити як виглядає форма для додавання нового гранту.

The screenshot shows the 'Додати грант' (Add Grant) form in the ZNU system. The form includes the following fields and sections:

- Повна назва гранту**: Text input field.
- Назва програми, в рамках якої наданий грант**: Text input field.
- Ким наданий грант**: Text input field.
- Обсяг фінансування**: Text input field with a unit dropdown set to 'грн'.
- Терміни виконання**: Date range selector with 'з' (from) and 'по' (to) options.
- Мета проекту**: Text input field.
- Основні наукові результати**: Text input field.
- Одержувач гранту**: Section with three input fields for 'Прізвище', 'Ім'я', and 'По батькові'.
- Співвиконавці**: Text input field.

At the bottom right, there are two buttons: 'Скасувати' (Cancel) and 'Зберегти' (Save).

Рисунок 4.35 – Форма додавання гранту

В розділі заплановані заходи, користувачі можуть запланувати нові події та переглянути вже існуючі(Рис. 4.36).

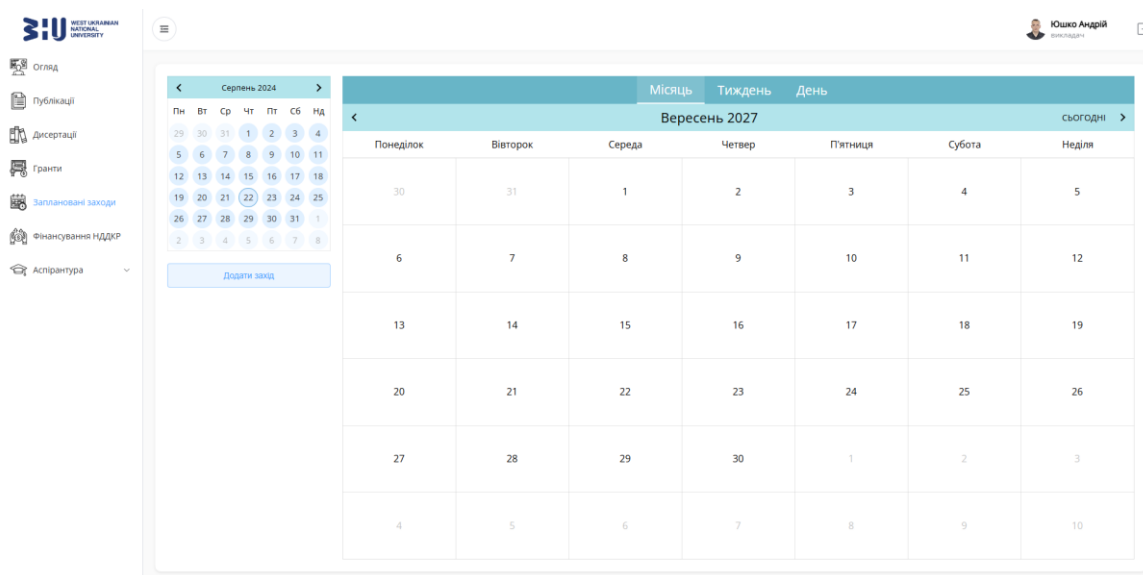


Рисунок 4.36 – Сторінка перегляду та додавання заходів

На рисунку 4.37 наведено форму додавання заходів, яка містить різноманітні обов'язкові поля, позначені зірочкою, для збору інформації. Вона включає такі поля, як «Назва», «Тип», «Факультет», «Кафедра», «Дата проведення», «Місце проведення», «Кількість учасників» та «Співорганізатори». Також є розділи для контактних даних організаційного комітету, де потрібно вказати поштову адресу, телефон та електронну адресу.

Рисунок 4.37 – Форма планування заходу

На сторінці фінансування НДДКР відображається перелік проєктів з фінансуванням науково-дослідних і дослідно-конструкторських робіт (НДДКР), включаючи назву, керівника, терміни, суму фінансування, тип фінансування та категорію кожного проєкту (Рис. 4.38).

№	Назва	Керівник	Терміни	Фінансування	Тип фінансування	Категорія	Дії
1	Концепція відбудови та зеленої реконструкції України	Крисоватий Андрій Ігорович	01-2024 - 12-2026	3,600,000.00 грн	Кошти державні	Фундаментальна	⋮
2	Удосконалення соціально-економічних аспектів планування, обліку та контролю в системі управління підприємством	Пивовць Віктор Михайлович	01-2020 - 12-2024	0.00 грн	Безкоштовно	Приміщення	⋮
3	Оптимізація фінансового забезпечення управління запасами на підприємстві	Чорний Роман Степанович	01-2024 - 01-2024	30,000.00 грн	Безкоштовно	Інше	⋮
4	Інтелектуально-вимірочні модулі для забезпечення високої якості інформативності, живучості та достовірності систем моніторингу об'єктів критичної інфраструктури	Конан Володимир Володимирович	01-2024 - 12-2024	1,080,000.00 грн	Кошти державні	Інше	⋮
5	Трансформація ринку логістичних послуг в умовах карантинації	Зварич Роман Євгенович	08-2024 - 06-2025	200,000.00 грн	Кошти підприємств	Приміщення	⋮
6	Удосконалення системи управління персоналом на підприємстві	Пивовць Віктор Михайлович	02-2024 - 12-2024	25,000.00 грн	Кошти підприємств	Інше	⋮
7	Розробка Стратегії розвитку Збаразької територіальної громади (інформаційно-аналітичний матеріал)	Шкільняк Михайло Михайлович	01-2024 - 01-2024	75,000.00 грн	Кошти місцеві	Інше	⋮
8	Модель регіональної безпеки: економічні й технічні аспекти сталого розвитку та цивільного захисту під час війни	Литун Марія Володимирівна	01-2024 - 12-2024	640,000.00 грн	Кошти державні	Фундаментальна	⋮
9	Формування екологічного бренду компанії на зарубіжних ринках	Зварич Роман Євгенович	04-2024 - 03-2025	200,000.00 грн	Кошти підприємств	Приміщення	⋮
10	Модель регіональної безпеки: економічні й технічні аспекти сталого розвитку на цивільного захисту під час війни	Литун Марія Володимирівна	01-2024 - 12-2026	2,400,000.00 грн	Кошти державні	Фундаментальна	⋮

Рисунок 4.38 – Сторінка перегляду фінансування НДДКР

Також передбачена можливість швидкого пошуку за факультетом, кафедрою та терміном виконання.

Як вже згадувалося раніше в системі передбачена можливість автоматичного формування профілю науковця, який в подальшому може використовуватися для задач фільтрування публікацій.

На рисунку 4.39 відображена сторінка перегляду профілю науковця.

Налаштування

Профіль науковця

Профіль користувача

Прізвище: Юшко, Ім'я: Андрій, По батькові: Васильович

Прізвище на англійській: Yushko, Ім'я на англійській: Andriy, По батькові на англійській: Vasylivich

Факультет: факультет комп'ютерних інформаційних технологій, Кафедра: Кафедра комп'ютерних наук

Scopus ID: 57288540800, Web of Science ID: JCE-4944-2023, Google Scholar ID: Ix_BwzqIAAAJ, ORCID: 0009-0003-6431-3479

Посада: викладач, Email: a.yushko@znu.edu.ua

Зберегти

Рисунок 4.39 – Сторінка профілю науковця

Як видно з зображення вище користувач має можливість не тільки переглянути свій профіль, але і відредагувати потрібну інформацію.

Висновки до розділу 4

1. У розділі обґрунтовано доцільність використання багаторівневої архітектури програмної системи, що поєднує клієнтську частину на основі Vue.js, серверну логіку з використанням Node.js та GraphQL, а також гнучке сховище MongoDB. Такий підхід забезпечує модульність, масштабованість, високу швидкість обробки даних і можливість інтеграції різномірних інформаційних джерел, включно з міжнародними наукометричними сервісами.

2. Розроблені механізми автоматизованого збору та синхронізації наукової інформації довели свою ефективність у роботі з великими масивами даних різної структури. Інтеграція інтелектуальних методів аналізу, зокрема модулів штучного інтелекту та алгоритмів структурного парсингу, забезпечила підвищення точності вилучення метаданих і формування наукових профілів викладачів.

3. Представлені UML-діаграми підтверджують логічну цілісність системи та демонструють оптимальну організацію взаємодії між користувачами різних ролей і підсистемами програмного комплексу. Вони засвідчують відповідність проєктних рішень сучасним стандартам моделювання складних інформаційних систем та дозволяють формалізувати функціональні вимоги до програмного забезпечення.

4. Результати модульного, інтеграційного, функціонального та навантажувального тестування підтверджують високу надійність, стабільність і продуктивність системи. Виконані тестові сценарії продемонстрували коректність бізнес-логіки, стійкість до помилкових умов, здатність системи ефективно функціонувати при одночасній обробці великої кількості запитів та відповідність вимогам інформаційної безпеки.

5. Створений графічний інтерфейс забезпечує інтуїтивність, ергономічність і зручність використання, що підвищує доступність програмного середовища для широкого кола користувачів. Інтерфейс успішно реалізує повний цикл роботи з даними наукової діяльності — від авторизації та

синхронізації до формування звітності, що підтверджує завершеність і практичну готовність системи до впровадження у реальних умовах діяльності закладу освіти.

ВИСНОВКИ

У дисертаційній роботі розв'язано науково-практичну проблему створення інтелектуалізованої системи збору, інтеграції, аналізу та прогнозування наукової діяльності науково-педагогічних працівників, яка поєднує сучасні підходи до багатоджерельної обробки наукових даних із математичними методами інтервального аналізу. Запропоновані моделі, інформаційні технології та архітектурні рішення забезпечують підвищення достовірності інформаційного забезпечення, усунення фрагментарності даних та формування обґрунтованих прогнозів сумарного наукового навантаження навіть за умов неповноти, нерегулярності та неоднорідності вибірок.

1. Проведено аналіз методів та засобів збору, інтеграції, аналізу та прогнозування показників наукової діяльності НПП, що дало змогу встановити ключові структурні та технологічні недоліки існуючих CRIS- і RIMS-систем. Виявлено відсутність персоналізованих механізмів прогнозування, недостатність інтелектуальних модулів обробки даних, низьку сумісність джерел інформації та брак математично обґрунтованих моделей, здатних працювати з обмеженими й зашумленими даними. Визначені обмеження стали підґрунтям для розробки нових методів і технічних рішень.

2. Розроблено ієрархічну систему показників наукової активності, яка формалізує структуру наукової діяльності НПП та забезпечує уніфіковане подання різних її видів. Побудована ієрархія охоплює узагальнені та деталізовані показники публікаційної, проектної, грантової, дисертаційної, патентної, експертної та іншої активності, створюючи цілісну модель наукового профілю дослідника. Сформована система стала теоретичною основою математичного моделювання наукового навантаження.

3. Розроблено інформаційну технологію збору, інтеграції, аналізу та прогнозування наукових показників, яка забезпечує інтелектуалізоване об'єднання різнорідних джерел у єдине уніфіковане середовище. Технологія включає алгоритми автоматизованого збору (API-взаємодія, веб-парсинг, ОАІ-

PMH), нормалізацію, семантичну верифікацію, видалення дублювань, фільтрацію нерелевантних публікацій на основі NLP та векторних подань, а також формування персоналізованих профілів науковців. Завдяки цьому створено узгоджену базу даних високої якості, придатну для прогнозного аналізу.

4. Розроблено математичну модель прогнозування сумарного наукового навантаження НПП із використанням інтервального аналізу даних. Модель описує наукову активність через інтервальні параметри та еліпсоїдні оцінки вагових коефіцієнтів, що відображають «портрет» науковця. Такий підхід забезпечує стійкість до неповноти та коливань у даних, дозволяє формувати прогноз у вигляді діапазону можливих значень і гарантує отримання реалістичних, індивідуалізованих планових показників наукової роботи.

5. Розроблено нові архітектурні рішення для інтелектуалізації процесів збору, аналізу та прогнозування, а також створено прикладне програмне забезпечення у вигляді інтелектуалізованої системи для планування наукової діяльності НПП. Система включає модулі ETL-обробки, семантичної верифікації, фільтрації даних, інтервального моделювання, формування профілів та генерації прогнозів. Використання сервісно-орієнтованої архітектури забезпечує модульність, масштабованість, сумісність з міжнародними інформаційними ресурсами та можливість інтеграції з національною інфраструктурою наукових даних.

6. Проведено апробацію розробленої системи на прикладі ЗВО, що підтвердило її практичну ефективність. Використання системи покращило повноту та якість даних, зменшило кількість дублювань, скоротило час підготовки аналітичних матеріалів та забезпечило достовірні прогнози сумарного наукового навантаження для різних категорій науково-педагогічних працівників. Отримані результати засвідчують, що запропоновані математичні, інформаційні та архітектурні засоби формують новий інтелектуалізований підхід до планування наукової діяльності.

Узагальнюючи, у дисертаційній роботі створено цілісну науково обґрунтовану концепцію інтелектуалізованої підтримки процесів збору, аналізу та прогнозування наукової діяльності НПП. Розроблений методологічний, математичний та програмний апарат усуває ключові недоліки існуючих систем, забезпечує персоналізовані оцінки наукової активності та окреслює перспективи подальшого розвитку інформаційних технологій для управління науковою діяльністю в закладах вищої освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрашко Ю. В. Огляд методів оцінювання діяльності науково-педагогічних працівників та вищих навчальних закладів // Управління розвитком складних систем. – 2017. – № 29. – С. 151–159.
2. Двухглавов Д. Е. Формалізація процесу підготовки плану навчального навантаження викладачів кафедри закладу вищої освіти // Вісник Національного технічного університету "ХПІ". Серія: Системний аналіз, управління та інформаційні технології. – 2022. – Т. 1, № 7. – С. 35–45.
3. Дзюба С. Облік робочого часу й оцінювання трудових затрат науково-педагогічних працівників закладів вищої освіти України / С. Дзюба // Вища освіта України. – 2018. – № 4 (71). – С. 15–24.
4. Дивак М. П. Задачі математичного моделювання статичних систем з інтервальними даними : монографія. – Тернопіль : Економічна думка ТНЕУ, 2011. – 215 с.
5. Дивак М. П. Ідентифікація дискретних моделей систем з розподіленими параметрами на основі аналізу інтервальних даних : монографія / М. П. Дивак, Н. П. Порплиця, Т. М. Дивак. – Тернопіль : Економічна думка ТНЕУ, 2018. – 220 с.
6. Дивак М. П. Прикладні задачі структурної та параметричної ідентифікації інтервальних моделей складних об'єктів : монографія / М. П. Дивак, А. В. Пукас, Н. П. Парплиця, А. М. Мельник. – Тернопіль : Університетська думка, 2021. – 212 с.
7. Дивак М. П., Мельник А. М., Кедрін Є. С., Отоо Ф. А. Інтервальна модель портрету користувачів тематичної групи з проблем екології у соціальній мережі // Оптико-електронні інформаційно-енергетичні технології. – 2022. – Вип. 41, № 1. – С. 78–88.
8. Дивак М. П., Мельник А. М., Манжула В. І. Знання-орієнтовані системи для ідентифікації інтервальних математичних моделей складних

динамічних та статичних об'єктів : монографія. – Тернопіль : ЗУНУ, 2024. – 288 с.

9. Дивак М. П., Пукас А. В., Олійник І. С. Особливості комп'ютерної реалізації методу локалізації параметрів інтервальних моделей із виділенням «насиченого блоку» // Інформаційні технології та комп'ютерна інженерія. – 2014. – № 2. – С. 59–71.

10. Дивак М., Манжула В., Мельник А., Юшко А. Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. *Measuring and computing devices in technological processes*. 2024. №1. С. 125–139. <https://doi.org/10.31891/2219-9365-2024-77-15>

11. Дивак М., Юшко А. Математична модель прогнозування показників наукової діяльності науково-педагогічних працівників з використанням інтервального аналізу даних. *Herald of Khmelnytskyi National University. Technical Sciences*. 2025. Т. 353(3.2). С. 375–386. <https://doi.org/10.31891/2307-5732-2025-353-53>

12. Мельник А., Дивак М. Метод структурної ідентифікації інтервальних дискретних моделей складних об'єктів із адаптивним налаштуванням вибору структурних елементів // *Measuring and Computing Devices in Technological Processes*. – 2022. – № 3. – С. 61–72. <https://doi.org/10.31891/2219-9365-2022-71-3-7>

13. Петренко М. Г., Палагін О. В., Бойко М. О., Матвейшин С. М. Knowledge-Oriented Tool Complex for Developing Databases of Scientific Publications and Taking into account Semantic Web Technology // *Control Systems and Computers*. – 2022. – № 3 (299). – С. 11–28. <https://doi.org/10.15407/csc.2022.03.011>

14. Положення про планування та облік роботи науково-педагогічних працівників Західноукраїнського національного університету (нова редакція) [Електронний ресурс]. – Тернопіль: ЗУНУ, 2024. – 27 с. – Режим доступу:

https://www.wunu.edu.ua/pdf/public_inf/Polozhennya%20pro%20planuvannya%20ta%20oblik%20roboty.pdf (Дата звернення: 06.07.2025). – Назва з екрана.

15. Про вищу освіту [Електронний ресурс] : Закон України від 01.07.2014 № 1556-VII : станом на 1 черв. 2025 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1556-18#Text> (дата звернення: 06.07.2025). – Назва з екрана.

16. Про наукову і науково-технічну діяльність [Електронний ресурс] : Закон України від 26.11.2015 № 848-VIII : станом на 9 квіт. 2025 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/848-19#Text> (дата звернення: 06.07.2025). – Назва з екрана.

17. Про освіту [Електронний ресурс] : Закон України від 05.09.2017 № 2145-VIII : станом на 1 черв. 2025 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2145-19#Text> (дата звернення: 06.07.2025). – Назва з екрана.

18. 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. <https://doi.org/10.31891/2219-9365-2025-81-45>

19. Спірін О. Основні складники цифрової компетентності наукових і науково-педагогічних працівників закладів вищої освіти України // *UNESCO Chair Journal "Lifelong Professional Education in the XXI Century"*. – 2024. – Т. 2, № 10. – С. 91–103. [https://doi.org/10.35387/ucj.2\(10\).2024.0007](https://doi.org/10.35387/ucj.2(10).2024.0007)

20. Юшко А. В., Сімак А. В. Інформаційна інтелектуальна система аналізу наукової та науково-педагогічної діяльності академічного колективу. *Оптико-електронні інформаційно-енергетичні технології*. 2024. Вип. 47(1). С. 7–16. <https://doi.org/10.31649/1681-7893-2024-47-1-7-16>

21. Abubakar H. D., Umar M. Sentiment classification: review of text vectorization methods: bag of words, tf-idf, word2vec and doc2vec, *SLU J. Sci. Technol.* 4.1&2 (2022) 27–33. <https://doi.org/10.56471/slujst.v4i.266>

22. Afzal W., Alone S., & Torkar R. (2020). Software test process improvement: A systematic mapping study. *Journal of Software: Evolution and Process*, 32(10), e2265. <https://doi.org/10.1002/smr.2265>
23. Al-Rubaye H., & Al-Badri A. (2021). Performance analysis of MongoDB versus relational databases in big data applications. *Journal of Engineering and Applied Sciences*, 16(5), 645–651.
24. Alharbi F. R., & Csala D. (2022). A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach. *Inventions*, 7(4), 94. <https://doi.org/10.3390/inventions7040094>
25. Artz M. (2023). Contract testing in microservices: Pact and modern API ecosystems. *IEEE Software*, 40(2), 67–75. <https://doi.org/10.1109/MS.2022.3221440>
26. Auhunas S., Hauschke C., Kampe B., Rudakova T., Shapovalov V., & Shapovalov Y. (2024). Persistent conference identifiers in CRIS – first implementations with VIVO and URIS. *Procedia Computer Science*, 249, 254–263. <https://doi.org/10.1016/j.procs.2024.11.071>
27. Azeroual O., Saake G., Abuosba M., Schöpfel J. Solving problems of research information heterogeneity during integration – using the European CERIF and German RCD standards as examples. *Information Services and Use*. 2019;39(1-2):105-122. <https://doi.org/10.3233/ISU-180030>
28. Bisht K., Kumar A. A method for fuzzy time series forecasting based on interval index number and membership value using fuzzy c-means clustering. *Evol. Intel.* 16, 285–297 (2023). <https://doi.org/10.1007/s12065-021-00656-0>
29. Bollinger T. Crossref 4.4.2 XML Elements and Attributes, TAO Pub. 2021, 0426 (2021). <https://doi.org/10.48034/20210426>
30. Brown M., Gruen A., Maldoff G., Messing S., Sanderson Z., Zimmer M. Web scraping for research: legal, ethical, institutional, and scientific considerations, 2024. <https://doi.org/10.48550/arXiv.2410.23432>

31. Casola V., De Benedictis A., & Rak M. (2020). Security assessment of JSON Web Token implementations. *Future Generation Computer Systems*, 104, 110–123. <https://doi.org/10.1016/j.future.2019.10.006>
32. Cerosh R., & Thangaraju S. (2022). Implementing GraphQL for microservice-based architectures. *IJACSA*, 13(5), 121–129. <https://doi.org/10.14569/IJACSA.2022.0130514>
33. Chaman Sab M., KK M. A., & Bagalkoti V. (2024). Unlocking Research Potential with Pure Portal: A Review of Elsevier’s RIMS Solution. *Journal of Data Science, Informetrics, and Citation Studies*, 3(3), 374–379. <https://doi.org/10.5530/jcitation.3.3.41>
34. Chen Y., & Wen X. (2022). Automated API testing based on REST constraints. *Journal of Systems Architecture*, 127, 102538. <https://doi.org/10.1016/j.sysarc.2022.102538>
35. Chodorow K. (2021). *MongoDB: The Definitive Guide* (3rd ed.). O’Reilly.
36. Desyatnyuk O., Dyvak M., Yushko A., Koruts U., Nadvynychnyy S., Ostroverkhov V. The Method of Forecasting a Researcher’s Project and Publication Activity Indicators Based on Interval Data Analysis. 2025 15th International Conference on ACIT. 2025. P. 701–707. <https://doi.org/10.1109/acit65614.2025.11185612>
37. Dutta A., & Roy S. (2021). REST API testing automation: Approaches, tools, and best practices. *Software Quality Journal*, 29, 1081–1103. <https://doi.org/10.1007/s11219-021-09555-5>
38. Dyvak M. et al. A new method identification of interval discrete models of objects with a randomized selection of structural elements // 2023 13th International Conference on Advanced Computer Information Technologies (ACIT). – 2023. <https://doi.org/10.1109/acit58437.2023.10275572>
39. Dyvak M. et al. Interval model of the efficiency of the functioning of information web resources for services on ecological expertise // *Mathematics*. – 2020. – T. 8, № 12. – C. 2116. <https://doi.org/10.3390/math8122116>

40. Dyvak M. et al. Method of structural identification of nonlinear interval models of static objects // Information Technology and Computer Engineering. – 2022. – T. 54, № 2. – С. 103–114. <https://doi.org/10.31649/1999-9941-2022-54-2-103-114>
41. Dyvak M. et al. Structure identification of difference equations with interval estimates of their parameters // 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM). – 2019. <https://doi.org/10.1109/cadsm.2019.8779308>
42. Dyvak M. et al. The task of parametric identification the interval models with nonlinear parameters // 2022 12th International Conference on Advanced Computer Information Technologies (ACIT). – 2022. <https://doi.org/10.1109/acit54803.2022.9913166>
43. Dyvak M. et al. The task of structural identification the interval models of static objects with multiple parameters // 2022 12th International Conference on Advanced Computer Information Technologies (ACIT). – 2022. <https://doi.org/10.1109/acit54803.2022.9913146>
44. Dyvak M. et al. Transformation of mathematical model for complex object in form of interval difference equations to a differential equation // International Journal of Computing. – 2023. – С. 219–224. <https://doi.org/10.47839/ijc.22.2.3091>
45. Dyvak M., Stakhiv P., Pukas A. Algorithms of parallel calculations in task of tolerance ellipsoidal estimation of interval model parameters, Bull. Pol. Acad. Sci. 60.1 (2012). <https://doi.org/10.2478/v10175-012-0022-9>
46. Dyvak M., Yushko A., Savenko O. та ін. An intelligent information system for generating a scientist's scientometrics using content analysis methods. AISTDS'2024, Kyiv, 1 October 2024. https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf
47. Dyvak M., Yushko A., Savenko O. та ін. Web-oriented application for student attendance accounting with automatic parsing of class schedules. YAISD'2025, Ternopil, 8 September 2025. <https://ceur-ws.org/Vol-3974/paper17.pdf>
48. ECMA International. (2022). ECMA-404: The JSON Data Interchange Syntax.

49. Fatima S., Hussain A., Amir S. B. et al. (2023). XGBoost and Random Forest Algorithms: An in Depth Analysis. *Pakistan Journal of Scientific Research*, 3(1), 26–31. <https://doi.org/10.57041/vol3iss1pp26-31>
50. Felder F., Granata V., Minotti C. et al. (2025). Recommendation on how to implement the DataCite Metadata Schema for research data. Eawag: Swiss Federal Institute of Aquatic Science and Technology. <https://doi.org/10.55408/eawag:35606>
51. Formanek M. (2023). DSpace 7 Benefits: Is It Worth Upgrading?. *Information Technology and Libraries*, 42(3). <https://doi.org/10.5860/ital.v42i3.16209>
52. Garay A., Ruví R. Digital competence in scientific research in higher education // *International Journal of Health Sciences*. – 2022. – C. 5778–5787. <https://doi.org/10.53730/ijhs.v6ns5.10242>
53. Gkevrou M., Stamovlasis D. Illustration of a software-aided content analysis methodology applied to educational research, *Educ. Sci.* 12.5 (2022) 328. <https://doi.org/10.3390/educsci12050328>
54. Gonzalez-Argote D., & Gonzalez-Argote J. (2023). Generación de grafos a partir de metadatos de revistas científicas con el sistema OAI-PMH. *Seminars in Medical Writing and Education*, 2, 43. <https://doi.org/10.56294/mw202343>
55. Hang C.-N., Yu P.-D., Morabito R., Tan C.-W. Large language models meet next-generation networking technologies: A review, *Future Internet* 16.10 (2024) 365. <https://doi.org/10.3390/fi16100365>
56. Hartig O., & Pérez J. (2022). Semantics and complexity of GraphQL. *ACM Transactions on Database Systems*, 47(3), 1–48. <https://doi.org/10.1145/3527313>
57. Hobson T. (2022). *Practical MongoDB Aggregations*. <https://doi.org/10.1007/978-1-4842-8086-8>
58. Islam M., & Akther S. (2023). Evaluating performance of microservice architectures using container-based load testing. *IEEE Access*, 11, 80233–80248. <https://doi.org/10.1109/ACCESS.2023.3290904>

59. Jana A., Rout R. (2022), "Designing metadata schema for a human library: a prototype". *Digital Library Perspectives*, Vol. 38 No. 3 pp. 346–361. <https://doi.org/10.1108/DLP-07-2021-0052>
60. Jones M. (2020). *Bearer Token Usage*. RFC 6750. <https://doi.org/10.17487/RFC6750>
61. Julrode P. & Jarusawat P. (2025). Development of a Semantic Ontology for Knowledge of Ancient Lanna Documents. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*. 19. pp. 168-181. <https://doi.org/10.37936/ecti-cit.2025192.256289>
62. Júnior E. W., Farias K., & da Silva B. (2022). On the use of UML in the Brazilian industry: A survey. *Journal of Software Engineering Research and Development*, 10, Article 10. <https://doi.org/10.5753/jserd.2022.2554>
63. Kim K., Lee G., & Park S. (2023). Optimizing GraphQL performance for large-scale applications. *Journal of Web Engineering*, 21(8), 1615–1636.
64. Koç H., Erdoğan A. M., Barjakly Y., & Peker S. (2021). UML diagrams in software engineering research: A systematic literature review. *Proceedings*, 74(1), 13. <https://doi.org/10.3390/proceedings2021074013>
65. Koundouri P. et al. (2021). Open Access in Scientific Information: Sustainability Model and Business Plan for the Infrastructure and Organization of OpenAIRE. *Journal of Benefit-Cost Analysis*. <https://doi.org/10.1017/bca.2020.26>
66. Kumar Dr, Kumar Dr, & Pandey S. (2023). Multilingual Repository on (DSpace): A Study of Open Archives Initiative and Protocol for Metadata Harvesting (OAI-PMH). 10. 19-23. <https://doi.org/10.26562/irjcs.2023.v1003.02>
67. Le N., Tran D., Sturgill R. Content analysis of three-dimensional model technologies and applications for construction: current trends and future directions, *Sensors* 24.12 (2024) 3838. <https://doi.org/10.3390/s24123838>
68. Lee H.-S., Shim H.-S. Implementation of generative AI using metaverse-based LLM, *Korea Ind. Technol. Converg. Soc.* 29.2 (2024) 123–132. <https://doi.org/10.29279/jitr.2024.29.2.123>

69. Li W., Zhou J., Chen L. et al. Upper and Lower Bound Interval Forecasting Methodology Based on Ideal Boundary and Multiple Linear Regression Models. *Water Resour Manage* 33, 1203–1215 (2019). <https://doi.org/10.1007/s11269-018-2177-0>
70. Lin G., Lin A., & Gu D. (2022). Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient. *Information Sciences*, 608, 517-531. <https://doi.org/10.1016/j.ins.2022.06.090>
71. Liu J. (2021). Digital Object Identifier (DOI) and DOI Services: An Overview. *Libri*, 71(4), 349-360. <https://doi.org/10.1515/libri-2020-0018>
72. Lutsiv N., Maksymyuk T., Beshley M. et al. Deep semisupervised learning-based network anomaly detection in heterogeneous information systems, *Comput., Mater. & Contin.* 70.1 (2022) 413–431. <https://doi.org/10.32604/cmc.2022.018773>
73. Lytvyn V., Vysotska V., Pukach P. et al. Analysis of the developed quantitative method for automatic attribution of scientific and technical text content written in Ukrainian, *Eastern-European J. Enterp. Technol.* 6.2 (96) (2018) 19–31. <https://doi.org/10.15587/1729-4061.2018.149596>
74. Ma J., Xia D., Guo H. et al. Metaheuristic-based support vector regression for landslide displacement prediction: a comparative study. *Landslides* 19, 2489–2511 (2022). <https://doi.org/10.1007/s10346-022-01923-6>
75. Majchrzak T. A., & Ernst S. (2021). Automated web testing: Tools, techniques, and trends. *Computers*, 10(7), 82. <https://doi.org/10.3390/computers10070082>
76. Majumdar S., & Chatterjee D. (2024). Efficient data retrieval using GraphQL over NoSQL databases. *Information and Software Technology*, 161, 107302. <https://doi.org/10.1016/j.infsof.2023.107302>
77. Makridakis S., Spiliotis E., Assimakopoulos V. et al. (2023). Statistical, machine learning and deep learning forecasting methods: Comparisons and ways

forward. *Journal of the Operational Research Society*, 74(3), 840–859.
<https://doi.org/10.1080/01605682.2022.2118629>

78. Manzhula V., Dyvak M., Zabchuk V. The improved method for identifying parameters of interval nonlinear models of static systems // *International Journal of Computing*. – 2024. – C. 19–25. <https://doi.org/10.47839/ijc.23.1.3431>

79. Mehta K., & Persaud K. (2020). *Building Applications with MongoDB*. <https://doi.org/10.1007/978-1-4842-5787-7>

80. Mienye I. D., Swart T. G., & Obaido G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information*, 15(9), 517. <https://doi.org/10.3390/info15090517>

81. Milani F., & Valetto G. (2023). Schema evolution and JSON-based data quality issues in modern NoSQL systems. *Information and Software Technology*, 158, 107171. <https://doi.org/10.1016/j.infsof.2023.107171>

82. Mohammadi Ostani M., Ebadollah Amoughin J., Jalili Manaf M. (2024), "Enrichment of Schema.org with cultural heritage context standards for the description and processing of electronic thesis and dissertations". *Digital Library Perspectives*, Vol. 40 No. 2 pp. 244–263. <https://doi.org/10.1108/DLP-07-2023-0063>

83. MongoDB Inc. (2023). *MongoDB Manual 7.0*.

84. Montesinos López O.A., Montesinos López A., Crossa J. (2022). Support Vector Machines and Support Vector Regression. In: *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Springer, Cham. https://doi.org/10.1007/978-3-030-89010-0_9

85. Nayem H. M., Aziz S., & Kibria B. M. G. (2024). Comparison among Ordinary Least Squares, Ridge, Lasso, and Elastic Net Estimators in the Presence of Outliers: Simulation and Application. *International Journal of Statistical Sciences*, 24(20), 25–48. <https://doi.org/10.3329/ijss.v24i20.78212>

86. Nguyen T., Nguyen H., & Menzies T. (2021). Performance testing for modern web systems: A systematic review. *ACM Transactions on Software Engineering and Methodology*, 30(4), 1–38. <https://doi.org/10.1145/3447582>

87. Novytskyi O. (2025), "Metadata harvesting for digital library integration in Ukraine: a comparative study of the OAI-PMH protocol and VuFind's efficacy". *Digital Library Perspectives*, Vol. 41 No. 1 pp. 74–90. <https://doi.org/10.1108/DLP-03-2024-0034>
88. Ozkaya M., & Erata F. (2020). A survey on the practical use of UML for different software architecture viewpoints. *Information and Software Technology*, 121, 106275. <https://doi.org/10.1016/j.infsof.2020.106275>
89. Pau D. P., Aymone F. M. Forward learning of large language models by consumer devices, *Electronics* 13.2 (2024) 402. <https://doi.org/10.3390/electronics13020402>
90. Petroșanu D.-M., Pîrjan A., Tăbușcă A. Tracing the influence of large language models across the most impactful scientific works, *Electronics* 12.24 (2023) 4957. <https://doi.org/10.3390/electronics12244957>
91. Pham L. M., The H. C. LNLB-BERT: transformer for long document classification with multiple attention levels, *IEEE Access* (2024) 1. <https://doi.org/10.1109/access.2024.3492102>
92. Pirnau M., Botezatu M. A., Priescu I. et al. Content analysis using specific natural language processing methods for big data, *Electronics* 13.3 (2024) 584. <https://doi.org/10.3390/electronics13030584>
93. Popova A., Lyndina Y., Matseiko N. Scientific activity of academic staff at the university: reflections on the impact of contemporary conditions // *Scientific Papers of Berdiansk State Pedagogical University. Series: Pedagogical Sciences.* – 2023. – T. 1, № 2. – C. 56–70. <https://doi.org/10.31494/2412-9208-2023-1-2-56-70>
94. Proksch S., et al. (2021). Mutation testing in modern software development. *IEEE Transactions on Software Engineering*. <https://doi.org/10.1109/TSE.2021.3075097>
95. Pukas A., Melnyk A., Voytyuk I., Yushko A., Romanyuk M., Honchar L. Transactional Business Application Based on Microservice Architecture. 2021 11th

International Conference on ACIT. 2021. P. 564–567.
<https://doi.org/10.1109/acit52158.2021.9548385>

96. Pukas A., Simak A., Yushko A., Nadvynychnyy S., Shandruk S., Vatslavskiy O. Automated Reporting Module in the Academic Staff Performance Appraisal System. 2024 14th International Conference on ACIT. 2024. Vol. 2021. P. 824–827. <https://doi.org/10.1109/acit62333.2024.10712573>

97. Putri M., Wijaya I. G. P. S., Adi Praja F. P. et al. (2023). The Comparison Study of Regression Models (Multiple Linear Regression, Ridge, Lasso, Random Forest, and Polynomial Regression) for House Price Prediction in West Nusa Tenggara. 1-6. <https://doi.org/10.1109/ICADEIS58666.2023.10270916>

98. Qaiser A., Farooq M. U., Nabeel Mustafa S. M., & Abrar N. (2023). Comparative Analysis of ETL Tools in Big Data Analytics. *Pakistan Journal of Engineering and Technology*, 6(1), 7–12. <https://doi.org/10.51846/vol6iss1pp7-12>

99. Rahman M. (2023). Building Real-World Web Applications with Vue 3. Apress. <https://doi.org/10.1007/978-1-4842-9381-3>

100. Rinjeni T. P., Indriawan A., & Rakhmawati N. A. (2024). Matching scientific article titles using Cosine Similarity and Jaccard Similarity algorithm. *Procedia Computer Science*, 234, 553-560. <https://doi.org/10.1016/j.procs.2024.03.039>

101. Rojas J. M., & Fraser G. (2020). Unit testing in modern software engineering: Current trends and challenges. *Software Quality Journal*, 28, 871–912. <https://doi.org/10.1007/s11219-019-09488-0>

102. Romeo J., Raglianti M., Nagy C., & Lanza M. (2025). UML is back. Or is it? Investigating the past, present, and future of UML in open source software. In *ICSE 2025* (pp. 2342–2354). IEEE. <https://doi.org/10.1109/ICSE55347.2025.00155>

103. Rudenko S., Pitera V., Kovtun T. Information-Analytical System for Evaluating the Activity of the Scientific-Pedagogical Staff Based on KPIs at Odesa National Maritime University // *Proceedings of the 5th International Workshop IT Project Management (ITPM 2024)*. – 2024. URL: <https://ceur-ws.org/Vol-3865/paper1>

104. Sachenko A., Kochan V., Turchenko V. et al. "Intelligent nodes for distributed sensor network," IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference. 1999, pp. 1479-1484 vol.3. <https://doi.org/10.1109/IMTC.1999.776072>

105. Salman H. A., Kalakech A., & Steiti A. (Trans.). (2024). Random Forest Algorithm Overview. *Babylonian Journal of Machine Learning*, 2024, 69-79. <https://doi.org/10.58496/BJML/2024/007>

106. Santana F., Cordeiro A., & Oliveira Jr E. (2023). Use of the Dublin Core Standard to Express Open Metadata Related to Software Engineering Experiments. *Anais do Workshop de Práticas de Ciência Aberta para Engenharia de Software (OpenScienSE)*. 1-5. <https://doi.org/10.5753/opensciense.2023.235672>

107. Sathish T., SaiKumar D., Patil S. et al. (2024). Exponential smoothing method against the gradient boosting machine learning algorithm-based model for materials forecasting to minimize inventory. *AIP Advances*, 14(6). <https://doi.org/10.1063/5.0208491>

108. Schaffer A. L., Dobbins T. A. & Pearson S. A. Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. *BMC Med Res Methodol* 21, 58 (2021). <https://doi.org/10.1186/s12874-021-01235-8>

109. Schirrwagen J., Bardi A., Czerniak A. et al. (2020). Data Sources and Persistent Identifiers in the Open Science Research Graph of OpenAIRE. *International Journal of Digital Curation*. 15. 5. <https://doi.org/10.2218/ijdc.v15i1.722>

110. Seenivasan D. ETL vs ELT: Choosing the right approach for your data warehouse (February 28, 2022). *International Journal for Research Trends and Innovation(IJRTI)*, Vol.7, Issue 2, page no.110 – 122. <http://dx.doi.org/10.2139/ssrn.5148194>

111. Shahin M., & Ali N. (2022). Continuous testing in DevOps: A systematic literature review. *Information and Software Technology*, 145, 106827. <https://doi.org/10.1016/j.infsof.2021.106827>

112. Sheffer Y., Hardt D., & Jones M. (2020). OAuth 2.0 Security Best Current Practice. IETF Draft.
113. Shevchuk R., Mykytiuk V., Miramontes D. A., Yushko A., Yurchyshyn T., Mokhun S. Software Service for Studying Words of the Ukrainian Language. 2021 11th International Conference on ACIT. 2021. P. 648–651. <https://doi.org/10.1109/acit52158.2021.9548448>
114. Silva V. S., Matas L., Moreira T., & Segundo W. C. (2022). An ETL strategy for integrating the LA Referencia platform and VIVO for the Brazilian CRIS. *Procedia Computer Science*, 211, 111–117. <https://doi.org/10.1016/j.procs.2022.10.182>
115. Sina L. B., Secco C. A., Blazevic M., & Nazemi K. (2023). Hybrid Forecasting Methods—A Systematic Review. *Electronics*, 12(9), 2019. <https://doi.org/10.3390/electronics12092019>
116. Singh A., Garg A. Named entity recognition (NER) and relation extraction in scientific publications, *Int. J. Recent Technol. Eng. (IJRTE)* 12.2 (2023) 110–113. <https://doi.org/10.35940/ijrte.b7846.0712223>
117. Sivertsen G. (2019). Developing Current Research Information Systems (CRIS) as Data Sources for Studies of Research. In: *Springer Handbook of Science and Technology Indicators*. https://doi.org/10.1007/978-3-030-02511-3_25
118. Spatharioti S. E., Rothschild D., Goldstein D. G., Hofman J. M. 2025. Effects of LLM-based Search on Decision Making: Speed, Accuracy, and Overreliance. In *Proceedings of the 2025 CHI Conference*. <https://doi.org/10.1145/3706598.3714082>
119. Steck H., Ekanadham C., & Kallus N. (2024, May). Is Cosine-Similarity of Embeddings Really About Similarity? In *Companion Proceedings of the ACM Web Conference 2024*. <https://doi.org/10.1145/3589335.3651526>
120. Steiner J. (2021). JSON Web Token Best Practices. RFC 8725. <https://doi.org/10.17487/RFC8725>
121. Stock P., Bremer H., Lachmann R. et al. (2025). Dashboard Filtering Using LLM-Based Interfaces. <https://doi.org/10.1109/IV64158.2025.11097665>

122. Sun Z., Wang G., Li P. et al. (2024). An improved random forest based on the classification accuracy and correlation measurement of decision trees. *Expert Systems with Applications*, 237, 121549.
123. Sunye M., Lima K., Abelló A., Ferreira E. (2025). Proposal for a Common Conceptual Schema for Metadata Integration and Reuse Across Diverse Scientific Domains. In *Proceedings of the 27th International Conference on Enterprise Information Systems*. <https://doi.org/10.5220/0013291500003929>
124. Taelman R., Van Sande M., & Verborgh R. (2021). GraphQL-LD: Linked Data Querying with GraphQL. *Future Internet*, 13(6), 146. <https://doi.org/10.3390/fi13060146>
125. Torcal J., Moreno V., Llorens J., & Granados A. (2024). Creating and validating a ground truth dataset of UML diagrams using deep learning techniques. *Applied Sciences*, 14(23), 10873. <https://doi.org/10.3390/app142310873>
126. Torres J., & Casillas J. (2020). Load testing for large-scale distributed systems: Tools and best practices. *Journal of Systems and Software*, 170, 110753. <https://doi.org/10.1016/j.jss.2020.110753>
127. Tran N. K., Kühle L. C., & Klau G. W. (2024). A critical review of multi-output support vector regression. *Pattern Recognition Letters*, 178, 69-75. <https://doi.org/10.1016/j.patrec.2023.12.007>
128. Varghese J., & George S. (2021). An approach for automated GraphQL API testing using schema-based fuzzing. *International Journal of Computer Applications*, 174(9), 1–8.
129. Wadud M. A. H., Mridha M. F., & Rahman M. M. (2022). Word Embedding Methods for Word Representation in Deep Learning for Natural Language Processing. *Iraqi Journal of Science*, 63(3), 1349-1361. <https://doi.org/10.24996/ijcs.2022.63.3.37>
130. Walha A., Ghazzi F. & Gargouri F. Data integration from traditional to big data: main features and comparisons of ETL approaches. *J Supercomput* 80, 26687–26725 (2024). <https://doi.org/10.1007/s11227-024-06413-1>

131. Wang Y. Research on the TF–IDF algorithm combined with semantics for automatic extraction of keywords from network news texts, *J. Intell. Syst.* 33.1 (2024). <https://doi.org/10.1515/jisys-2023-0300>
132. Williams E., & Nagle F. (2020). Empirical analysis of GraphQL adoption in modern web APIs. *IEEE Access*, 8, 226393–226407. <https://doi.org/10.1109/ACCESS.2020.3045729>
133. Wright A., Andrews H., & Ben-Kiki O. (2020). JSON Schema Specification.
134. Wu T., Chang Y., & Li C. (2021). Schema stitching in GraphQL for aggregating heterogeneous data sources. *Software: Practice and Experience*, 51(11), 2251–2268. <https://doi.org/10.1002/spe.3032>
135. Xiong J., Engelhard G., & Cohen A. S. (2025). Analysis of Mixed-Format Assessments Using Measurement Models and Topic Modeling. *Measurement: Interdisciplinary Research and Perspectives*, 23(2), 101–115. <https://doi.org/10.1080/15366367.2023.2298135>
136. Xu L., Xu H. & Ding F. Adaptive Multi-Innovation Gradient Identification Algorithms for a Controlled Autoregressive Autoregressive Moving Average Model. *Circuits Syst Signal Process* 43, 3718–3747 (2024). <https://doi.org/10.1007/s00034-024-02627-z>
137. Yan Z., et al. (2022). AI-assisted performance testing: A survey and future directions. *Journal of Systems Architecture*, 131, 102742. <https://doi.org/10.1016/j.sysarc.2022.102742>
138. Yu Y., Si X., Hu C., & Zhang J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architecture *Neural computation*, 31(7), 1235–1270. https://doi.org/10.1162/neco_a_01199
139. Yushko A., Dyvak M., Koruts U., Taraj M., Hrabar E., Sprinsyan V. The Information-Intellectual System for Evaluating Scientific and Scientific-Pedagogical Activities of the Academic Community. 2024 14th International Conference on ACIT. 2024. Vol. 2136. P. 828–833. <https://doi.org/10.1109/acit62333.2024.10712561>

140. Yushko A., Dyvak M., Melnyk A., Trufanova Y., Kobytisia V., Martsenyuk Y. AI-Based Vector Filtering of Irrelevant Academic Publications Using Research Interest Modeling. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT). 2025. P. 779–784. <https://doi.org/10.1109/acit65614.2025.11185809>
141. Yushko A., Shevchuk R., Łopaciński K., Leszczynska M., Yashchyk O., Yurchyshyn T. Shielding Web Application against Cyber-Attacks using SIEM. 2023 13th International Conference on ACIT. 2023. P. 393–396. <https://doi.org/10.1109/acit58437.2023.10275630>
142. Zaki Ahmed A., Rodríguez Díaz M. A methodology for machine-learning content analysis to define the key labels in the titles of online customer reviews with the rating evaluation, *Sustainability* 14.15 (2022) 9183. <https://doi.org/10.3390/su14159183>
143. Zhang D., Zhang N., Ye N., Fang J., & Han X. (2020). Hybrid learning algorithm of radial basis function networks for reliability analysis. *IEEE Transactions on reliability*, 70(3), 887-900. <https://doi.org/10.1109/TR.2020.3001232>
144. Zhang M., Harman M., & Jia Y. (2020). Search-based unit test generation: A survey. *ACM Computing Surveys*, 53(2), 1–35. <https://doi.org/10.1145/3379344>
145. Zhao N., 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>
146. Zhu H., Sun Y., Zhou H., Guan Y., Wang N., Ma W. Intelligent clustering-based interval forecasting method for photovoltaic power generation using CNN–LSTM neural network. *AIP Advances* 1 June 2024; 14 (6): 065329. <https://doi.org/10.1063/5.0213067>
147. Zuniga-Garcia M. A., Santamaría-Bonfil G., Arroyo-Figueroa G., & Batres R. (2019). Prediction Interval Adjustment for Load-Forecasting using Machine Learning. *Applied Sciences*, 9(24), 5269. <https://doi.org/10.3390/app9245269>

148. Tohalino, J. A. V., & Amancio, D. R. (2022). On predicting research grants productivity via machine learning. *Journal of Informetrics*, 16(2), 101260. <https://doi.org/10.1016/j.joi.2022.101260>
149. Ma, A., Liu, Y., Xu, X., & Dong, T. (2021). A deep-learning based citation count prediction model with paper metadata semantic features. *Scientometrics*, 126(8), 6803–6823. <https://doi.org/10.1007/s11192-021-04033-7>
150. Wang, B., Wu, F., & Shi, L. (2023). AGSTA-NET: Adaptive graph spatiotemporal attention network for citation count prediction. *Scientometrics*, 128(1), 511–541. <https://doi.org/10.1007/s11192-022-04541-0>

ДОДАТОК А

ПОРІВНЯЛЬНА ТАБЛИЦЯ МЕТОДІВ ПРОГНОЗУВАННЯ НАУКОВОЇ ДІЯЛЬНОСТІ

Таблиця А.1

Порівняльна таблиця методів прогнозування наукової діяльності

№	Метод	Тип моделі	Облік нелінійності	Урахування часової динаміки	Стійкість до шуму	Інтерпретованість результатів	Необхідність попередньої обробки даних	Складність реалізації	Вимоги до обсягу даних	Основні переваги	Основні недоліки	Типові сфери застосування
1	Лінійна регресія	Статистична	Немає (лінійна залежність)	Частково (через тренд)	Низька	Висока	Мінімальна	Низька	Невеликий	Проста, швидкість, зрозумілі коефіцієнти	Не враховує складні взаємозв'язки, чутлива до викидів	Початковий трендовий аналіз
2	Метод ковзного середнього	Статистична	Немає	Так (середнє вікно)	Висока	Висока	Не потребує	Дуже низька	Малий	Згладжує флуктуації, стабілізує ряд	Затримка прогнозу, втрата короткотрокових змін	Стабільні часові ряди без різких коливань
3	Експоненційне згладжування	Статистична	Частково (через вагові коефіцієнти)	Так	Середня	Висока	Мінімальна	Низька	Малий–середній	Добре відстежує тренд, просто реалізується	Не враховує зовнішніх факторів, можливе запізнення	Короткотроковий прогноз при стабільному тренді
4	Random Forest	Машинне навчання (ансамбль дерев)	Так (через розгалуження рішень)	Ні (немає пам'яті у часі)	Висока	Середня	Потрібна нормалізація та кодування ознак	Середня	Середній–великий	Стійкість до шуму, точність, можливість оцінки важливості ознак	«Чорна скринька», велика кількість параметрів	Багатофакторні моделі з незалежними змінними

Продовження таблиці А.1

5	Gradient Boosting	Машинне навчання (последовний ансамбль дерев)	Так	Ні	Висока	Низька	Бажана стандартизація даних	Висока	Середній–великий	Висока точність, врахування складних залежностей	Ризик перенавчання, потребує налаштування параметрів	Точні прогнози на структурованих наборах даних
6	Support Vector Regression (SVM)	Машинне навчання (метод опорних векторів)	Так (через ядро)	Частково (через тренд)	Середня	Середня	Обов'язкове масштабування ознак	Середня	Малий–середній	Висока узагальнювальна здатність, робота з малою вибіркою	Чутливість до вибору ядра, складне налаштування параметрів	Прогнозування за невеликими, але багатомірними вибірками
7	LSTM (Recurrent Neural Network)	Глибинне навчання (рекурентна мережа)	Так	Так (довготривала пам'ять)	Висока	Низька	Потрібна нормалізація та формування послідовностей	Висока	Великий	Враховує часові залежності, навчається на послідовностях, здатна до самокорекції	Вимоглива до обчислювальних ресурсів, потребує багато даних	Прогнозування часових рядів та трендів у динамічних системах
8	Інтервальний підхід (паралельна / еліпсоїдна модель)	Інтервальний аналіз	Частково (через межі інтервалів)	Так (через історію змін)	Дуже висока	Висока	Потрібна попередня оцінка невизначеності	Висока	Малий	Оцінює діапазон прогнозу, працює з неточними або неповними даними	Не дає точкового результату, складність обчислень	Системи зі значною невизначеністю даних або малими вибірками

ДОДАТОК Б

ЛІСТИГ КОДУ ОСНОВНИХ МОДУЛІВ ПРОГРАМИ

```

import mongoose from 'mongoose';
import crypto from "crypto"
import fs from "fs";
import csvParser from "csv-parser";
import { signToken, hashPassword, verifyPassword, extractUserInfo } from
"../utils/index.js";
import { User, Entrant, Publication, Dissertation, Event, Job, Grant, Project,
Employee, Scientist, Supervisor, Interview, ScientificProfile } from
'../db/dbConnector.js'
import { sendEmail } from '../utils/mailer.js';
import { fullSyncQueue } from "../queues/fullSynchronization.js"
import { generateReport } from "../report/generate-report.js"
import dayjs from 'dayjs';
import customParseFormat from 'dayjs/plugin/customParseFormat.js';
import { formatAuthorMatchPipeline } from "../utils/authorMatchPipeline.js";
import { getUsersWithDetails, countUsersByCriteria } from
"../helpers/getUserDetails.js"
import { getTeachersMetadata } from "../helpers/parseTeachersMetadata.js"
import { fullSynchronization } from '../jobs/fullSynchronization.js';
import { findFacultyByDepartmentLabel, findDepartmentByLabel,
findDepartmentBySpeciality } from '../constants/organizationStructure.js';
import { logger } from "../utils/logger.js"
import { redis } from "../queues/redisConnection.js";

dayjs.extend(customParseFormat);

const convertToISO = (dateString) => {
  return dayjs(dateString, 'DD.MM.YYYY').toISOString();
}

const getFilterPipelines = async (input, ctx, periodField = "date") => {
  const { userId } = extractUserInfo(ctx)
  let { filter = {} } = input;
  const defaultPeriod = [dayjs().startOf("year"), dayjs().endOf("year")];
  const { scopusID, role, firstName, lastName, middleName, enFirstName, enLastName,
department } = await User.findOne({ _id: new mongoose.Types.ObjectId(userId) })
  let pipelines = [];
  if (role === "TEACHER") {
    const authorConditions = [
      ...formatAuthorMatchPipeline({ firstName, lastName, middleName, enFirstName,
enLastName }),
      { addedByUserId: new mongoose.Types.ObjectId(userId) }
    ];
    if (scopusID) {
      authorConditions.push({ scopusID });
    }
    pipelines.push({
      $match: {
        $or: authorConditions
      },
    });
  }
  if (role === "HEAD_OF_DEPARTMENT") {

```

```

    pipelines.push({
      $match: {
        department: department,
      }
    })
  }
  if (!filter.period || filter?.period?.length === 0) {
    filter = {
      ...filter,
      period: defaultPeriod
    }
  }
  Object.keys(filter).forEach(field => {
    if (field === "period" && periodField === "date") {
      pipelines.push({
        $match: {
          date: {
            $gte: new Date(filter.period[0]),
            $lte: new Date(filter.period[1])
          }
        }
      })
    } else if (field === "period" && periodField !== "date") {
      pipelines.push({
        $match: {
          $or: [
            {
              $and: [
                { [`${periodField}.0`]: { $lte: new
Date(filter.period[1]) } }, // Початок проєкту менший або рівний кінцевій даті фільтру
                { [`${periodField}.1`]: { $gte: new
Date(filter.period[0]) } } // Кінець проєкту більший або рівний початковій даті
                фільтру
              ]
            }
          ]
        }
      })
    } else if (field === "author" && filter?.author) {
      let authorMatch = [];
      filter.author.forEach(author => {
        const formattedAuthorMatch = formatAuthorMatchPipeline(author)
        authorMatch = [...authorMatch, ...formattedAuthorMatch]
      })
      if (authorMatch.length > 0) {
        pipelines.push({
          $match: {
            $or: [
              ...authorMatch,
            ]
          }
        })
      }
    } else {
      if (Array.isArray(filter[field])) {
        const matchOr = filter[field].map(item => ({
          [field]: item
        }));
        if (matchOr.length > 0) {
          pipelines.push({
            $match: {

```

```

        $or: matchOr
      }
    });
  }
} else {
  pipelines.push({
    $match: {
      [field]: filter[field]
    }
  });
}
});
}
});
});

pipelines.push({
  $group: {
    _id: { lowerCaseName: { $toLower: "$name" } },
    doc: { $first: "$$ROOT" }
  }
});
pipelines.push({
  $replaceRoot: { newRoot: "$doc" }
});

return pipelines
}

const getPaginationPipelines = (input) => {
  const { pagination = {} } = input;
  return [
    { $skip: pagination?.offset || 0 },
    { $limit: pagination?.limit || 10 },
  ]
}

export const resolvers = {
  Query: {
    getAllUsers: (root) => {
      return new Promise((resolve, reject) => {
        User.find((err, users) => {
          if (err) reject(err);
          else resolve(users);
        })
      })
    },
    getEvents: async (_, { input }, ctx) => {
      const filterPipelines = await getFilterPipelines(input, ctx)
      const pipelines = [
        ...filterPipelines,
      ]
      const events = await Event.aggregate(pipelines);
      return events
    },
    getProjects: async (_, { input }, ctx) => {
      const filterPipelines = await getFilterPipelines(input, ctx, "terms")
      const totalCount = await Project.aggregate([
        ...filterPipelines,
        {
          $group: {
            _id: null,
            total: { $sum: 1 }
          }
        }
      ])
    }
  }
}

```

```

    }
  });
  const pipelines = [
    ...filterPipelines,
    { $sort: { date: 1 } },
    ...getPaginationPipelines(input)
  ]
  const projects = await Project.aggregate(pipelines)
  return {
    projects,
    total: totalCount[0]?.total || 0
  };
},
getGrants: async (_, { input }, ctx) => {
  const filterPipelines = await getFilterPipelines(input, ctx,
"executionTerms")
  const totalCount = await Grant.aggregate([
    ...filterPipelines,
    {
      $group: {
        _id: null,
        total: { $sum: 1 }
      }
    }
  ]);
  const pipelines = [
    ...filterPipelines,
    { $sort: { date: 1 } },
    ...getPaginationPipelines(input)
  ]
  const grants = await Grant.aggregate(pipelines)
  return {
    grants,
    total: totalCount[0]?.total || 0
  };
},
getEvent: async (_, { id }, ctx) => {
  return await Event.findOne({ _id: new mongoose.Types.ObjectId(id) });
},
getGrant: async (_, { id }, ctx) => {
  return await Grant.findOne({ _id: new mongoose.Types.ObjectId(id) });
},
getPublications: async (_, { input }, ctx) => {
  const filterPipelines = await getFilterPipelines(input, ctx)
  const totalCount = await Publication.aggregate([
    ...filterPipelines,
    {
      $group: {
        _id: null,
        total: { $sum: 1 }
      }
    }
  ]);
  const pipelines = [
    ...filterPipelines,
    { $sort: { date: 1 } },
    ...getPaginationPipelines(input)
  ]
  const publications = await Publication.aggregate(pipelines)
  return {

```

```

        publications,
        total: totalCount[0]?.total || 0
    });
},
getDissertations: async (_, { input }, ctx) => {
    const filterPipelines = await getFilterPipelines(input, ctx)
    const totalCount = await Dissertation.aggregate([
        ...filterPipelines,
        {
            $group: {
                _id: null,
                total: { $sum: 1 }
            }
        }
    ]);
    const pipelines = [
        ...filterPipelines,
        { $sort: { date: 1 } },
        ...getPaginationPipelines(input)
    ]
    const dissertations = await Dissertation.aggregate(pipelines)
    return {
        dissertations,
        total: totalCount[0]?.total || 0
    };
},
getPublicationActivityCount: async (_, { input }, ctx) => {
    const filterPipelines = await getFilterPipelines(input, ctx)
    const byCategoryPipelines = [
        ...filterPipelines,
        {
            $unwind: "$category"
        },
        {
            $group: {
                _id: "$category",
                count: { $sum: 1 }
            }
        },
        {
            $project: {
                _id: 0,
                category: "$_id",
                count: 1
            }
        }
    ]
    const byTypePipelines = [
        ...filterPipelines,
        {
            $group: {
                _id: "$type",
                count: { $sum: 1 }
            }
        },
        {
            $project: {
                _id: 0,
                type: "$_id",
                count: 1
            }
        }
    ]
}

```

```

    }
  ]
  const byCategoryCount = await Publication.aggregate(byCategoryPipelines)
  const byTypeCount = await Publication.aggregate(byTypePipelines)
  return [...byCategoryCount, ...byTypeCount].filter(activity =>
activity.type || activity.category)
  },
  getPublicationActivity: async (_, { input }, ctx) => {
    const { atLeastNumber = false } = input;
    let pipelines = [];
    const employees = await getUsersWithDetails(input)
    const totalCount = await countUsersByCriteria(input)
    pipelines = [
      ...pipelines,
      {
        $group: {
          _id: "$department",
          publicationCount: { $sum: 1 },
          totalPageAthorNumbers: { $sum: "$pageAthorNumbers" }
        }
      },
    ];

    if (atLeastNumber) {
      pipelines.push({
        $match: {
          publicationCount: { $gte: 5 }, // Filter authors with at least
5 publications
        },
      });
    }

    pipelines.push({
      $project: {
        publicationCount: 1,
        totalPageAthorNumbers: 1,
        _id: 0,
      },
    });

    const publicationsCountResult = await Promise.all(employees.map(async
(employee) => {
      const userDetails = employee?.userDetails || employee;
      const filterPipelines = await getFilterPipelines({
        ...input,
        filter: {
          ...input.filter,
          author: [userDetails]
        }
      }, ctx)
      const publications = await Publication.aggregate([
        ...filterPipelines,
        ...pipelines,
        { $sort: { publicationCount: -1 } }
      ])
      return {
        author: {
          firstName: employee.userDetails?.trimmedFirstName ||
employee.firstName,
          lastName: employee.userDetails?.trimmedLastName ||
employee.lastName,

```

```

        middleName: employee.userDetails?.trimmedMiddleName ||
employee.middleName
    },
    publicationCount: publications[0]?.publicationCount || 0,
    totalPageAthorNumbers: publications[0]?.totalPageAthorNumbers || 0,
  }
  )))
  if (input?.ignoreZeroResults) {
    const noneZeroResults = publicationsCountResult
      .filter(result => result.publicationCount !== 0 &&
result.totalPageAthorNumbers)
    return {
      publications: noneZeroResults,
      total: totalCount - noneZeroResults.length
    }
  }
  return {
    publications: publicationsCountResult,
    total: totalCount
  }
},
getPublicationActivityContribution: async (_, { input = {} }, ctx) => {
  const { atleastNumber = false } = input;
  const employees = await getUsersWithDetails(input);
  const totalCount = await countUsersByCriteria(input);

  const publicationsCountResult = await Promise.all(
    employees.map(async (employee) => {
      const userDetails = employee?.userDetails || employee;
      const filterPipelines = await getFilterPipelines({
        ...input,
        filter: {
          ...input.filter,
          author: [userDetails]
        }
      }, ctx);

      const publications = await Publication.aggregate([
        ...filterPipelines,
        {
          $project: {
            coAuthors: 1 // !!! тут тепер coAuthors
          }
        }
      ]);

      const publicationCount = publications.length;

      let totalContribution = 0;
      for (const pub of publications) {
        const coAuthorsCount = Array.isArray(pub.coAuthors) ?
pub.coAuthors.length : 0;
        const totalAuthors = coAuthorsCount + 1; // Співавтори + сам
автор

        totalContribution += 1 / totalAuthors;
      }

      return {
        author: {
          firstName: employee.userDetails?.trimmedFirstName ||
employee.firstName,

```



```

        lastName: employee.userDetails?.trimmedLastName ||
employee.lastName,
        middleName: employee.userDetails?.trimmedMiddleName ||
employee.middleName
    },
    publicationCount,
    totalContribution: Number(totalContribution.toFixed(3)) //
округлюємо як і треба
    });
  })
);

let finalResult = publicationsCountResult;
if (atLeastNumber) {
  finalResult = finalResult.filter(result => result.publicationCount >=
5);
}

if (input?.ignoreZeroResults) {
  finalResult = finalResult.filter(result => result.publicationCount !==
0 && result.totalContribution !== 0);
  return {
    publications: finalResult,
    total: totalCount - finalResult.length
  };
}

return {
  publications: finalResult,
  total: totalCount
};
},

checkImportStatus: async (_, { input }) => {
  const job = await Job.findOne(input);
  return {
    status: job?.status || null
  }
},

getActiveSyncProgress: async (_, { input }) => {
  // Отримуємо дані поточного користувача з контексту
  const { userId, department: userDepartment } = input;

  // === 1. Перевірка АКТИВНОЇ МАСОВОЇ СИНХРОНІЗАЦІЇ (Redis GET) ===
  const departmentKey = `:${userDepartment}`;
  const syncRunId = await redis.get(departmentKey);

  if (syncRunId) {
    // Знайдено активну групову синхронізацію.
    const syncKey = `:${syncRunId}`;
    const statusData = await redis.hgetall(syncKey);

    const total = parseInt(statusData.total) || 0;
    const completed = parseInt(statusData.completed) || 0;
    const failed = parseInt(statusData.failed) || 0;
    const totalProcessed = completed + failed;

    let progress = 0;
    if (total > 0) progress = (totalProcessed / total) * 100;

    return {

```

```

        syncRunId: syncRunId,
        status: statusData.status,
        progress: Math.min(100, progress.toFixed(2)),
        jobsCompleted: completed,
        jobsFailed: failed,
        totalJobs: total,
        isBatch: true,
    });
}

// === 2. Перевірка АКТИВНОЇ ІНДИВІДУАЛЬНОЇ СИНХРОНІЗАЦІЇ (Пошук у Bull)
===

const jobStatuses = ['active', 'waiting', 'delayed'];
const activeJobs = await fullSyncQueue.getJobs(jobStatuses);
// Шукаємо завдання, яке належить поточному користувачеві
const userJob = activeJobs.find(job =>
    job.data.isSingleUser && job.data.user.userId.toString() ===
    userId.toString()
);

if (userJob) {
    const activeJobId = userJob.id.toString();
    const status = await userJob.getState();

    let progress = 0;
    if (status === 'active') progress = 50;
    if (status === 'completed' || status === 'failed') progress = 100;

    // Перевіряємо, чи Bull не видалив завдання (стан completed/failed)
    if (status === 'completed' || status === 'failed') {
        return {
            status: 'NO_ACTIVE_SYNC',
            progress: 100,
            totalJobs: 0,
        };
    }

    return {
        syncRunId: activeJobId,
        status: status.toUpperCase(),
        progress: progress.toFixed(2),
        jobsCompleted: (status === 'completed' ? 1 : 0),
        jobsFailed: (status === 'failed' ? 1 : 0),
        totalJobs: 1,
        isBatch: false,
    };
}

// === 3. Немає активної синхронізації ===
return {
    status: 'NO_ACTIVE_SYNC',
    progress: 100,
    totalJobs: 0,
};
},
getScientists: async (_, { input }) => {
    const totalCount = await Scientist.aggregate([
        {
            $group: {
                _id: null,

```

```

        total: { $sum: 1 }
      }
    }
  });
const scientists = await Scientist.aggregate([
  {
    $lookup: {
      from: "supervisors",
      localField: "supervisor",
      foreignField: "_id",
      as: "supervisorInfo"
    }
  },
  {
    $unwind: {
      path: "$supervisorInfo",
      preserveNullAndEmptyArrays: true // Залишити вчених без
керівника
    }
  },
  {
    $lookup: {
      from: "employees",
      let: { employeeId: "$supervisorInfo.employeeID" },
      pipeline: [
        {
          $match: {
            $expr: {
              $eq: ["$_id", { $toObjectId: "$$employeeId" }]
            }
          }
        }
      ],
      as: "employeeInfo"
    }
  },
  {
    $unwind: {
      path: "$employeeInfo",
      preserveNullAndEmptyArrays: true // Залишити вчених без
інформації про співробітника
    }
  },
  {
    $addFields: {
      supervisor: "$employeeInfo"
    }
  },
  {
    $project: {
      supervisorInfo: 0, // Приховати непотрібні поля
      employeeInfo: 0
    }
  },
  ...getPaginationPipelines({ pagination: input })
]);

return {
  scientists,
  total: totalCount[0]?.total || 0
};

```

```

},
getSupervisors: async (_, { input }) => {
  let filterPipelines = [];
  const preProjectPipelines = []

  if (input?.filter?.filterbyQuota) {
    filterPipelines.push({
      $match: {
        $expr: {
          $and: [
            { $lte: ['$scientistCount', '$quota'] },
            { $gt: ['$quota', 0] }
          ]
        }
      }
    });
  }

  if (input?.filter?.faculty) {
    preProjectPipelines.push({
      $match: {
        'employeeInfo.faculty': input.filter.faculty
      }
    });
  }

  if (input?.filter?.firstName && input?.filter?.firstName.length > 0) {
    filterPipelines.push({
      $match: {
        firstName: { $regex: input.filter.firstName, $options: 'i' }
      }
    });
  }

  if (input?.filter?.speciality && input.filter.speciality.length > 0) {
    preProjectPipelines.push({
      $match: {
        specialities: { $in: input.filter.speciality }
      }
    });
  }

  const totalCount = await Supervisor.aggregate([
    {
      $lookup: {
        from: 'employees',
        localField: 'employeeID',
        foreignField: '_id',
        as: 'employeeInfo'
      }
    },
    {
      $unwind: {
        path: '$employeeInfo',
        preserveNullAndEmptyArrays: true
      }
    },
    ...preProjectPipelines,
    ...filterPipelines,
    {
      $group: {
        _id: null,

```

```

        total: { $sum: 1 }
      }
    });

filterPipelines = [
  ...filterPipelines,
  ...getPaginationPipelines(input)
]

const supervisors = await Supervisor.aggregate([
  {
    $lookup: {
      from: 'scientists',
      localField: '_id',
      foreignField: 'supervisor',
      as: 'scientists'
    }
  },
  {
    $lookup: {
      from: 'employees',
      localField: 'employeeID',
      foreignField: '_id',
      as: 'employeeInfo'
    }
  },
  {
    $unwind: {
      path: '$employeeInfo',
      preserveNullAndEmptyArrays: true
    }
  },
  ...preProjectPipelines,
  {
    $project: {
      _id: 1,
      firstName: 1,
      lastName: 1,
      middleName: 1,
      quota: 1,
      faculty: '$employeeInfo.faculty',
      degree: '$employeeInfo.degree',
      scientistCount: { $size: '$scientists' },
      scientists: 1,
      specialities: 1
    }
  },
  ...filterPipelines
]);

return {
  supervisors,
  total: totalCount[0]?.total || 0
}
},
getEntrants: async (_, { input }) => {
  const totalCount = await Entrant.aggregate([
    {
      $group: {

```

```

        _id: null,
        total: { $sum: 1 }
      }
    ]);
const entrants = await Entrant.aggregate([
  ...getPaginationPipelines(input)
]);

return {
  entrants,
  total: totalCount[0]?.total || 0
};
},
searchSupervisor: async (_, { input }) => {
  const searchResult = await Supervisor.aggregate([
    {
      $lookup: {
        from: 'scientists',
        localField: '_id',
        foreignField: 'supervisor',
        as: 'scientists'
      }
    },
    {
      $project: {
        _id: 1,
        firstName: 1,
        lastName: 1,
        middleName: 1,
        quota: 1,
        scientistCount: { $size: '$scientists' }
      }
    },
    {
      $match: {
        $and: [
          { firstName: { $regex: input.firstName, $options: 'i' } },
          { $expr: { $lt: ['$scientistCount', '$quota'] } }
        ]
      }
    }
  ])
  return searchResult
},
searchAuthor: async (_, { input }, ctx) => {
  const { author } = input;
  const employees = await Employee.aggregate([
    {
      $match: {
        $and: [
          { firstName: { $regex: author.firstName, $options: 'i' } },
        ]
      }
    },
    {
      $lookup: {
        from: "users",
        let: {
          employeeFirstName: { $trim: { input: "$firstName" } },
          employeeLastName: { $trim: { input: "$lastName" } },

```

```

        employeeMiddleName: { $trim: { input: "$middleName" } }
      },
      pipeline: [
        {
          $addFields: {
            trimmedFirstName: { $trim: { input: "$firstName" } },
            trimmedLastName: { $trim: { input: "$lastName" } },
            trimmedMiddleName: { $trim: { input: "$middleName" } }
          }
        },
        {
          $match: {
            $expr: {
              $and: [
                { $eq: ["$trimmedFirstName", "$$employeeFirstName" ] },
                { $eq: ["$trimmedLastName", "$$employeeLastName" ] },
                { $eq: ["$trimmedMiddleName", "$$employeeMiddleName" ] }
              ]
            }
          }
        }
      ],
      as: "userDetails"
    }
  },
  {
    $unwind: {
      path: '$userDetails',
      preserveNullAndEmptyArrays: true
    }
  },
  { $group: { _id: "$externalID", document: { $first: "$$ROOT" } } },
  { $replaceRoot: { newRoot: "$document" } },
  { $sort: { firstName: 1 } }
]);
return employees.map(employee => {
  return {
    firstName: employee.firstName,
    lastName: employee.lastName,
    middleName: employee.middleName,
    degree: employee.degree,
    enFirstName: employee?.userDetails?.enFirstName || null,
    enLastName: employee?.userDetails?.enLastName || null,
    enMiddleName: employee?.userDetails?.enMiddleName || null,
    externalID: employee.externalID,
    id: employee._id
  }
})
},
getInterviews: async (_, { input }, ctx) => {
  const { userId, role } = extractUserInfo(ctx)

  const pipelines = [];
  const interviewsPipelines = [
    {
      $lookup: {

```

```

        from: 'entrants',
        localField: 'entrantID',
        foreignField: '_id',
        as: 'entrantInfo'
      }
    },
    {
      $unwind: {
        path: '$entrantInfo',
        preserveNullAndEmptyArrays: true
      }
    },
    {
      $lookup: {
        from: 'supervisors',
        localField: 'supervisorID',
        foreignField: '_id',
        as: 'supervisorInfo'
      }
    },
    {
      $unwind: {
        path: '$supervisorInfo',
        preserveNullAndEmptyArrays: true
      }
    },
    {
      $addFields: {
        supervisor: "$supervisorInfo",
        entrant: "$entrantInfo"
      }
    },
    {
      $project: {
        supervisorInfo: 0,
        entrantInfo: 0,
      }
    }
  }
];

if (role === "TEACHER" || role === "HEAD_OF_DEPARTMENT") {
  const { firstName, lastName, middleName } = await User.findOne({ _id:
new mongoose.Types.ObjectId(userId) })
  pipelines.push({
    $match: {
      $and: [
        { "supervisor.firstName": { $regex: firstName, $options:
'i' } },
        { "supervisor.lastName": { $regex: lastName, $options: 'i'
} },
        { "supervisor.middleName": { $regex: middleName, $options:
'i' } }
      ]
    }
  });
}

if (role === "ENTRANT") {
  const { firstName, lastName, middleName } = await Entrant.findOne({
_id: new mongoose.Types.ObjectId(userId) })
  pipelines.push({

```



```

        $match: {
          $and: [
            { "entrant.firstName": { $regex: firstName, $options: 'i' } },
            { "entrant.lastName": { $regex: lastName, $options: 'i' } },
            { "entrant.middleName": { $regex: middleName, $options: 'i' } }
          ]
        }
      });
    }

    const totalCount = await Interview.aggregate([
      ...interviewsPipelines,
      ...pipelines,
      {
        $group: {
          _id: null,
          total: { $sum: 1 }
        }
      }
    ]);
    const interviews = await Interview.aggregate([
      ...interviewsPipelines,
      ...pipelines,
      ...getPaginationPipelines(input)
    ]);

    return {
      interviews,
      total: totalCount[0]?.total || 0
    };
  },
  hasActivity: async (_, { }, ctx) => {
    const filterPipelines = await getFilterPipelines({}, ctx)

    const publicationsTotalCount = await Publication.aggregate([
      ...filterPipelines,
      {
        $group: {
          _id: null,
          total: { $sum: 1 }
        }
      }
    ]);
    const hasPublications = publicationsTotalCount[0]?.total > 0
    return hasPublications
  }
},
Mutation: {
  registration: async (root, { input }) => {
    const { password, email, ...other } = input;
    const userExist = await User.findOne({ email });
    if (!userExist) {
      const hashedPassword = await hashPassword(password);
      const newUser = new User({
        ...other,
        email,
        password: hashedPassword,
        role: "TEACHER"
      });
    }
  }
}

```

```

    });

    return new Promise((resolve, reject) => {
      newUser.save((err) => {
        if (err) reject(err);
        else resolve({
          user: {
            ...other,
            email,
            id: newUser._id,
          },
          token: signToken({ userId: newUser._id, role: newUser.role,
department: newUser.department })
        });
      })
    })
  }
  throw new Error("Користувач з такою електронною поштою вже
зареєстрований");
},
entrantRegistration: async (root, { input }) => {
  const { password, email, ...other } = input;
  const entrantExist = await Entrant.findOne({ email });
  if (!entrantExist) {
    const hashedPassword = await hashPassword(password);
    const newEntrant = new Entrant({
      ...other,
      email,
      password: hashedPassword,
      role: "ENTRANT"
    });

    return new Promise((resolve, reject) => {
      newEntrant.save((err) => {
        if (err) reject(err);
        else resolve({
          entrant: {
            ...other,
            email,
            id: newEntrant._id,
          },
          token: signToken({ userId: newEntrant._id, role:
newEntrant.role })
        });
      })
    })
  }
  throw new Error("Користувач з такою електронною поштою вже
зареєстрований");
},
resetPassword: async (_, { email }) => {
  const resetToken = crypto.randomBytes(32).toString('hex');
  const user = await User.findOneAndUpdate({ email }, {
    resetToken,
    expireToken: Date.now() + 3600000
  })
  if (user) {
    sendEmail("../emails/reset-password", email, {
      resetLink: `${process.env.CLIENT_URL}/reset-
password/${resetToken}`,
    }).then(() => {

```

```

        return {
          success: true
        }
      }).catch(e => {
        return {
          success: false
        }
      })
    }
    return {
      success: !!user
    }
  },
  updatePassword: async (_, { input }) => {
    const { password, token: resetToken } = input;
    const hashedPassword = await hashPassword(password);
    const user = await User.findOneAndUpdate({ resetToken, expireToken: { $gt:
Date.now() } }, {
      password: hashedPassword,
      resetToken: null,
      expireToken: null
    })
    return {
      success: !!user
    }
  },
  addPublication: async (_, { input }, ctx) => {
    const { userId } = extractUserInfo(ctx)
    const { department, faculty } = await User.findOne({ _id: new
mongoose.Types.ObjectId(userId) })
    const newPublication = new Publication({
      ...input,
      department,
      faculty,
      addedByUserId: userId
    });

    return new Promise((resolve, reject) => {
      newPublication.save((err) => {
        if (err) reject(err);
        else resolve({
          id: newPublication._id,
          department,
          faculty,
          ...input
        });
      });
    })
  },
  addDissertation: async (_, { input }, ctx) => {
    const { userId } = extractUserInfo(ctx)
    const { department, faculty } = await User.findOne({ _id: new
mongoose.Types.ObjectId(userId) })
    const newDissertation = new Dissertation({
      ...input,
      department,
      faculty,
      addedByUserId: userId
    });

    return new Promise((resolve, reject) => {

```

```

        newDissertation.save((err) => {
          if (err) reject(err);
          else resolve({
            id: newDissertation._id,
            ...input
          });
        })
      })
    },
    addEvent: async (_, { input }, ctx) => {
      const { userId } = extractUserInfo(ctx)
      const { department, faculty } = await User.findOne({ _id: new
mongoose.Types.ObjectId(userId) })
      const newEvent = new Event({
        ...input,
        department,
        faculty,
        addedByUserId: userId
      });

      return new Promise((resolve, reject) => {
        newEvent.save((err) => {
          if (err) reject(err);
          else resolve({
            id: newEvent._id,
            ...input
          });
        })
      })
    },
    addGrant: async (_, { input }, ctx) => {
      const { userId } = extractUserInfo(ctx)
      const { department, faculty } = await User.findOne({ _id: new
mongoose.Types.ObjectId(userId) })
      const newGrant = new Grant({
        ...input,
        department,
        faculty,
        addedByUserId: userId
      });

      return new Promise((resolve, reject) => {
        newGrant.save((err) => {
          if (err) reject(err);
          else resolve({
            id: newGrant._id,
            ...input
          });
        })
      })
    },
    addProject: async (_, { input }, ctx) => {
      const { userId } = extractUserInfo(ctx)
      const { department, faculty } = await User.findOne({ _id: new
mongoose.Types.ObjectId(userId) })
      const newProject = new Project({
        ...input,
        department,
        faculty,
        addedByUserId: userId
      });
    }
  }
}

```

```

    return new Promise((resolve, reject) => {
      newProject.save((err) => {
        if (err) reject(err);
        else resolve({
          id: newProject._id,
          ...input
        });
      });
    })
  },
  addScientist: async (_, { input }, ctx) => {
    const newScientist = new Scientist(input);
    return new Promise((resolve, reject) => {
      newScientist.save((err) => {
        if (err) reject(err);
        else resolve({
          id: newScientist._id,
          ...input
        });
      });
    })
  },
  addSupervisor: async (_, { input }, ctx) => {
    const newSupervisor = new Supervisor(input);
    return new Promise((resolve, reject) => {
      newSupervisor.save((err) => {
        if (err) reject(err);
        else resolve({
          id: newSupervisor._id,
          ...input
        });
      });
    })
  },
  updatePublication: async (_, { input }) => {
    const { id, publication } = input;
    const updatedPublication = await Publication.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, publication);
    return updatedPublication;
  },
  updateDissertation: async (_, { input }) => {
    const { id, dissertation } = input;
    const updatedDissertation = await Dissertation.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, dissertation);
    return updatedDissertation;
  },
  updateEvent: async (_, { input }) => {
    const { id, event } = input;
    const updatedEvent = await Event.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, event);
    return updatedEvent;
  },
  updateGrant: async (_, { input }) => {
    const { id, grant } = input;
    const updatedGrant = await Grant.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, grant);
    return updatedGrant;
  },
  updateProject: async (_, { input }) => {
    const { id, project } = input;

```

```

        const updatedProject = await Project.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, project);
        return updatedProject;
    },
    updateScientist: async (_, { input }) => {
        const { id, scientist } = input;
        const updatedScientist = await Scientist.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, scientist);
        return updatedScientist;
    },
    updateSupervisor: async (_, { input }) => {
        const { id, supervisor } = input;
        const updateSupervisor = await Supervisor.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, supervisor);
        return updateSupervisor;
    },
    updateInterview: async (_, { input }) => {
        const { id, interview } = input;

        try {
            const updateInterview = await Interview.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, interview, { returnOriginal: true });

            const supervisor = await Supervisor.aggregate([
                {
                    $match: { _id: new
mongoose.Types.ObjectId(updateInterview.supervisorID) }
                }
            ]);

            const entrant = await Entrant.aggregate([
                {
                    $match: { _id: new
mongoose.Types.ObjectId(updateInterview.entrantID) }
                }
            ]);

            const supervisorName = `${supervisor[0].firstName}
${supervisor[0].lastName} ${supervisor[0].middleName}`;
            const entrantName = `${entrant[0].firstName} ${entrant[0].lastName}
${entrant[0].middleName}`;
            const date = dayjs(interview.date).format("DD.MM.YYYY");
            const time = dayjs(interview.date).format("HH:mm");
            const place = interview?.place || updateInterview.place;
            const mailData = {
                supervisorName,
                entrantName,
                date,
                time,
                place
            };

            if (!updateInterview.date && interview.date) {
                sendEmail("../emails/interview/set-date", entrant[0].email,
mailData);
            }

            if (updateInterview.date && interview.date &&
!dayjs(updateInterview.date).isSame(interview.date)) {
                sendEmail("../emails/interview/update-date", entrant[0].email,
mailData);
            }
        }
    }
}

```

```

    }

    if (interview?.status === "Скасовано") {
      sendEmail("../emails/interview/cancel", entrant[0].email,
mailData);
    }

    return {
      success: true
    }
  } catch (err) {
    return {
      success: false
    }
  }
},
deletePublication: async (_, { id }) => {
  const result = await Publication.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
deleteGrant: async (_, { id }) => {
  const result = await Grant.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
deleteProject: async (_, { id }) => {
  const result = await Project.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
deleteScientist: async (_, { id }) => {
  const result = await Scientist.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
deleteSupervisor: async (_, { id }) => {
  const result = await Supervisor.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
deleteDissertation: async (_, { id }) => {
  const result = await Dissertation.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
login: async (_, args) => {
  const { email } = args.input;

```

```

const result = await User.findOne({ email }).lean();
let entrantLogin = null;

if (!result) {
  entrantLogin = await Entrant.findOne({ email }).lean();
}

if (!result && !entrantLogin) {
  throw new Error("Неправильный пароль або email");
}

const isValidPassword = await verifyPassword(result ? result.password :
entrantLogin.password, args.input.password);

if (!isValidPassword) {
  throw new Error("Неправильный пароль або email");
}
const { _id,
  password,
  ...other
} = result ? result : entrantLogin

const token = result ?
  signToken({ userId: _id, role: result.role, department:
result.department }) :
  signToken({ userId: _id, role: entrantLogin.role });

return {
  user: {
    id: _id,
    ...other
  },
  token,
};
},
deleteUser: async (_, args) => {
  const id = args.id;
  const result = await User.findOneAndDelete({ _id: new
mongoose.Types.ObjectId(id) });
  return {
    success: !!result
  }
},
updateUser: async (_, args) => {
  const { id, user } = args.input;
  const updatedUser = await User.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(id) }, user);
  return updatedUser;
},
fullSynchronization: async (_, { input }, ctx) => {
  const { userId } = extractUserInfo(ctx)
  const { email, scopusID, firstName, lastName, middleName, faculty,
department } = await User.findOne({ _id: new mongoose.Types.ObjectId(userId) })
  const dateFrom = input?.from ? dayjs(input.from).format("YYYY-MM-DD") :
dayjs().subtract(10, "year").startOf('year').format("YYYY-MM-DD")
  const dateTo = input?.to ? dayjs(input.to).format("YYYY-MM-DD") :
dayjs().format("YYYY-MM-DD")
  const isScopus = input?.isScopus
  const isCrossref = input?.isCrossref
  const isNRAT = input?.isNRAT
  try {

```



```

const job = await fullSyncQueue.add({
  emailNotification: true,
  isSingleUser: true,
  user: {
    userId,
    firstName,
    lastName,
    middleName,
    email,
    scopusID,
    faculty,
    department
  },
  dateFrom,
  dateTo,
  isScopus,
  isCrossref,
  isNRAT
});

const syncRunId = job.id.toString();

return {
  success: true,
  syncRunId: syncRunId
}
} catch (e) {
  logger.error(`Error adding single sync job for user ${userId}:
${e.message}`);

  return {
    success: false,
    syncRunId: null
  }
}
},
allUsersDataSync: async (_, { input }, ctx) => {
  const dateFrom = input?.from ? dayjs(input.from).format("YYYY-MM-DD") :
dayjs().startOf('year').format("YYYY-MM-DD")
  const dateTo = input?.to ? dayjs(input.to).format("YYYY-MM-DD") :
dayjs().endOf('year').format("YYYY-MM-DD")
  const users = await User.find({ department: input.department });
  const totalUsers = users.length;
  const { isScopus, isCrossref, isNRAT } = input;

  if (totalUsers === 0) {
    return { success: true, syncRunId: null };
  }

  const syncRunId = new mongoose.Types.ObjectId().toString();
  const departmentKey = `:${input.department}`;
  const syncKey = `:${syncRunId}`;

  const startTime = Date.now();

  await redis.hmset(syncKey,
    'status', 'IN_PROGRESS',
    'total', totalUsers,
    'completed', 0,
    'failed', 0,
    'department', input.department,

```

```

        'startedAt', startTime
    );
    await redis.expire(syncKey, 60 * 60 * 24 * 7);
    await redis.set(departmentKey, syncRunId);

    const jobPromises = users.map((user) => {
        const { email, scopusID, firstName, lastName, middleName, faculty,
department } = user;
        return fullSyncQueue.add({
            emailNotification: false,
            syncRunId: syncRunId,
            isSingleUser: false,
            user: {
                userId: user._id,
                firstName,
                lastName,
                middleName,
                email,
                scopusID,
                faculty,
                department
            },
            dateFrom,
            dateTo,
            isScopus,
            isCrossref,
            isNRAT
        });
    });

    await Promise.all(jobPromises);

    return {
        success: true,
        syncRunId: syncRunId
    }
},
updateUserAvatar: async (_, { avatar }, ctx) => {
    const { userId } = extractUserInfo(ctx);
    const updatedUser = await User.findOneAndUpdate({ _id: new
mongoose.Types.ObjectId(userId) }, {
        avatar
    });
    if (!updatedUser) {
        return {
            success: false
        }
    }
    return {
        success: true
    }
},
exportReport: async (_, { input }, ctx) => {
    const { filter } = input
    const { userId } = extractUserInfo(ctx)
    const { faculty, department } = await User.findOne({ _id: new
mongoose.Types.ObjectId(userId) })
    if (!filter.faculty && !filter.department) {
        input.filter.faculty = faculty;
        input.filter.department = department;
    }
}

```

```

        return {
            file: await generateReport(input)
        }
    },
    importEmployees: async () => {
        const promises = [];

        const importStream = fs.createReadStream('./tmp/employees.csv', { encoding:
'' })
            .pipe(csvParser({ separator: ';' }));

        return new Promise((resolve, reject) => {
            importStream
                .on('data', (row) => {
                    const { fullName, ...other } = row;
                    const fullNameSplited = fullName.split(" ");
                    const currentDate = new Date();
                    const department =
findDepartmentByLabel(other.department.replace(/\s+/g, ' ').replace(/-\s+/g, '-
').trim());

                    const employeeData = {
                        ...other,
                        firstName: fullNameSplited[0],
                        lastName: fullNameSplited[1],
                        middleName: fullNameSplited[2],
                        dateOfBirth: convertToISO(row.dateOfBirth),
                        faculty:
findFacultyByDepartmentLabel(other.department.replace(/\s+/g, ' ').replace(/-\s+/g, '-
').trim()),

                        department: department || other.department,
                        importDate: [currentDate]
                    };

                    const promise = (async () => {
                        try {
                            const existingEmployee = await Employee.findOne({
                                externalID: employeeData.externalID,
                                workPlace: employeeData.workPlace,
                                department: employeeData.department
                            });
                            if (existingEmployee) {
                                employeeData.importDate =
[...existingEmployee.importDate, currentDate];
                            } else {
                                employeeData.importDate = [currentDate];
                            }

                            await Employee.findOneAndUpdate(
                                {
                                    externalID: employeeData.externalID,
                                    workPlace: employeeData.workPlace,
                                    department: employeeData.department
                                },
                                employeeData,
                                { new: true, upsert: true }
                            );
                        } catch (error) {
                            throw error;
                        }
                    })();
                })();
        });
    }
}

```

```

        promises.push(promise);
    })
    .on('end', async () => {
        try {
            await Promise.all(promises);
            resolve({ success: true });
        } catch (error) {
            reject({ success: false, error: error.message });
        }
    })
    .on('error', (error) => {
        reject({ success: false, error: error.message });
    });
});
},
importScientists: async () => {
    const promises = [];

    const importStream = fs.createReadStream('./tmp/scientists_dr.csv', {
encoding: '' })
        .pipe(csvParser({ separator: ';' }));

    return new Promise((resolve, reject) => {
        importStream
            .on('data', (row) => {
                const { fullName, speciality, ...other } = row;
                const fullNameSplited = fullName.split(" ");
                const currentDate = new Date();
                const specialitySplited = speciality.split(" ")
                const department =
findDepartmentBySpeciality(specialitySplited[0]);
                const scientistData = {
                    ...other,
                    firstName: fullNameSplited[0],
                    lastName: fullNameSplited[1],
                    middleName: fullNameSplited[2],
                    dateOfBirth: convertToISO(dayjs(row.dateOfBirth,
"M/D/YYYY")),
                    faculty: findFacultyByDepartmentLabel(department),
                    speciality: specialitySplited[0],
                    department: findDepartmentByLabel(department) ||
other.department,
                    importDate: [currentDate],
                    degree: "Доктор наук"
                };

                const promise = (async () => {
                    try {
                        const existingScientist = await Scientist.findOne({
                            firstName: scientistData.firstName,
                            lastName: scientistData.lastName,
                            middleName: scientistData.middleName,
                            dateOfBirth: scientistData.dateOfBirth
                        });
                        if (existingScientist) {
                            scientistData.importDate =
[...existingScientist.importDate, currentDate];
                        } else {
                            scientistData.importDate = [currentDate];
                        }
                    }
                });
            });
    });
}

```

```

        await Scientist.findOneAndUpdate(
            {
                firstName: scientistData.firstName,
                lastName: scientistData.lastName,
                middleName: scientistData.middleName,
                dateOfBirth: scientistData.dateOfBirth
            },
            scientistData,
            { new: true, upsert: true }
        );
    } catch (error) {
        throw error;
    }
}());

    promises.push(promise);
})
.on('end', async () => {
    try {
        await Promise.all(promises);
        resolve({ success: true });
    } catch (error) {
        reject({ success: false, error: error.message });
    }
})
.on('error', (error) => {
    reject({ success: false, error: error.message });
});
});
},
crossrefSearch: async (_, { input }, ctx) => {
    const { userId } = extractUserInfo(ctx)
    const { firstName, lastName, middleName, department, faculty } = await
User.findOne({ _id: new mongoose.Types.ObjectId(userId) })

    try {
        // const parsedResults = await nratParser({ firstName, lastName,
middleName, faculty, department, userId}, "2024-01-01", "2024-12-01")
        const res = await fullSynchronization({ firstName, lastName,
middleName, faculty, department, userId }, "2024-01-01", "2024-12-01", false, false,
true)

        return {
            success: true
        }
    } catch (e) {
        return {
            success: false
        }
    }
},
requestInterview: async (_, { input }) => {
    try {
        const newInterview = new Interview({
            ...input,
            createdAt: new Date(),
            status: "Запит подано"
        });

        const supervisor = await Supervisor.aggregate([
            {

```



```
    },
    update: {
      $set: {
        scientificInterests: teacher.scientificInterests,
        orcid: teacher.orcid,
        wosId: teacher.wosId,
        scopusId: teacher.scopusId,
        scholarId: teacher.scholarId
      }
    },
    upsert: true
  }
}));

try {
  const bulkWriteResult = await ScientificProfile.bulkWrite(bulkOps);
  return {
    success: bulkWriteResult.result.ok === 1
  };
} catch (error) {
  return {
    success: false
  };
}
},
};
```

ДОДАТОК В

СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. Дивак М., Манжула В., Мельник А., Юшко А. Архітектура програмного забезпечення для математичного моделювання на основі аналізу інтервальних даних з використанням хмарних технологій. *Measuring and computing devices in technological processes*. 2024. №1. С. 125–139. <https://doi.org/10.31891/2219-9365-2024-77-15>
2. Дивак М., Юшко А. Математична модель прогнозування показників наукової діяльності науково-педагогічних працівників з використанням інтервального аналізу даних. *Herald of Khmelnytskyi National University. Technical Sciences*. 2025. Т. 353(3.2). С. 375–386. <https://doi.org/10.31891/2307-5732-2025-353-53>
3. 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. <https://doi.org/10.31891/2219-9365-2025-81-45>
4. Юшко А. В., Сімак А. В. Інформаційна інтелектуальна система аналізу наукової та науково-педагогічної діяльності академічного колективу. *Оптико-електронні інформаційно-енергетичні технології*. 2024. Вип. 47(1). С. 7–16. <https://doi.org/10.31649/1681-7893-2024-47-1-7-16>
5. Desyatnyuk O., Dyvak M., Yushko A., Koruts U., Nadvynychnyy S., Ostroverkhov V. The Method of Forecasting a Researcher's Project and Publication Activity Indicators Based on Interval Data Analysis. 2025 15th International Conference on ACIT. 2025. P. 701–707. <https://doi.org/10.1109/acit65614.2025.11185612>
6. Dyvak M., Yushko A., Savenko O. та ін. An intelligent information system for generating a scientist's scientometrics using content analysis methods. *AISTDS'2024, Kyiv, 1 October 2024*. https://ceur-ws.org/Vol-3942/S_06_Dyvak.pdf

7. Dyvak M., Yushko A., Savenko O. та ін. Web-oriented application for student attendance accounting with automatic parsing of class schedules. YAISD'2025, Ternopil, 8 September 2025. <https://ceur-ws.org/Vol-3974/paper17.pdf>
8. Pukas A., Melnyk A., Voytyuk I., Yushko A., Romanyuk M., Honchar L. Transactional Business Application Based on Microservice Architecture. 2021 11th International Conference on ACIT. 2021. P. 564–567. <https://doi.org/10.1109/acit52158.2021.9548385>
9. Pukas A., Simak A., Yushko A., Nadvynychnyy S., Shandruk S., Vatslavskiy O. Automated Reporting Module in the Academic Staff Performance Appraisal System. 2024 14th International Conference on ACIT. 2024. Vol. 2021. P. 824–827. <https://doi.org/10.1109/acit62333.2024.10712573>
10. Shevchuk R., Mykytiuk V., Miramontes D. A., Yushko A., Yurchyshyn T., Mokhun S. Software Service for Studying Words of the Ukrainian Language. 2021 11th International Conference on ACIT. 2021. P. 648–651. <https://doi.org/10.1109/acit52158.2021.9548448>
11. Yushko A., Dyvak M., Koruts U., Taraj M., Hrabar E., Sprinsyan V. The Information-Intellectual System for Evaluating Scientific and Scientific-Pedagogical Activities of the Academic Community. 2024 14th International Conference on ACIT. 2024. Vol. 2136. P. 828–833. <https://doi.org/10.1109/acit62333.2024.10712561>
12. Yushko A., Dyvak M., Melnyk A., Trufanova Y., Kobytzia V., Martsenyuk Y. AI-Based Vector Filtering of Irrelevant Academic Publications Using Research Interest Modeling. 2025 15th International Conference on Advanced Computer Information Technologies (ACIT). 2025. P. 779–784. <https://doi.org/10.1109/acit65614.2025.11185809>
13. Yushko A., Shevchuk R., Łopaciński K., Leszczynska M., Yashchuk O., Yurchyshyn T. Shielding Web Application against Cyber-Attacks using SIEM. 2023 13th International Conference on ACIT. 2023. P. 393–396. <https://doi.org/10.1109/acit58437.2023.10275630>

ДОДАТОК Г
АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОГО
ДОСЛІДЖЕННЯ



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/ 98-2025

« 27 11 2025 року

ДОВІДКА
про участь у виконанні науково-дослідних робіт

Видана **ЮШКУ Андрію Васильовичу** про участь у підготовці НДР, які виконував Західноукраїнський національний університет. Зокрема, у 2020-2021 роках на посадах молодшого наукового співробітника він брав участь у виконанні держбюджетного дослідження «Математичне та програмне забезпечення для ідентифікації та моніторингу особливо небезпечних джерел забруднення ґрунту та ґрунтових вод» (держреєстраційний номер 0120U102040), у 2021 році – госпдоговірної роботи «Оптимізація веб-порталу “Газпостач”» (0121U110387); на посадах наукового співробітника у 2023 році – роботи з надання науково-професійних послуг «Комплексна розробка та SEO-налаштування публічного сайту ПрАТ “Тернопільміськгаз” з модулями особистого кабінету та подачі показників побутовими споживачами», у 2024 році – держбюджетного дослідження «Математичне та комп’ютерне моделювання об’єктів з розподіленими параметрами на основі поєднання онтологічного та інтервального аналізу» (0122U001497), роботи з надання науково-професійних послуг «Послуги з розробки модуля відслідковування заявок користувачів на розробку технічної документації».

Начальник
 науково-дослідної частини



Віта СЕМАНЮК



**ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
«КОМПАНІЯ «ГАЗПОСТАЧ»**

46008, м.Тернопіль, вул. Г.Коллонтая, буд.2, приміщення А, Код ЄДРПОУ 43747741
тел.093 169 02 96 www.gazpostach.com.ua e-mail: admin@gazpostach.com.ua

«17» 09.25. № 9/25

АКТ
про впровадження результатів дисертаційної роботи
Юшка Андрія Васильовича
«Математичне та програмне забезпечення для планування наукової
діяльності науково-педагогічних працівників»

Даний акт складений про те, що при реалізації програмних проєктів у ТОВ «КОМПАНІЯ «ГАЗПОСТАЧ» використано результати дисертаційної роботи аспіранта кафедри комп'ютерних наук Західноукраїнського національного університету Юшка Андрія Васильовича.

Автором запропоновано:

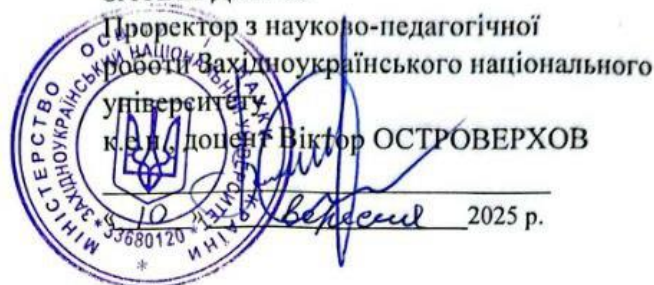
- підхід до інтеграції багатоджерельних даних, який включає автоматизоване отримання, верифікацію та нормалізацію інформації з різних платформ (міжнародних реєстрів, репозитаріїв, API та слабоструктурованих джерел), що уможливило використання єдиних узгоджених даних у внутрішніх корпоративних проєктах компанії;
- архітектуру інтелектуалізованої програмної системи, яка передбачає модульність, можливість масштабування та інтеграцію з існуючими інформаційними сервісами компанії, що забезпечило впровадження сучасних технологій аналізу даних у практичну діяльність підприємства.

Директор



Ірина КИРДЕЛЬ

ЗАТВЕРДЖУЮ



АКТ

про впровадження в освітній процес Західноукраїнського національного університету результатів дисертаційної роботи
ЮШКА Андрія Васильовича
«Математичне та програмне забезпечення для планування наукової діяльності науково-педагогічних працівників»

Даний акт складений про те, що результати дисертаційної роботи ЮШКА Андрія Васильовича на тему: «Математичне та програмне забезпечення для планування наукової діяльності науково-педагогічних працівників» використана в освітньому процесі факультету комп'ютерних інформаційних технологій Західноукраїнського національного університету для студентів спеціальностей «Інженерія програмного забезпечення» та «Інформаційні системи та технології».

При викладанні дисципліни «Інтелектуальні інформаційні системи» розглядаються розроблені автором інформаційна технологія інтегрованого збору наукових даних науково-педагогічних працівників, а також математичні методи та моделі для прогнозування показників наукового навантаження. Особливу увагу приділено застосуванню великих мовних моделей та методів семантичного ранжування для визначення релевантності наукових публікацій і формування індивідуальних профілів дослідників.

Запропоновані автором наукові результати реалізовано в програмній системі, що використовується для планування показників наукової діяльності науково-педагогічних працівників ЗУНУ.

Декан факультету комп'ютерних інформаційних технологій, к.т.н., доцент

Ігор ЯКИМЕНКО

Завідувач кафедри комп'ютерних наук, д.т.н., професор

Андрій ПУКАС