MJI
Media Jurnal Informatika
Vol.17, No.2, December 2025
p-issn : 2088-2114, e-issn : 2477-2542

Development of a Scientific Article Recommendation Web Application Using a Hybrid Recommender System: A Case Study in Computer Science

Alfien Sukma Prawira[a,1], Adinda Raisa Az-zahra [a,2], Mahesya Setia Nugraha [a,3], Ade Hodijah [a,4,*], Urip Teguh Setijohatmo [a,5], Siti Dwi Setiarini [a,6], Bambang Wisnuadhi [a,7]

[a]Department of Computer and Informatics Engineering, Politeknik Negeri Bandung
[1] *alfien.sukma.tif22@polban.ac.id*; [2] *adinda.raisa.tif22@polban.ac.id*; [3] *mahesya.setia.tif22@polban.ac.id*; [4]* *adehodijah@jtk.polban.ac.id,*
[5] *urip@jtk.polban.ac.id,*[6] *siti.dwi@polban.ac.id,* [7] *bwisnuadhi@polban.ac.id*
* corresponding author

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
hybrid recommender system
scientificarticle
graph data science
graph database
waterfall methodology

ABSTRACT

The rapid increase in scientific publications has created significant challenges for researchers in finding relevant literature. Conventional citation-based recommender applications often have drawbacks, such as bias toward popular articles and vulnerability to manipulation through citation cartels, which reduce objectivity. To address these limitations, this development aimed to design and develop a web-based scientific article recommendation application using a hybrid recommender system approach. The development followed the waterfall methodology, covering requirements analysis, design, implementation, and testing stages. The hybrid approach combines Content-based filtering by analyzing content similarity and Collaborative filtering based on user interaction history. Scientific articles and user preferences were modeled in a graph database to map relationships, with the implementation of Graph Data Science Library using algorithms named K-Nearest Neighbor, Degree centrality, and PageRank. Based on 101 black-box unit test cases, the application successfully delivered three main recommendation features by integrating content analysis—based on access history and currently viewed articles—with user preference modeling through peer institutions. The testing results confirm that all recommendation functions operated as intended across various user scenarios. Overall, the developed application provides multiple recommendation features that enhance objectivity and relevance, supporting researchers, students, and practitioners in discovering scientific references more effectively.

1. Introduction

The digital era has witnessed a rapid growth in online scientific publications. According to the Scimago Journal & Country Rank, more than 10 million documents were indexed in Scopus between 2022 and 2023, providing unprecedented access to scholarly references. However, this rapid expansion has also created significant challenges for researchers in identifying genuinely relevant articles using conventional keyword-based search methods [1][2].

To address this issue, scientific article recommender systems have been widely adopted. Traditional citation-based approaches, including citation network analysis, co-citation, and bibliographic coupling, are effective in modeling scholarly influence but often suffer from limitations such as bias toward highly cited articles (the Matthew effect), delayed recognition of recent research, and susceptibility to citation manipulation practices such as citation cartels [3][4][5][6]. These shortcomings highlight the need for more objective and adaptive recommendation strategies.

Recent studies have explored hybrid recommender systems that combine Content-Based Filtering (CBF) and Collaborative Filtering (CF) to overcome the individual limitations of each method [7][8][9][10][11]. While these approaches have shown promise, many existing systems treat content similarity and user interaction patterns as separate components, without fully exploiting the semantic relationships between articles, users, and institutions.

A mixed strategy is used in this development, which presents results from various methods simultaneously [12]. In this configuration, CBF addresses the new item problem inherent in CF, while CF, in turn, enhances the diversity of the recommendations. To manage and connect data between entities such as articles, authors, institutions, and user interactions, a graph-based database using Neo4j is used. This data representation enables the system to recognize relationships and preferences more efficiently[13],thus providing more contextual recommendations.

For the CBF implementation, content analysis is performed on titles, abstracts, and citation networks. The content is represented using vector embeddings from the SPECTER model, which is specifically designed to capture the semantic relationships between scientific articles [14]. In addition, Sentence Transformers were also used to process keyword-based searches, by transforming keywords and content into vector representations for similarity calculation[15]. Similarity is calculated using cosine similarity, a metric within the K-Nearest Neighbors algorithm in Neo4j's Graph Data Science. Cosine similarity measures the directional similarity between vectors, not the magnitude, making it effective for comparing articles semantically[16].

In addition, CBF also utilizes the PageRank algorithm to measure the importance of scientific articles based on citation networks [17]. This approach considers not only the volume of citations but also the quality of the citing sources. Consequently, articles with significant influence in the knowledge network receive higher scores in the recommendation system [18][17].

The CF component, on the other hand, is implemented through an analysis of user interactions using Degree Centrality. This method measures the popularity of an article based on how many users interact with it [19]. Articles that are frequently read are assumed to hold high value within the community and are therefore recommended to other users.

This study addresses this gap by proposing a hybrid scientific article recommendation system that integrates content-based and collaborative filtering within a unified graph-based data model. Scientific articles, user interactions, and institutional relationships are represented in a Neo4j graph database, enabling the system to capture contextual and semantic relationships more effectively. The recommendation process leverages Graph Data Science algorithms, including K-Nearest Neighbor with cosine similarity for semantic content matching, PageRank for citation-based importance estimation, and Degree Centrality for modeling collective user interest. Unlike existing hybrid recommender systems that primarily rely on citation counts or isolated similarity measures, the proposed approach explicitly combines semantic embeddings derived from scientific text with graph-based structural analysis to generate multiple recommendation strategies. This design allows recommendations to be ordered and interpreted based on both semantic relevance and relational context. The objective of this research is to develop a scientific article recommendation system that enhances relevance, diversity, and objectivity by integrating content analysis and user preferences within a graph-based hybrid framework. The proposed system is expected to assist students and researchers, particularly in the field of computer science, in discovering high-quality and contextually relevant scientific literature amid the growing volume of academic publications.

2. Method

The method used in this research is the Waterfall software development model, a classic approach that is sequential and systematic. This model was selected due to the project's well-defined and stable requirements from the outset, coupled with time and resource constraints that necessitated a linear and structured approach[20]. The Waterfall method consists of four main stages, namely Requirement Definition, System and Software Design, Implementation, and Integration and Testing [21].

2.1 Type and Approach of Research

This study is classified as software engineering research, utilizing a Research and Development (R&D) approach. The objective is to design and build a web-based scientific article recommender system and subsequently evaluate its effectiveness and accuracy. The R&D approach was selected based on the need to systematically develop a digital product, proceeding from requirements analysis through to functional testing.

2.2 Object and Scope of Research

The object of this research is a scientific article recommendation system web application. This application is designed to provide article recommendations based on various aspects, such as user access history, institutional links, and articles being read. The study falls within the domain of education and information technology, with a scope encompassing user interface (UI) design, the utilization of a graph database, and the implementation of recommendation algorithms.

2.3 Data Collection Techniques

Data collection in this study was conducted using two primary techniques: a literature review and dataset construction from external scholarly sources. The literature review aimed to establish a theoretical foundation by examining core concepts related to recommender systems, including vector embeddings, similarity measurement, graph centrality, and hybrid recommender system architectures. In parallel, a comparative analysis of the Connected Papers application was performed to identify its core functionalities, operational workflow, and limitations, which informed the formulation of both functional and non-functional system requirements.

The dataset used for system development consists of scientific articles sourced exclusively from Semantic Scholar, a large-scale academic literature provider. The scope of the dataset focuses on publications in the computer science domain, covering approximately 6.1 million articles. Due to limitations in automated access, data updates were performed manually by retrieving article metadata directly from the Semantic Scholar website. At the time of development, fully automated data harvesting was not implemented.

To support structured data acquisition, keyword-based retrieval was guided by the Computer Science Ontology (CSO). The ontology provides a hierarchical representation of computer science topics, with 20 direct child concepts under the “Computer Science” node and a total of 10,362 descendant concepts. These ontology-derived keywords were used as search queries to identify relevant scientific articles within Semantic Scholar.

Data collection involved a series of preprocessing steps to prepare the dataset for graph-based analysis. These steps consisted of four main stages: (1) graph generation, in which articles and their relationships were modeled as nodes and edges in a Neo4j graph database; (2) embedding generation, where textual content from article titles and abstracts was transformed into vector representations; (3) similarity relationship construction, where semantic similarity between articles was calculated and encoded as graph relationships; and (4) PageRank score computation, which assessed the relative importance of articles within the citation network.

Unlike the official Semantic Scholar API, which imposes strict rate limits and authentication constraints on several endpoints, this study utilized the Semantic Scholar Python library, which modifies API interaction mechanisms to facilitate large-scale data retrieval. This library enables automated searching and metadata extraction based on supported parameters, allowing article data to be processed and stored efficiently in the Neo4j database. The same data source is also used by Connected Papers through API-based integration, ensuring consistency with existing scholarly recommendation platforms.

2.4 Tools and Materials Used

The following key tools and technologies were used in the system's development:

- Programming Languages: Python (backend), HTML/CSS/JavaScript (frontend).
- Framework: Django (based on the Model-View-Template pattern).

- Database: Neo4j, with the Graph Data Science (GDS) library to support graph algorithms.
- Design Diagrams: Use Case Diagrams, Activity Diagrams.
- Architecture: Client-Server.

The development and testing processes were also supported by additional software tools, including Visual Studio Code and Neo4j Desktop.

2.5 Research Procedures or Stages

This research proceeded through the following stages of the Waterfall model:

- Requirement Definition: Requirements were identified through a literature review and an analysis of similar applications. The findings were documented in Use Case Diagrams, which served as the foundation for the development phase.
- System and Software Design: This stage involved designing the User Interface (UI), the graph database schema, and the system's operational workflows. These designs were visualized using Activity Diagrams.
- Implementation: Code was implemented in a modular fashion, using the Django framework for the backend and HTML/CSS/JS for the frontend. The Neo4j graph database was utilized to implement the algorithm-based recommendation features.
- Integration and Testing: All components were integrated and subsequently tested using a black-box testing approach. The testing process involved various input scenarios to evaluate the system's reliability and robustness.

2.6 Data Analysis Techniques

Data analysis in this study was conducted through unit testing using the black-box testing approach, with the primary objective of verifying that all recommendation features operate in accordance with the specified functional requirements. The evaluation focused on the logical correctness, consistency, and stability of the recommendation outputs across different user interaction scenarios, rather than on predictive accuracy metrics.

The proposed system adopts a mixed hybrid recommender approach, in which Content-Based Filtering (CBF) and Collaborative Filtering (CF) are applied in separate recommendation features rather than combined into a single scoring function. Each recommendation feature is designed to address a specific recommendation context and utilizes different analytical mechanisms accordingly.

First, scientific paper recommendations based on access history and recommendations based on the currently viewed paper are generated using a content-based approach. In these features, semantic similarity between articles is calculated using vector embeddings derived from titles and abstracts, with cosine similarity applied within the K-Nearest Neighbor algorithm. Degree Centrality is then used to prioritize articles that are frequently connected within the content similarity graph, ensuring that recommended papers are both semantically relevant and structurally significant.

Second, scientific paper recommendations based on peer institutions are generated using a collaborative filtering perspective. In this feature, the PageRank algorithm is applied to the institutional interaction graph to identify influential articles based on citation and institutional relationships. Unlike the content-based features, PageRank is used independently and is not combined with similarity or Degree Centrality scores.

The final recommendation lists are produced independently for each feature, without aggregating similarity, Degree Centrality, and PageRank scores into a unified numerical value. System analysis confirms that this mixed hybrid design enables flexible and context-aware recommendations, while black-box unit testing verifies that each recommendation feature functions correctly and produces logically ordered results.

3. Results and Discussion

The web application for scientific article recommendation was developed following the Waterfall model, which consists of four primary stages: Requirement Definition, System and Software Design, Implementation, and Integration and Testing. Each stage yielded distinct, interconnected outputs that contributed to the formation of the complete system.

3.1 Presentation of Research Results

- Requirement Definition

  At this stage, system requirements are represented in the form of a Use Case Diagram to illustrate the interactions between users and the system (Fig. 1).
- System and Software Design

  Based on the defined requirements, a system design process is carried out which includes the preparation of a Graph Data Model for database design (Fig. 2). The process design is described using Activity Diagrams, including the process of searching for scientific articles based on keywords or titles (Fig. 3), the recommendation process for scientific articles based on peer institutions (Fig. 4), the process of recommending scientific articles based on scientific articles being read (Fig. 5), and the recommendation process for scientific articles based on access history (Fig. 6). Additionally, the interface design for the main pages of the application was designed, including the search page for scientific articles with recommendations based on access history and peer institutions (Fig. 7), the recommendation page for scientific articles based on peer institutions (Fig. 8), the recommendation page for scientific articles based on access history (Fig. 9), the search results page for scientific articles (Fig. 10), and the recommendation page for scientific articles based on the currently read article (Fig. 11).
- Implementation

  The implementation was carried out comprehensively using a full-stack approach with the Django framework. Table 3, Table 4, Table 5, and Table 6 provide examples of the implementation of the designed system, showcasing screenshots of the implemented interface and the results of database validation.

- Integration and Testing

  Following the implementation phase, the next step involved system integration. Upon completion of integration, comprehensive functional testing of the application was conducted. A total of 101 test cases were executed, with each test case evaluated using the Unit Testing method based on a black-box approach (Fig. 12). Table 1 shows summary of the test cases and Table 2 shows examples of particular test performed on the application.

Table 1. Summary of Unit Testing Results Using the Black-Box Approach

User Function	Number of Test Cases
Account registration	10
Email verification for the registration process	3
Email verification for the password change process	3
Account login	8
Resend email verification code during registration	1
Resend email verification code during password change	1
Update personal data	12
Store access history	2
Scientific paper recommendations based on access history	5
Ordering of scientific paper recommendations based on access history	2
Scientific paper recommendations based on peer institutions	7
Ordering of scientific paper recommendations based on access history	3
Scientific paper search	3
Ordering of scientific paper search results	1
Filtering scientific paper search results	3
Access to scientific paper search	2
Access to scientific paper details	2
Scientific paper search results	1
Details of a scientific paper	1

Scientific paper recommendations based on the currently viewed paper	1
Ordering of scientific paper recommendations based on the currently viewed paper	1
Visiting the scientific paper provider platform	4
Save scientific paper	4
Saved scientific paper list	2
Filter saved scientific paper list	4
Manage scientific papers	15

Table 2. Test Case Example of User Function of Ordering of scientific paper recommendations based on access history

Test Case Name	Steps to execute	Test Data	Expected Result	Result	Test Case Result
Validation of recommendation ordering based on similarity scores from access history	1. Ensure that the scientific paper search page is displayed 2. Ensure that the recommendation list is displayed	**p.title**: "*An Overview of Augmented Reality*"	Each displayed scientific paper recommendation for each access history is selected based on the highest similarity score	For each access history, the displayed scientific paper recommendation is selected based on the highest similarity score, considering only papers that have a *HIGHEST_SIMILARITY* relationship with the sample paper	Pass

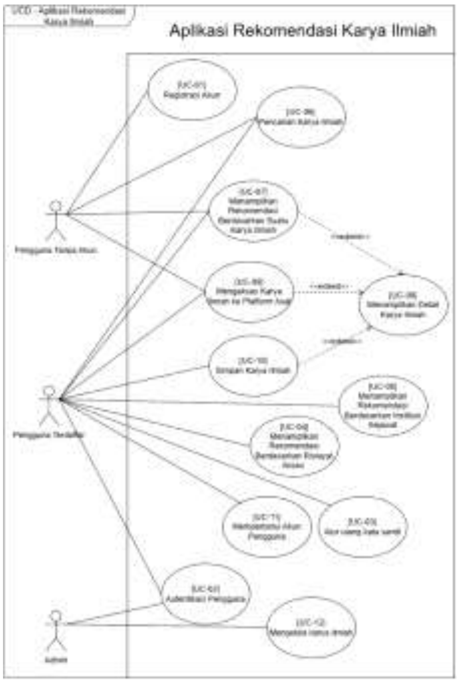

Fig. 1. Use Case Diagram of the Scientific Article Recommendation System

Fig. 1. Use case diagram illustrating the main interactions between users and the scientific article recommendation system, including article search, access history tracking, multiple recommendation features, and article management functionalities. This diagram represents the functional scope of the system and serves as the basis for defining system requirements and unit testing scenarios.

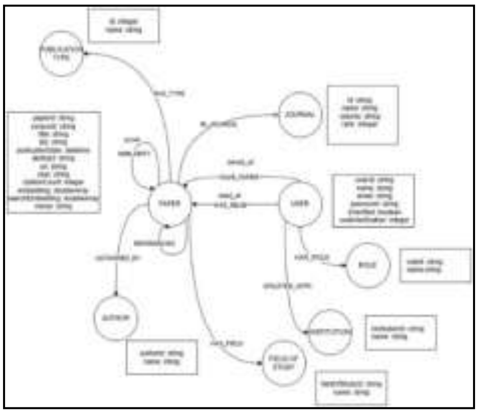

Fig. 2. Graph Data Model for Scientific Article Recommendation

Fig. 2. Graph data model representing the relationships between scientific articles, users, institutions, and interaction history within the recommendation system. This model illustrates how content similarity, access history, and institutional relationships are structured in the Neo4j graph database to support content-based and collaborative recommendation features.

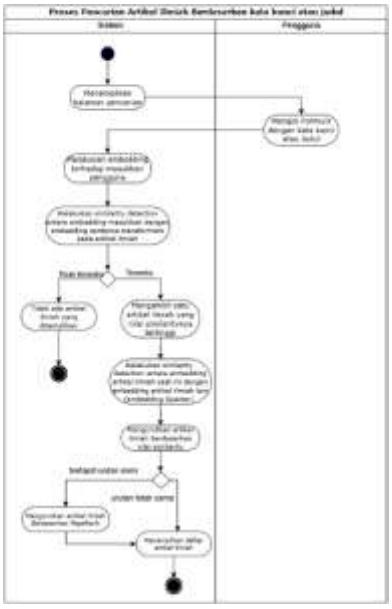

Fig. 3. Activity diagram of the process of searching for scientific articles based on keywords or titles

Fig. 3. Activity diagram illustrating the workflow of searching scientific articles based on user-input keywords or article titles, from query submission and content matching to the presentation of ranked search results. This diagram explains the logical flow of the search feature used in content-based recommendations.

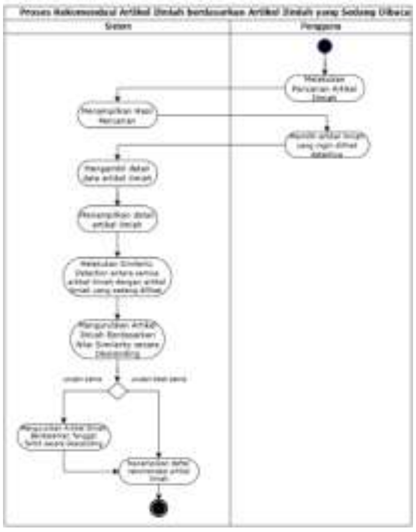

Fig. 5. Activity Diagram of the process of recommending scientific articles based on scientific articles being read

Fig. 5. Activity diagram showing the recommendation workflow triggered when a user views a specific scientific article. The system computes content similarity between the viewed article and candidate articles to generate contextually relevant recommendations.

Fig. 4. Activity diagram of the recommendation process for scientific articles based on peer institutions

Fig. 4. Activity diagram depicting the recommendation process based on peer institutions, where institutional relationships and interaction data are analyzed using PageRank to identify influential articles. This process supports collaborative filtering by leveraging institutional-level user preferences.

Fig. 6. Interface design for recommendation based on access history and peer institutions

Fig. 6. User interface design illustrating the presentation of recommendation results based on user access history and peer institutions. This interface demonstrates how multiple recommendation features are delivered to users within a unified interaction environment.

Fig. 8. Interface design for recommendation based on access history

Fig. 8. User interface design showing how access history–based recommendations are presented to users, including the ordering of recommended articles according to similarity relevance.

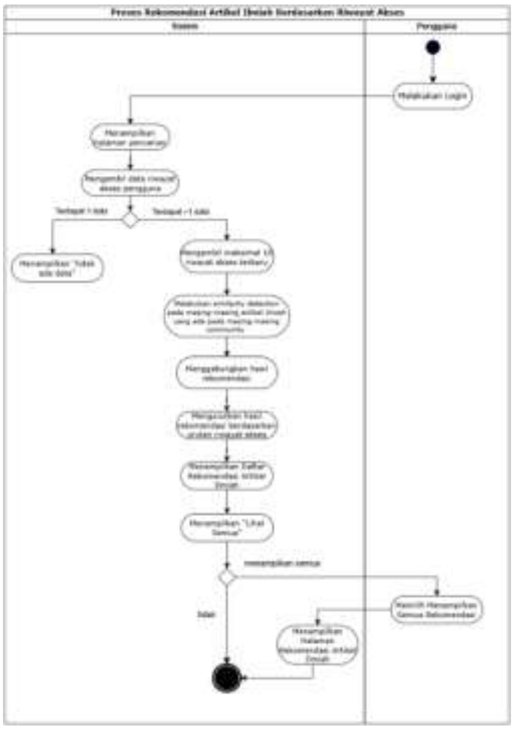

Fig. 7. *Activity Diagram* of the recommendation process based on access history

Fig. 7. Activity diagram describing the recommendation process based on user access history, where previously accessed articles are analyzed to generate personalized content-based recommendations using similarity and graph-based prioritization.

Fig. 9. Interface design for recommendation based on peer institutions

Fig. 9. User interface design illustrating recommendations generated from peer institutional relationships, highlighting articles identified as influential through graph-based PageRank analysis.

Fig. 10. Interface design for recommendation based on articles being read

Fig. 10. User interface design presenting recommendations related to the article currently being read by the user, enabling seamless exploration of semantically similar scientific literature.

Fig. 11. Interface design for searching article feature

Fig. 11. User interface design of the scientific article search feature, demonstrating how users input keywords or titles and receive ranked search results to support efficient literature discovery.

Table 3. Implementation of the Process of Displaying Recommendations Based on Scientific Articles Being Read.

<table>
<tr><td>Process name</td><td>The Process of Displaying Recommendations Based on Scientific Articles Being Read.</td></tr>
<tr><td colspan="2">Interface</td></tr>
<tr><td colspan="2">
</td></tr>
<tr><td colspan="2">Database checking</td></tr>
<tr><td colspan="2"></td></tr>
</table>

Table 4. Implementation of the Process of Displaying Recommendations Based on Scientific Article Search Results

<table>
<tr><td>Process name</td><td>The Process of Displaying Recommendations Based on Scientific Article Search Results</td></tr>
<tr><td colspan="2">Interface</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">Database checking</td></tr>
<tr><td colspan="2"></td></tr>
</table>

Table 5. Implementation of the Process of Displaying Scientific Article Recommendations Based on Access History

Process name	The Process of Displaying Scientific Article Recommendations Based on Access History
Interface	
	
Database checking	
	

Table 6. Implementation of the Process of Displaying Scientific Article Recommendations Based on Peer Institutions

Process name	The Process of Displaying Scientific Article Recommendations Based on Peer Institutions
Interface	
	
Database checking	
	

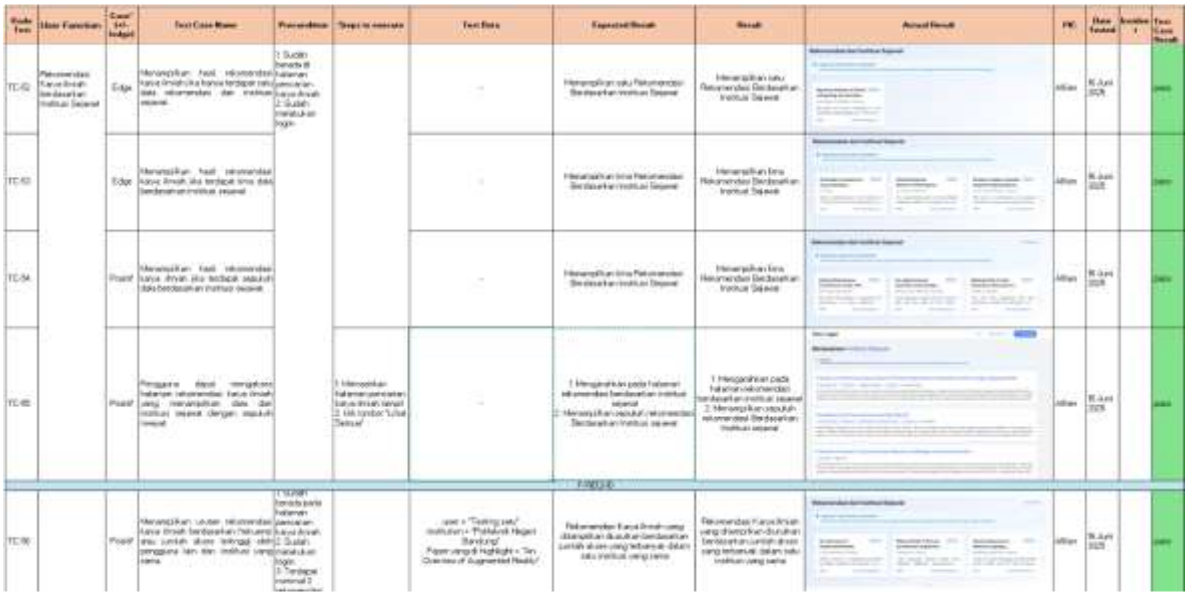

Fig. 12. Testing with unit testing method and black box approach

Fig. 12. Illustration of unit testing conducted using the black-box testing approach, showing the validation of system functionalities against predefined test cases. This testing process focuses on verifying the correctness of inputs, outputs, and expected behaviors of the recommendation features without examining the internal implementation details.

3.2 Analysis of Findings

The analysis results demonstrate that the hybrid recommendation system, integrating Content-Based Filtering (CBF) and Collaborative Filtering (CF), delivers more relevant and personalized outcomes compared to single-method approaches.

- CBF utilizes titles, abstracts, and citation networks to detect semantic similarities between articles. The application of vector embeddings using SPECTER and Sentence Transformers, combined with cosine similarity, supports accurate content matching.
- CF uses user access history, which is analyzed using Degree Centrality in a graph database to generate recommendations based on other users' behavior patterns
- PageRank is used within citation networks to determine the influence weight of articles in the recommendation ranking.

This approach successfully addresses users' needs in discovering relevant scientific references based on both content and community behavior. Although the Results section primarily presents system workflows and interface visualizations, these outputs provide insights into the functional performance of the proposed hybrid recommender system. Based on the unit testing results using a black-box approach, all core functionalities—article search, content-based recommendations, and peer-institution-based recommendations—operated as intended, indicating functional correctness and system stability under predefined test scenarios.

From a recommendation perspective, the system demonstrates a clear distinction from conventional citation-based recommender systems such as Connected Papers. Instead of relying solely on citation networks, the proposed system integrates semantic similarity through content-based filtering and structural importance through graph-based metrics. Degree centrality is employed to recommend articles closely related to user access history and currently viewed papers, while PageRank is specifically utilized to identify influential articles within peer institution networks. This separation of recommendation logic allows each algorithm to be applied in a context where it is most meaningful, rather than combining multiple scores into a single hybrid value.

Compared with related works that typically merge collaborative filtering and content-based filtering into a unified scoring mechanism, the mixed hybrid approach in this study emphasizes feature-level hybridization rather than score-level fusion. This design choice enhances interpretability, as users can clearly understand the rationale behind each recommendation feature. However, this approach also introduces limitations in quantitatively comparing recommendation quality across features, as no unified ranking metric is currently defined.

3.3 Implications of the Results

The development of the hybrid-based scientific article recommendation system demonstrates that the application can provide article recommendations aligned with user preferences. This feature is expected to help students, researchers, and practitioners find relevant references more quickly and efficiently.

The hybrid approach, combining Content-Based Filtering (CBF) and Collaborative Filtering (CF), s proven to provide advantages over using only one method. The CBF approach utilizes the analysis of titles, abstracts, and citation networks to detect similarities between articles, enabling the system to recommend relevant works even for articles with few citations. Meanwhile, CF approach utilizes user interaction data, such as article reading history. This information is processed using Degree Centrality in a graph database to identify articles that are widely accessed and considered interesting by the user community. The combination of these approaches results in more adaptive and personalized recommendations. Figure 13 below illustrates the implementation results of the CF approach.

The application of vector embedding models, such as SPECTER and Sentence Transformers, facilitates the measurement of semantic similarities between article contents. The use of cosine similarity as a similarity metric consistently supports the identification of articles with content closely matching user preferences. Figures 14 and 15 below present the implementation results of the CBF approach.

Additionally, the implementation of PageRank on citation networks adds value to the recommendation ranking process. By utilizing citation information, the system not only considers the frequency of citations but also evaluates the influence of citing articles. This ensures the display of highly relevant works. Figure 16 below illustrates the implementation results of the CBF approach utilizing vector embeddings and PageRank scores.

In the future, this recommendation system has potential for further development to provide greater benefits. One potential enhancement is the addition of a topic or disciplinary classification feature for scientific articles. By classifying articles into specific categories, such as machine learning or computer vision, the system can filter and group works based on these categories, resulting in more specific and tailored recommendations. This classification could also enhance recommendation accuracy and enrich the context of the reference search process.

Lastly, the current system is limited to scientific articles in the field of computer science. To reach a broader user base from diverse academic backgrounds, system development could expand to include other disciplines, such as engineering, medicine, economics, or social sciences. This expansion would strengthen the system's role as an inclusive, cross-disciplinary scientific recommendation platform adaptable to the academic needs of various communities.

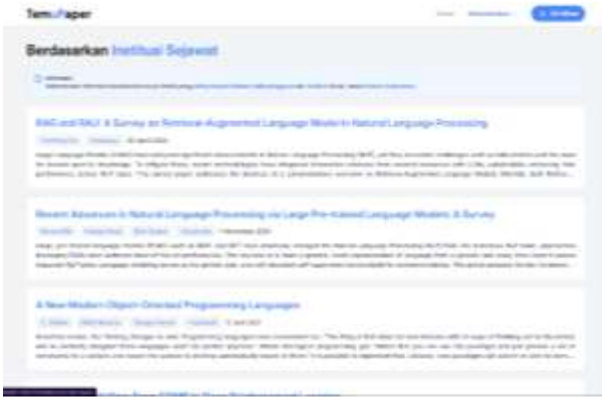

Fig. 13. Scientific articles recommendation based on peer institutions page

Fig. 14. Scientific articles recommendation based on article being read page

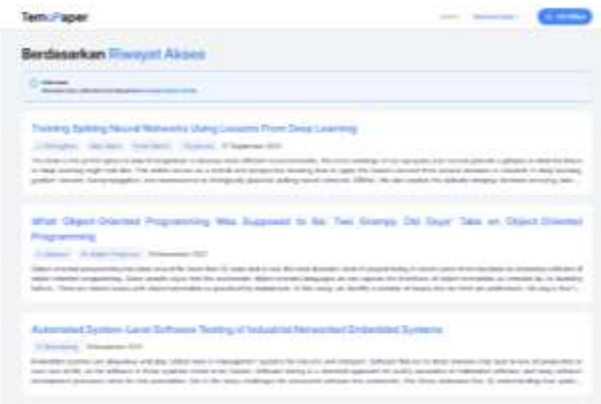

Fig. 15. Scientific articles recommendation based on access history page

Fig. 15. User interface displaying scientific article recommendations generated from the user's access history. The recommendations are ordered based on semantic similarity, highlighting articles that are most relevant to the user's previously accessed content.

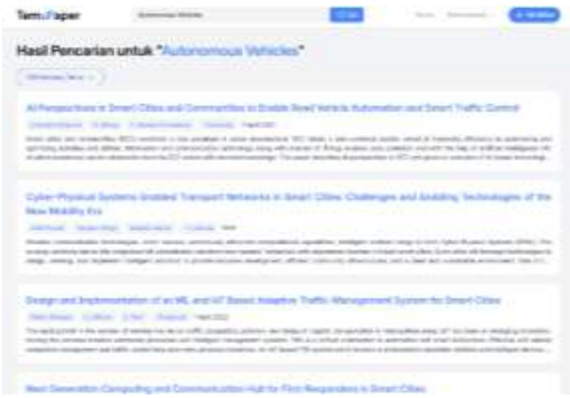

Fig. 16. Scientific article search page

Fig. 16. User interface of the scientific article search feature, where users can search for articles using keywords or titles and obtain ranked search results to support efficient literature exploration.

3.4 Limitations of the Study

The development and implementation of this recommendation application are subject to several limitations that should be acknowledged, which also indicate directions for future research.

First, the process of collecting scientific article data was performed manually and limited in scope, resulting in a dataset that does not fully represent the breadth of available scientific literature. This limitation may impact the quality and diversity of the recommendations generated by the application. Future work should consider automated data acquisition techniques, such as web scraping or API-based harvesting from large scholarly databases, to construct a more comprehensive and representative dataset.

Second, the scope of the application in this study is confined to scientific articles within the field of computer science. Consequently, the system is not yet capable of providing cross-disciplinary recommendations that may be needed by users from other academic domains, such as medicine, social sciences, or engineering. Future research could extend the ontology and knowledge graph to support multi-domain or interdisciplinary recommendations, thereby improving the system's generalizability.

Third, although the system integrates content-based and graph-based approaches, it does not yet accommodate real-time user interaction aspects, such as explicit feedback or user click behavior. The absence of these interactive components limits the system's ability to adapt dynamically to evolving user preferences. Future studies may integrate user interaction data and feedback mechanisms to enhance personalization and recommendation accuracy.

Finally, constraints in resources—including development time, computational capacity, and supporting data—restricted the scope of exploration and comprehensive evaluation of the system. As a result, the findings of this study remain preliminary. Future work should involve large-scale experiments, performance benchmarking, and user-based evaluations to validate the effectiveness and practicality of the proposed approach.

4. Conclusion

This study developed a web-based scientific article recommendation application to address the challenges posed by the rapid growth of scientific publications and the limitations of conventional citation-based recommender systems. By reducing reliance on citation counts alone, the proposed system aims to mitigate bias toward popular articles and improve the objectivity of the recommendation process.

The main scientific contribution of this work lies in the implementation of a hybrid recommender system that integrates content-based filtering and collaborative filtering within a graph-based data model. Content similarity is derived from the analysis of article titles and abstracts, while user preferences are modeled through interaction history and institutional relationships. These components are represented in a graph database, enabling structured relationship mapping and advanced analysis using the Graph Data Science Library.

Unlike existing systems that treat similarity and user behavior independently, this application leverages graph-based algorithms—including K-Nearest Neighbor, Degree Centrality, and PageRank—to generate multiple recommendation strategies that are context-aware and interpretable. The system provides several recommendation features, such as search-based recommendations, recommendations related to the currently viewed article, access history–based recommendations, and peer institution–based recommendations, allowing users to select recommendations according to their specific information needs.

Based on the unit testing results using a black-box approach, the results demonstrate that combining content-based and collaborative filtering approaches within a graph database framework is a promising direction for developing more effective and objective scientific article recommendation systems. This hybrid and graph-driven implementation contributes to the advancement of recommendation systems by enhancing relevance, transparency, and flexibility in academic literature discovery.

Overall, the current implementation establishes a foundational architecture that demonstrates how graph-based technologies and semantic modeling can be integrated into scientific article recommendation systems. Future work should focus on benchmarking the system against baseline methods and incorporating quantitative performance metrics to strengthen empirical validation.

Acknowledgment

The authors express their deepest gratitude to Politeknik Negeri Bandung for the funding support provided, which enabled the successful execution of this research and the writing of this scientific paper. Sincere appreciation is also extended to the academic community of the Department of Computer and Informatics Engineering, particularly the lecturers who have provided guidance, feedback, and motivation throughout the process of preparing this paper. All forms of assistance, whether in the form of knowledge, direction, or encouragement, have been invaluable to the authors in completing this work.

Declarations

This paper involves contributions from multiple authors. Mahesya, Adinda, and Alfien contributed as students who completed this work as part of their final assignment. Ade and Urip acted as supervisors, while Rini and Bambang served as evaluators during the assessment process.This research was funded by PoliteknikNegeri Bandung (POLBAN) under the Director's Decree Number 317/PL1/HK.02/2025. The authors declare no conflict of interest. No additional information is available for this publication.

Data and Software Availability Statements

All data and software supporting the findings of this study are publicly available at the following GitHub repository: https://github.com/alfiensukma/temupaper_app.git.

References

[1] A. Martín-Martín, M. Thelwall, E. Orduna-Malea, and E. Delgado López-Cózar, "Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: a multidisciplinary comparison of coverage via citations," *Scientometrics*, vol. 126, no. 1, pp. 871–906, Jan. 2021, doi: 10.1007/S11192-020-03690-4/METRICS.

[2] M. Gusenbauer, "Google Scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases," *Scientometrics*, vol. 118, no. 1, pp. 177–214, Jan. 2019, doi: 10.1007/S11192-018-2958-5/TABLES/4.

[3] I. Shabanov, "Litmaps vs ResearchRabbit vs Connected Papers - alattinjauanliteraturterbaik di tahun 2025 - akademisi yang mudah." Accessed: Feb. 14, 2025. [Online]. Available: https://effortlessacademic.com/litmaps-vs-researchrabbit-vs-connected-papers-the-best-literature-review-tool-in-2025/

[4] I. Zupic and T. Čater, "Bibliometric methods in management and organization," *Organ Res Methods*, vol. 18, no. 3, pp. 429–472, Jul. 2015, doi: 10.1177/1094428114562629;JOURNAL:JOURNAL:ORMA;REQUESTEDJOURNAL:JOURNAL:ORMA;WGROUP:STRING:PUBLICATION.

[5] S. Kojaku, G. Livan, and N. Masuda, "Detecting anomalous citation groups in journal networks," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-93572-3.

[6] Y. Yaniasih, "Teorikritisterhadapanalisissitasiuntukkajiankuantitatifsainsdanevaluasikinerjariset," *BerkalaIlmuPerpustakaandanInformasi*, vol. 16, no. 1, pp. 127–141, Jun. 2020, doi: 10.22146/bip.v16i1.72.

[7] R. Widayanti, M. Heru, R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, "Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering," *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 289–302, 2023.

[8] M. D. D. Maristha, A. J. Santoso, and F. K. S. Dewi, "Sistemrekomendasipembelianprodukkesehatanpada e-commerce abcberbasis graph database amazon neptunemenggunakanmetode hybrid content-collaborative filtering," 2021.

[9] A. L. Pradana and A. Wibowo, "User demographic information and deep neural network in film recommendation system based on collaborative filtering," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 5, pp. 139–146, May 2022, doi: 10.46338/ijetae0522_16.

[10] M. F. Aljunid and M. D. Huchaiah, “An efficient hybrid recommendation model based on collaborative filtering recommender systems,” *CAAI Trans IntellTechnol*, vol. 6, no. 4, pp. 480–492, Dec. 2021, doi: 10.1049/cit2.12048.

[11] M. Uta *et al.*, “Knowledge-based recommender systems: overview and research directions,” 2024, *Frontiers Media SA*. doi: 10.3389/fdata.2024.1304439.

[12] R. Burke, “Hybrid recommender systems: survey and experiments,” *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002, doi: 10.1023/A:1021240730564/METRICS.

[13] S. Sakr *et al.*, “The future is big graphs,” *Commun ACM*, vol. 64, no. 9, pp. 62–71, Sep. 2021, doi: 10.1145/3434642.

[14] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, “SPECTER: Document-level representation learning using citation-informed transformers,” Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.07180

[15] M. T. Colangelo, M. Meleti, S. Guizzardi, E. Calciolari, and C. Galli, “A comparative analysis of sentence transformer models for automated journal recommendation using pubmed metadata,” *Big Data and Cognitive Computing*, vol. 9, no. 3, p. 67, Mar. 2025, doi: 10.3390/BDCC9030067/S1.

[16] M. Wijewickrema, V. Petras, and N. Dias, “Selecting a text similarity measure for a content-based recommender system: A comparison in two corpora,” *Electronic Library*, vol. 37, no. 3, pp. 506–527, Aug. 2019, doi: 10.1108/EL-08-2018-0165.

[17] I. Kanellos, T. Vergoulis, D. Sacharidis, T. Dalamagas, and Y. Vassiliou, “Impact-based ranking of scientific publications: a survey and experimental evaluation,” 2019. [Online]. Available: https://www.budapestopenaccessinitiative.org/

[18] J. Zhou, A. Zeng, Y. Fan, and Z. Di, “Ranking scientific publications with similarity-preferential mechanism,” *Scientometrics*, vol. 106, no. 2, pp. 805–816, Feb. 2016, doi: 10.1007/s11192-015-1805-1.

[19] V. Latora and M. Marchiori, “A measure of centrality based on network efficiency,” *New J Phys*, vol. 9, no. 6, p. 188, Jun. 2007, doi: 10.1088/1367-2630/9/6/188.

[20] I. Sommerville, *Software Engineering (9th Edition)*. Addison-Wesley, 2010. Accessed: Jun. 30, 2025. [Online]. Available: http://www.amazon.com/Software-Engineering-9th-Edition-Sommerville/dp/0137035152

[21] A. F. Syarifaldi, O. V. Berhitu, R. Apriansyah, and H. Kuswanto, “Penerapansisteminformasi data order berbasis web menggunakanmetode waterfall,” *Media JurnalInformatika*, vol. 15, no. 2, p. 185, Dec. 2023, doi: 10.35194/mji.v15i2.3508.