Ahliya Journal of Business Technology and MEAN Economies
[Published by Palestine Ahliya University]
Vol. 02 Issue 02 (2025)
ISSN: 3007-9691

Recommender System for Delivering Learning Contents from a Digital Library to a Learning Management System

Khitam Sabateen[1], Jamil A. Itmazi[2]
[1] M.Sc. Students' Research Club, Palestine Ahliya University (Palestine)
alabedkhitam@gmail.com
[2] Palestine Ahliya University (Palestine)
j.itmazi@paluniv.edu.ps

Received:22/05/2025 Accepted:15/07/2025 Published:31/12/2025

Abstract:

This study enhances recommender systems (RSs) in digital libraries (DLs) integrated with learning management systems (LMSs), specifically Moodle at Palestine Ahliya University, to provide students with personalized educational resource recommendations. A hybrid RS combining content-based (CBF), rule-based (RBF), demographic-based (DBF), and collaborative filtering (CF) was developed. The methodology included content analysis of Moodle and DL resources, expert consultations, and user testing. The system, designed using a waterfall model, successfully prioritized recommendations based on title, keywords, subject, and user demographics, displaying results with relevance scores. User testing highlighted the need for continuous refinement. Practical implications include improved resource discovery efficiency and relevance, saving time and effort. The study underscores the importance of integrating RSs into LMSs and DLs to enhance learning experiences. The originality of this study lies in the integrated deployment of a multi-technique hybrid recommender system within a real university LMS–digital library environment, moving beyond isolated recommendation models. It also introduces a priority-based academic relevance mechanism that aligns recommendations with course content, learner characteristics, and digital library metadata.

Keywords: *Recommender Systems; University Courses; Digital Library; Learning Management System; Collaborative Filtering; Content-Based Filtering.*

Ahliya Journal of Business Technology and MEAN Economies [Published by Palestine Ahliya University]
Vol. 02 Issue 02 (2025) ISSN: 3007-9691

نظام توصية لتوصيل محتويات التعلم من مكتبة رقمية إلى نظام إدارة التعلم

ختام سباتين[1]، جميل اطميزي[2]

[1] نادي أبحاث طلاب الماجستير، جامعة أهلية فلسطين (فلسطين)
alabedkhitam@gmail.com
[2] جامعة فلسطين الأهلية (فلسطين)
j.itmazi@paluniv.edu.ps

تاريخ الاستلام:2025/05/22 تاريخ القبول:2025/07/15 تاريخ النشر:2025/12/31

ملخص:

تعزز هذه الدراسة أنظمة التوصية في المكتبات الرقمية المدمجة مع أنظمة إدارة التعلم، وتحديدًا نظام Moodle في جامعة فلسطين الأهلية، بهدف تزويد الطلبة بتوصيات تعليمية مخصّصة للموارد الأكاديمية. وقد تم تطوير نظام توصية هجين يجمع بين أساليب الترشيح المعتمد على المحتوى، والترشيح القائم على القواعد، والترشيح الديموغرافي، والترشيح التعاوني. شملت المنهجية تحليل محتوى موارد Moodle والمكتبة الرقمية، والاستعانة بآراء الخبراء، واختبار المستخدمين. وقد صُمّم النظام وفق نموذج الشلال، ونجح في ترتيب أولويات التوصيات اعتمادًا على العنوان والكلمات المفتاحية والموضوع والخصائص الديموغرافية للمستخدمين، مع عرض النتائج مصحوبة بدرجات للملاءمة. وأظهرت اختبارات المستخدمين الحاجة إلى التحسين المستمر للنظام. وتتمثل الآثار العملية للدراسة في تحسين كفاءة وملاءمة اكتشاف الموارد التعليمية، مما يوفر الوقت والجهد. وتؤكد الدراسة أهمية دمج أنظمة التوصية في أنظمة إدارة التعلم والمكتبات الرقمية لتعزيز التجربة التعليمية. وتكمن الأصالة العلمية لهذه الدراسة في التطبيق المتكامل لنظام توصية هجين متعدد التقنيات ضمن بيئة جامعية حقيقية تجمع بين نظام إدارة التعلم والمكتبة الرقمية، متجاوزة بذلك نماذج التوصية المعزولة. كما تقدم آلية أولوية قائمة على الملاءمة الأكاديمية تعمل على مواءمة التوصيات مع محتوى المقررات الدراسية وخصائص المتعلمين وبيانات المكتبة الرقمية.

الكلمات المفتاحية: *أنظمة التوصية؛ المقررات الجامعية؛ المكتبة الرقمية؛ نظام إدارة التعلم؛ التصفية التعاونية؛ التصفية القائمة على المحتوى.*

1. Introduction

1.1 Digital Library (DL)

1.1.1 Concept and Definition

The term "Digital Library (DL)" appeared in the literature as early as 1988 in a technical report published by the Corporation for National Research Initiatives (CNRI) (Kahn & Cerf, 1988). It gained prominence with the NSF/DARPA/NASA Digital Library Initiative in 1994 (Paulsen & Rekkedal, 2001). Alternative terms like "electronic library" or "virtual library" are also used. A DL stores and manages collections in digital formats (e.g., text, images, audio, and video) rather than traditional print media, enabling organized, computer-based storage, retrieval, and networked access (Borgman, 2000). It functions as an online repository, hosting diverse digital media such as documents, multimedia files, and social media content (Witten et al., 2009). DLs vary in scale, from personal collections to institutional archives, and support remote access via networks while ensuring interoperability (Lanagan & Smeaton, 2012).

Unlike the unstructured internet, DLs are systematically curated, with content selected for reliability and relevance. They provide structured access to information, mitigating data overload.

Key Characteristics of a DL:

1. A service-oriented system.
2. A repository of digital objects.
3. User-centric support for information retrieval.
4. Organized presentation of resources.
5. Direct or indirect accessibility.
6. Digital availability.

DLs streamline the creation, storage, and dissemination of digital content, facilitating efficient search and retrieval (see Figure 1). Their core functions include acquisition, cataloging, access management, and resource organization through standardized protocols.

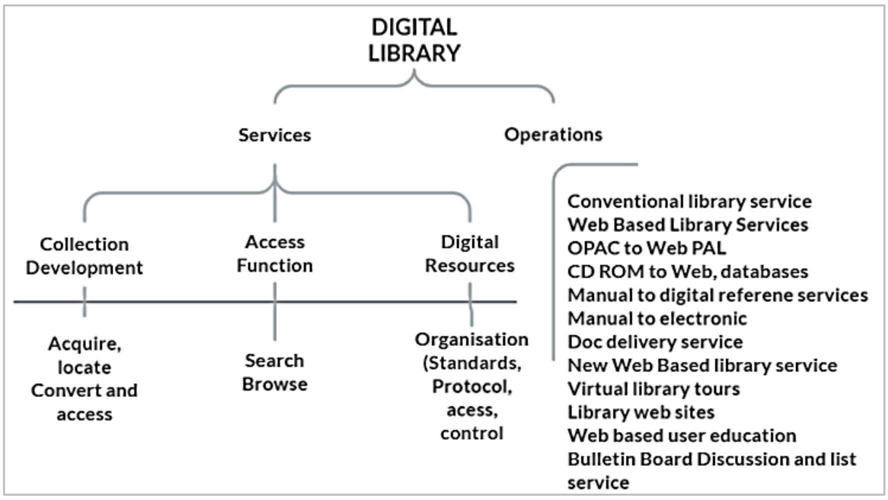

Figure 1: The operations and services of Digital Libraries

1.1.2 Architecture of DL System

Digital Library (DL) systems comprise four core components: the user interface, search system, handling system, and repository. These operate across various internet-connected computer systems, as illustrated in Figure 2 (Papi, 2023).

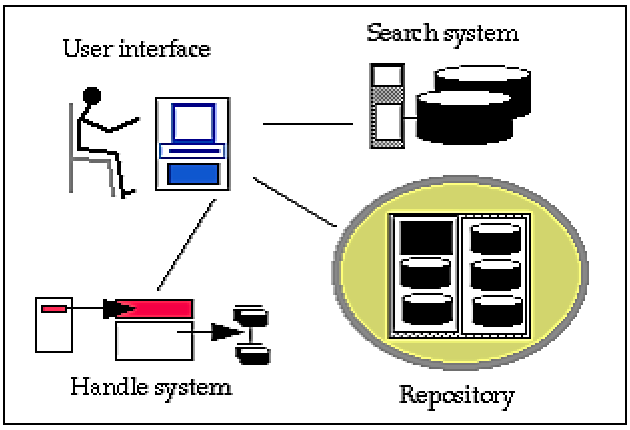

Figure 2: Main Components of The Digital Library System

1. User Interface
   Digital libraries provide organized collections of resources in digital formats that are accessed through web-based user interfaces and supported by backend services responsible for storage, retrieval, and management (Borgman, 2000).
2. Repository
   Digital repositories store and manage digital objects using standardized architectures that support ingestion, storage, retrieval, access control, and rights management through defined service interfaces (ISO, 2012).
3. Handle System
   This system assigns persistent identifiers to digital objects, enabling long-term resource tracking across repositories. It maps objects to their storage locations for retrieval.
4. Search System
   Digital libraries use indexed metadata and standardized protocols such as OAI-PMH to enable users to query repository records prior to accessing full digital objects (Open Archives Initiative, 2015).

1.1.3 Types of Digital Libraries

According to Kadury and Frank (2007), digital libraries (DLs) can be categorized into 3 main architectural models based on how resources are stored, integrated, and accessed:

A. Stand-alone Digital Libraries (SDLs): refer to independent, self-contained digital collections that manage and provide access to their own locally stored resources without relying on external systems. Typical examples include the ACM Digital Library (www.acmdl.org), IEEE Computer Society Digital Library (www.ieeedl.com), and the Library of Congress (www.loc.gov), where all content and services are maintained within a single platform.
B. Federated Digital Libraries (FDLs): integrate multiple autonomous SDLs through a unified interface that enables simultaneous searching across distributed repositories. These systems emphasize interoperability and metadata harmonization, although they face technical challenges related to heterogeneous standards and protocols. Examples include Networked Computer Science Technical Reference Library (NCSTRL) and Networked Digital Library of Theses and Dissertations (NDLTD).
C. Harvest Digital Libraries (HDLs): function as virtual or aggregated libraries that collect and index metadata from distributed sources rather than storing the full content locally. This approach supports centralized discovery and streamlined access to curated, subject-specific

resources. Representative systems include the Internet Public Library (IPL) and other virtual library portals.

1.2 Learning Management Systems (LMSs)

1.2.1 History of LMS

The evolution of LMS began with computer-based education concepts (Watson & Watson, 2012). The first integrated LMS, EKKO, launched in 1991 by Norway's NKI Distance Education Network, followed by DOS-based systems like NB Learning Network (Paulsen & Rekkedal, 2001). The UK's Open University adopted First-class in the 1990s, pioneering European online learning (Oxagile, 2016; Sharma, 2015).

Moodle, the first open-source LMS (2002), revolutionized e-learning with its plug-and-play functionality, enabling content portability and cost-effective training delivery. SCORM (2004) standardized content packaging, while the Experience API (2013) advanced tracking capabilities. Post-2012, customizable LMS platforms emerged, replacing legacy systems across education and corporate training (Ellis, 2013).

1.2.2 Definition of LMS

An LMS is an internet-based software for delivering educational content, tools, and assessments (Adobe, 2009; Srichanyachon, 2014). It supports cross-device access, course management, and student-teacher interaction (Coates et al., 2005). Institutions use LMSs to streamline training, maintain records, and facilitate assignments, grading, and exams (Jurubescu, 2008).

1.2.3 LMS Platforms

Two primary types exist:

- Open-source: Moodle (developed in 2001), Sakai, ATutor.
- Commercial: Blackboard, SuccessFactors, Litmos.

Moodle's flexibility, user-friendly interface, and integration capabilities make it widely adopted. For example, Oman's Majan University College customizes Moodle as "MOVE" for 3,500 students

1.3 Recommender System (RS)

1.3.1 Definition of Recommendation Systems (RS)

RSs are software tools that suggest relevant items to users (students, customers, etc.) based on their historical data and behavior (Ricci et al., 2021). As a machine learning application, they filter information to predict user preferences efficiently (Leiva et al., 2020).

The rise of RSs addresses internet data overload, helping users discover content aligned with their interests—from products (Amazon) to educational resources. RSs analyze clicks, purchases, and social interactions to refine suggestions (Han et al., 2018).

1.3.2 Types of RSs

1. Collaborative Filtering (CF):
   - Predicts preferences by comparing user behavior (Fayyaz et al., 2020).
   - User-to-User: Matches similar users.
   - User-to-Product: Recommends items liked by similar users.
2. Content-Based Filtering (CBF):
   - CBF recommends items by analyzing the characteristics and features of content that align with users' past preferences and interaction history, such as movie genres or product attributes. The method constructs user profiles based on previously selected items and measures similarity between new and existing items using mathematical techniques, most commonly cosine similarity. This similarity metric enables the system to identify and suggest items that closely match the user's interests (Abdurrafi & Ningsih, 2023).

- *Limitations*: Requires rich metadata; lacks novelty (Gong & Cheng, 2008).

3. Hybrid RSs:

- Combines CF and CBF to mitigate cold-start/data sparsity issues. Methods include weighted scoring and feature augmentation (Patel & Patel, 2015).

1.3.3 Industrial Applications

- E-commerce: Amazon, Alibaba (personalized product suggestions).
- Entertainment: Netflix, Spotify (content recommendations).
- Transportation: Route optimization (GPS/RFID data) (Fayyaz et al., 2020).
- Healthcare: Treatment suggestions based on medical history.

1.3.4 Benefits

- Boosts sales and user engagement.
- Enhances personalized experiences.

1.3.5 Challenges

1. Data Sparsity: Sparse user-item interactions.
   - *Solution*: Matrix factorization (e.g., SVD).
2. Cold Start: New users/items lack data.
   - *Solution*: CBF for initial recommendations.
3. Scalability: Handled via distributed systems (Spark, Hadoop).
4. Privacy: Requires anonymization techniques (encryption).

1.4 The Role of AI in RSs

AI enhances recommendation systems by analyzing user behavior and preferences through advanced pattern recognition. Key capabilities include:

- Big Data Processing: AI handles large datasets (e.g., Netflix's viewing history analysis (Ricci et al., 2015).
- Personalization: Deep learning tailors' suggestions (e.g., Amazon's review-based recommendations (Zhang et al., 2019)).
- Unstructured Data Handling: CNNs/RNNs process images/text (e.g., Pinterest's visual recommendations).
- Continuous Learning: Systems like Spotify's *Discover Weekly* adapt dynamically (Aggarwal, 2016).

1.4.1 Key Components

1. Collaborative Filtering:
   - *User-Based*: This approach predicts a user's preferences by identifying other users with similar past behaviors and recommending items based on aggregated patterns among these similar users (Koren, Bell, & Volinsky, 2009).
   - *Item-Based*: Analyzes item relationships via matrix factorization.
2. Content-Based Filtering: Matches item features (e.g., genre/keywords) to user profiles (Aggarwal, 2016).
3. Hybrid Algorithms: Combine CF/CBF (e.g., Netflix's hybrid model (Zhang et al., 2019)).
4. Deep Learning Models:
   - CNNs for images, RNNs for sequential data.
   - Embeddings for unstructured data.

1.4.2 Applications

Table 1: AI Algorithms in Key Applications

Application	Platforms	Algorithms Used
E-Commerce	Amazon, eBay	CF (KNN, Matrix Factorization), CBF (TF-IDF), Hybrid
Streaming	Netflix, Spotify	CF (KNN, SVD), DL (RNNs, CNNs), Embeddings
Online Education	Coursera, edX	CF (KNN), CBF (Naive Bayes, SVM), DL (Neural Nets)
Healthcare	-	CF (KNN), DL (Multi-layer NNs), Embeddings
Personal Finance	-	CF (KNN), DL (RNNs), Traditional ML (Decision Trees)

1.4.3 Technical Specifications

Key Components:

- **Interfaces**:
  - *Student*: Search block with relevance scores.
  - *Teacher*: Custom recommendation controls.
- **Algorithms**: Cosine Similarity, SVD, TF-IDF.
- **Data Sources**: Moodle DBs, DL resources, user profiles.

Workflow:

1. Login → 2. Search → 3. Algorithmic analysis → 4. Ranked results.

Programming: Python (data processing), PHP (Moodle), SQL (DBs).

2. Literature Review

This chapter establishes the theoretical foundation for Recommender Systems (RSs) in Digital Libraries (DLs) and Learning Management Systems (LMSs). It examines key recommendation approaches including Collaborative Filtering (CF), Content-Based Filtering (CBF), Rule-Based Filtering (RBF), and hybrid models, while addressing academic challenges like cold start problems and data sparsity. The integration of RSs with educational platforms is shown to improve resource accessibility and student engagement. See Table 2.

Table 2 summarizes relevant studies:

Study	Year	Key Contribution	Algorithms
Talaghzi et al.	2023	Multi-criteria RS for e-learning	CF, CBF, Hybrid
Tolety et al.	2022	Hybrid CF-CBF for adaptive e-learning	CF, CBF
Monsalve-Pulido et al.	2020	Autonomous RS for virtual learning	CF, CBF, Knowledge-based
Nugraha et al.	2020	DL book recommendations	User-based CF

2.1 Study Problem and Questions

Core Problem: Information overload in DLs creates challenges for Moodle users seeking relevant academic resources.

Key Questions:

1. Algorithm design for optimal educational resource recommendations
2. Critical factors in DL content recommendation systems
3. Comparative analysis of RS algorithms in higher education

2.2 Objectives & Importance

Primary Objectives:

1. Develop Moodle-integrated DL recommendation algorithm
2. Evaluate academic applicability of CF, CBF, and RBF
3. Optimize recommendation precision and relevance

Significance:

- Enhances personalized learning experiences
- Streamlines academic resource discovery

- Improves institutional knowledge management

2.3 Research Methodology and Scope

Approach:

- Case study analysis
- Document/content examination
- Waterfall development with UML modeling

Focus: Moodle and DSpace implementation at Palestine Ahliya University, excluding non-educational applications.

3. Methodology and Procedures

The higher education landscape is undergoing rapid transformation, driven by the integration of Learning Management Systems (LMSs), Digital Libraries (DLs), and Recommendation Systems (RSs). These technologies collectively enhance the learning experience by streamlining content delivery, improving accessibility, and personalizing educational resources. Platforms like Moodle enable seamless material sharing, assignment management, and online collaboration, while DLs act as comprehensive knowledge hubs, offering diverse digital assets such as scholarly articles, e-books, and multimedia content. The incorporation of RSs further refines this ecosystem by leveraging user behavior and academic preferences to suggest tailored resources. This integration not only boosts search accuracy but also optimizes time efficiency for both students and educators by delivering content that aligns with individual academic needs, course demands, and learning histories.

The collaboration between LMSs, DLs, and RSs fosters a unified educational framework that surpasses conventional Classroom-Based Learning. Functioning as intelligent decision-support tools, RSs assess multiple student profile factors—including interests, enrolled courses, and instructor recommendations—to curate the most pertinent digital materials. This adaptive approach enhances educational efficiency, ensuring learners receive resources that directly support their academic progress. By harnessing the combined strengths of these systems, institutions can cultivate a more interactive and customized learning environment, ultimately enriching the educational experience for all stakeholders.

3.1 Methodology

This research employs two primary analytical methods: content analysis and document analysis, both aimed at refining recommender systems (RS) within Moodle and digital libraries (DL). Additionally, the study follows a waterfall model for system and prototype development, supplemented by UML diagrams to enhance design clarity.

1. Content Analysis

   This method involves scrutinizing textual data from Moodle and DLs to uncover patterns and themes relevant to the study. Key focuses include:

- Assessing academic materials such as research papers, articles, and educational resources.
- Identifying trends related to RSs and their role in resource recommendation.
- Evaluating titles, topics, and metadata to determine their alignment with user requirements. This process reveals insights into digital content structure, aiding in the development of more effective RSs.

2. Document Analysis

   A deeper investigation into selected references and resources, emphasizing:

- The quality and features of available digital content.
- Metadata, contextual details, and keyword relevance to gauge resource usefulness.

- How well materials meet user expectations, ensuring precise recommendations. Findings from this analysis contribute to a thorough understanding of educational resources and their potential to enhance recommendation algorithms.

3.2 Algorithm and Prototype Development Methodology

1. Waterfall Model

The waterfall model was selected for its systematic, phase-based structure—requirements analysis, design, implementation, testing, deployment, and maintenance. Its linear approach suits projects with clearly defined objectives, prioritizing stability and thoroughness over iterative flexibility.

2. Unified Modeling Language (UML) Diagrams

These diagrams serve to:

- Visualize system architecture, component interactions, and workflows.
- Demonstrate RS algorithm functionality through use cases, sequence diagrams, and class diagrams.
- Enhance design accuracy and stakeholder communication during development.

3. Data Collection and Analysis

The study gathers textual and document-based data from Moodle and distance learning sources, selected for their research relevance. Analysis employs qualitative techniques, including:

Content Analysis:

- Objective: Extract insights on RSs by examining digital content structure, metadata, and search relevance.
- Outcome: Assess resource quality and alignment with user needs, refining the recommendation algorithm.

By merging content and document analysis, this methodology offers a holistic approach to improving RSs within Moodle and DLs. The structured use of the waterfall model and UML diagrams ensures precision, coherence, and alignment with research goals, ultimately advancing RSs to better serve academic users.

3.3 Data Collection Tools

A. Document Analysis:

This involves reviewing textual content from DL resources in Moodle, applying thematic and content analysis techniques to identify patterns and extract key information. The process deepens understanding of academic materials and evaluates RS effectiveness.

B. Expert Meetings:

Collaborations with RS, Moodle, DL, and LMS specialists through structured discussions and surveys help shape the research direction. These sessions assess the feasibility of proposed improvements and strategies to optimize RS integration in Moodle.

C. Testing Tool Feedback:

Post-implementation evaluations of the RS algorithm involve gathering input from users (students and instructors) via surveys or brief interviews. This feedback is instrumental in refining the system to better meet user needs.

4. Model Design and Building

This chapter outlines the methodology for developing a hybrid recommendation system tailored for Learning Management Systems (LMS) and Digital Libraries (DLs). The proposed model integrates multiple filtering techniques—content-based, rule-based, demographic-based, and collaborative filtering—to enhance recommendation accuracy for students.

The system architecture is examined in detail, explaining how each algorithm functions within the framework. Pseudocode is provided to illustrate the implementation steps, along with an analysis of how final recommendations are merged and evaluated for relevance to user needs.

4.1 Hybrid RS Method

A hybrid recommendation system (RS) is more effective than a single-method approach for suggesting learning materials from a DL within an LMS. The research proposes a general RS algorithm, as depicted in Figure 3, along with its pseudocode.

Pseudocode for Recommendation System (RS) Algorithm

1. Access LMS Website (Moodle):
   - Connect to the LMS to retrieve student and course data.
2. Retrieve List of Digital Resources:
   - Query the DL to fetch available materials (e.g., books, papers, videos).
3. Normal Searching:
   - Conduct a basic search based on student queries or course requirements.
4. Apply Content-Based Filtering:
   - Analyze resource metadata (title, subject, type).
   - Match resources to the student's courses, interests, and academic history.
5. Apply Rule-Based Filtering:
   - Enforce predefined rules (e.g., exclude outdated materials, prioritize recent publications).
6. Apply Demographic-Based Filtering:
   - Filter resources based on student demographics (academic level, department).
7. Apply Collaborative Filtering:
   - Analyze behavior of similar students (same course or interests).
   - Recommend resources frequently accessed or highly rated by peers.
8. Apply Duplicates Filtering:
   - Remove duplicate entries to streamline recommendations.
9. Generate Final Recommendation List:
   - Combine results from all filtering methods.
   - Rank resources by relevance, popularity, and student preferences.
10. Display Final List of Recommended Resources:
    - Present the top N recommendations to the student via the LMS.

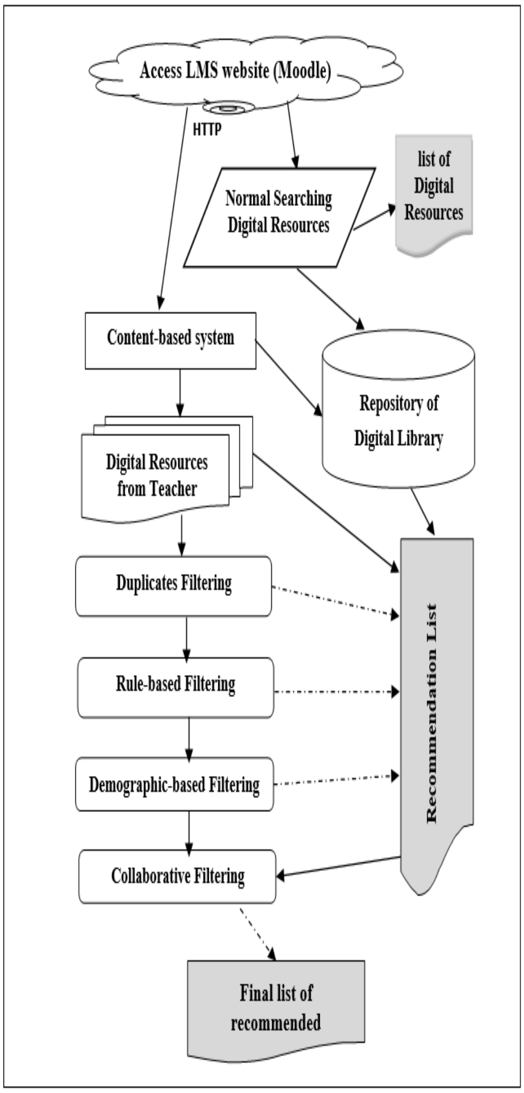

Figure 3: The General RS Structure

4.2 Algorithm Steps for Creating an RS in Moodle

The proposed algorithm enhances resource recommendations in Moodle by analyzing user queries and academic needs. A dedicated search block is integrated into the Moodle interface, allowing users to easily find relevant materials.

Example of Integration in Moodle:

1. User Login: Students log in to their accounts, where a custom recommendation block appears.

2. User Interface Block: The block provides a simple search interface (Figure 4) with parameters for refining results.
3. Real-Time Updates: The algorithm adjusts recommendations based on user feedback (e.g., skipped or poorly rated content).

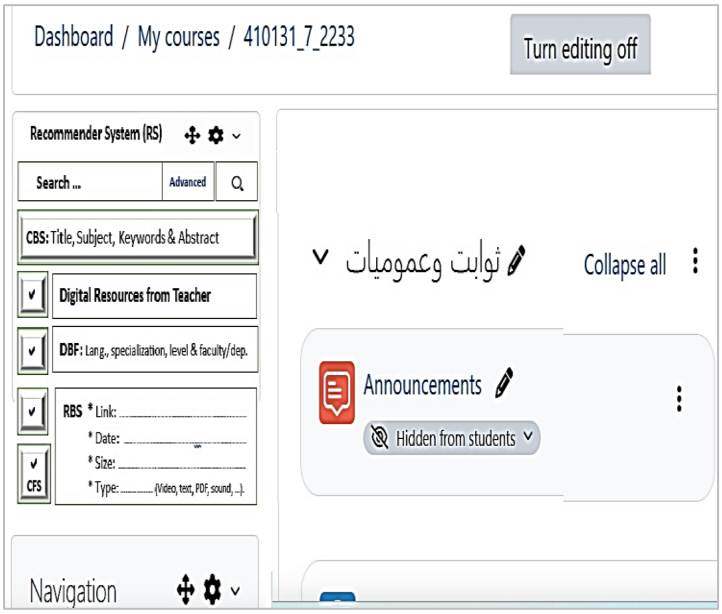

Figure 4: The user interface (block) for a student.

4.3 The "RS Block"

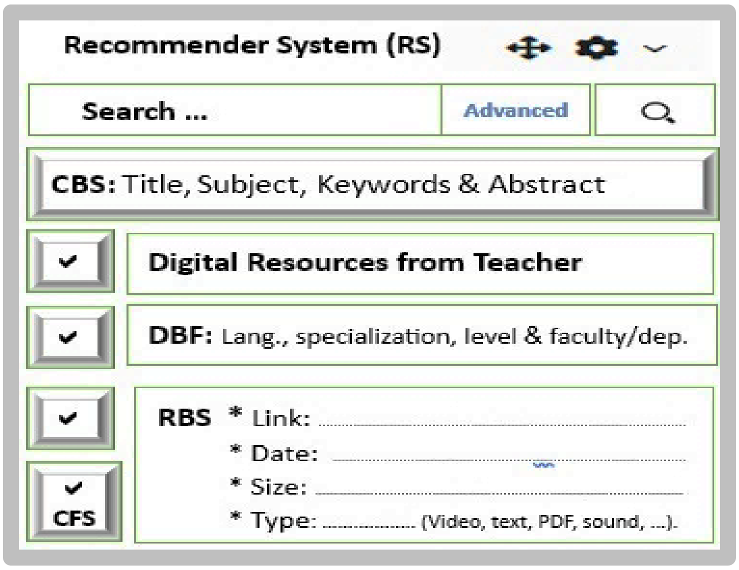

Figure 5: The RS block

The RS block (Figure 5) consists of multiple levels:

Level 1: Search Box

– Users enter keywords, and the system retrieves matching resources.

Level 2: Content-Based Recommendation Algorithm

– Resources are scored based on metadata (title, keywords, topics).
– A mathematical formula calculates relevance:

$$S_{\text{core}} = \frac{\sum_{i=1}^{n} w_i \cdot \text{match}(a_i, b_i)}{\sum_{i=1}^{n} w_i} \times 100$$

Where:

- Score: Final relevance percentage.
- *Wi*: Weight assigned to each criterion (e.g., title = 0.4, topic = 0.3).
- Match (*ai*, *bi*): Returns 1 if criteria match, 0 otherwise.
- *n*: The number of criteria used in the recommendation.

Example Calculation:

If a student searches for *"Machine Learning"* with weights Title (0.4), Topic (0.3), Keywords (0.3), a perfect match yields:

$$\text{Score} = \frac{(0.4 \quad 1)+(0.3x \quad 1)+(0.3x \quad 1)}{0.4+0.3+0.3} \times 100$$

$$\text{Score} = \frac{0.4+0.3+0.3}{1} \times 100 = 1\times 100 = 100\%$$

Results Display:

Resources are listed in descending order by score (Table 3).

Table 3: Example of Results

Resource Title	Percentage
Machine Learning	100%
Modern Neural Networks	95%
Deep Data Analysis	90%

Challenges & Improvements:

- Some results may be superficially relevant but lack depth.
- Weight adjustments can improve accuracy (e.g., prioritizing keywords over titles).
- Rule-based filtering is introduced to refine results further.

CBR Algorithm steps

In this stage, the system performs deep search between the active eCourse of LMS and the resources in the DL to obtain the closest results. The criteria could be Title, Subject, Keywords and Abstract. This Algorithm will make the following as it in figure 6:

- Search the current eCourse attributes and restore the matched resources in the DL. The course attributes include: Reference/resource title, Subject, Keywords and Abstract.
- Getting the attributes of the digital resources in the DL database, (Title, Subject, Keywords and Abstract).
- Analyzing the attributes of the digital resources in the digital library (DL) database, such as Title, Subject, Keywords, and Abstract.
- Comparing the attributes of the current eCourse with those of the digital resources.
- Adding the link to the digital resource in the recommendation list if it is relevant to the current eCourse.
- Repeating the above process until all digital resources have been evaluated.
- Finally, the "recommended resources" are moved to the next stage.

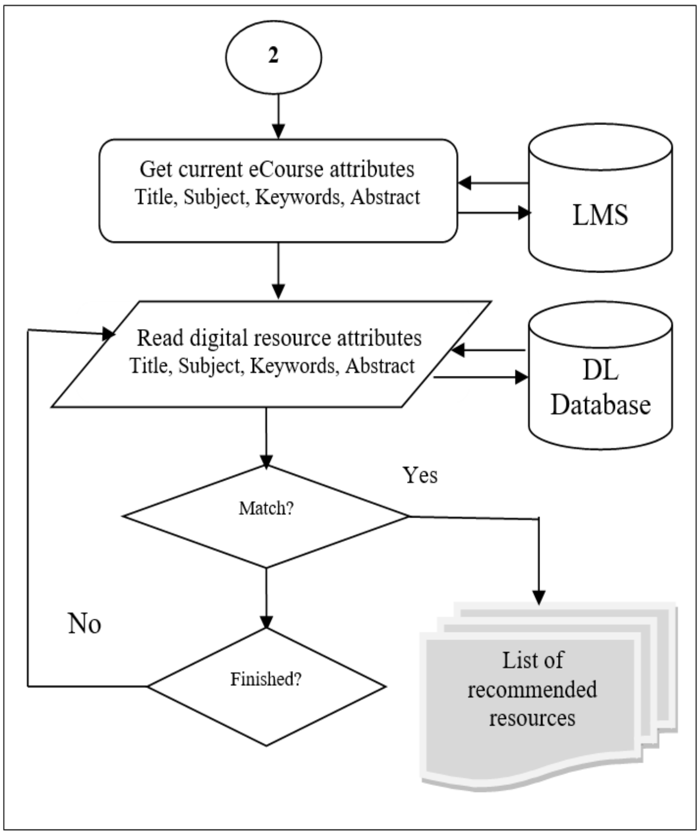

Figure 6: Flowchart of the Content-based system

Level 3: Teacher Recommendations (figure 7)

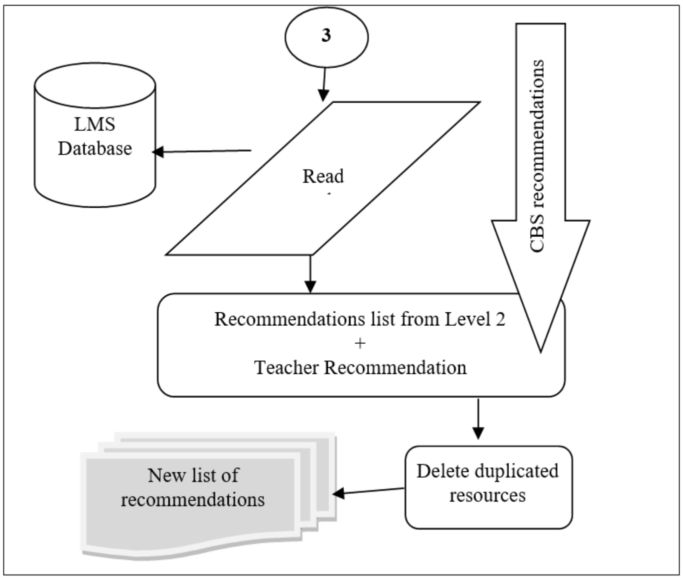

Figure 7: Flowchart of Teacher recommendations

- Instructors can manually recommend resources (internal/external).
- Teacher-suggested materials are given higher priority.
- Duplicates are removed, retaining the highest-scored version.

Level 4: Demographic & Rule-Based Filtering

- Demographic filtering excludes resources mismatching student profiles (e.g., language, department).
- Rule-based filtering applies additional constraints (e.g., publication date, file type) see figure 8.

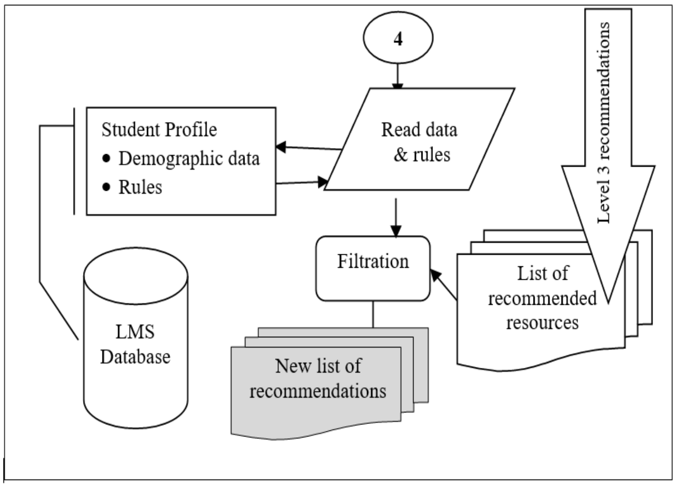

Figure 8: Flowchart of Demographic-Based & Rule-Based Filtering

Level 5: Collaborative Filtering (CF)

- Recommends resources highly rated by similar students (same department/course). See figure 9.
- Uses a rating matrix (Table 4) to calculate peer preferences.

Table 4: Rating matrix

Digital Resource Student	DR_1	DR_2		DR_m
Std_1				2
Std_2	5	3	3	
			3	5
Std_n	3		5	

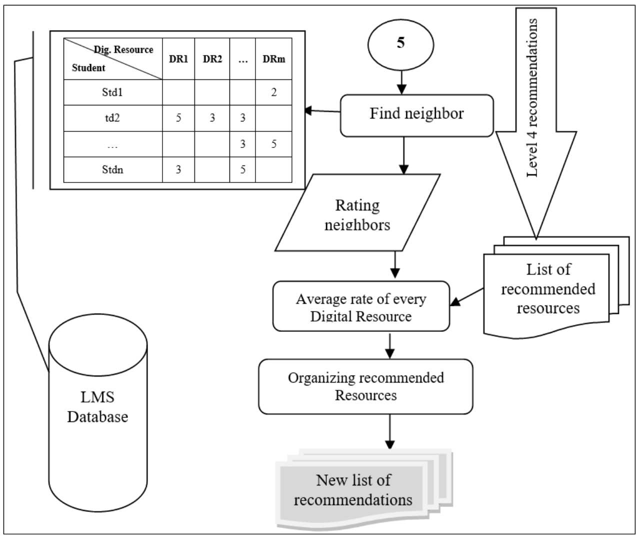

Figure 9: The general step of the CF system stage.

5. Discussion

5.1 Analysis of the Algorithms Used to Produce a Modified Algorithm

This chapter reviews the rationale behind selecting four key algorithms:

1. Content-Based Filtering (CBF): Analyzes educational resource characteristics to provide recommendations aligned with user preferences.
2. Rule-Based Filtering (RBF): Applies user-defined parameters (e.g., students, professors) to refine recommendation accuracy.
3. Demographic Filtering (DBF): Personalizes suggestions based on user demographics like academic specialization, education level, and language.

4. Collaborative Filtering (CF): Customizes recommendations by analyzing interaction patterns of similar users.

These algorithms were chosen to address common challenges in traditional recommendation systems, such as cold starts, limited diversity, and data sparsity. By retaining their core elements without modification, we better understand their individual strengths and limitations, paving the way for a hybrid algorithm that combines their advantages.

The modified hybrid algorithm integrates these four approaches to enhance recommendation accuracy and efficiency for educational resources. This integration aims to overcome individual weaknesses while providing a comprehensive system tailored to the needs of students and faculty. This chapter will analyze each algorithm's components, justify their selection, and explain their innovative integration into the system.

Additionally, we discuss technical and practical considerations in algorithm design, focusing on challenges like recommendation diversity and system efficiency. A sequential experiment serves as an illustrative example of the system's functionality.

This discussion serves as a reference for the analytical process behind the system's design, reflecting a deep understanding of recommendation system foundations in Moodle and their alignment with modern educational demands and digital library resources.

1. An Overview of the Login Mechanism in the Moodle System

As shown in Figure 3, the first step involves students logging into their university's Moodle system using their credentials. Upon login, they access their dashboard, which displays academic courses and other relevant information.

A dedicated "Recommended Resources" block has been proposed for both student and faculty interfaces. For students, this block includes search options—either standard searches within the university's digital library or specialized searches using recommendation algorithms. Faculty members can use their block to add recommended resources, which then appear in student suggestion lists.

Key features of Moodle that support this research include:

- User-centric design: Moodle's interface is tailored for students, making it ideal for integrating recommendation tools.
- Customization tools: The platform allows seamless addition of specialized search blocks with algorithm-based options.
- Enhanced search functionality: Combining traditional and algorithm-driven searches improves resource discovery.

The digital library serves as the primary data source, providing metadata (titles, keywords, abstracts) for recommendation algorithms. Its integration with Moodle ensures easy access to diverse academic resources, including e-books, videos, and research papers.

2. Explanation of the Student and Faculty Blocks in Moodle

- Student Block: Recommends resources based on academic profile (specialization, current courses, past interactions), streamlining resource discovery.
- Faculty Block: Enables professors to suggest resources for specific courses, which are prioritized in student recommendations.

Example Workflow:

- Student: Logs in → Accesses recommendation block → Selects search type → Receives personalized resource list.
- Faculty: Adds course-specific resources → Algorithm prioritizes these in student suggestions.

Figure 4 illustrates the Moodle interface for a Palestine Ahliya University student, showing the recommendation block.

3. First Search Level: Regular Search Box

The block includes a standard search box where users can enter keywords or resource names. Results are displayed as an organized list, similar to search engines like Google. This feature benefits users who prefer quick searches without algorithm assistance.

4. Second Search Level: Recommendation Algorithms

A. Content-Based Filtering (CBF)

CBF analyzes resource metadata (titles, keywords, abstracts) to match user queries. Students specify search criteria through checkboxes, and results are ranked by relevance scores (calculated via a predefined formula).

Example: A student searching for "Machine Learning" receives resources scored based on title, topic, and keyword matches. Faculty suggestions are weighted heavily (e.g., 90%) to ensure prominence.

B. Demographic-Based Filtering (DBF)

DBF refines recommendations using student demographics (language, specialization, academic level). For instance:

- Language: Filters resources by preferred language.
- Specialization: Prioritizes resources aligned with the student's field of study.
- Academic Level: Tailors suggestions to the student's year (e.g., second-year engineering).

C. Rule-Based Filtering (RBF)

RBF applies user-defined rules (e.g., file type, publication date) to exclude irrelevant resources. Key filters include:

- Link: Excludes specific resources by URL.
- Date: Filters by publication year (e.g., "AI conferences since 2020").
- File Size/Type: Excludes large files or undesired formats (e.g., videos).

D. Collaborative Filtering (CF)

CF enhances personalization by incorporating ratings from similar users (neighbors). Evaluations from same-college peers carry more weight, while faculty suggestions retain high priority. The algorithm dynamically adjusts resource weights based on feedback, ensuring relevance.

Chapter 6: Conclusion and Future Work

6. Conclusion

This research developed a hybrid recommendation system for Moodle-integrated digital libraries, combining CBF, RBF, DBF, and CF to address information overload and diverse learner needs. Key achievements include:

- CBF: Scored resources by metadata alignment.
- RBF: Applied user rules for refined results.
- DBF: Ensured academic relevance via demographics.
- CF: Leveraged peer ratings for personalization.

The system's Moodle integration and user-friendly interface demonstrated improved recommendation accuracy and satisfaction in testing.

6.1 Future Work

- Algorithm Enhancement: Integrate AI (e.g., deep learning) to improve CBF/CF accuracy.
- Recommendation Expansion: Include non-academic content (conferences, workshops).

- Database Growth: Collaborate with libraries/publishers to diversify resources.
- Global Integration: Connect with platforms like Blackboard and IEEE Xplore via APIs.
- UI/UX Improvement: Design intuitive interfaces with faster response times.
- Interaction Analysis: Use predictive analytics to anticipate student needs.

Implementation Priorities:

- Immediate focus on deep learning integration.
- Expand database and global LMS/DL integration.
- Refine UI/UX based on usability testing.

References

Abdurrafi, M. F., & Ningsih, D. H. U. (2023). *Content-based filtering using cosine similarity algorithm for alternative selection on training programs*. Journal of Soft Computing Exploration.

Adobe. (2009). Adobe solutions for learning management systems (LMS). Retrieved July 3, 2024, from www.adobe.com/content/dam/Adobe/en/education/pdfs/lms-primer-102909.pdf

Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer.

Borgman, C. L. (2000). *From Gutenberg to the global information infrastructure: Access to information in the networked world*. MIT Press.

Coates, H., et al. (2005). A critical examination of the effects of learning management systems on university teaching and learning. *Tertiary Education and Management*, 11(1), 19–36.

Ellis, R. K. (2013). A Field Guide to Learning Management Systems. ASTD Learning Circuits. Retrieved on January 25, 2025 from https://www.td.org

Fayyaz, Z., Ebrahimian, M., Nawara, D., Ibrahim, A., & Kashef, R. (2020). Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences*, 10(21), 7748.

Gong, S., & Cheng, G. H. (2008). Mining User Interest Change for Improving Collaborative Filtering. *2008 Second International Symposium on Intelligent Information Technology Application*, 3, 24–27.

Han, S., Qiao, Y., Zhang, Y., Lin, W., & Yang, J. (2018). Analyze users' online shopping behavior using interconnected online interest-product network. *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 15–18 April 2018 (pp. 1–6). IEEE.

ISO-International Organization for Standardization. (2012). *Reference model for an Open Archival Information System (OAIS) (ISO 14721:2012)*. https://public.ccsds.org/pubs/650x0m2.pdf

Jurubescu, T. (2008). Learning Content Management Systems. *Informatica Economica*, 12(4), 91–94.

Kadury, A., & Frank, J. (2007). *Harvesting and aggregation of digital libraries using the OAI framework*. Proceedings of the Third International Conference on Web Information Systems and Technologies (WEBIST).

Kahn, R. E., & Cerf, V. G. (1988). *The digital library project: Volume 1: The world of knowbots and information utilities*. Corporation for National Research Initiatives. www.cnri.reston.va.us/kahn-cerf-88.pdf

Koren, Y., Bell, R., & Volinsky, C. (2009). *Matrix factorization techniques for recommender systems*. IEEE Computer, 42(8), 30–37.

Kovač, M., Phillips, A., Weel, A., & Wischenbart, R. (2017). Book statistics: What are they good for? *Logos*, 28(1), 7–17.

Lanagan, J., & Smeaton, A. F. (2012). Video digital libraries: Contributive and decentralised. *International Journal on Digital Libraries*, 12(4), 159–178.

Leiva, M., Budán, M. C. D., & Simari, G. I. (2020). Guidelines for the analysis and design of argumentation-based recommendation systems. *IEEE Intelligent Systems*, 35(1), 28–37.

Monsalve-Pulido, J. A., Aguilar, J., Montoya, E., & Salazar, C. (2020). Autonomous recommender system architecture for virtual learning environments. *Applied Computing and Informatics*, 20(1/2), 69–88.

Nugraha, E., Ardiansyah, T., Junaeti, E., & Riza, L. S. (2020). Enhanced digital library with book recommendations based on collaborative filtering. *Journal of Engineering Education Transformations*, 34(0).

Open Archives Initiative. (2015). *The Open Archives Initiative protocol for metadata harvesting* (OAI-PMH). www.openarchives.org/OAI/openarchivesprotocol.html

Oxagile. (2016, April 12). History and trends of learning management system (Infographic). Retrieved July 3, 2024, from www.oxagile.com/article/history-and-trends-of-learning-management-system-infographics/

Papi, Z. (2023). A critical review and evaluation of the textbook entitled *Designing the Digital Library Based on the Standards and Criteria of University Textbooks. International Journal of Digital Content Management*, 4(1), 395–413. https://doi.org/10.22054/dcm.2022.70058.1156

Patel, Y. G., & Patel, V. (2015). A Survey on Various Techniques of Recommendation System in Web Mining. *International Journal of Engineering Development and Research*, 3(4), 696–700. ISSN: 2321-9939.

Paulsen, M., & Rekkedal, T. (2001). The NKI Internet College: A review of 15 years delivery of 10,000 online courses. *International Review of Research in Open and Distance Learning*, 1(2).

Ricci, F., Rokach, L., & Shapira, B. (2021). Recommender Systems: Techniques, Applications, and Challenges. In B. Shapira, L. Rokach, & R. Francesco (Eds.), *Recommender Systems Handbook: Third Edition* (pp. 1–35). Springer US.

Sharma, A. (2015, May 4). The history of distance learning and the LMS. *ELH Online Learning Made Simple.*

Srichanyachon, N. (2014). EFL learners' perceptions of using LMS. *Turkish Online Journal of Educational Technology*, 13(4), 30–35.

Talaghzi, J., Bellafkih, M., Bennane, A., Himmi, M. M., & Amraouy, M. (2023). Combined e-learning course recommender system. *International Journal of Emerging Technologies in Learning (iJET)*, 18(06), 53–70.

Tolety, V. B. P., & Prasad, E. V. (2022). Hybrid content and collaborative filtering-based recommendation system for e-learning platforms. *Bulletin of Electrical Engineering and Informatics*, 11(3), 1543–1549.

Watson, R., & Watson, S. (2012). An argument for clarity: What are learning management systems, what are they not, and what should they become? *TechTrends*, 51(2), 28–34.

Witten, I. H., Bainbridge, D., & Nichols, D. (2009). *How to build a digital library* (2nd ed.). Morgan Kaufman. ISBN 9780080890395

Zhang, L., & Rangwala, H. (2022). Deep Learning-Based Personalized Course Recommendation in MOOCs. Journal of Educational Data Mining, 14(1), 45-67.

Zhang, Y., Qian, Y., Gan, M., Tang, X., & Lin, Z. (2019). Service recommendation based on user dynamic preference extraction and prediction. *Proceedings of the 2019 IEEE World Congress on Services (SERVICES)*, 8–13 July 2019 (Vol. 2642, pp. 121–126). IEEE.