Hybridizing Collaborative Filtering and Knowledge: How do they Work Together? A Scoping Review

Alex Martínez-Martínez, Raul Montoliu, Inmaculada Remolar
Institute of New Imaging Technologies, Universitat Jaume, I Castellón de la Plana, Spain, 12071

Abstract—**The rapid expansion of digital platforms and the increasing complexity of user preferences have driven the need for more sophisticated recommendation systems. While Collaborative Filtering and Knowledge-Based Filtering have been widely adopted as core techniques for personalized recommendations, their individual limitations have led to the rise of hybrid approaches. Despite significant advancements, a comprehensive understanding of hybridization methodologies, their technical implementations, and emerging challenges remains unsolved. The purpose of this research is to systematically examine and synthesize the domain of Hybrid Recommender Systems to address this. This study presents a scoping review, following the PRISMA-ScR guidelines, to systematically examine the domain of hybridizing Collaborative Filtering and Knowledge-Based Filtering. A total of 62 hybrid recommenders across various application domains were analyzed, and categorized into three primary hybridization strategies: Model Fusion, Transfer Learning, and Hierarchical Models. The review explores technical characteristics, hybridization techniques, data sources, evaluation methodologies, and domain-specific applications. Key findings indicate that most hybrid approaches focus on leveraging graph-based models, deep learning architectures, and causal inference techniques to enhance recommendation outcomes. However, despite these advancements, critical gaps remain. The review identifies key challenges, including computational complexity, lack of explainability, bias in recommendations, and reliance on offline evaluation metrics. Additionally, scalability issues in knowledge graph maintenance and the need for user-centered evaluation frameworks highlight important directions for future research. Addressing these gaps will be crucial in making hybrid recommendation systems more efficient, interpretable, and adaptable across diverse domains. This study contributes to the field by providing a structured synthesis of existing hybridization techniques, pinpointing success factors, and proposing future research avenues to advance hybrid recommendation systems.**

Keywords*—*Hybrid Recommender Systems; collaborative filtering; knowledge-based recommenders; personalized recommendations; scoping review

I. INTRODUCTION

The rapid expansion of digital platforms and the pervasive availability of online services have transformed how users access information, enabling quick and efficient retrieval of vast datasets. However, this abundance, combined with its complexity and diversity, often overwhelms users, limiting their ability to process information and make decisions effectively [1]. To address this challenge, the development of advanced Recommendation Systems (RSs) has become a critical point. These systems act as essential filtering tools, delivering personalized content precisely when users need it, across a wide range of domains, such as what movie to see, what book to read and many others [2]. Research in this area has led to the emergence of various recommendation techniques, such as Collaborative Filtering (CF) [3], Knowledge-Based Filtering (KBF) [4], and others [5], [6], [7], [8].

RSs are transforming the generation of personalized suggestions through advanced machine learning and deep learning techniques [9]. By analyzing vast amounts of real-time data, these systems effectively identify user preferences, behaviors, and patterns, enabling them to deliver highly relevant and efficient recommendations. This level of personalization enhances user satisfaction and engagement by ensuring that suggestions are tailored to individual needs and remain accurate as preferences evolve. With their ability to continuously learn and adapt, these RSs have become indispensable in today's data-driven landscape.

CF and KBF are two of the main approaches used in RSs, each offering unique strengths and addressing different aspects of recommendation tasks. CF techniques select recommended items based on the user's past interactions with other items. These methods are broadly categorized into memory-based and model-based recommenders [10]. Memory-based approaches rely on the entire interaction history, generating recommendations by assessing similarities between users or items [11]. In contrast, model-based approaches build predictive models that uncover underlying patterns in the data [12]. KBF approaches leverage domain knowledge about items, as well as potential user preferences or requirements, to make personalized suggestions [13]. A well-known example of this type of recommender is the ontology-based system, which creates formal representations of knowledge about items, users, and their relationships [14]. In 2012, Google introduced the concept of the Knowledge Graph, a structured representation that organizes entities, their attributes, and the relationships between them, enabling meaningful connections and insights [15].

In today's industrial landscape, leading companies have demonstrated the strategic value of these systems. For example, Netflix makes extensive use of Collaborative Filtering to suggest content based on viewing patterns of users with similar profiles, with a large proportion of its consumption coming from personalized recommendations [16]. On the other hand, platforms such as Amazon have integrated knowledge graphs to enrich the shopping experience, allowing the system to understand the semantic relationships between products and brands, which significantly improves accuracy in domains with massive inventories [17]. Similarly, in the music sector, Spotify employs hybrid approaches that combine user behavior analysis (CF) with metadata and genre tags (KBF) to create playlists such as "Weekly Discovery", mitigating problems of overspecialization and encouraging the discovery of new artists [18].

Despite their widespread use, CF methods still face notable challenges, including the cold-start problem and data sparsity, which limit their effectiveness in certain scenarios [19]. In a similar vein, KBF also faces challenges such as knowledge acquisition and the difficulty of completing knowledge representations [20]. Hybrid Recommender Systems (HRSs) combine two or more recommendation techniques to improve their performance and overcome their single limitations. Previous reviews in this area have highlighted the benefits of combining different recommendation techniques [21], [22], [23]. However, these reviews generally focused on early studies exploring how multiple recommendation strategies can be effectively combined. Many of these reviews only cover research up to 2017, and recent developments in the field have introduced new insights that require further analysis. Other similar reviews in the field also tend to concentrate their analysis on specific areas of research, as well as, covering non HRSs, which may result in not covering the complete set of current HRSs [24], [25], [26].

Recent advancements in recommendation technologies present exciting opportunities for improving hybrid systems, yet they have not been included in earlier reviews. This scoping review addresses this gap by systematically exploring recent literature on HRSs that combine CF with KBF approaches. The articles considered for this review are up to October 2024, covering an important amount of articles that were not considered in previous research. With this, this scoping review aims to analyze the current state of hybridizing CF and KBF, focusing on how both techniques together seek to improve the results of the HRSs. To achieve this, the following research questions have been formulated:

- RQ1: What methodologies are currently used to hybridize collaborative filtering and knowledge-based recommendation systems?
- RQ2: What common challenges arise when integrating collaborative filtering with knowledge-based techniques?
- RQ3: How do hybrid recommendation systems address data sparsity and cold start issues across different domains and applications?
- RQ4: How effective are hybrid approaches in improving recommendation accuracy compared to standalone collaborative filtering or knowledge-based systems?

The main contribution of this work lies in its comprehensive analysis of recent advancements in HRSs that integrate CF and KBF approaches. By addressing the limitations of prior reviews, this scoping review provides an updated perspective on how explicit knowledge can be effectively incorporated into CF to enhance recommendation performance. Additionally, the review offers insights into the comparative effectiveness of hybrid approaches in improving recommendation accuracy, filling a critical gap in the literature, and guiding future research in this evolving field.

The rest of the study is organized as follows: Section II provides an overview of CF and KBF, outlining their current state and the most common challenges they face. Section III details the methodology used to gather and select the literature analyzed in this review. Section IV presents a comprehensive analysis of the selected articles, highlighting key findings and trends. Section V discusses the research questions, synthesizes the insights obtained from the analysis, identifies existing gaps in the literature, and suggests future research directions. Finally, Section VI concludes with a summary of key takeaways and final remarks.

II. BACKGROUND

This section explores how collaborative filtering and knowledge-based recommender systems are built, elucidating their underlying mechanisms, the prevalent challenges encountered during their development, and how they are evaluated.

A. Collaborative Filtering

CF is one of the most widely used approaches for building RSs [27]. Using past users' interactions, these systems can predict their future preferences or interests. Typically, this information is organized into a user-item matrix, denoted by M. In this matrix, the rows represent users of the set U, while the columns correspond to items of the set V, with the matrix having dimensions $W_u \times W_v$. Eq. (1) and Eq. (2) correspond to the definition of the user and item set, respectively.

$$U \mid u_i \in U \;\; \forall \;\; i \in [1, \ldots, W_u] \tag{1}$$

$$V \mid v_j \in V \;\; \forall \;\; j \in [1, \ldots, W_v] \tag{2}$$

Each cell of the matrix M corresponds to a user's rating or interaction with an item. Specifically, each element m_{ij} represents the rating given by user u_i to item v_j. These ratings can usually be represented on a Likert scale (e.g., ranging from 1 to 5) or as binary values indicating interaction, where $m_{ij} = 1$ denotes an interaction and $m_{ij} = 0$ indicates no interaction.

This matrix allows the system to compute the similarity between users (rows) or items (columns), based on the ratings it collects. These two possible similarities give rise to the two major versions of CF, which are User-Based Filtering (UBF) and Item-Based Filtering (IBF). UBF focuses on finding users similar to a specific user to recommend items that this set of similar users has found interesting. IBF identifies items similar to those with which the target user has previously interacted to recommend new items that they may find interesting.

To determine whether two users or items are similar or not, it is necessary to implement a metric that is capable of evaluating the distance between two or more users or items. Distance and similarity can be seen as opposites, so if two users have a very large distance D_u their similarity S_u will be very small, so we can say that they are very different users. In contrast, if this distance is very small, their similarity will be very large, so we can say that they are very similar users. This can also be applied to the items, where their distance is defined by D_v and their similarity by S_v. Although typically ranging between 0 and 1, the actual values of these distances may vary depending on the specific metrics used. Eq. (3) showcases the definition of the distance between users, where D_u represents the distance between the two users u_i and u_k. Eq. (4) presents the definition of the distance between items, where D_v represents the distance between the two items v_j and v_k.

$$D_u : (u_i, u_k) \to \mathbb{R}; \forall u_i, u_k \in U \tag{3}$$

$$D_v : (v_j, v_k) \to \mathbb{R}; \forall v_j, v_k \in V \tag{4}$$

There are various metrics for evaluating distances or similarities between users, with two of the most common being Euclidean Distance (ED) and Cosine Similarity (CS). The ED between two users u_1 and u_2 is defined in Eq. (5), where $m_{u_1 j}$ and $m_{u_2 j}$ represent the ratings of the $j-th$ feature of the two users being compared. This metric measures the geometric distance between users' rating vectors, yielding values ranging from 0 to 1.

$$ED(u_1, u_2) = \sqrt{\sum_{j=1}^{W_v} (m_{u_1 j} - m_{u_2 j})^2} \tag{5}$$

The CS to compare between u_1 and u_2 is defined in Eq. (6), where u_1 and u_2 represent the feature vectors of two users, corresponding to their respective rows in the rating matrix, the symbol $\cdot$ denotes the dot product between the feature vectors, $\parallel u_1 \parallel$ and $\parallel u_1 \parallel$ represent the norms (magnitudes) of the vectors and $\times$ symbol indicates the scalar multiplication of these two norms. This metric evaluates the cosine of the angle between the two vectors, producing similarity scores between -1 and 1.

$$CS(u_1, u_2) = \frac{u_1 \cdot u_2}{\parallel u_1 \parallel \times \parallel u_2 \parallel} \tag{6}$$

Building on the example of users, once similarities between all pairs of users have been calculated, the system identifies a subset of users most similar to the target user. This subset is often referred to as the *neighborhood* and is defined in Eq. (7), where θ is a threshold that determines the minimum similarity required for a user to be included in the neighborhood. Here, N_{u_i} represents the neighborhood of user u_i, and u_k is another user from the set U.

$$N_{u_i} = \{u_k \in U \mid u_k \neq u_i, S_u(u_i, u_k) \geq \theta\} \tag{7}$$

Using this neighborhood, the system predicts how the target user would rate items they have not yet interacted with. This prediction is typically computed as a weighted average of the ratings within the neighborhood, where the similarity between users determines the weights. The predicted rating for user u_i on item v_j is defined in Eq. (8), where N_{u_i} represents the neighborhood of user u_i, $\hat{m}_{ij}$ represents the predicted rating of item v_j for the user u_i and m_{kj} represents the rating of item v_j from user u_k. Finally, based on the predicted ratings, the system recommends items with the highest predicted scores to the user.

$$\hat{m}_{ij} = \frac{\sum_{u_k \in N_{u_i}} S_u(u_k, u_i) \times m_{kj}}{\sum_{u_k \in N_{u_i}} \mid S_u(u_k, u_i) \mid} \tag{8}$$

In CF, an important concept to highlight is matrix factorization (MF) [28]. This technique is used to analyze and predict user-item interactions by decomposing a large user-item interaction matrix into two lower-dimensional matrices, F_u and F_v. These matrices typically represent the set of users U and the set of items V, where each user and item is described by a set of latent factors of dimension W_l. These latent factors capture underlying features or characteristics of users and items. For instance, in a movie recommendation system, latent factors might represent genres (e.g., action, comedy) or other attributes that describe user preferences and item characteristics. By identifying these patterns, MF helps uncover the relationships between users and items even in sparse datasets.

Once the matrices F_u and F_v are computed using techniques such as Singular Value Decomposition (SVD) [29], the system predicts missing ratings by taking the dot product of corresponding latent factors from F_u and F_v. The predicted rating is calculated as showed in Eq. (9), where $\hat{m}_{ij}$ is the predicted rating for user u_i on item v_j, W_l is the number of latent factors, and u_{il} and v_{jl} are the $l-th$ latent factor of user u_i and the item v_j in the matrices F_u and F_v, respectively. MF thus enables the system to predict missing interactions efficiently by leveraging the latent relationships between users and items.

$$\hat{m}_{ij} = \sum_{l=1}^{W_l} u_{il} \cdot v_{jl} \tag{9}$$

B. Knowledge-Based Filtering

KBF leverages structured knowledge about a specific domain to generate recommendations [30]. The process begins with acquiring domain-specific knowledge from a variety of sources, such as domain experts, existing literature, databases, or other structured information repositories. Once this knowledge is gathered, it must be organized into a structured format that the RS can interpret. Common approaches to represent such knowledge include ontologies and Knowledge Graphs (KGs), which effectively capture the domain's concepts, entities, relationships, rules, and constraints. These structured representations provide the foundation for reasoning and inference, enabling the system to offer tailored recommendations based on the domain's intrinsic properties and user preferences.

On one hand, an ontology offers a structured framework for organizing and representing knowledge within a specific domain. Fig. 1 illustrates an example of an ontology in the movie domain. The key components of an ontology include:

- Entities are the fundamental objects, concepts, or items that exist within the domain, such as a Person or a Movie.
- Relationships define the connections between entities, specifying how they interact or are associated within the domain. For example, a Person might have a direct relationship with a Movie.
- Constraints impose rules or restrictions on relationships and entity properties to ensure consistency and

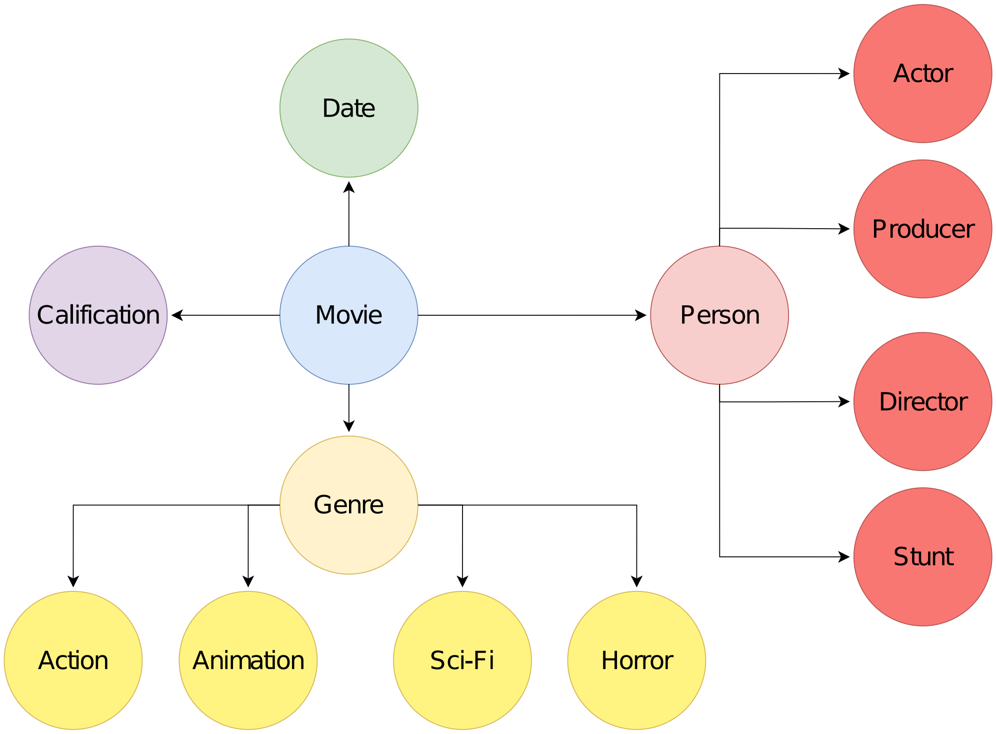


Fig. 1. Example ontology of the movies domain.

integrity within the ontology. These rules might specify, for instance, that a Person can only direct a Movie if they belong to the Director category.

Ontologies are often represented using formal languages like the Resource Description Framework (RDF) or the Web Ontology Language (OWL), which enable precise modeling and facilitate reasoning over the domain knowledge [31].

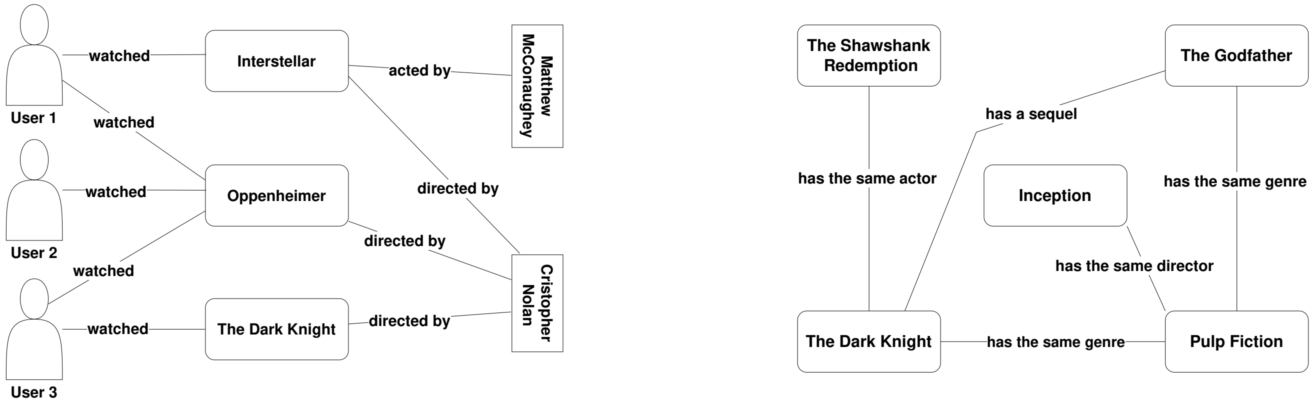
On the other hand, a KG is a structured representation of knowledge that organizes information into a graph format, where entities are depicted as nodes and the relationships between them as edges connecting these nodes [15]. This format enables the integration and connection of information in a way that is both human-readable and machine-understandable. Fig. 2 illustrates two examples of different types of KGs, the Item Knowledge Graph (IKG) and the User-Item Knowledge Graph (UIKG). It is important to note that the UIKG can be used as the source of both, CF and KBF. These graphs are typically represented as a collection of triples, with each triple defined as (h, r, t). In this representation:

- The elements h (head) and t (tail) represent the entities.
- The element r (relation) denotes the relationship connecting the entities.

For example, a triple like (Interstellar, Directed By, Christopher Nolan) captures the relationship between the movie Interstellar and its director, Christopher Nolan. Knowledge graphs thus provide a flexible and powerful framework for structuring and querying domain-specific information.

Once the source data is defined, it must be transformed into a format the recommendation algorithm can process. This transformation can be achieved through various techniques, such as feature extraction or embedding mechanisms. For instance, an embedding mechanism maps the source data into numerical vectors that capture the semantic meanings and relationships within the data. Fig. 3 provides an example of how an embedding operates for an entity in a knowledge graph using a neural network-based approach. The resulting numerical vectors represent the items, enabling the calculation of distances D_v between any pair of items. These distances serve as the foundation for the recommender system to compute similarities between entities and generate personalized recommendations for users. It is important to note that not all embedding methods function identically. Their underlying mechanisms, training objectives, and mathematical operations can vary widely, leading to differences in how effectively they capture and represent the relationships in the data.

The training process is a fundamental aspect of embedding methods, particularly in scenarios where the objective is to learn a feature space that clusters similar items together while separating dissimilar items. One example of this is the Contrastive Loss (CL) training approach [32]. CL evaluates how effectively an embedding model distinguishes between similar and dissimilar pairs of data points and is defined in Eq. (10), where x_1 and x_2 are the two data points being compared, such as two triples from a KG, y indicates whether the pair is similar $(y = 1)$ or dissimilar $(y = 0)$, D_x is the computed distance between x_1 and x_2, and ϵ is a margin parameter that specifies the minimum separation required between dissimilar pairs.



(a) User item knowledge graph.

(b) Item knowledge graph.

Fig. 2. Example knowledge graphs of the movies domain. Fig. 2a represents a complete user-item interaction knowledge graph. Fig. 2b represents an item knowledge graph managing the relations between the entities.

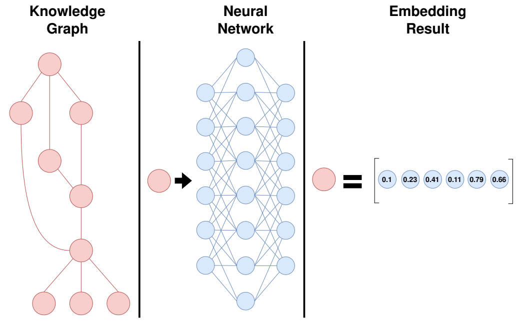


Fig. 3. Process of embedding using a neural network. Each node of the knowledge graph serves as an input of the neural network. The output of the NN is the embedding of the selected item of the KG.

$$CL(x_1, x_2, y) = y \cdot \frac{1}{2}D_x^2 + (1-y) \cdot \frac{1}{2}max(0, \epsilon - D_x)^2 \quad (10)$$

CL plays a vital role in training embedding models as it directly optimizes the spatial arrangement of data points in the embedding space. By adjusting the distances between pairs based on their similarity, it helps create a meaningful feature space where similar items are tightly clustered and dissimilar items are well-separated. This makes the method highly effective for tasks that demand robust and discriminative feature representations.

C. Common Challenges

Despite their significant advantages in efficiently managing the vast amount of information available, RSs still face notable challenges that limit their effectiveness [33]. Two of the most prominent issues are:

1) Cold start problem: This challenge arises when there is insufficient historical data about either users or items [34]. For new users, the system lacks insights into their preferences or past behavior, making it difficult to provide accurate, personalized recommendations. Similarly, new items introduced into the system suffer from a lack of interaction data, which prevents the system from effectively incorporating them into recommendations. This limits the ability of the recommender to adapt quickly to dynamic changes in the user base or item catalog. Although much research has focused on trying to overcome this limitation, there is still no clear solution to this problem [35], [36], [37].

2) Data sparsity: This occurs when user-item interaction data is limited or incomplete, which is a common scenario in systems with large user and item sets [38]. Sparse data restricts the system's ability to identify meaningful patterns or similarities between users and items, thereby impairing its capacity to make personalized and accurate recommendations. This often results in less relevant suggestions for users and diminishes the overall performance and reliability of the recommender system. As with cold start, there are many approaches to solving the data sparsity problem but none have been able to find a suitable solution to overcome this limitation [39], [40].

Addressing these challenges is critical for enhancing the adaptability and robustness of recommender systems in real-world applications.

D. Hybridization

HRSs emerge as a solution to address the inherent limitations of individual recommendation methods. By integrating multiple approaches, these systems aim to enhance the accuracy and diversity of recommendations, providing users with a more personalized experience. By combining multiple recommendation methods, HRSs can:

- Mitigate Limitations: Overcome the shortcomings of individual approaches, such as the cold start problem and data sparsity.
- Enhance Accuracy: By considering various information sources, recommendations better align with user preferences and needs.

- Increase Diversity: Offer a broader range of recommended items, preventing over-specialization and promoting the discovery of new content.

The choice of hybridization strategy depends on the application domain, the nature of the available data, and the specific objectives of the recommender system. Studies have shown that hybrid systems often outperform individual methods in terms of accuracy and user satisfaction by leveraging the complementary strengths of each approach [41]. Several strategies exist to combine recommendation methods, from which some of the most common are:

- Weighted Hybridization: This method combines the scores of different recommendation techniques by assigning specific weights to each. For instance, a system might allocate 70% weight to CF and 30% to KBF, adjusting these values based on observed effectiveness.
- Switching Hybridization: In this approach, the system dynamically selects the most appropriate recommendation technique based on the current context or data availability. For example, it might use KBF for new users with no interaction history and switch to CF as more data becomes available.
- Cascade Hybridization: This strategy applies one recommendation method first and then refines its results using another. For example, the system might initially use KBF to generate a list of relevant items and then apply CF to further rank or filter that list.

E. Evaluation Method and Metrics

Evaluating RSs is a critical step to ensure they provide accurate and relevant recommendations, ultimately influencing user satisfaction and engagement across various domains [42]. A robust evaluation not only highlights the system's strengths and weaknesses but also informs optimization efforts to enhance performance. Without proper evaluation, these systems risk delivering unsatisfactory recommendations, potentially leading to user frustration, reduced engagement, and more.

The most widely used evaluation methods for recommender systems are listed as follows:

- Offline evaluation: This evaluation method relies on historical data to assess the system's performance. Common metrics include precision, recall, mean absolute error (MAE), and normalized discounted cumulative gain (NDCG). These metrics are defined at the end of this section. They are often reported as $metric@N$, which refers to an evaluation of the performance of the recommender system based on the top-N items in the recommendation list, where N represents the number of recommendations considered for evaluation. Although offline evaluation is efficient and repeatable, it may not always reflect user behavior in real-world scenarios.
- Online evaluation: In this evaluation approach, the recommender system is deployed in a live environment, and performance is measured based on real-time user interactions and feedback. This method provides dynamic and realistic insights but requires careful experimentation, such as A/B testing, a controlled experiment used to compare two recommendation algorithms by splitting users into groups and exposing each group to a different version, to minimize the risk of negative user experiences during the process.
- User studies: This procedure gathers direct feedback from users through surveys, interviews, or usability tests. These studies assess subjective factors like user satisfaction, perceived relevance, and the overall usefulness of recommendations. While time-consuming and resource-intensive, user studies offer valuable qualitative insights that complement the quantitative data from offline and online evaluations.

Evaluation metrics are essential tools for assessing and optimizing the performance of RSs. They provide a structured way to measure how well the system meets its objectives, whether it's delivering accurate recommendations, covering a broad range of relevant items, or ranking recommendations effectively. Each metric focuses on a distinct aspect of performance, helping to identify specific strengths and weaknesses in the system.

- Precision: Ensures the system is making accurate recommendations, showing the proportion of suggested items that are genuinely relevant to the user.
- Recall: Complements precision by focusing on coverage, evaluating how well the system retrieves all relevant items.
- MAE: Captures how closely the system's predicted ratings match the actual ratings, providing a clear measure of predictive accuracy that supports trust in the system.
- NDCG: Goes a step further by considering both relevance and ranking quality, ensuring that the most relevant items appear at the top of the recommendation list, where users are most likely to notice them.

Precision and recall evaluate the relevance and effectiveness of the recommendations made by the system. Eq. (11) defines these metrics, where Q represent the set of relevant items, and R represents the set of the recommended items:

$$Precision = \frac{\mid Q \bigcap R \mid}{\mid R \mid} \quad Recall = \frac{\mid Q \bigcap R \mid}{\mid Q \mid} \tag{11}$$

These sets are defined in Eq. (12), where $\mid R \mid$ and $\mid Q \mid$ represent the size of both sets of items. A relevant item is typically defined as an item that is of interest or value to the user, usually measured by the user's previous interactions with the item.

$$Q \subseteq V \mid \forall q \in Q, q \in V \quad R \subseteq V \mid \forall r \in R, r \in V \tag{12}$$

MAE evaluates the difference between predicted ratings and actual ratings. Eq. (13) defines MAE for a given user u_i, where $\hat{m}_{u_i j}$ is the predicted rating of item j, $m_{u_i j}$ is the actual rating of item j, and N_v is the total number of items rated by the user:

$$MAE = \frac{\sum_{j=1}^{N_v} \hat{m}_{u_i j} - m_{u_i j}}{N_v} \tag{13}$$

NDCG evaluates the quality of the ranking of the recommended items. Eq. (14) presents the definition of the DCG, where rel_j is the relevance of item j, and W_v represents the number of items in the ranked list that are considered when calculating the score.

$$DCG = \sum_{j=1}^{W_v} \frac{rel_j}{\log_2(j+1)} \tag{14}$$

The value of rel_j is typically user-defined; for instance, it could be 1 if the item is considered relevant and 0 if it is not relevant. The NDCG is obtained by normalizing the DCG score by the ideal DCG score of the best possible permutation of items.

By using these evaluation metrics, developers can track the effectiveness of RSs and fine-tune them for better accuracy, coverage, ranking, and predictive power. Proper evaluation ensures the system not only provides accurate recommendations but also enhances user satisfaction, engagement, and platform success.

III. METHODS

This scoping review investigated the existing literature on the hybridization of CF and KBF recommenders. The authors followed the guidelines of Preferred Reporting Items for Systematic Review and Meta-Analysis- Extension for Scoping Reviews (PRISMA-ScR) [43] to conduct this review. The protocol followed for this review was registered with the Open Source Foundation [44].

A. Search Strategy

To cover the maximum number of relevant articles, the SCOPUS, Web of Science (all databases), and ACM Library (The ACM Guide to Computing Literature) databases were used to retrieve articles. The search was conducted on October 8th, 2024. For this review, a combination of recommendation engine-related search terms was used. The search terms combined the three main aspects of this review: collaborative filtering, knowledge-based filtering, and recommender systems. The formulation of the search query was informed by an examination of prior reviews about collaborative and knowledge-based RSs [45], [46]. The created common search query is as follows: (“ontology” OR “knowledge” OR “knowledge-based” OR “ontology-based”) AND (“recommender” OR “recommendation” AND “recommend”) AND (“collaborative” AND (“filter” OR “filtering”)). All specific search queries for the databases are included in Appendix A.

B. Eligibility Criteria

To be included in the scoping review, papers had to meet the following criteria: they must be written in English, available as Open Access, and present a novel development of a RS rather than a review or similar study. Additionally, they must have been published between 2012 and October 8, 2024. The review specifically included papers that focused on developing a HRS by combining only CF and KBF. Eligible papers had to clearly describe the functioning of the recommendation engine and introduce an innovation in either the technologies used for any of the recommenders or auxiliary techniques aimed at improving recommendations. Papers were excluded if they represented a second version of a previously published recommender or incorporated any additional recommendation approach beyond CF and KBF.

C. Study Selection

The study selection process was conducted in two phases to ensure accuracy and thoroughness. In the first phase, articles were screened based on their titles and abstracts. Any disagreements were resolved through discussion until a consensus was reached. If no agreement was found, a third reviewer was consulted to make the final decision. In the second phase, the articles that passed the initial screening underwent a full review. Each article was evaluated independently against the eligibility criteria, with reasons for exclusion carefully documented. Any remaining discrepancies were addressed through further discussion, involving a third reviewer when necessary.

D. Data Extraction and Analysis

For the selected studies, four key characteristics were extracted for analysis. These are as follows:

- Recommender Domain: The specific domain where the recommender is applied (e.g., movies, education, locations).
- Data Sources: The types of data used in the recommendation process (e.g., ratings, user profiles, graphs).
- Evaluation Methods: The methods used to assess the performance of the recommendation system (e.g., accuracy, likelihood).
- Recommender Techniques: A description of the techniques used in the recommender system, including any additional methods employed to enhance the recommendation process.

Following this, the studies were grouped into categories based on their similarities, and an analysis was conducted to explore recurring themes and relationships among the different characteristics of hybrid recommendation systems. This analysis aimed to identify current gaps, limitations, and potential areas for future research and innovation in the field of Hybrid Recommender Systems.

IV. RESULTS

Fig. 4 illustrates the complete search process conducted for this scoping review. Initially, 690 records were retrieved from the selected databases, and after removing duplicates,

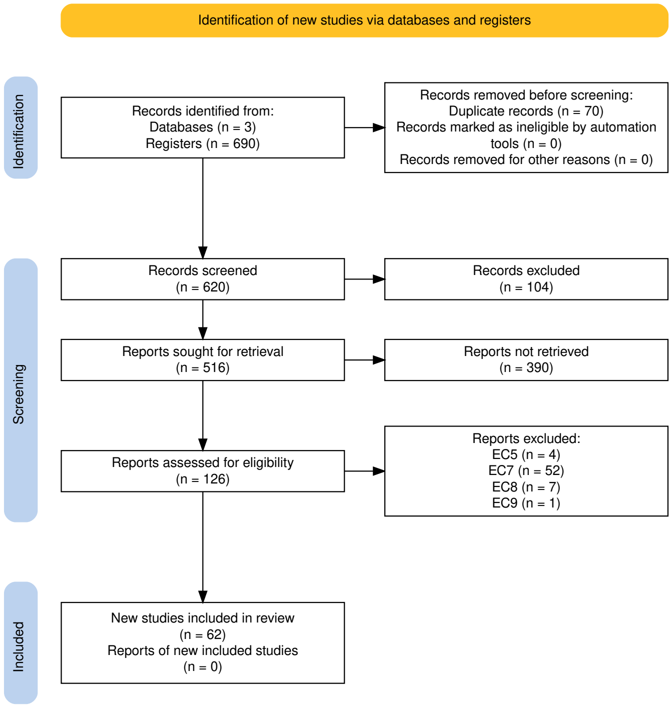


Fig. 4. PRISMA 2020 flow diagram for scoping reviews, which included searches of databases and registers only.

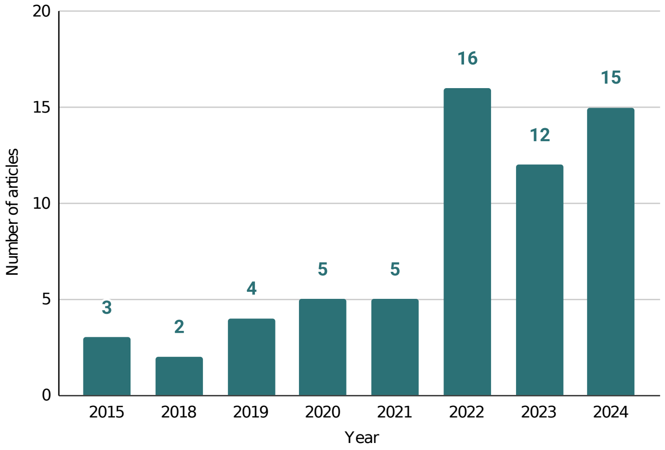


Fig. 5. Number of fully read articles per year.

only journal articles were retained for further analysis. Subsequently, a screening process based on abstracts and titles narrowed down the selection to 126 articles for full-text review. Following thorough examination, 62 articles were deemed suitable for detailed analysis. The publication dates of these articles are depicted in Fig. 5, revealing that the majority of the included articles (69.35%) were published between 2022 and 2024. Notably, no articles from 2012, 2013, 2014, 2016, and 2017 were included in the final analysis. The complete version of the data extraction process is found in Supplementary Material (Appendix).

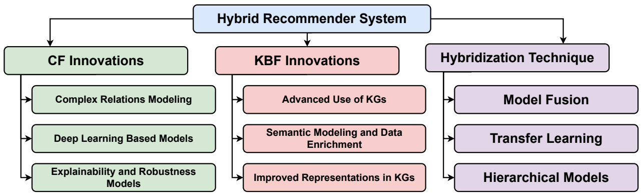


Fig. 6. Classification framework of the revised articles.

TABLE I. CLASSIFICATION OF THE REVIEWED ARTICLES BASED ON THEIR CF INNOVATION

Innovation Classification	Citation
Complex Relations Modeling	[47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77]
Deep Learning Based Models	[78], [79], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100], [101]
Explainability and Robustness Models	[102], [103], [104], [105], [106], [107], [108]

A. *Article Classification*

After carefully reading and extracting the information from all the articles, the developed systems were classified based on how the articles were innovative with respect to both CF and KBF. The classification framework can be seen in Fig. 6.

Innovation in HRS has led to three major approaches in both CF and KBF. In CF, advancements have enhanced the representation of user-item interactions, integrated deep learning models, and ensured more explainable and robust recommendations. Meanwhile, in KBF, innovation has optimized the use of KGs, enriched the semantic representation of data, and improved the efficiency of embeddings and graph-based representations. Besides this, the included articles have been also classified depending on how they have been hybridized.

Table I depicts the classification of all included articles based on their CF innovation. The three categories of this classification are as follows:

- Complex relationship modeling: Improves the representation of user-item interactions by incorporating semantic information, graph networks, and contextual cues.
- Deep learning-based models: Applies deep neural networks (CNN, RNN, GCN, GAT) to capture complex patterns in recommendation data.
- Explainability and robustness models: Ensures more interpretable and reliable recommendations through causal inference, fuzzy reasoning (fuzzy reasoning, path-based), and privacy.

Table II shows the classification of all included articles based on their KBF innovation. The three categories of this classification are as follows:

TABLE II. CLASSIFICATION OF THE REVIEWED ARTICLES BASED ON THEIR KBF INNOVATION

Innovation Classification	Citation
Advance Use of KGs	[51], [56], [57], [58], [59], [60], [61], [80], [67], [89], [92], [71], [93], [94], [95], [97], [77], [108]
Semantic Modeling and Data Enrichment	[47], [48], [49], [78], [50], [52], [53], [54], [62], [63], [64], [65], [66], [88], [68], [69], [70], [72], [73], [74], [75], [76]
Improved Representations in KGs	[102], [103], [55], [79], [81], [82], [83], [84], [85], [86], [87], [90], [104], [91], [105], [106], [96], [107], [98], [99], [100], [101]

TABLE III. CLASSIFICATION OF THE REVIEWED ARTICLES BASED ON THEIR HYBRIDIZATION STRATEGY

Hybridization Strategy	Citation
Model Fusion	[47], [48], [49], [102], [50], [51], [52], [54], [103], [55], [56], [58], [79], [59], [61], [62], [63], [65], [81], [82], [83], [84], [66], [67], [86], [87], [88], [68], [69], [89], [90], [105], [92], [71], [72], [106], [73], [74], [75], [94], [95], [96], [98], [99], [77], [100], [108]
Transfer Learning	[78], [53], [57], [60], [64], [104], [91], [70], [76], [107]
Hierarchical Models	[80], [85], [93], [97], [101]

- Advanced Use of KGs: Exploits KGs with multi-hop embeddings, deep learning, and complex relationship modeling to improve recommendation.
- Semantic Modeling and Data Enrichment: Uses ontologies, pattern mining, and semantic rules to represent concepts and improve the personalization of recommendations.
- Improved Representations in KGs: Optimizes the representation and processing of knowledge graphs through advanced embeddings, hyperbolic models, and noise reduction.

Table III shows the classification of all included articles based on their hybridization strategy. Three major hybridization techniques have been obtained, which are as follows:

- Model Fusion: CF and KBF models work separately to generate predictions and are then combined into a single output.
- Transfer Learning: A model learns information in one domain or technique (e.g. KBF) and transfers that knowledge to improve another model (e.g. CF).
- Hierarchical Models: CF and KBF operate at different levels of a multi-level or deep learning model, rather than being directly combined in a single stage.

B. Recommender Domain

In this review, the categorization of RSs was based on explicit mentions of their respective domains. Articles mentioning a specific domain were classified accordingly, while those lacking such specificity were considered applicable to any knowledge domain. It is important to note that despite saying any domain, it would be necessary to have data characteristics required by the systems that may not be available in all domains. Following the review process, it was found that many articles did not specify a particular domain (33). Among those that did, the education field emerged as the second most common domain (9). Tourism occupies the third position of the most treated domains (4). Other domains represented include entertainment, composed of music and movies domain (3), e-commerce (3), cloud services (3) , micro-blogging (1), manufacturing (1), healthcare (1), and additional categories. Fig. 7 shows a graphical representation of the recommender domains, a detailed list of the domains with their corresponding articles can be found in Table IV of Appendix B.

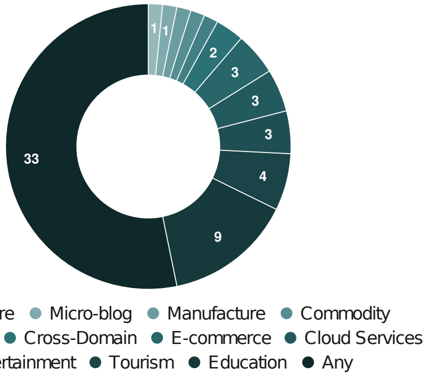


Fig. 7. Number of articles of each domain.

C. Data Sources

The foundation of this review lies in CF and KBF. Deepening into their development and combination underscores the pivotal role of data in shaping both filtering methodologies.

The data used for CF in the examined systems primarily involves user-item interactions, represented in various forms. These range from intricate user-item graph structures (9) to comprehensive item rating matrices (41), the latter being the most commonly used approaches in the current state-of-the-art. These two data sources were explained in Section II. Additionally, some articles incorporate other sources of information, such as knowledge-based results (7), Quality of Service matrices (1), or Average trustworthiness scores (1). Fig. 8 provides a graphical representation of the data sources utilized for CF.

Among the various data sources for KBF, two source types dominate among the others. These data sources are KGs (38) and ontologies (17), which were introduced in Section II, accounting for 83% of the reviewed articles. Additional data sources include knowledge maps (2), CF results (2), and linked open data (1). Fig. 9 illustrates a graphical representation of the data sources utilized for knowledge-based filtering.

D. Evaluation Methods

Evaluating the effectiveness of the proposed recommendation engine is a critical component of this review. The studies

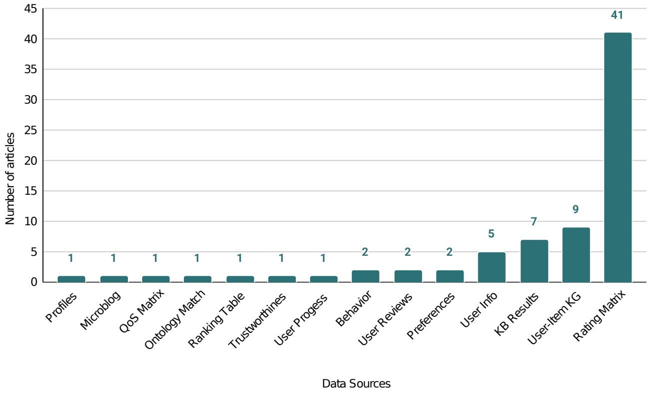


Fig. 8. Number of times each collaborative data source was used.

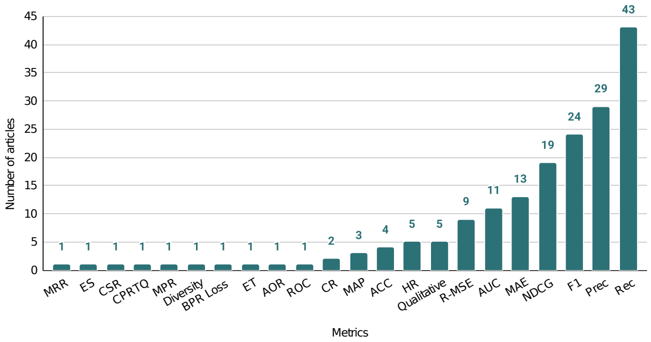


Fig. 10. Number of times each metric was employed.

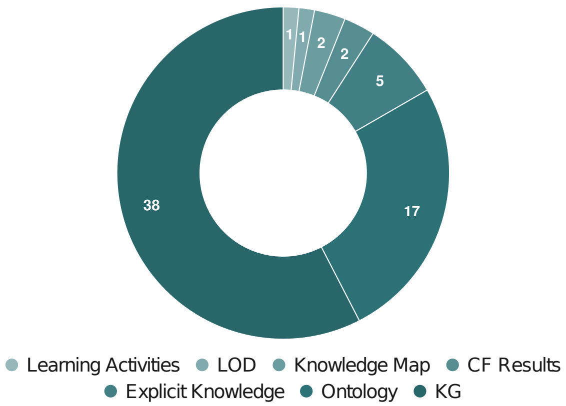


Fig. 9. Number of times each knowledge data source was used.

analyzed employed a variety of metrics to assess system performance thoroughly. Most studies used between one and four quantitative metrics (60), including measures such as MAE (13), Root Mean Square Error (RMSE) (9), Precision (29), Recall (43), F1-Score (23), and NDCG (20), among others. Additionally, some articles incorporated qualitative metrics (5), focusing on aspects like user satisfaction (4) and recommendation effect analysis (1). It is worth noting that [68] did not specify the evaluation metrics used, so only the remaining 61 articles were included in this analysis. Fig. 10 illustrates the frequency of metric usage across all articles, while a detailed breakdown of these metrics and their sources can be found in Table V to Table VII of Appendix C.

E. Recommender Techniques

This section examines the implementation of CF and KBF, focusing on the different implemented techniques, the side techniques used to improve the recommender accuracy, and how all these approaches are joined together to build HRS.

1) Implementation of collaborative filtering: As previously noted, CF is a central focus of this review. A detailed analysis of its implementation highlights the importance of innovative recommendation techniques. The implementation of CF has centered on three critical aspects, which are, 1) modeling complex relations, 2) using deep learning techniques, and 3) seeking explainability and robustness for the recommendations.

Complex relationship modeling leverages graph structures, proximity networks, and semantic modeling to overcome the limitations of similarity calculations between users or items. These approaches capture implicit relationships, use knowledge networks to enhance predictions, and adapt better to data-scarce environments. A notable example is [56], where the authors introduce a neighbor aggregation network that expands user-item interactions using knowledge graphs. Instead of relying solely on direct similarities, this approach utilizes multi-hop relationships within the graph to improve recommendations. This results in more accurate recommendations in scenarios with sparse data, expanding the similarity space and mitigating the data sparsity problem.

Another interesting example is [67], which introduces a method of modeling tourism relationships based on KGs, allowing the identification of implicit connections between users and destinations. Rather than relying solely on explicit interactions (e.g., hotel ratings or visits to attractions), this model extends relationships through structured information in a tourism network. Using indirect relationships to improve the accuracy of recommendations, even when there are few direct interactions, it shows that graph structures can improve the predictive capacity of CF by expanding the network of relationships.

Deep Learning based models allow the systems to learn more abstract representations of user-item relationships. In particular, Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT) have shown great effectiveness in preference prediction. They can capture non-linear and complex relationships in data, increase generalizability in data-poor scenarios, and improve model learning capability by integrating additional signals. In [94], the authors use a GCN to improve the accuracy of recommendations by combining collaborative information with KG representations, which allows for multi-level feature extraction in the user-item interaction.

Another notable approach is presented in [80], where user and course information are combined within a bilateral knowledge graph to enhance the accuracy and personalization of recommendations on educational platforms. Fig. 11 illustrates the architecture of the HRS. This system constructs a

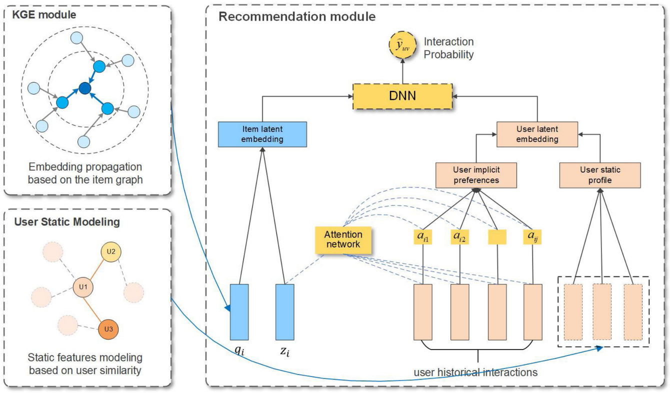


Fig. 11. Architectural representation of the POCR framework, as presented in [80].

bilateral knowledge graph where courses are linked through semantic relationships, and users are connected based on their learning profiles. Additionally, it incorporates an attention mechanism that assigns different weights to each connection in the network, helping identify the most relevant information for recommendations. This approach improves personalization by tailoring course suggestions to the learner's profile and learning history. It also captures implicit relationships between courses, reducing reliance on explicit interactions alone and leading to more accurate recommendations.

Explainability and robustness models focus on making recommendations more interpretable, a feature often missing in traditional CF algorithms. Additionally, many conventional systems are vulnerable to data biases, which can impact the fairness of recommendations. To address these challenges, these models integrate causal inference mechanisms to reduce bias, use explainability techniques such as rule-based reasoning or attention to pathways, and apply differential privacy to protect user data without compromising recommendation quality. For example, [108] presents an approach based on graph paths, where specific routes between users and items are analyzed to provide more transparent recommendations. This helps users understand why they received a particular suggestion, increasing their trust in the system and aiding in bias detection.

In [106], the authors propose a new approach to enhance recommendation reliability by integrating trust metrics into the KG representation. Their method employs a noise-tolerant graph model that filters out unreliable information before incorporating it into the recommender system. Additionally, it introduces a confidence-based weight adjustment mechanism, which evaluates the reliability of user-item interactions before using them for predictions. This improves the system's robustness and accuracy, making it more resilient to inconsistent data. This approach is especially valuable in sensitive fields like healthcare and education, where the reliability of recommendations is crucial.

2) Implementation of knowledge-based approach: Similar to CF, the development of knowledge-based filtering recommendations highlights the importance of the techniques used to generate effective recommendations. The implementation of KBF has centered on three critical aspects, which are, the advanced use of KGs, semantic modeling and data enrichment, and the improved representations in KGs.

The advanced use of KGs enables the representation of complex relationships between items, allowing information to be extracted from the structured connections within the graph. This enhances the ability to uncover implicit relationships, improves recommendation quality, supports generalization, and enables personalization. As a result, recommendations can be generated even in scenarios with limited historical data, as the model can infer similarities between items beyond their explicit attributes. A notable example is [92], where the authors apply attention networks on graphs to enhance recommendation quality. Their approach models the influence of neighboring nodes in a KG, assigning dynamic weights to each connection. This leads to more accurate recommendations by identifying the most relevant nodes in the graph. The study demonstrates that attention mechanisms in graphs are effective for improving personalization and reducing bias in recommendations.

Another noteworthy approach is [89], which introduces a new embedding method that combines Word2Vec and TransR to capture deeper relationships between entities. By modeling web services and their connections as a graph, this method enables more accurate recommendations in complex systems. This embedding technique enhances the representation of items and services, capturing dependencies that traditional models might overlook. Its structured representation makes it particularly well-suited for domains like e-commerce and personalized services, where understanding intricate relationships is essential for improving recommendations.

Semantic modeling and data enrichment—through techniques like pattern mining and semantic rules—enhance recommendation quality by representing concepts in a richer and more interpretable way. Traditional systems often overlook the semantic structure of data, but these approaches improve explainability, better organize knowledge, and enable more meaningful personalization. By leveraging semantic understanding, these models help reduce redundancy and noise in data, avoiding reliance solely on statistical correlations. A compelling example is [76], where the authors use ontologies to represent products and their relationships, combined with sequential pattern mining to model user preferences. This integration of structured knowledge with sequential models improves personalization, demonstrating its potential for generating more accurate recommendations.

Another notable example is [48], which applies semantic analysis to enhance the categorization of tourism services and provide more accurate recommendations. Fig. 12 illustrates the architecture of the HRS, which improves the organization of relevant information and increases recommendation accuracy. By using ontologies to structure knowledge about activities and user preferences, this approach incorporates semantic relationships between services, enabling the discovery of more meaningful connections for users. This demonstrates that integrating semantic knowledge with CF can overcome data sparsity limitations and enhance personalization.

KGs can be challenging to manage due to their complexity and size. To address this, advanced representation techniques optimize how they are structured and processed, improving the

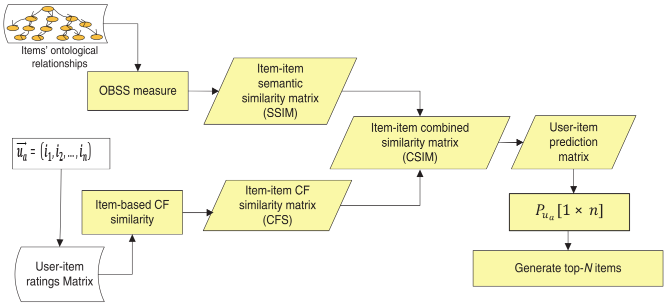


Fig. 12. Architecture of the developed recommender, as shown in [48].

efficiency and accuracy of HRSs. These techniques help reduce noise by filtering out irrelevant information, enhance computational efficiency by optimizing graph structures for faster queries, and better capture complex relationships, allowing for more subtle similarities between items. A notable example is [96], which employs a Lorenz-based hyperbolic model to improve entity representation and capture high-dimensional relationships with minimal information loss. This optimization enhances the learning of large-scale graphs, reducing distortion in their representation and ultimately improving recommendation accuracy. Such advancements are especially valuable in domains with highly complex relationships, where effectively managing massive KGs is crucial.

Another example is [105], which introduces an optimized neural network to enhance the interpretability of recommendations. It employs a hierarchical structure to represent relationships at multiple levels within the KG. This is achieved through a self-supervision mechanism that improves the quality of the generated graph embeddings. By refining how connections between entities are represented, this approach increases the accuracy of graph-based recommendations while making them more informative. Additionally, it enhances interpretability, helping users understand the reasoning behind each recommendation. This demonstrates that neural networks with hierarchical structures can improve knowledge organization within KGs, leading to more transparent and effective recommendations.

3) Implementation of auxiliary technique: In this review, an auxiliary technique refers to a method used to enhance the recommendation capabilities of an HRS. These techniques can be applied at different stages of the recommendation process and must be separate from the primary recommendation methods. Among all the reviewed articles, only 16 incorporate auxiliary techniques. These include various approaches such as Attention Mechanisms (4), Fuzzification (2), Feature Dimension Adjustments (2), Preference Generation Strategies (1), PrefixSpan and TopseqRule (1), Pattern Mining (1), Weighting Enhancement Techniques (1), Multiple Space Embedding Methods (1), Incremental Learning Approaches (1), Causal Inference (1), and High-Order Preference Extraction (1). These auxiliary techniques help address specific limitations within the main recommendation models, acting as complementary enhancements that improve performance across different dimensions.

An interesting example is [64], where the authors propose using Sequential Pattern Mining (SPM) to analyze the order in which users read books and identify recurring preference patterns. Additionally, they apply Rule Mining (RM) to extract meaningful relationships between books from different domains. This process takes place before generating recommendations, enabling the system to suggest relevant items that users have not yet explored. By capturing the evolution of user interests, this approach improves recommendation accuracy, ensuring that suggestions adapt to individual reading patterns over time.

Another notable example is [107], which implements a system that uses causal reasoning to analyze the impact of different factors in the recommendation process and correct potential biases from historical data. This helps reduce the influence of such biases, making recommendations fairer and more personalized. Additionally, by leveraging causal inference, the system improves the reliability of recommendations by providing clear explanations for why certain items are suggested to users. This is especially important in applications where fairness is a key concern, such as in education. This study also demonstrates that combining recommender systems with causal techniques can enhance both the accuracy and interpretability of recommendations.

4) Hybridization process: This review focuses on the hybridization of CF and KBF together with various auxiliary techniques. The analysis of these hybrid implementations highlights their capability to improve the accuracy and personalization of the given recommendations and to reduce the cold start and sparsity problems. The reviewed papers have implemented this hybridization in a variety of ways, which are, model-based fusion, hybridization through transfer learning, and hybridization in hierarchical models.

In [98], the authors propose a model fusion approach by developing a hybrid architecture with two layers that combine user-item interaction representation and semantic information. The fusion occurs at two levels. First, direct relation modeling is applied, where the system learns the basic structure of the UIKG. Then, semantic knowledge is incorporated by integrating semantic relationships extracted from the KG. This enhances item representation and captures implicit relationships, leading to more precise recommendations. This study highlights that combining CF with KBF improves recommendation accuracy, making it particularly effective for well-structured domains like education. Another example of this hybrid approach is [50], where the authors implement a CF model to generate predictions based on user behavior and a KBF model that uses a product ontology to establish semantic relationships between items. Both models are then combined using a weighted aggregation function, where each technique contributes based on its relevance. This fusion improves recommendation accuracy by reducing dependence on historical data, making the system more adaptable and robust.

In [53], the authors propose a transfer learning approach that first utilizes tourism ontologies to model relationships between different categories of attractions. Then, the KBF technique generates semantic representations of the items, which are transferred to the CF model. This allows CF to generate recommendations even with limited data, making it an effective strategy for mitigating the cold start problem. By leveraging structured knowledge from another technique, this approach enhances recommendation quality in data-scarce scenarios. Another interesting transfer learning approach is presented in [104], where KBF is first used to construct a KG that maps relationships between users and items. Causal inference techniques are then applied to correct potential biases in user-item interactions. This refined knowledge is subsequently transferred to the CF model, enabling it to generate fairer and more explainable recommendations. By reducing the impact of biases in the data, this approach enhances both the fairness and interpretability of the HRS.

In [93], the authors propose a hierarchical hybridization consisting of three levels. The first level uses CF to learn initial representations of users and items. The second level applies a GCN in a KG to refine item representations. Finally, the third level fuses the processed information to generate the final recommendation. This approach enhances the accuracy and robustness of the HRS by integrating user interactions with semantic knowledge in a scalable model. Another notable approach is the system introduced in [85], which features a bi-layer architecture where each layer serves a distinct function in the HRS. First, a bipartite graph is constructed, linking users with items, and CF is applied to capture relationships based on explicit interactions. Then, a KG is integrated to represent semantic relationships between items, which are grouped using KBF based on their characteristics and connections within the graph. Each model then applies a Light GCN to refine user and item representations, which are subsequently combined. This approach reduces the computational complexity of the HRS while improving its ability to capture user-item relationships by integrating both direct interactions and structured knowledge.

V. DISCUSSION

The data presented in Section IV offers valuable insights into the hybridization of CF and KBF approaches. This section explores these findings in greater depth, aiming to address the research questions outlined in Section I.

A. *Current State of Hybridization*

HRS emerged as a response to the limitations of traditional implementations of CF and KBF. While CF has proven to be highly effective in personalizing content based on previous user interactions, it suffers from problems such as data sparsity, cold start, and lack of explainability. On the other hand, KBF uses structured knowledge, such as knowledge graphs and ontologies, to improve recommendations, but can be rigid and costly to build and maintain. Hybridization seeks to combine the best of both worlds, integrating CF and KBF into a single model, giving rise to a variety of ways of doing it. Over the last few years, hybridization research has evolved with the integration of deep learning, graph neural networks (GCNs), causal inference, and self-supervised learning (SSL) techniques, which implies that a detailed analysis of current hybridization techniques is needed. This provides a comprehensive foundation for partially addressing some of the research questions introduced earlier, offering valuable insights into the evolution and impact of hybrid approaches.

1) *RQ1: What methodologies are currently used to hybridize collaborative filtering and knowledge-based recommendation systems?:* This review highlights that most of the included articles focus on improving both types of recommenders, as well as the strategies for their combination, leading to various innovations in the implementation of CF and KBF. CF has primarily focused on enhancing the models' ability to analyze collaborative data, while KBF has concentrated on representing structured and semantic knowledge. The combination of these approaches has driven advancements in the hybridization process, resulting in three main methods for achieving it:

- Model Fusion: CF and KBF generate recommendations separately and then combine them using techniques like aggregation, heightening, or neural networks. This approach adapts well to different data types and integrates multiple information sources. However, it may increase computational complexity.
- Transfer Learning: One model (CF or KBF) learns representations and transfers this knowledge to the other. It is commonly used in cross-domain scenarios or to enhance item representations in CF through KBF. This approach reduces bias in recommendations and improves generalization in data-sparse domains but requires precise alignment between CF and KBF representations.
- Hierarchical Models: CF and KBF operate at different levels within a deep learning model and are combined in a hierarchical structure, where each level refines the previous one. This method captures complex relationships across multiple levels while leveraging the graph structure without significantly increasing computational costs.

2) *RQ2: What common challenges arise when integrating collaborative filtering with knowledge-based techniques?:* The fusion of the different recommenders is a key component of the development of HRS, however the integration of CF and KBF still presents significant challenges:

- Computational Complexity: Models such as GCNs and deep neural networks require large computational power, which limits their applicability in real-time systems.
- Interpretability and Explainability: Combining CF with KBF often generates more complex and less interpretable models. This is critical in sensitive areas such as health, education, and finance, where users need to understand why an item is recommended to them.
- Data Sparsity and Cold Start: Although KBF helps mitigate this problem, hybrid systems still rely on prior interactions to fit their models.
- Knowledge Graph Maintenance: Building and updating ontologies or KGs is costly and requires human intervention, which can limit the scalability of the system.

Despite these challenges, the integration of CF and KBF continues to drive advancements in HRS by leveraging both collaborative interactions and structured knowledge. Addressing these issues will enable more scalable, accurate, and explainable HRS, improving their applicability across various domains.

3) RQ3: How do hybrid recommendation systems address data sparsity and cold start issues across different domains and applications?: Addressing the challenge of data integration is crucial for overcoming two key issues in traditional RS: the cold start problem and data sparsity. HRS have developed different strategies to face these limitations, which are as follows:

- Incorporation of KGs: using ontologies or KGs to generate semantic representations of the items helps to reduce the dependency on historical data which leads to a mitigation of the data sparsity and cold start.
- Transfer of Learning in Cross-Domain Scenarios: transferring knowledge from one domain to another can reduce the impact of having numerous missing data.
- Self-Supervised Learning: self-supervised learning techniques can generate good representations of both items and users without requiring large volumes of labeled data.
- Causal Inference Methods: using causal inference allows for improved personalization without relying exclusively on explicit user data.

B. Evaluating Hybridization

Combining CF and KBF has been shown to significantly improve recommendation performance compared to individual RSs. Most evaluations of HRS still rely on offline quantitative methods. A common approach is to compare the new system against well-established ones to assess its effectiveness. Only a few studies conduct online evaluations or user studies, likely due to the added complexity and the challenge of gathering enough participants for meaningful results. However, these evaluations provide valuable insights into the advantages of hybrid approaches and help answer key research questions.

1) RQ4: How effective are hybrid approaches in improving recommendation accuracy compared to standalone collaborative filtering or knowledge-based systems?: Offline comparisons between traditional RSs and HRSs show that HRSs generally perform better across different evaluation metrics. However, the improvement is not always significant, as some performance gains are minor. Still, certain articles show increases of 20% or more compared to traditional systems. A clear example of this trend is presented in [81], where hybrid systems achieve notably better evaluation results than previous state-of-the-art recommenders. These improvements are reported in several studies, comparing HRSs with leading systems like RippleNet, KGAT, KGCN, and CKAN [109], [110], [111], [112].

Comparing different systems and datasets can be challenging, making evaluations more complex. Common metrics like precision and recall have limitations that can affect the reliability of results. One issue with these metrics is their inconsistency in handling the number of recommended elements and relevant elements. For example, if recall@3 is used but only two relevant elements exist, achieving 100% recall is impossible. This can lead to misinterpretations if such constraints are not considered. Since most studies rely on these metrics, reproducibility becomes difficult, both within the same domain and across different domains. Discrepancies in defining relevant elements can further complicate result comparisons. Furthermore, the lack of standardized benchmarking—characterized by heterogeneous datasets and varying metric parameters—means that while individual hybrid models may report performance gains exceeding 20%, these results cannot be statistically synthesized in a unified manner at this stage.

In addition to these limitations of offline metrics, there is a critical disconnect between algorithmic performance and actual user experience. While the majority of the articles reviewed use quantitative validations based on historical data, only a tiny fraction incorporate user studies or deployments in real environments to measure qualitative factors. This reliance on static metrics such as RMSE or precision ignores fundamental dimensions of hybrid recommendation, such as serendipity (discovery of valuable but unexpected items), model transparency, and the trust that users place in the system after receiving a knowledge-based explanation. Therefore, the success of a hybrid system should not be measured solely by its predictive power, but by its impact on long-term user behavior and satisfaction.

Evaluating qualitative metrics is another major challenge, as it requires detailed analysis to confirm that HRSs truly outperform traditional systems. In the studies reviewed, user satisfaction with these systems is generally high. However, the lack of in-depth qualitative evaluations makes it difficult to measure their real-world effectiveness. Additionally, these metrics have limitations, such as the number of participants and the reliability of their responses, which can affect the accuracy of the results.

Overall, these findings suggest that HRSs have the potential to outperform traditional systems, offering improved recommendation quality and adaptability. Their ability to combine different approaches allows them to address limitations such as data sparsity and cold start problems more effectively. However, despite their promising performance, further research is needed to fully understand their advantages, limitations, and real-world applicability.

C. Domain Preference

Although many of the HRSs analyzed do not target specific domains, they often rely on commonly used datasets such as MovieLens [113] or Last.FM [114]. Despite their domain-agnostic nature, creating their respective KGs or ontologies still requires customization based on the domain. For this reason, despite saying that they can be taken to any domain, it is important to keep in mind that specific data requirements may need to be met. This means that certain domains may present certain limitations when it comes to obtaining certain data, or transforming it so that the developed techniques can be carried out. However, certain specific domains tend to receive more attention.

Among the domains receiving significant attention, educational recommendation stands out as an important area of study, possibly driven by the rapid growth of online learning platforms and online education [115]. Other domains, such as cloud services and entertainment, are also gaining prominence, fueled by the increasing adoption of cloud-based software solutions and the continuous expansion of the online entertainment industry [78], [54]. Additionally, the tourism sector has seen a surge in focus, especially in the wake of the COVID-19 pandemic. As the tourism industry adapts and looks for ways to rejuvenate, RSs tailored to this sector hold great potential [116].

Cross-domain recommenders are also emerging as a powerful solution for transferring knowledge between domains, making them particularly valuable [53], [64]. However, these systems also come with several limitations that need to be addressed. One major challenge is the difference in data distributions between domains, which can lead to inconsistencies when transferring knowledge. Additionally, user preferences may not always align across domains, making it difficult to accurately model their interests. Another limitation is the need for large and diverse datasets to effectively learn meaningful relationships between domains, which may not always be available.

Overall, HRSs have proven to be effective across a wide range of domains, demonstrating their flexibility and adaptability in different recommendation scenarios. Their ability to integrate multiple techniques makes them a valuable alternative to traditional recommenders. This review provides a solid foundation for revisiting and further addressing some of the research questions that were only partially answered earlier in this section, offering deeper insights into the strengths and limitations of hybrid approaches.

1) RQ2: What common challenges arise when integrating collaborative filtering with knowledge-based techniques?: One major difficulty in HRSs is the need for domain-specific customization. Building and maintaining KGs or ontologies requires specialized expertise, which may not always be available, making the integration process complex and resource-intensive. Additionally, ensuring compatibility between the data used for both approaches is critical but not always straightforward. CF relies on user interaction data, while KBF depends on structured domain knowledge, and aligning these two types of information can be challenging, especially in domains where relevant knowledge representations are scarce or difficult to standardize. Moreover, balancing the strengths of both methods while mitigating their individual weaknesses requires careful algorithmic design to avoid biases or inconsistencies in recommendations.

Despite these challenges, successfully integrating CF and KBF can lead to more accurate, explainable, and personalized recommendations. By leveraging the strengths of both approaches, hybrid systems can provide richer insights, improve user satisfaction, and enhance recommendation quality across diverse domains.

2) RQ3: How do hybrid recommendation systems address data sparsity and cold start issues across different domains and applications?: HRSs tackle data sparsity and cold start issues by leveraging the strengths of multiple recommendation techniques. By combining CF with KBF, HRSs can generate accurate recommendations even when user interaction data is limited. In domains such as education, cloud services, and tourism, where data availability varies, HRSs improve recommendation quality by incorporating structured knowledge, user profiles, or contextual information.

Additionally, in domains with fewer types of available data, advanced data processing techniques—such as feature engineering, transfer learning, or matrix factorization—help optimize the integration of both recommenders. This allows HRSs to mitigate sparsity and cold start issues by utilizing alternative sources of information, such as metadata, semantic relationships, or content-based attributes. As a result, HRSs enhance personalization and adaptability across different applications, making them a powerful solution for recommendation systems facing data limitations.

D. Current Gaps in Hybridization

Despite recent advances, hybridization in recommender systems still faces several challenges that limit its applicability in real-world scenarios. These challenges affect not only the efficiency and scalability of the models but also their interpretability and generalizability across domains. The following gaps are listed:

1) Computational complexity: Modern hybrid models have incorporated advanced techniques such as GCNs, GATs and deep learning to improve the representation of items and users. However, these architectures are often computationally intensive, which can make them difficult to implement in environments with limited hardware or real-time systems.

2) Balance between accuracy and explainability: While hybridization has improved the accuracy of recommendations by integrating multiple sources of information, many hybrid models still operate as black boxes, making it difficult to understand why a user receives a specific recommendation. This problem is critical in sensitive domains such as health, education, and finance, where users and regulators need to understand the logic behind system decisions.

3) Handling sparse data and cold start: One of the main benefits of hybridization is that it can mitigate the problem of data sparsity and cold start, but it still does not solve it completely. CF needs prior interactions to make accurate predictions, so it still has problems when there is not enough user history. KBF helps alleviate this problem by providing structured information (ontologies, knowledge graphs), but it depends on the graph being well constructed and up-to-date.

4) Generalizability and adaptability: Hybrid models have proven effective in e-commerce, entertainment, and education, but their performance varies depending on the domain and the types of data available.

5) Lack of evaluation with online metrics and user tests: One of the big problems in current hybridization research is that most evaluations are based on offline metrics, such as accuracy, recall, and NDCG, without considering their actual impact on user experience. Offline metrics may not reflect how users actually interact with the system. Few studies incorporate A/B experiments or user tests in real environments and little attention is paid to diversity, novelty, trust, and user satisfaction metrics.

E. Future Research Recommendations

The continuous evolution of HRSs has significantly improved the accuracy, adaptability, and robustness of recommendations. However, the several challenges that remain unresolved require further research to enhance the scalability, explainability, and generalization of hybrid models. Based on the current gaps identified, the following research directions are proposed:

1) Development of more efficient and scalable hybrid models: There is a growing need to create lightweight GCNs and graph-based approximation techniques to reduce the high computational costs associated with deep learning models. Research into low-rank approximations, model pruning, and quantization could significantly enhance efficiency. Another promising direction is the integration of federated learning and distributed computing to support large-scale HRSs. These advancements could greatly expand the applicability of HRSs, making them more viable for mobile and edge computing environments.

2) Improving explainability without compromising accuracy: The integration of explainable AI techniques is essential, particularly in domains like education and e-commerce. Approaches such as attention mechanisms, rule-based explanations, and counterfactual reasoning can improve transparency without sacrificing performance. Further research into graph-based path analysis could help track how recommendations are generated from both CF and KBF sources. Additionally, incorporating human-in-the-loop systems would allow users to adjust and better understand recommendations, increasing trust and facilitating adoption in regulated sectors.

3) Implementing contrastive learning techniques: In hybrid systems, it can enhance the learning of meaningful item representations, reducing reliance on labeled data. Additionally, leveraging multi-task learning to jointly optimize representation learning and recommendation objectives can improve the adaptability and robustness of HRSs. These approaches can lead to more efficient and generalizable recommendation models, particularly in data-scarce environments.

4) Enhancing fairness, bias mitigation, and diversity in hybrid models: Integrating causal inference techniques can help differentiate correlation from causation in user preferences, leading to more equitable recommendations. Implementing fairness-aware loss functions ensures a balance between accuracy and fairness, reducing biases that may disproportionately affect certain user groups. Additionally, promoting diversity-aware recommendations can prevent over-personalization and filter bubbles, fostering user satisfaction and enabling the discovery of new content. These advancements contribute to making HRSs more transparent, inclusive, and user-centric.

5) Automating knowledge graph construction and maintenance: Leveraging Natural Language Processing (NLP) and Large Language Models (LLMs) enables the extraction of structured knowledge from unstructured data sources, reducing the reliance on manual curation. Reinforcement learning can dynamically update knowledge graphs based on user interactions, ensuring they remain relevant over time. Additionally, incorporating crowdsourcing and user feedback mechanisms enhances the accuracy and adaptability of these graphs. These advancements streamline KG maintenance, making hybrid models more scalable and responsive to evolving domains.

6) Shifting from offline to online evaluation metrics and user testing: Expanding the use of online evaluation methods, such as A/B testing and reinforcement learning-based optimization, enables real-time assessment of HRSs. Incorporating user-centered metrics like engagement rate, diversity, novelty, and trust ensures that evaluations align with user satisfaction. Developing hybrid frameworks that integrate both offline and online evaluations provides a more comprehensive performance analysis. This shift leads to more accurate optimizations based on real user interactions rather than static datasets, improving the adaptability and effectiveness of HRSs in dynamic environments.

By delving into these aspects of the recommendation process, researchers can uncover deeper insights and pave the way for enhanced efficiency in recommender performance, ultimately leading to improved results and heightened user satisfaction. This concerted effort towards advancing HRSs stands to revolutionize the field, offering more robust solutions to the evolving needs and expectations of users in various contexts.

F. Limitations

This scoping review is subjected to several limitations inherent to reviews of this nature. First, only articles published in journals were analyzed, which may have led to the exclusion of relevant studies presented at conferences, workshops, or other sources such as preprints and technical reports. Given the rapid evolution of hybrid recommendation systems, conference proceedings often include cutting-edge research that may not yet be published in journals. Second, articles may have been excluded if they did not explicitly provide the necessary details for assessing their eligibility based on the predefined selection criteria. Some studies may contain relevant methodologies but lack clear descriptions of their hybridization strategies, making it difficult to categorize them accurately. Third, despite employing search terms derived from previous reviews, variations in terminology and indexing practices across different databases may have caused some relevant articles to be omitted. Authors often describe hybridization techniques using diverse terminologies, making it challenging to ensure comprehensive coverage of all relevant works.

Additionally, this review focused on methodological aspects rather than empirical performance comparisons. While it identifies and categorizes existing hybridization strategies, it does not conduct a quantitative meta-analysis of their effectiveness across different domains, which could provide deeper insights into their real-world applicability. Finally, the screening and data extraction process was conducted systematically, but it remains susceptible to potential inaccuracies and subjective interpretations. Although efforts were made to ensure consistency in data classification, some articles may have been misclassified or overlooked due to ambiguities in their descriptions. Future studies could address this limitation by incorporating multiple reviewers and inter-rater reliability assessments to enhance the robustness of the review process.

VI. CONCLUSION

HRSs have become a fundamental approach to overcome the limitations of traditional CF and KBF methods. By combining CF's ability to leverage user interactions with KBF's structured knowledge representation, hybrid models provide more accurate, personalized, and robust recommendations. However, the growing complexity of hybridization strategies has led to the need for a comprehensive understanding of the methodologies used, the challenges faced, and the gaps that remain.

This scoping review systematically analyzed the state-of-the-art HRSs, classifying them into three primary hybridization strategies: Model Fusion, Transfer Learning, and Hierarchical Models. Model Fusion techniques combine the outputs of CF and KBF into a single recommendation, while Transfer Learning enables knowledge learned from one approach to enhance the other, particularly in cross-domain scenarios. Hierarchical Models, on the other hand, use multi-level architectures, where CF and KBF operate at different stages within deep learning frameworks, optimizing the representation of users and items at different levels of abstraction.

The findings of this review highlight that hybridization significantly improves recommendation accuracy, personalization, and the ability to handle data sparsity and cold start issues. Recent advancements, such as GCNs, GATs, and deep learning architectures, have become dominant in modern hybrid systems. However, these techniques introduce new computational challenges and raise concerns regarding their interpretability and explainability. Additionally, emerging approaches like causal inference, self-supervised learning, and reinforcement learning offer promising solutions but remain underexplored in the context of HRSs.

Moving forward, HRSs must address several key areas to maximize their impact. Scalability and efficiency should be improved through the development of lightweight GCNs, federated learning approaches, and model compression techniques. Explainability should be prioritized, integrating interpretable AI techniques such as attention-based path analysis and rule-based hybrid models. Additionally, automated knowledge graph construction and maintenance could significantly enhance the adaptability of KBF-based approaches, reducing reliance on manual ontology updates. Finally, a shift toward holistic evaluation frameworks, incorporating both offline and online metrics, will be crucial to assess the true effectiveness of hybrid models in real-world applications.

By addressing these challenges, HRSs will become more adaptable, transparent, and effective across diverse applications, ranging from e-learning and healthcare to e-commerce and content streaming. Continued research and innovation in this field will be essential to unlocking the full potential of hybrid models, ensuring that they remain scalable, interpretable, and user-centric in the years to come.

ACKNOWLEDGMENT

This work was supported by the SEEN project, funded by the Erasmus+ Programme of the European Union under Grant Agreement No. 2024-1-BG01-KA220-HED-000255491. The views and opinions expressed are those of the authors and do not necessarily reflect those of the European Union or the Erasmus+ Programme. Neither the European Union nor the granting authority can be held responsible for them.

Funded by
the European Union

SUPPLEMENTARY INFORMATION

A supplementary file (Appendix) is provided with this review, detailing the full process followed during the analysis of the reviewed articles, including selection criteria, categorization methodology, and extracted data for each study.

REFERENCES

[1] D. Bawden and L. Robinson, "Information overload: An overview," in *Oxford Encyclopedia of Political Decision Making*. Oxford: Oxford University Press, June 2020.

[2] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Techniques, applications, and challenges," *Recommender systems handbook*, pp. 1–35, 2021.

[3] Y. Koren, S. Rendle, and R. Bell, "Advances in collaborative filtering," *Recommender systems handbook*, pp. 91–142, 2021.

[4] R. L. Rosa, G. M. Schwartz, W. V. Ruggiero, and D. Z. Rodríguez, "A knowledge-based recommendation system that includes sentiment analysis and deep learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2124–2135, 2019.

[5] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowledge-Based Systems*, vol. 157, pp. 1–9, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705118302107

[6] M. Y. H. Al-Shamri, "User profiling approaches for demographic recommender systems," *Knowledge-Based Systems*, vol. 100, pp. 175–187, 2016.

[7] S. Kulkarni and S. F. Rodd, "Context aware recommendation systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, p. 100255, 2020.

[8] M. Zihayat, A. Ayanso, X. Zhao, H. Davoudi, and A. An, "A utility-based news recommendation system," *Decision Support Systems*, vol. 117, pp. 14–27, 2019.

[9] Q. Zhang, J. Lu, and Y. Jin, "Artificial intelligence in recommender systems," *Complex & Intelligent Systems*, vol. 7, no. 1, pp. 439–457, 2 2021.

[10] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012, recommender Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157312000828

[11] D. Valcarce, A. Landin, J. Parapar, and Álvaro Barreiro, "Collaborative filtering embeddings for memory-based recommender systems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 347–356, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197619301599

[12] M. Yu, T. Quan, Q. Peng, X. Yu, and L. Liu, "A model-based collaborate filtering algorithm based on stacked autoencoder," *Neural Computing and Applications*, vol. 34, no. 4, pp. 2503–2511, 2 2022. [Online]. Available: https://doi.org/10.1007/s00521-021-05933-8

[13] M. Dong, X. Zeng, L. Koehl, and J. Zhang, "An interactive knowledge-based recommender system for fashion product design in the big data environment," *Information Sciences*, vol. 540, pp. 469–488, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025520304278

[14] C. Obeid, I. Lahoud, H. El Khoury, and P.-A. Champin, "Ontology-based recommender system in higher education," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1031–1034.

[15] A. Singhal, "Introducing the knowledge graph: things, not strings," *Official Google Blog*, 5 2012. [Online]. Available: https://blog.google/products/search/introducing-knowledge-graph-things-not/

[16] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2015.

[17] C. Yu, X. Liu, J. Maia, Y. Li, T. Cao, Y. Gao, Y. Song, R. Goutam, H. Zhang, B. Yin *et al.*, "Cosmo: A large-scale e-commerce common sense knowledge generation and serving system at amazon," in *Companion of the 2024 International Conference on Management of Data*, 2024, pp. 148–160.

[18] J. Jiang, A. Ponnada, A. Li, B. Lacker, and S. F. Way, "A genre-based analysis of new music streaming at scale," in *Proceedings of the 16th ACM Web Science Conference*, ser. WEBSCI '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 191–201.

[19] R. Sharma, D. Gopalani, and Y. Meena, "Collaborative filtering-based recommender system: Approaches and research challenges," in *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 2017, pp. 1–6.

[20] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, "Knowledge graphs: Opportunities and challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13 071–13 102, 11 2023.

[21] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 11 2002.

[22] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent data analysis*, vol. 21, no. 6, pp. 1487–1524, 2017.

[23] V. Danilova and A. Ponomarev, "Hybrid recommender systems: The review of state-of-the-art research and applications," in *Proceedings of the 20th Conference of FRUCT Association*, 2017.

[24] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, no. 1, p. 59, May 2022. [Online]. Available: https://doi.org/10.1186/s40537-022-00592-5

[25] I. Saifudin and T. Widiyaningtyas, "Systematic literature review on recommender system: Approach, problem, evaluation techniques, datasets," *IEEE Access*, vol. 12, pp. 19 827–19 847, 2024.

[26] R. De Croon, L. Van Houdt, N. N. Htun, G. Štiglic, V. Vanden Abeele, and K. Verbert, "Health recommender systems: Systematic review," *J Med Internet Res*, vol. 23, no. 6, p. e18035, 6 2021.

[27] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 421425, 10 2009.

[28] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 21st International Conference on Neural Information Processing Systems*, ser. NIPS'07. Red Hook, NY, USA: Curran Associates Inc., 2007, p. 1257–1264.

[29] K. Baker, "Singular value decomposition tutorial," *The Ohio State University*, vol. 24, p. 22, 2005.

[30] L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, and J. J. Samper-Zapater, "Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1202–1222, 2015.

[31] J. Z. Pan, *Resource Description Framework*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 71–90.

[32] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[33] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705113001044

[34] J. Yuan, W. Shalaby, M. Korayem, D. Lin, K. AlJadda, and J. Luo, "Solving cold-start problem in large-scale recommendation engines: A deep learning approach," 2016. [Online]. Available: https://arxiv.org/abs/1611.05480

[35] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, ser. ICUIMC '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 208–211.

[36] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 2065–2073, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417413007240

[37] J. Feng, Z. Xia, X. Feng, and J. Peng, "Rbpr: A hybrid model for the new user cold start problem in recommender systems," *Knowledge-Based Systems*, vol. 214, p. 106732, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120308613

[38] H. Wang, Z. Wang, and W. Zhang, "Quantitative analysis of matthew effect and sparsity problem of recommender systems," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE, Apr. 2018, p. 78–82. [Online]. Available: http://dx.doi.org/10.1109/ICCCBDA.2018.8386490

[39] H. Yin, Q. Wang, K. Zheng, Z. Li, and X. Zhou, "Overcoming data sparsity in group recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 7, pp. 3447–3460, 2022.

[40] S. Ahmadian, N. Joorabloo, M. Jalili, and M. Ahmadian, "Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach," *Expert Systems with Applications*, vol. 187, p. 115849, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421012100

[41] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent data analysis*, vol. 21, no. 6, pp. 1487–1524, 2017.

[42] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? a survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, May 2019.

[43] A. C. Tricco, E. Lillie, W. Zarin, K. K. O'Brien, H. Colquhoun, D. Levac, D. Moher, M. D. Peters, T. Horsley, L. Weeks, S. Hempel, E. A. Akl, C. Chang, J. McGowan, L. Stewart, L. Hartling, A. Aldcroft, M. G. Wilson, C. Garritty, S. Lewin, C. M. Godfrey, M. T. Macdonald, E. V. Langlois, K. Soares-Weiser, J. Moriarty, T. Clifford, Ö. Tunçalp, and S. E. Straus, "Prisma extension for scoping reviews (prisma-scr): Checklist and explanation," *Annals of Internal Medicine*, vol. 169, no. 7, pp. 467–473, 2018, pMID: 30178033.

[44] A. Martinez-Martinez, R. Montoliu, and I. Remolar, "Collaborative filter and knowledge-based hybrid recommenders - scoping review protocol," 2024.

[45] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim, "A literature review and classification of recommender systems research," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10 059–10 072, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417412002825

[46] J. K. Tarus, Z. Niu, and G. Mustafa, "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 21–48, 6 2018. [Online]. Available: https://doi.org/10.1007/s10462-017-9539-5

[47] D. Wu, J. Lu, and G. Zhang, "A fuzzy tree matching-based personalized e-learning recommender system," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2412–2426, 2015.

[48] M. Al-Hassan, H. Lu, and J. Lu, "A semantic enhanced hybrid recommendation approach: A case study of e-government tourism service recommendation system," *Decision Support Systems*, vol. 72, pp. 97–109, 2015.

[49] J. Zheng, B. Zhang, X. Yue, G. Zou, J. Ma, and K. Jiang, "Neighborhood-user profiling based on perception relationship in the micro-blog scenario," *Journal of Web Semantics*, vol. 34, pp. 13–26, 2015.

[50] M. Guia, R. R. Silva, and J. Bernardino, "A hybrid ontology-based recommendation system in e-commerce," *Algorithms*, vol. 12, no. 11, 2019.

[51] Z. Cao, X. Qiao, S. Jiang, and X. Zhang, "An efficient knowledge-graph-based web service recommendation algorithm," *Symmetry*, vol. 11, no. 3, 2019.

[52] L. Liu, T. Yuan, H. Song, and H. Wang, "Manufacturing technology knowledge retrieval system based on ose and ucf," *IEEE Access*, vol. 7, pp. 39 623–39 638, 2019.

[53] C. Valliyammai and S. Ephina Thendral, “Ontology matched cross domain personalized recommendation of tourist attractions,” *Wireless Personal Communications*, vol. 107, no. 1, pp. 589–602, 7 2019.

[54] L. O. Colombo-Mendoza, R. Valencia-García, R. Colomo-Palacios, and G. Alor-Hernández, “A knowledge-based multi-criteria collaborative filtering approach for discovering services in mobile cloud computing platforms,” *Journal of Intelligent Information Systems*, vol. 54, no. 1, pp. 179–203, 2 2020.

[55] Y. Zhang, J. Wang, and J. Luo, “Knowledge graph embedding based collaborative filtering,” *IEEE Access*, vol. 8, pp. 134 553–134 562, 2020.

[56] D. Zhang, L. Liu, Q. Wei, Y. Yang, P. Yang, and Q. Liu, “Neighborhood aggregation collaborative filtering based on knowledge graph,” *Applied Sciences*, vol. 10, no. 11, 2020.

[57] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, “Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data,” *Expert Systems with Applications*, vol. 149, p. 113248, 2020.

[58] B. Jiang, J. Yang, Y. Qin, T. Wang, M. Wang, and W. Pan, “A service recommendation algorithm based on knowledge graph and collaborative filtering,” *IEEE Access*, vol. 9, pp. 50 880–50 892, 2021.

[59] Y. Abuzir and M. Dwieb, “Hotel recommender system based on knowledge graph and collaborative approach,” *International Journal of Computing*, vol. 20, no. 1, pp. 63–71, Mar. 2021.

[60] J. Tang, “Optimization of english learning platform based on a collaborative filtering algorithm,” *Complexity*, vol. 2021, p. 6624012, 4 2021.

[61] G. Xu, G. Jia, L. Shi, and Z. Zhang, “Personalized course recommendation system fusing with knowledge graph and collaborative filtering,” *Computational Intelligence and Neuroscience*, vol. 2021, p. 9590502, 9 2021.

[62] H. Bouazza, B. Said, and F. Zohra Laallam, “A hybrid iot services recommender system using social iot,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, Part B, pp. 5633–5645, 2022.

[63] C. Obeid, C. Lahoud, K. El, and P.-A. Champin, “A novel hybrid recommender system approach for student academic advising named cohrs, supported by case-based reasoning and ontology,” *Computer Science and Information Systems*, vol. 19, no. 2, p. 979–1005, 2022.

[64] T. Anwar and V. Uma, “Cd-spm: Cross-domain book recommendation using sequential pattern mining and rule mining,” *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 793–800, 2022.

[65] H. Jia, L. Yang, and B. Cui, “Collaborative filtering recommendation of online learning resources based on knowledge association model,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 17, no. 02, p. pp. 158–172, Jan. 2022.

[66] A. Alsobhi and N. Amare, “Ontology-based relational product recommendation system,” *Computational and Mathematical Methods in Medicine*, vol. 2022, p. 1591044, 9 2022.

[67] X. Su, J. He, J. Ren, and J. Peng, “Personalized chinese tourism recommendation algorithm based on knowledge graph,” *Applied Sciences*, vol. 12, no. 20, 2022.

[68] I. M. G. Cardoso, J. L. V. Barbosa, B. M. Alves, L. P. S. Dias, and L. C. Nesi, “Vulcont: A recommender system based on context history ontology,” *IET Software*, vol. 16, no. 1, pp. 111–123, 2022.

[69] C. Lahoud, S. Moussa, C. Obeid, H. E. Khoury, and P.-A. Champin, “A comparative analysis of different recommender systems for university major and career domain guidance,” *Education and Information Technologies*, vol. 28, no. 7, pp. 8733–8759, 7 2023.

[70] N. Liu, “Improving online ideological and political precision education in universities based on knowledge mapping,” *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1, pp. –, 2024.

[71] Y. Niu, R. Lin, and H. Xue, “Research on learning resource recommendation based on knowledge graph and collaborative filtering,” *Applied Sciences*, vol. 13, no. 19, 2023.

[72] L. Wei and X. Zhu, “Semantic recommendation model via fusing knowledge graph and formal concept analysis,” *IEEE Access*, vol. 11, pp. 62 337–62 347, 2023.

[73] J. Tian and B. Tian, “Computer aided music instructional resource recommendation system based on knowledge map and collaborative filtering,” *Computer-Aided Design and Applications*, p. 32–45, 10 2023.

[74] M. Al-Hassan, B. Abu-Salih, E. Alshdaifat, A. Aloqaily, and A. Rodan, “An improved fusion-based semantic similarity measure for effective collaborative filtering recommendations,” *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 45, 3 2024.

[75] P. Bahrani, B. Minaei-Bidgoli, H. Parvin, M. Mirzarezaee, and A. Keshavarz, “A hybrid semantic recommender system based on an improved clustering,” *The Journal of Supercomputing*, vol. 80, no. 9, pp. 13 341–13 385, 6 2024.

[76] G. Mustafa, N. A. Jhamat, Z. Arshad, N. Yousaf, M. N. Abdal, M. Maray, D. Alqahtani, M. A. Merhabi, M. A. Aziz, and T. Ahmad, “Ontocommerce: Incorporating ontology and sequential pattern mining for personalized e-commerce recommendations,” *IEEE Access*, vol. 12, pp. 42 329–42 342, 2024.

[77] M. G. Ayadi, H. Mezni, R. Alnashwan, and H. Elmannai, “Effective healthcare service recommendation with network representation learning: A recursive neural network approach,” *Data & Knowledge Engineering*, vol. 148, p. 102233, 2023.

[78] M. Nilashi, O. Ibrahim, and K. Bagherifard, “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques,” *Expert Systems with Applications*, vol. 92, pp. 507–520, 2018.

[79] Z. Pan and H. Chen, “Collaborative knowledge-enhanced recommendation with self-supervisions,” *Mathematics*, vol. 9, no. 17, 2021.

[80] S. Yang and X. Cai, “Bilateral knowledge graph enhanced online course recommendation,” *Information Systems*, vol. 107, p. 102000, 2022.

[81] E. Elahi and Z. Halim, “Graph attention-based collaborative filtering for user-specific recommender system using knowledge graph and deep neural networks,” *Knowledge and Information Systems*, vol. 64, no. 9, pp. 2457–2480, 9 2022.

[82] F. Wang, Y. Li, Y. Zhang, and D. Wei, “Klgcn: Knowledge graph-aware light graph convolutional network for recommender systems,” *Expert Systems with Applications*, vol. 195, p. 116513, 2022.

[83] S. Zhang, X. Wang, R. Wang, J. Gu, and J. Li, “Knowledge graph recommendation model based on feature space fusion,” *Applied Sciences*, vol. 12, no. 17, 2022.

[84] M. Jian, C. Zhang, X. Fu, L. Wu, and Z. Wang, “Knowledge-aware multispace embedding learning for personalized recommendation,” *Sensors*, vol. 22, no. 6, 2022.

[85] P. Chen, J. Zhao, and X. Yu, “Lighterkgcn: A recommender system model based on bi-layer graph convolutional networks,” *Journal of Internet Technology*, vol. 23, no. 3, pp. 621–629, 2022. [Online]. Available: https://jit.ndhu.edu.tw/article/view/2707

[86] Q. Dai, X.-M. Wu, L. Fan, Q. Li, H. Liu, X. Zhang, D. Wang, G. Lin, and K. Yang, “Personalized knowledge-aware recommendation with collaborative and attentive graph convolutional networks,” *Pattern Recognition*, vol. 128, p. 108628, 2022.

[87] F. Qian, Y. Zhu, H. Chen, J. Chen, S. Zhao, and Y. Zhang, “Reduce unrelated knowledge through attribute collaborative signal for knowledge graph recommendation,” *Expert Systems with Applications*, vol. 201, p. 117078, 2022.

[88] P. M. T. Do and T. T. S. Nguyen, “Semantic-enhanced neural collaborative filtering models in recommender systems,” *Knowledge-Based Systems*, vol. 257, p. 109934, 2022.

[89] F. Xie, Y. Zhang, K. Przystupa, and O. Kochan, “A knowledge graph embedding based service recommendation method for service-based system development,” *Electronics*, vol. 12, no. 13, 2023.

[90] X. Zhang and S. Kuang, “A lightweight method of knowledge graph convolution network for collaborative filtering,” *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 19, no. 1, pp. 1–21, 2023.

[91] Y. Ma, X. Zhang, C. Gao, Y. Tang, L. Li, R. Zhu, and C. Yin, “Enhancing recommendations with contrastive learning from collaborative knowledge graph,” *Neurocomputing*, vol. 523, pp. 103–115, 2023.

[92] Q. Wang, H. Cui, J. Zhang, Y. Du, Y. Zhou, and X. Lu, “Neighbor-augmented knowledge graph attention network for recommendation,” *Neural Processing Letters*, vol. 55, no. 6, pp. 8237–8253, 12 2023.

[93] F. Wang, Z. Zheng, Y. Zhang, Y. Li, K. Yang, and C. Zhu, “To see further: Knowledge graph-aware deep graph convolutional network for recommender systems,” *Information Sciences*, vol. 647, p. 119465, 2023.

[94] J. Peng, J. Gong, C. Zhou, Q. Zang, X. Fang, K. Yang, and J. Yu, “Kgcfrec: Improving collaborative filtering recommendation with knowledge graph,” *Electronics*, vol. 13, no. 10, 2024.

[95] H. Yang and C. Yang, “Graph convolutional recommendation system based on bilateral attention mechanism,” *Journal of Engineering*, vol. 2024, no. 1, p. 2978680, 2024.

[96] B. Huang, W. Yu, R. Xie, J. Luo, J. Xiao, and J. Huang, “Lorentz equivariant model for knowledge-enhanced hyperbolic collaborative filtering,” *Knowledge-Based Systems*, vol. 291, p. 111590, 2024.

[97] D. Sakong, V. H. Vu, T. T. Huynh, P. L. Nguyen, H. Yin, Q. V. H. Nguyen, and T. T. Nguyen, “Higher-order knowledge-enhanced recommendation with heterogeneous hypergraph multi-attention,” *Information Sciences*, vol. 680, p. 121165, 2024.

[98] G. Li, L. Yang, S. Bai, X. Song, Y. Ren, and S. Liu, “Bikagcn: Knowledge-aware recommendations under bi-layer graph convolutional networks,” *Neural Processing Letters*, vol. 56, no. 1, p. 20, 2 2024.

[99] N. Sathya, S. Duraisamy, and T. Latha Maheshwari, “Time-aware attention and knowledge graph embedding in deep learning model for improving customer preference based recommendations,” *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, p. 808–818, 8 2024. [Online]. Available: http://dx.doi.org/10.22266/IJIES2024.0831.61

[100] A. Li, X. Liu, and B. Yang, “Item attribute-aware graph collaborative filtering,” *Expert Systems with Applications*, vol. 238, p. 122242, 2024.

[101] L. e Wang, Y. Qi, Z. Sun, and X. Li, “Mknbl: Joint multi-channel knowledge-aware network and broad learning for sparse knowledge graph-based recommendation,” *Neurocomputing*, vol. 575, p. 127277, 2024.

[102] Q. Ai, V. Azizi, X. Chen, and Y. Zhang, “Learning heterogeneous knowledge base embeddings for explainable recommendation,” *Algorithms*, vol. 11, no. 9, 2018. [Online]. Available: https://www.mdpi.com/1999-4893/11/9/137

[103] B. Yu, C. Zhou, C. Zhang, G. Wang, and Y. Fan, “A privacy-preserving multi-task framework for knowledge graph enhanced recommendation,” *IEEE Access*, vol. 8, pp. 115 717–115 727, 2020.

[104] Y. Wei, X. Wang, L. Nie, S. Li, D. Wang, and T.-S. Chua, “Causal inference for knowledge graph based recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 153–11 164, 2023.

[105] Z. Lyu, Y. Wu, J. Lai, M. Yang, C. Li, and W. Zhou, “Knowledge enhanced graph neural networks for explainable recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4954–4968, 2023.

[106] Y. Ge, J. Ma, L. Zhang, X. Li, and H. Lu, “Trustworthiness-aware knowledge graph representation for recommendation,” *Knowledge-Based Systems*, vol. 278, p. 110865, 2023.

[107] X. Huang, J. Wang, and J. Cui, “A personalized collaborative filtering recommendation system based on bi-graph embedding and causal reasoning,” *Entropy*, vol. 26, no. 5, 2024.

[108] L. Eytan, V. Bogina, I. Ben-Gal, and N. Koenigstein, “Kpar: Knowledge-aware path-based attentive recommender with interpretability,” *ACM Trans. Recomm. Syst.*, 6 2024, just Accepted.

[109] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, “Ripplenet: Propagating user preferences on the knowledge graph for recommender systems,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 417–426.

[110] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, “Kgat: Knowledge graph attention network for recommendation,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 950–958.

[111] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 974–983.

[112] Z. Wang, G. Lin, H. Tan, Q. Chen, and X. Liu, “Ckan: Collaborative knowledge-aware attentive network for recommender systems,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 219–228.

[113] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems*, vol. 5, no. 4, p. 1–19, 12 2015. [Online]. Available: http://dx.doi.org/10.1145/2827872

[114] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[115] C. J. Tanis, “The seven principles of online learning: Feedback from faculty and alumni on its importance for teaching and learning,” *Research in Learning Technology*, vol. 28, no. 0, 3 2020.

[116] A. A. Lew, J. M. Cheer, M. Haywood, P. Brouder, and N. B. Salazar, “Visions of travel and tourism after the global covid-19 transformation of 2020,” *Tourism Geographies*, vol. 22, no. 3, pp. 455–466, 2020.

APPENDIX A
SEARCH QUERIES

A. SCOPUS Query

For SCOPUS, the results were limited to *articles* published during the pre-established years, written in English, open access (all, gold, hybrid gold, and green), and the query was searched within the articles' title, abstract, and keywords. The complete query is as follows: TITLE-ABS-KEY (("ontology" OR "knowledge" OR "knowledge-based" OR "ontology-based") AND ("recommender" OR "recommendation" AND "recommend") AND ("collaborative" AND ("filter" OR "filtering"))) AND PUBYEAR > 2011 AND PUBYEAR < 2025 AND (LIMIT-TO (DOCTYPE, "ar")) AND (LIMIT-TO (LANGUAGE, "English")) AND (LIMIT-TO (OA, "all") OR LIMIT-TO (OA, "repository") OR LIMIT-TO (OA, "publisherfullgold"))

B. Web of Science Query

For Web of Science, the results were limited to *articles* published during the pre-established years, written in English, and open access (all, gold, hybrid gold, and green). The complete query is as follows: ("ontology" OR "knowledge" OR "knowledge-based" OR "ontology-based") AND ("recommender" OR "recommendation" AND "recommend") AND ("collaborative" AND ("filter" OR "filtering")) (Topic) OR ("ontology" OR "knowledge" OR "knowledge-based" OR "ontology-based") AND ("recommender" OR "recommendation" AND "recommend") AND ("collaborative" AND ("filter" OR "filtering")) (Title) OR ("ontology" OR "knowledge" OR "knowledge-based" OR "ontology-based") AND ("recommender" OR "recommendation" AND "recommend") AND ("collaborative" AND ("filter" OR "filtering")) (Abstract) and Article (Document Types) and English (Languages) and All Open Access or Gold or Gold-Hybrid or Free to Read or Green Published (Open Access) and 2024 or 2023 or 2022 or 2021 or 2020 or 2019 or 2018 or 2017 or 2016 or 2014 or 2015 or 2013 or 2012 (Publication Years)

C. ACM Library Query

For the ACM Library, the results were limited to *research articles* published during the pre-established years, and the query was searched within the articles' title, abstract, and keywords. The complete query is as follows: [[[Title: "ontology"] OR [Title: "knowledge"] OR [Title: "knowledge-based"] OR [Title: "ontology-based"]] AND [[Title: "recommender"] OR [[Title: "recommendation"] AND [Title: "recommend"]]] AND [Title: "collaborative"] AND [[Title: "filter"] OR [Title: "filtering"]]] OR [[[Abstract: "ontology"] OR [Abstract: "knowledge"] OR [Abstract: "knowledge-based"] OR [Abstract: "ontology-based"]] AND [[Abstract: "recommender"] OR [[Abstract: "recommendation"] AND [Abstract: "recommend"]]] AND [Abstract: "collaborative"] AND [[Abstract: "filter"] OR [Abstract: "filtering"]]] OR [[[Keywords: "ontology"] OR [Keywords: "knowledge"] OR [Keywords: "knowledge-based"] OR [Keywords : "ontology-based"]] AND [[Keywords: "recommender"] OR [[Keywords: "recommendation"] AND [Keywords: "recommend"]]] AND [Keywords: "collaborative"] AND [[Keywords: "filter"] OR [Keywords: "filtering"]]] AND [E-Publication Date: (01/01/2012 TO 12/31/2024)]

APPENDIX B
RECOMMENDER DOMAINS

Table IV presents a comprehensive breakdown of the relationship between each individual article and its respective domains. The included domain *Any* refers to all articles that did not explicitly mention their domain or that prove that it works for several domains.

APPENDIX C
EVALUATION METRICS

Table V, Table VI and Table VII present a comprehensive breakdown of the relationship between each individual article and the evaluation metrics they use. It is important to mention that article [68] did not properly indicate the evaluation metrics used for evaluating their recommender.

A. Quantitative Evaluation Metrics

Table V and Table VI present the relationship between all the quantitative evaluation metrics used per article. The included metrics are Mean Absolute Error, Root Mean Square Error and Mean Square Error, Precision, Recall, F1-score, Normalized Discounted Cumulative Gain, Hit Ratio, Average Overall Rating, ROC Curve, Area Under Curve, Accuracy, Mean Average Precision, Coverage Rate, Execution Time, Correct Rate of Pretest and Test Questions (CRPTQ), Comparing the success results (CSR), and Efficiency of Students (ES).

B. Qualitative Evaluation Metrics

Table VII presents the relationship between all the qualitative evaluation metrics used per article. The included metrics are User Satisfaction and Recommendation Effect Analysis.

TABLE IV. ARTICLES PERTAINING TO EACH DOMAIN

Domain	Citation
Any	[102], [103], [55], [56], [57], [79], [81], [82], [83], [84], [86], [87], [88], [68], [90], [104], [91], [105], [92], [93], [106], [74], [75], [94], [95], [96], [97], [107], [98], [99], [100], [101], [108]
Education	[47], [60], [61], [63], [80], [65], [69], [70], [71]
Tourism	[48], [51], [59], [67]
Entertainment	[78], [72], [73]
Cloud Services	[54], [58], [89]
E-commerce	[50], [66], [76]
Cross-domain	[53], [64]
Micro-blog	[49]
Manufacturing	[52]
IoT Services	[62]
Commodity entities	[85]
Healthcare	[77]

TABLE V. QUANTITATIVE METRICS USED PER ARTICLE (FIRST PART). THE INCLUDED METRICS ARE MEAN ABSOLUTE ERROR (MAE), ROOT MEAN SQUARE ERROR AND MEAN SQUARE ERROR (R-MSE), PRECISION (PREC), RECALL (REC), F1-SCORE (F1), NORMALIZED DISCOUNTED CUMULATIVE GAIN (NDCG), HIT RATIO (HR), AVERAGE OVERALL RATING (AOR), ROC CURVE (ROC), AREA UNDER CURVE (AUC), ACCURACY (ACC), MEAN AVERAGE PRECISION (MAP), COVERAGE RATE (CR), EXECUTION TIME (ET), CORRECT RATE OF PRETEST AND TEST QUESTIONS (CRPTQ), COMPARING THE SUCCESS RESULTS (CSR), BPR LOSS (BPR), DIVERSITY (DIV), MEAN PERCENTILE RANK (MPR), EFFICIENCY OF STUDENTS (ES), AND MEAN RECIPROCAL RANK (MRR)

Citation	MAE	R-MSE	Prec	Rec	F1	NDCG	HR	AOR	ROC	AUC	ACC	MAP	CR	ET	CRPTQ	CSR	BPR	DIV	MPR	ES	MRR
[47]	✓																				
[48]	✓																				
[49]			✓	✓																	
[78]	✓		✓	✓																	
[102]			✓	✓		✓	✓														
[50]								✓													
[51]			✓	✓									✓								
[52]			✓	✓	✓																
[53]			✓	✓					✓				✓								
[54]			✓	✓	✓																
[103]		✓	✓	✓						✓	✓										
[55]						✓	✓														
[56]			✓	✓		✓															
[57]	✓	✓	✓	✓	✓																
[58]			✓	✓	✓																
[79]				✓		✓															
[59]			✓	✓	✓																
[60]															✓					✓	
[61]			✓	✓	✓																
[62]	✓		✓	✓	✓						✓										
[63]	✓	✓																			
[80]					✓						✓										
[64]		✓	✓	✓	✓																
[65]			✓	✓						✓											
[81]				✓	✓	✓				✓											
[82]				✓		✓															
[83]					✓					✓											
[84]				✓		✓															
[85]					✓					✓											
[66]																✓					

TABLE VI. QUANTITATIVE METRICS USED PER ARTICLE (SECOND PART). THE INCLUDED METRICS ARE MEAN ABSOLUTE ERROR (MAE), ROOT MEAN SQUARE ERROR AND MEAN SQUARE ERROR (R-MSE), PRECISION (PREC), RECALL (REC), F1-SCORE (F1), NORMALIZED DISCOUNTED CUMULATIVE GAIN (NDCG), HIT RATIO (HR), AVERAGE OVERALL RATING (AOR), ROC CURVE (ROC), AREA UNDER CURVE (AUC), ACCURACY (ACC), MEAN AVERAGE PRECISION (MAP), COVERAGE RATE (CR), EXECUTION TIME (ET), CORRECT RATE OF PRETEST AND TEST QUESTIONS (CRPTQ), COMPARING THE SUCCESS RESULTS (CSR), BPR LOSS (BPR), DIVERSITY (DIV), MEAN PERCENTILE RANK (MPR), EFFICIENCY OF STUDENTS (ES), AND MEAN RECIPROCAL RANK (MRR)

Citation	MAE	R-MSE	Prec	Rec	F1	NDCG	HR	AOR	ROC	AUC	ACC	MAP	CR	ET	CRPTQ	CSR	BPR	DIV	MPR	ES	MRR
[67]			✓	✓	✓																
[86]			✓	✓						✓											
[87]					✓					✓											
[88]	✓	✓	✓																		
[69]	✓	✓																			
[89]						✓						✓									
[90]					✓					✓											
[104]				✓		✓															
[91]				✓		✓															
[105]	✓	✓	✓	✓	✓																
[92]				✓	✓	✓				✓											
[71]			✓	✓	✓																
[72]	✓	✓				✓															
[93]				✓		✓															
[106]			✓	✓	✓		✓														
[73]				✓							✓										
[74]	✓																				
[75]			✓	✓										✓							
[76]	✓		✓	✓																	
[94]				✓		✓											✓				
[95]				✓	✓					✓											
[96]				✓		✓															
[97]				✓		✓															
[107]			✓	✓		✓	✓														
[98]				✓		✓															
[99]			✓	✓								✓									
[77]			✓	✓		✓						✓									✓
[100]			✓	✓	✓																
[101]	✓	✓		✓	✓						✓							✓			
[108]			✓	✓		✓	✓												✓		

TABLE VII. QUALITATIVE METRICS USED PER ARTICLE. THE INCLUDED METRICS ARE USER SATISFACTION (SATISFACTION), AND RECOMMENDATION EFFECT ANALYSIS (REA)

Citation	Satisfaction	REA
[54]	✓	
[60]		✓
[88]	✓	
[70]	✓	
[76]	✓	