Advancements in Hybrid Recommendation Systems

Mr Vikas Arya
Assistant Professor
Department of Computer Science Engineering
Galgotias University, Greater Noida
vikas.arya@galgotiasuniversity.edu.in

Harshit Srivastava
Department of Computer Science Engineering
Galgotias University, Greater Noida
harshitsrivastava2902@gmail.com

Keshav Kumar Jha
Department of Computer Science Engineering
Galgotias University, Greater Noida
jhakeshav261@gmail.com

Abstract—Recommendation systems have become indispensable tools for navigating the vast digital landscape, enhancing user experience by personalizing content and product suggestions. While foundational techniques like Collaborative Filtering (CF) and Content-Based Filtering (CBF) have been influential, they exhibit inherent weaknesses such as data sparsity, the cold-start problem, and limited recommendation diversity. Hybrid Recommendation Systems (HRS) represent a significant advancement, strategically combining multiple recommendation approaches to overcome these limitations and improve overall performance. This paper presents a comprehensive review of the key advancements in HRS. We trace the evolution from foundational hybridization strategies to the integration of sophisticated machine learning techniques, including Matrix Factorization and various Deep Learning architectures (e.g., NCF, RNNs, GNNs). The paper further explores the crucial role of incorporating external knowledge through Knowledge Graphs (KGs) and leveraging contextual information for more relevant and timely suggestions. Finally, we examine contemporary challenges and future research directions, encompassing explainability, fairness, scalability, cross-domain applications, and the critical need for evaluation metrics that capture aspects beyond predictive accuracy, such as novelty and diversity. This work synthesizes findings from numerous academic studies to provide a cohesive understanding of the state-of-the-art and trajectory of hybrid recommendation systems.

Keywords—*Hybrid Recommendation Systems, Collaborative Filtering, Content-Based Filtering, Deep Learning, Knowledge Graphs, Matrix Factorization, Recommender Systems, Cold Start, Data Sparsity, Context-Awareness, Explainability, Fairness, Evaluation Metrics.*

I. INTRODUCTION

The exponential growth of online information, products, and services has led to a situation often termed "information overload." Users struggle to find items that match their interests amidst a sea of possibilities. Recommendation Systems (RS) emerged as a critical solution, acting as personalized filters that predict user preferences and suggest relevant items [1]. Initial efforts focused on single-algorithm approaches, primarily Collaborative Filtering (CF), which leverages user-item interaction patterns [2], and Content-Based Filtering (CBF), which utilizes item attributes and user profiles [3].

However, these monolithic approaches face significant challenges. CF systems are notoriously hampered by the "cold-start" problem – difficulty recommending to new users or new items lacking sufficient interaction data – and data sparsity, where the user-item interaction matrix is mostly empty [4], [5]. CBF systems, while mitigating the new item problem, often suffer from overspecialization, limiting the discovery of novel items, and depend heavily on the availability and quality of item metadata [3], [6].

Recognizing these limitations, researchers proposed Hybrid Recommendation Systems (HRS) that combine two or more recommendation techniques [7], [8]. The core principle is synergy: leveraging the strengths of one method to compensate for the weaknesses of others. This often results in improved accuracy, robustness against data limitations, increased recommendation diversity, and better handling of cold-start scenarios [9], [10]. Early HRS often involved straightforward combinations of CF and CBF, guided by foundational hybridization frameworks [7].

Since these initial efforts, the field of HRS has undergone substantial evolution, driven largely by advancements in machine learning, particularly the rise of latent factor models, deep learning, graph representation learning, and the increasing availability of structured knowledge and contextual data. This paper surveys these critical advancements, charting the progress from basic hybrids to the complex, data-rich systems being developed today. Section II revisits the fundamentals of HRS. Section III discusses the impact of Matrix Factorization and early Deep Learning models. Section IV delves into modern deep learning architectures used in hybrids. Section V explores the integration of Knowledge Graphs and Context-Awareness. Section VI examines current challenges and future research directions, followed by a concluding summary in Section VII.

II. FOUNDATIONS OF HYBRID RECOMMENDATION

The development of effective HRS builds upon an understanding of the core recommendation paradigms and the structured ways in which they can be combined.

A. Core Recommendation Techniques Revisited

1. *Collaborative Filtering (CF)*: Exploits similarities in user behavior. User-based CF finds similar users; Item-based CF finds similar items based on user interaction patterns [2]. Strengths include domain independence and potential for serendipity. Weaknesses are cold-start, sparsity, and scalability [5], [11].
2. *Content-Based Filtering (CBF)*: Matches item attributes (e.g., genres, keywords) to user profiles built from past preferences [3]. Strong for new items if content is available, avoids user cold-start. Weaknesses include overspecialization and reliance on feature engineering [6], [11].
3. *Knowledge-Based Recommendation (KBR)*: Uses explicit domain knowledge, constraints, and potentially user requirements gathered through interaction [7]. Effective for items requiring deep user understanding (e.g., financial services) and less frequent purchases. Can handle cold-start but requires knowledge engineering.

B. Why Hybridize? Addressing Core Limitations

Hybridization is primarily motivated by the need to overcome the specific weaknesses inherent in single-method systems [8], [9], [10]:

- **Cold Start & Sparsity:** Combining interaction-based methods (CF) with attribute-based (CBF) or knowledge-based methods allows recommendations even with limited interaction data [4], [5].
- **Accuracy & Robustness:** Integrating signals from different data sources (interactions, content, context, knowledge) often yields more accurate and reliable preference predictions [9].
- **Diversity & Novelty:** Merging CF's ability to cross genres with CBF's specificity can lead to recommendations that are both relevant and diverse, avoiding filter bubbles [11], [12].
- **Synergy:** Leveraging intermediate outputs, such as using content features to improve CF similarity calculations or using CF predictions to augment user profiles for CBF.

C. Foundational Hybridization Strategies

Burke's taxonomy provides a widely adopted framework for classifying how recommendation components can be combined [7]:

1. **Weighted:** Combines scores/ranks from multiple recommenders using a formula (static or dynamic weights).
2. **Switching:** Selects one recommender based on the current context or data characteristics (e.g., use CBF for new users, CF for established ones).
3. **Mixed:** Presents recommendations from different systems together (e.g., separate lists on a webpage).
4. **Feature Combination:** Uses features derived from one technique as input for another (e.g., CF rating predictions as input to a CBF model).
5. **Cascade:** One recommender filters or refines the candidate set generated by another.
6. **Feature Augmentation:** Uses one technique to generate additional data points (e.g., predicted ratings) to make the input for another technique denser.
7. **Meta-Level:** Uses the entire learned model of one recommender as input for another (e.g., using a content-based profile in a CF algorithm).

These strategies provide the architectural blueprints upon which more advanced machine learning techniques have been integrated.

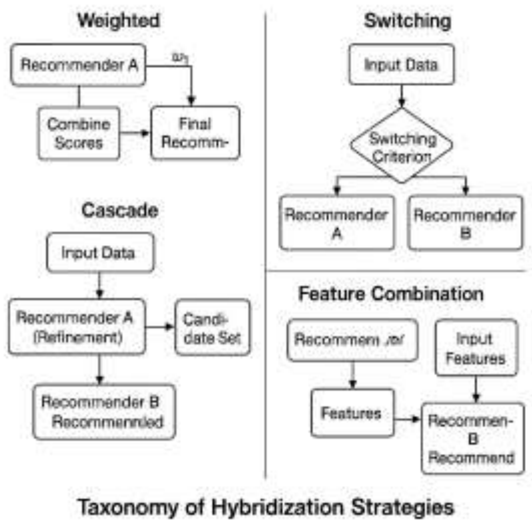

Fig 1. Taxonomy of Hybridization Strategies

III. KEY ADVANCEMENTS: LATENT FACTORS AND EARLY DEEP LEARNING

The integration of more sophisticated machine learning models marked a significant leap forward for HRS.

A. The Role of Matrix Factorization (MF)

Matrix Factorization techniques fundamentally changed collaborative filtering by modeling users and items through low-dimensional latent vectors learned from the interaction matrix [13]. MF models like Singular Value Decomposition (SVD) variants and Probabilistic Matrix Factorization (PMF) proved highly effective at predicting user ratings, especially in sparse datasets. Their integration into hybrid systems became widespread [14]:

- **MF + Content/Attributes:** Incorporating item features or user demographics directly into the MF optimization process (e.g., Factorization Machines [15], SVD++).
- **MF + Neighborhood Models:** Combining the global latent factor view of MF with the local neighborhood perspective of traditional CF [13].
- **MF as Feature Generator:** Using the learned latent factors from MF as rich input features for other recommendation algorithms (e.g., classification or ranking models).

B. Initial Integration of Deep Learning (DL)

Deep learning began influencing HRS by offering powerful tools for automatic feature representation learning and capturing non-linear relationships [16], [17]:

1. *Autoencoders for CF:* Autoencoders were used to learn compact, non-linear representations of the high-dimensional and sparse user-item interaction vectors, improving upon linear methods like MF for collaborative filtering tasks [18]. Denoising autoencoders were particularly useful for handling noisy interaction data.
2. *Neural Collaborative Filtering (NCF):* The NCF framework explicitly generalized MF by using neural networks (specifically Multi-Layer Perceptrons - MLPs) to learn the complex interaction function between user and item latent factors, going beyond the simple dot product used in MF [19]. NCF architectures often combine linear (GMF) and non-linear (MLP) interaction modeling, representing an inherently hybrid deep learning approach.
3. *DL for Content Representation:* Deep learning models, especially Convolutional Neural Networks (CNNs) for text (reviews, descriptions) and images, and embedding layers for categorical attributes, enabled the learning of dense, semantic representations of item content [20]. These deep content features could be fused with collaborative signals far more effectively than traditional sparse features (e.g., TF-IDF).

These early DL applications demonstrated the potential to move beyond linear assumptions and handcrafted features, paving the way for more complex hybrid architectures.

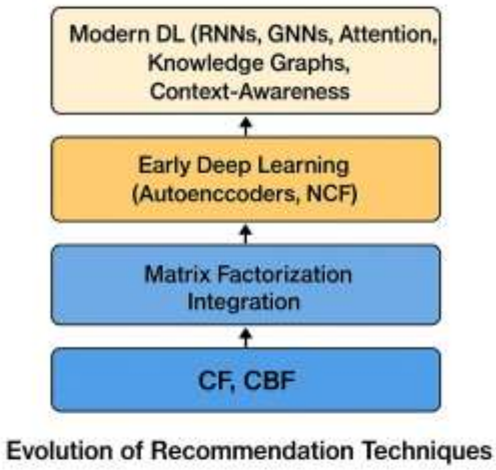

Fig 2. Evolution of Recommendation Techniques

IV. MODERN DEEP LEARNING ARCHITECTURES IN HYBRIDS

Recent advancements leverage more specialized deep learning architectures to capture complex dependencies and data modalities within HRS.

A. Modeling Sequences with RNNs

User interactions are often sequential (e.g., Browse history, playlist listening order). Recurrent Neural Networks (RNNs), particularly LSTMs and GRUs, are well-suited to model these temporal dependencies [21]. Session-based recommenders use RNNs to predict the next item a user might interact with based on their current session activity. In hybrid contexts, RNNs can model user dynamics, and their output (e.g., predicted next item probability, user state vector) can be combined with other signals like long-term preferences derived from CF or content features [22].

B. Leveraging Graph Structures with GNNs

The user-item interaction data, along with item attributes and user relationships, can often be naturally represented as graphs. Graph Neural Networks (GNNs) have emerged as powerful tools for learning from such graph-structured data [23]. GNNs learn node embeddings (for users and items) by iteratively aggregating feature information from neighboring nodes. This process inherently captures collaborative filtering effects (information propagation between similar users/items) and can easily incorporate node features (content, attributes) [24]. GNN-based recommenders (using GCNs, GraphSAGE, GATs) often form the core of modern hybrid systems, integrating interaction patterns, content features, and potentially knowledge graph information in an end-to-end manner [25], [26].

C. Attention Mechanisms for Contextual Weighting

Attention mechanisms allow models to dynamically focus on the most relevant parts of the input when making a prediction [27]. In HRS, attention can be used to:

- Weight the importance of different items in a user's interaction history when modeling sequential preferences (e.g., in RNN or Transformer-based models).
- Assign importance scores to different neighbors when aggregating information in GNNs.
- Selectively attend to relevant features from item content or user profiles.
- Adapt recommendations based on the current context.

Attention adds a layer of interpretability and often improves performance by allowing the model to adapt its focus dynamically.

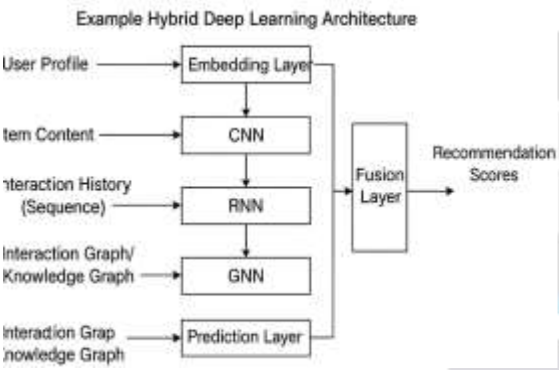

Fig 3. Hybrid Deep Learning Architecture

V. INCORPORATING RICHER INFORMATION: KNOWLEDGE GRAPHS & CONTEXT

Moving beyond user-item interactions and basic content, modern HRS increasingly leverage structured external knowledge and dynamic contextual information.

A. Hybrid Systems with Knowledge Graphs (KGs)

Knowledge Graphs provide structured relational data about items, attributes, and related entities (e.g., movie -> director -> genre). Integrating KGs into HRS offers significant benefits [28], [29]:

- **Data Enrichment:** KGs provide rich side information, helping to alleviate sparsity and cold-start problems, especially for long-tail items.
- **Enhanced Reasoning:** Systems can leverage multi-hop relationships in the KG to find connections between users and items that are not apparent from interaction data alone.
- **Explainability:** The paths traced in the KG can serve as justifications for recommendations, enhancing user trust [28].

Hybrid KG-based methods include [29], [30]:

1. *Embedding-Based:* Learn embeddings for entities and relations in the KG and combine them with user/item embeddings from CF models (e.g., using multi-task learning frameworks like MKR [31]).
2. *Path-Based:* Explicitly model and score paths between users and items in the KG (e.g., RippleNet [32] propagates user preferences along KG paths; KGAT [25] uses attention over paths).
3. *GNN-Based:* Apply GNNs directly to the KG or a combined user-item-KG graph to learn representations that fuse structural, relational, and collaborative information [26].

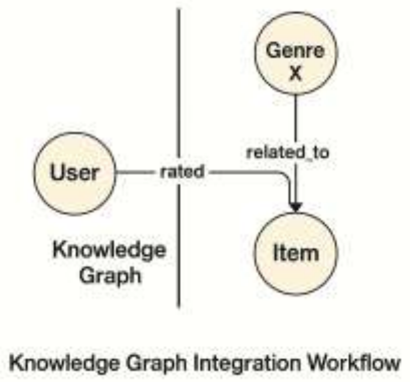

Fig 4. Knowledge Graph Workflow

B. Context-Aware Hybrid Recommendation

User preferences are rarely static; they often depend on context, such as time of day, location, device, current activity, or social setting [33]. Context-Aware Recommender Systems (CARS) explicitly incorporate such contextual information. Hybrid approaches are prevalent in CARS, as context needs to be combined with core user preferences derived from CF or CBF [33], [34]. Common strategies include [35]:

1. **Contextual Pre-filtering:** Use context to select or filter the relevant data *before* applying a standard recommendation algorithm.
2. **Contextual Post-filtering:** Generate recommendations using a standard algorithm and then filter or re-rank them based on the current context.
3. **Contextual Modeling:** Integrate contextual factors directly into the recommendation model itself (e.g., as

additional features in MF or DL models, using tensor factorization).

Deep learning models are increasingly used for contextual modeling, allowing for complex interactions between context variables and user/item features [36].

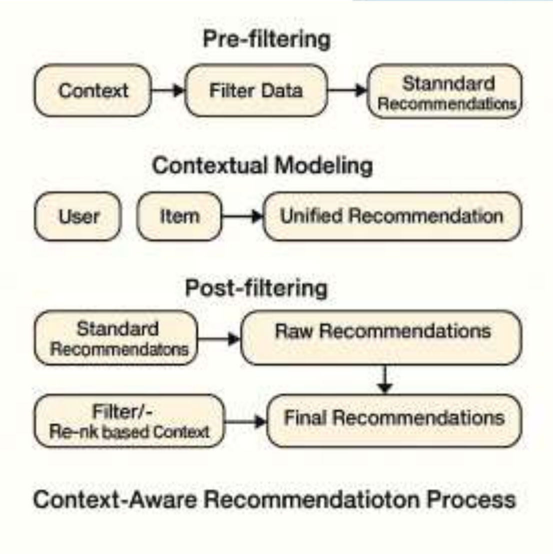

Fig 5. Context-Aware Recommendation Process

VI. CONTEMPORARY CHALLENGES & FUTURE DIRECTIONS

Despite remarkable progress, several critical challenges shape the current research landscape and future directions for HRS.

A. Explainability and Interpretability (XAI)

As hybrid models, especially those using deep learning, become increasingly complex "black boxes," providing meaningful explanations for recommendations is vital for user trust, debugging, and system transparency [28], [37]. Research focuses on generating post-hoc explanations or designing inherently interpretable hybrid models, often leveraging KG paths or attention weights.

B. Fairness, Bias, and Transparency

HRS can inherit and amplify biases present in historical data, leading to unfair outcomes for certain user groups or item providers (e.g., popularity bias, demographic bias) [38], [39]. Developing fairness-aware HRS involves defining fairness metrics, detecting bias, and designing mitigation strategies (e.g., data augmentation, adversarial training, re-ranking algorithms) without unduly compromising recommendation quality [39].

C. Scalability and Real-time Adaptation

Handling web-scale datasets with millions of users/items and adapting to rapidly changing user interests and item catalogs in real-time remain significant engineering challenges [11], [5]. Research explores distributed training, efficient indexing, incremental model updates, and architectures optimized for low-latency inference.

D. Cross-Domain Recommendation

Leveraging knowledge from auxiliary domains to improve recommendations in a target domain (where data might be sparser) is a promising direction [40]. Hybrid models using transfer learning, multi-task learning, or shared latent representations are key enablers for effective cross-domain recommendation.

E. Evaluation Beyond Accuracy

Over-reliance on prediction accuracy metrics (like RMSE, Precision@k) can lead to systems that are accurate but boring or unhelpful [41], [42]. Evaluating and optimizing for metrics like diversity (variety of recommended items), novelty (recommending unknown items), serendipity (surprising yet relevant items), and coverage (proportion of the item catalog recommended) is crucial for better user experience [42], [43], [44]. Developing reliable online and offline evaluation protocols that capture these multi-faceted goals remains an active area.

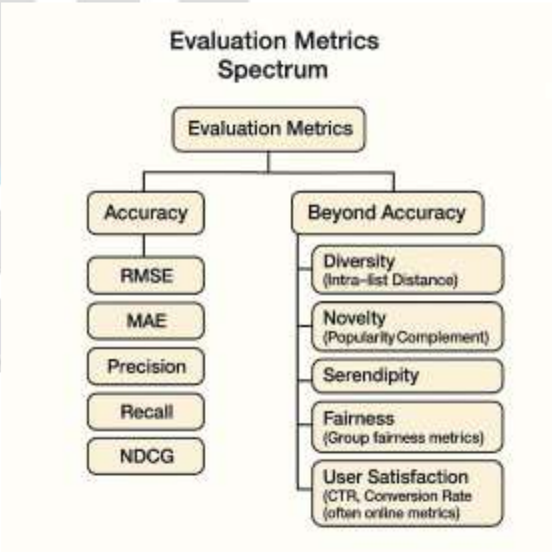

Fig 6. Evaluation Metrics Spectrum

VII. CONCLUSION

Hybrid Recommendation Systems represent a mature yet continually evolving field, driven by the need to overcome the limitations of traditional recommendation algorithms. Starting from foundational strategies combining collaborative filtering and content-based approaches, the field has embraced significant advancements from machine learning, integrating Matrix Factorization, a diverse array of Deep Learning architectures (NCF, RNNs, GNNs, Attention), and external knowledge sources like Knowledge Graphs. These innovations have enabled HRS to model increasingly complex user preferences, sequential behaviors, item attributes, contextual nuances, and relational information, leading to substantial improvements in accuracy, robustness, and the ability to handle challenges like data sparsity and cold starts.

Modern research increasingly focuses not only on predictive accuracy but also on crucial qualitative aspects such as explainability, fairness, diversity, and novelty. Incorporating context-awareness and enabling cross-domain recommendations are also key frontiers. Addressing these challenges while ensuring scalability and real-time adaptability for massive online platforms remains paramount. The future of hybrid recommendation systems lies in developing more intelligent, transparent, fair, and contextually adaptive systems that provide truly personalized and valuable experiences, effectively balancing accuracy with other user-centric quality dimensions. The continued synthesis of diverse algorithmic approaches and data sources promises to further enhance the capabilities and impact of these essential information filtering tools.

BIBLIOGRAPHY

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734-749, 2005. [2] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th International Conference on World Wide Web (WWW '01), 2001, pp. 285-295. [3] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in The Adaptive Web, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer Berlin Heidelberg, 2007, pp. 325-341. [4] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02), 2002, pp. 253–260. [5] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based Systems, vol. 46, pp. 109-132, 2013. [6] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in Recommender Systems Handbook, F. Ricci et al., Eds. Springer US, 2011, pp. 73-105. [7] R. Burke, "Hybrid recommender systems: Survey and experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, 2002. [8] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," Intelligent Data Analysis, vol. 21, no. 6, pp. 1487-1524, 2017. [9] K. C. Bodduluri et al., "Exploring the Landscape of Hybrid Recommendation Systems in E-Commerce: A Systematic Literature Review," IEEE Access, vol. 12, pp. 28276-28293, 2024. [10] F. Z. Trabelsi, A. Khtira, and B. El Asri, "Hybrid Recommendation Systems: A State of Art," in Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2021), 2021, pp. 281-288. [11] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009. [12] P. Castells, S. Vargas, and J. Wang, "Novelty and diversity metrics for recommender systems: choice, discovery and relevance," arXiv preprint arXiv:1101.4571, 2011. (Also see CIKM Workshop) [13] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 30-37, 2009. [14] K. K. Yadav, H. K. Soni, G. Yadav, and M. Sharma, "Collaborative Filtering Based Hybrid Recommendation System Using Neural Network and Matrix Factorization Techniques," International Journal of Intelligent Systems and Applications in Engineering, vol. 11, no. 10s, pp. 447-457, 2023. [15] S. Rendle, "Factorization machines," in 2010 IEEE International Conference on Data Mining, 2010, pp. 995-1000. [16] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM Computing Surveys (CSUR), vol. 52, no. 1, pp. 1-38, 2019. [17] M. Batmaz, Z., Yurekli, A., Bilge, A. et al. "A review on deep learning for recommender systems: challenges and remedies." Artif Intell Rev 52, 1–37 (2019). [18] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion), 2015, pp. 111–112. [19] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in Proceedings of the 26th International Conference on World Wide Web (WWW '17), 2017, pp. 173-182. [20] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15), 2015, pp. 1235–1244. [21] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), 2016. [22] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '16), 2016, pp. 729–732. [23] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017, pp. 1024-1034. [24] 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 (KDD '18),

2018, pp. 974–983. [25] X. Wang et al., "KGAT: Knowledge Graph Attention Network for Recommendation," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19), 2019, pp. 950-958. [26] Y. Gao, Y. Li, Y. Lin, H. Gao, L. Khan, "Deep Learning on Knowledge Graph for Recommender System: A Survey," arXiv preprint arXiv:2004.00387, 2020. [27] F. Guo, X. Zhang, Y. Wang, et al., "A Survey on Knowledge Graph-Based Recommender Systems," IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3549-3568, 2022. [28] C. Li, M. An, M. Li, Y. Liu, and S. Xie, "Explainable recommendation: A survey and new perspectives," ACM Computing Surveys (CSUR), vol. 55, no. 5, pp. 1-39, 2023. [29] G. Rizzo, F. Tommasi, and A. Cena, "Knowledge Graph-Based Recommender Systems: A Survey," ACM Computing Surveys, vol. 55, no. 9, pp. 1-37, 2023. [30] C. Xu et al., "Knowledge graph embedding based graph convolutional networks," in Proceedings of the Web Conference 2020 (WWW '20), 2020, pp. 2068-2078. [31] T. Wang, D. Lian, J. Yuan, X. Ge, and X. Xie, "Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation," in The World Wide Web Conference (WWW '19), 2019, pp. 2000-2010. [32] H. Wang et al., "RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems," in Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18), 2018, pp. 417-426. [33] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in Recommender Systems Handbook, F. Ricci et al., Eds. Springer US, 2011, pp. 217-253. [34] A. Katarya, S. K. Verma, "Context Aware Recommender Systems: A Review of Recent Trends, Challenges, and Techniques," arXiv preprint arXiv:2301.09485, 2023. [35] M. Unger, A. Bar, B. Shapira, and L. Rokach, "Towards latent context-aware recommendation systems," Knowledge-Based Systems, vol. 104, pp. 165-178, 2016. [36] A. Raza, C. Ding, "Deep Learning Based Context-Aware Recommender Systems: A State-of-the-Art Survey and Future Directions," ACM Computing Surveys (CSUR), vol. 54, no. 11s, pp. 1–36, 2022. [37] D. Jannach, L. P. Jing, M. Zanker, "Explainable Recommendation: A Survey of Recent Progress," arXiv preprint arXiv:2309.06517, 2023. [38] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," ACM Computing Surveys (CSUR), vol. 54, no. 6, pp. 1-35, 2021. [39] S. M. T. Islam, A. T. T. Chakraborti, M. T. M. Dip, and M. G. R. Alam, "A Survey on Fairness in Recommender Systems," arXiv preprint arXiv:2106.08488, 2021. [40] Z. Li, J. Zhang, Y. Zhao, X. Lian, and X. Xie, "A Survey on Cross-Domain Recommendation," ACM Transactions on Information Systems (TOIS), vol. 40, no. 4, pp. 1-42, 2022. [41] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in Recommender Systems Handbook, F. Ricci et al., Eds. Springer US, 2011, pp. 257-297. [42] S. Vargas and P. Castells, "Rank and relevance in novelty and diversity metrics for recommender systems," in Proceedings of the fifth ACM conference on Recommender systems (RecSys '11), 2011, pp. 121-128. [43] C. L. A. Clarke et al., "Overview of the TREC 2009 Web Track," in The Eighteenth Text REtrieval Conference (TREC 2009) Proceedings, 2009. [Example of evaluation focusing on diversity] [44] A. Said, B. Fields, D. Tikk, D. Jannach, "Evaluating Recommender Systems: Survey and Framework," ACM Computing Surveys (CSUR), vol. 51, no. 5, pp. 1-38, 2019.