International Journal of Computing and Digital Systems
2024, VOL. 17, NO. 1, 1–10
http://dx.doi.org/10.12785/ijcds/XXXXXX

Enhancing Personalization with Graph Neural Networks in Agile Recommendation Systems

Minal Patil[1], R H Goudar[1], Dhananjaya. G. M[1], Vijayalaxmi N. Rathod[1] and Anjanabhargavi A. Kulkarni[1]

[1]*Department CSE, Visvesvaraya Technological University, Belagavi, Karnataka, India 590018*

Abstract: Traditional recommender systems often struggle to adapt to rapidly changing user preferences and dynamic contexts. Agile Recommendation Systems (ARS) address this by incorporating real-time learning and context awareness. This survey explores recent ARS advancements, focusing on deep learning techniques for user modeling and context-aware recommendations. It highlights the challenges of evaluating ARS, particularly the lack of explainability and the potential of Graph Neural Networks (GNNs) and Reinforcement Learning (RL).This study proposes a novel ARS framework that combines GNNs, RL, and explainability for personalized and trustworthy recommendations. GNNs capture complex user-item relationships, and RL enables real-time adaptation. Explainability techniques are integrated to enhance user trust by providing insights into the recommendation rationale. It includes real-world situations that demonstrate how these techniques can be applied in practical recommendation systems, making the concepts more tangible and relevant.
This research offers:
A comprehensive review of ARS advancements and deep learning techniques.
A proposed framework with GNNs, RL, and explainability for enhanced personalization and user understanding.
A focus on addressing explainability challenges in GNN-RL based systems.
A discussion of potential biases within GNN and RL models and mitigation strategies.
By leveraging GNNs and Reinforcement Learning with explainability, this ARS framework has the potential to revolutionize the recommendation landscape, delivering more personalized and trustworthy experiences for users, ultimately enhancing user satisfaction , engagement and overall user experience.

Keywords: Recommender systems, agile recommendation system, context-awareness, user feedback integration, deep learning technique, graph neural network, reinforcement learning.

1. INTRODUCTION

The exponential growth of digital information has necessitated the development of sophisticated recommender systems (RS) to assist users in navigating the overwhelming volume of content. Content Recommendation Systems (CRS) have become integral to various online platforms, curating personalized experiences that enhance user engagement and satisfaction [1]. Traditional RS, however, often rely on static user profiles and inflexible models, leading to suboptimal recommendations.

Agile Recommendation Systems (ARS) have emerged as a promising solution to address these limitations. By continuously adapting to evolving user preferences and incorporating real-time feedback, ARS delivers more relevant and timely recommendations [2]. This paper explores the potential of ARS in revolutionizing the recommendation landscape.

Our study is focused on creating a revolutionary ARS framework that employs Graph Neural Networks (GNNs) and Reinforcement Learning (RL) to capture complicated user-item relationships and optimize suggestions in real time. By incorporating Explainable AI (XAI) techniques, we aim to enhance user trust and transparency. This paper contributes to the field by proposing a holistic approach that addresses the challenges of personalization, adaptability, explainability, and user experience in recommendation systems.

Importance of Explainability : Deep learning models have demonstrated impressive performance in recommendation tasks, their complex nature often hinders transparency and interpretability. To tackle this issue, explainable AI (XAI) has emerged as a vital component for building trust in recommendation systems. By providing insights into the factors influencing recommendations, XAI can enhance user satisfaction and acceptance [3].

E-mail address: minalangolkar1008@gmail.com, rhgoudar.vtu@gmail.com, gm.dhananjaya99@gmail.com,vijaylaxmirathod@gmail.com,anjanapramod2022@gmail.com

ARS demands explainable models due to dynamic recommendations and complex algorithms. Explainability is vital in ARS to build trust and improve recommendations [4].

Recent studies have highlighted the importance of user-centric explanations in XAI for recommendation systems.Several XAI techniques, including Local Interpretable Model-Agnostic Explanations (LIME), attention mechanisms, counterfactual explanations, and feature importance analysis, have been explored in this context [5]. Although these techniques demonstrate the potential for explaining recommendations, dedicated XAI methods are required for ARS to address the complexities introduced by GNNs and RL [6].

2. Literature Review

This survey conducted a comprehensive literature review of peer-reviewed articles published between January 2018 and December 2023 on Google Scholar, Scopus, and Web of Science using keywords such as agile recommendation systems, graph neural networks, reinforcement learning, explainability, and user feedback.

The review of the literature was completed through a systematic search of Google Scholar, Scopus, and Web of Science using the following keywords: agile recommendation systems, graph neural networks, reinforcement learning, explainability, user feedback, and personalization as shown in Figure 2. Inclusion criteria focused on peer-reviewed articles published in English that specifically addressed ARS, GNNs, or RL within the context of recommendation systems. Exclusion criteria included studies primarily focused on traditional recommender systems without an explicit ARS component.

Traditional Recommender Systems and their Limitations: Recommender systems (RS) have become a vital tool for navigating the overwhelming volume of digital information. Traditional RS, such as collaborative filtering and content-based approaches, are being used widely but face significant limitations. Collaborative filtering, recommends items based on user-item similarities, suffers from the cold-start problem and struggles to capture nuanced user preferences. Content-based methods, reliant on item attributes, often provide overspecialized recommendations and fail to adapt to evolving user tastes [7]. These limitations hinder the ability of traditional RS to deliver truly personalized and up-to-date recommendations.

The Emergence of Agile Recommendation Systems (ARS): To overcome the shortcomings of traditional RS, Agile Recommendation Systems (ARS) have emerged as a promising alternative. Recognizing the need for real-time adaptation, user feedback integration, and context awareness, ARS strives to deliver more personalized and relevant recommendations. By continuously learning from user interactions and evolving preferences, ARS aims to provide a superior user experience. Early research on ARS focused on incorporating user context, such as location, time, and device, into recommendation models. However, these initial efforts often relied on static user profiles and limited contextual information. Numerous recent investigations have examined the use of implicit feedback, such as click-through rates and browsing behaviour, to increase the precision of recommendations [8].

The Role of Deep Learning in ARS Building on the foundation laid by ARS, Deep learning methods have emerged as powerful tools to address the growing challenges of recommender systems. By leveraging the ability of deep learning models to capture complex patterns and relationships in large datasets, scholars have achieved noteworthy advancements in improving the capabilities of ARS.

Recurrent Neural Networks (RNNs) have been used to model temporal dependencies in user behaviour, allowing the system to adapt to changing preferences [9]. By extracting features from various content modalities using Convolutional Neural Networks (CNNs), recommendations become more comprehensive and informative [10]. However, the full potential of deep learning in ARS is yet to be realized, and further research is necessary to unlock its capabilities.

To thoroughly examine the challenges and capitalize on the opportunities within ARS, it is essential to compare existing approaches with proposed solutions. The following table presents a comparative analysis of different ARS approaches, highlighting their strengths, weaknesses, and focus areas.

Table I provides a broad overview of the key features and differences between traditional ARS and proposed ARS incorporating GNNs and RL. To provide a more detailed comparison, Table II offers a side-by-side look at various recommendation system approaches across different dimensions.
Existing: Traditional ARS approaches rely on collaborative filtering or content-based filtering techniques. These techniques struggle to capture complex user-item relationships and adapt to real-time user feedback.
Research Gap: The need for advancements in ARS that can handle complex user interactions and provide real-time personalization. Existing surveys might not place enough emphasis on emerging techniques or discuss the challenges of explainability, fairness, and privacy in ARS. Solution: This paper highlights the potential of emerging techniques like Graph Neural Networks (GNNs) and Reinforcement Learning (RL) in order to solve the shortcomings of existing ARS approaches.

(i) GNNs can model complex user-item relationships based on user interactions and social connections.

(ii) RL allows ARS to learn and adapt to user feedback

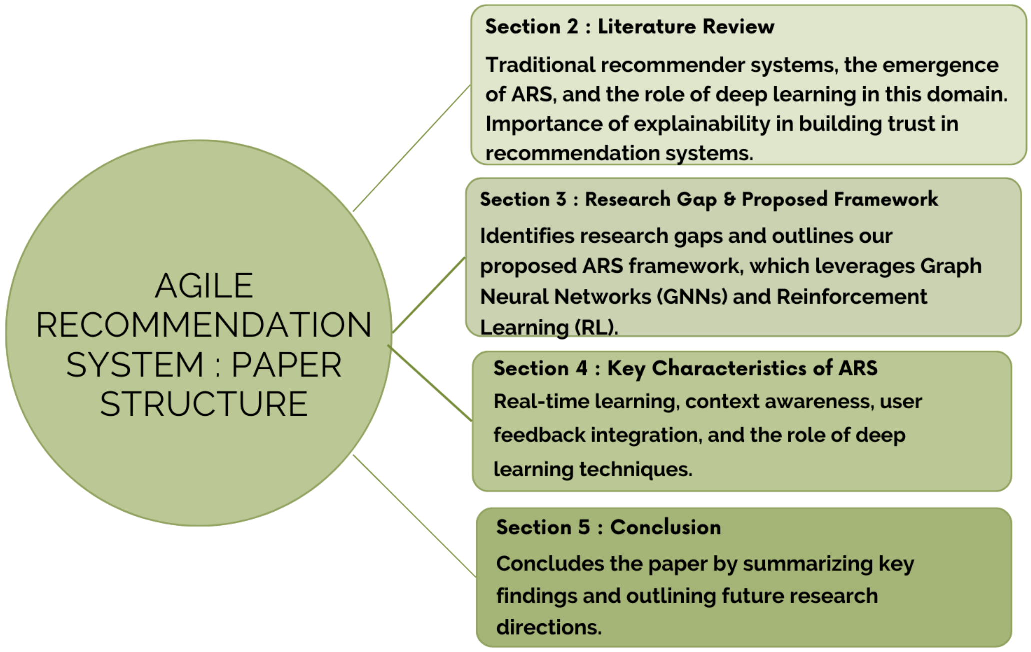

Figure 1. Structure of the paper

TABLE I. Existing vs. Proposed Approaches in Agile Recommendation Systems (ARS)

Aspect	Existing Approaches	Proposed Approach
User-Item Modeling	Collaborative Filtering, Matrix Factorization (Limitations: cold start, data sparsity) [11]	Graph Neural Networks (GNNs) with attention mechanisms for capturing complex relationships, combined with Reinforcement Learning (RL) for real-time adaptation [12]
Adaptability	Static models, limited real-time updates [13]	RL-based continuous learning and adaptation, leveraging dynamic graph structures [14]
Explainability	Lack of interpretability, difficulty in understanding recommendations [15]	User-centric explainability through attention-based visualizations and counterfactual explanations
Fairness	Potential biases due to data imbalances [16]	Fair representation learning techniques and bias mitigation strategies
Privacy	Data sharing concerns, privacy leakage [17]	Differential privacy, federated learning, and homomorphic encryption

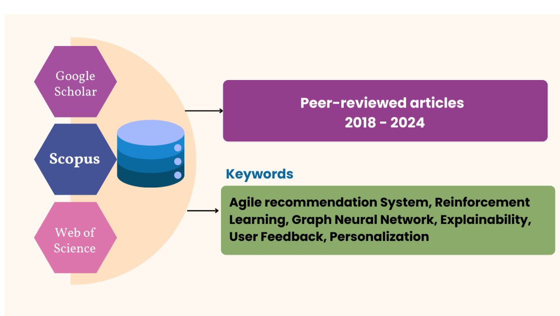

Figure 2. Literature Review Process

in real-time, personalizing recommendations dynamically.

The paper acknowledges the importance of explainability, fairness, and privacy in ARS, discussing limitations of current methods and highlighting ongoing research in these areas.

3. Research Gap and Proposed Approach

Building upon the insights gained from the comparative analysis presented in Tables I and Table II, this section explores the research gap and outlines a proposed framework for Agile Recommendation Systems (ARS).

Figure 3 illustrates a proposed approach for an Agile Recommendation System (ARS). It outlines a cyclical process involving various components, emphasizing user feedback and iterative improvement.

A. *Agile Recommendation Systems (ARS)*

Key Components and Processes are as follows: User: The starting point of the system. The user interacts with

TABLE II. Comparison of Recommendation System Approaches

Feature	Agile Recommendation Systems (ARS)	ARS with GNNs	ARS with RL	ARS with GNNs & RL (Proposed)
Adaptability	Real-time adaptation (implicit & explicit feedback, external data)	Adapts to evolving user-item relationships (GNNs) [18]	Adapts to real-time user feedback (RL) [19]	Combines real-time adaptation (RL) with user-item modeling (GNNs)
User-Item Modeling	Implicit & explicit interactions (clicks, views, ratings, reviews) [20]	Complex user-item relationships (user-item graph, GNNs)	User interaction rewards [21]	Combines user-item graph (GNNs) with user interaction rewards (RL)
Explainability	Limited understanding of recommendation rationale [22]	Explainability challenges with GNNs	Difficulty explaining RL agent decisions [23]	Combines XAI methods with GNNs and RL for explainability and bias detection
Focus	Dynamically adapting recommendations (user behaviour & feedback)	Leveraging GNNs for complex user-item relationships [24]	Optimizing recommendations (user interaction rewards) [25]	Combines GNNs and RL for personalized and dynamic recommendations

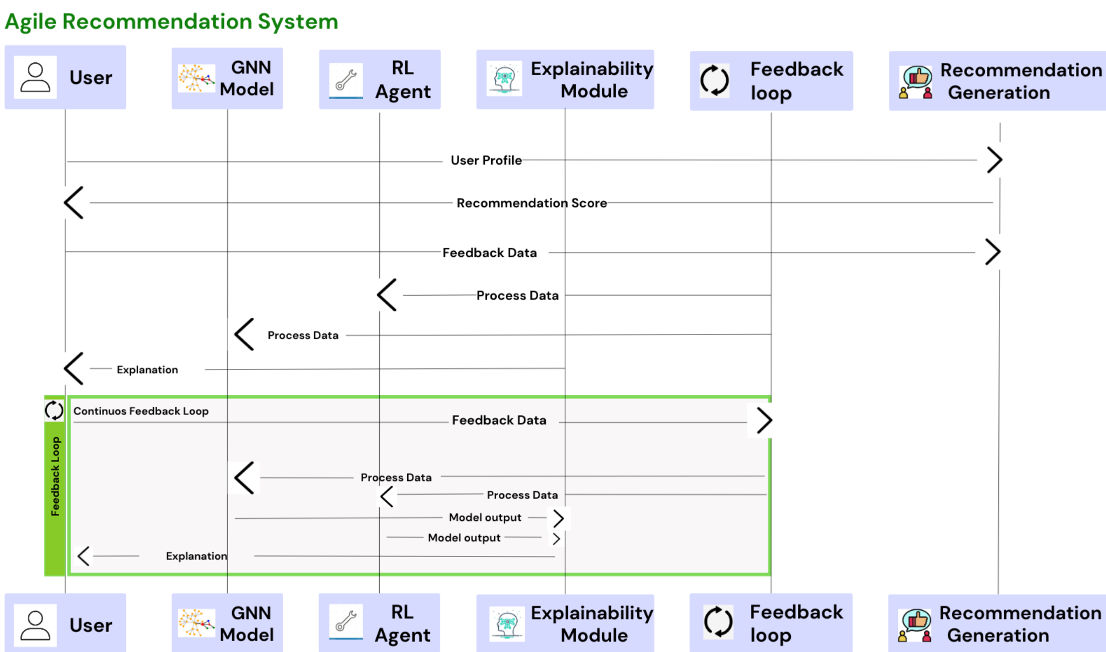

Figure 3. Agile Recommendation System Framework

the ARS, providing input or preferences. GNN Model: A Graph Neural Network (GNN) model is used to process and analyze user data and item information. GNNs are particularly suitable for recommendation systems because they can capture complex relationships and dependencies within the data. RL Agent: A Reinforcement Learning (RL) Agent is responsible for personalized learning. It learns to make decisions through trial and error, interacting with the environment to maximize a reward signal (user satisfaction or engagement).

Recommendation Generation: This process involves generating personalized recommendations for the user based on the output of the GNN model and the RL Agent. Explanation: A key component of the ARS is the ability to provide explanations for the generated recommendations. This could involve techniques like LIME, attention mechanisms, or counterfactual explanations. Feedback Generation: The user is prompted to provide feedback on the recommendations. This feedback can include ratings, likes, dislikes, or more detailed comments. Feedback Loop: The collected feedback is fed back into the system to improve the recommendation process. This could involve updating the GNN model, refining the RL Agent's policy, or adjusting other components based on the user's preferences. Stages of Agile Recommendation System are shown below in Figure 4.

Stage 1: Data Acquisition and Preprocessing
• Data Collection: Gather user-item interaction data (e.g., ratings, purchases, views).
• Data Cleaning: Preprocess data to handle missing values,

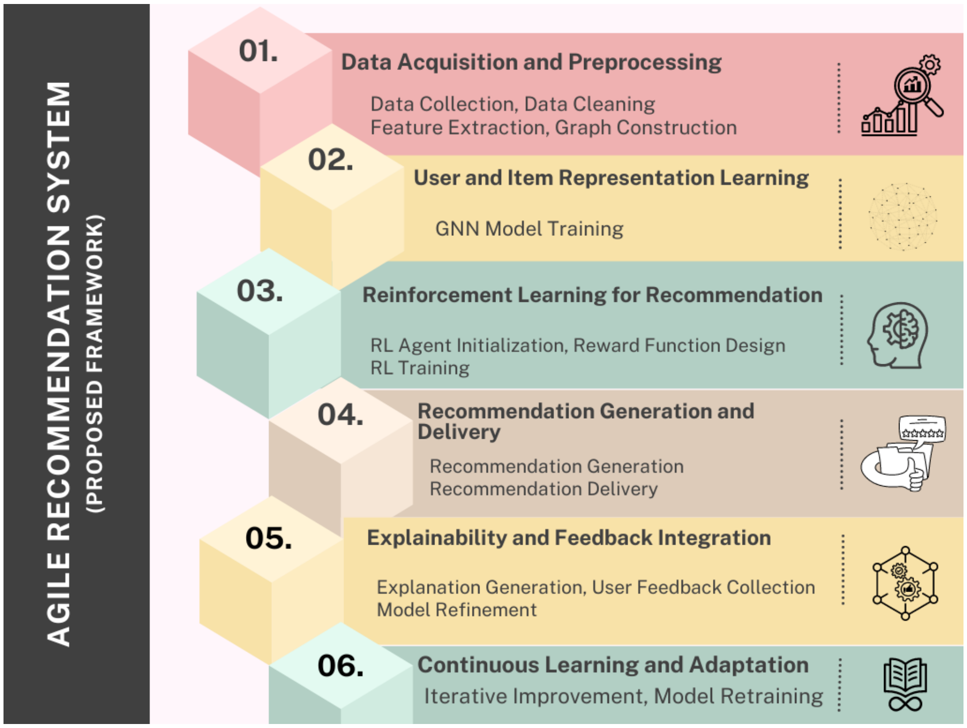

Figure 4. Stages of Agile Recommendation System

outliers, and inconsistencies.
• Feature Extraction: Take pertinent elements out of the user and item data.
• Graph Construction: Create a user-item interaction graph, incorporating extracted features as node attributes.

Stage 2: User and Item Representation Learning
• GNN Model Training: Train the GNN on the constructed graph to become familiar with the user and item embeddings capturing complex relationships.

Stage 3: Reinforcement Learning for Recommendation
• RL Agent Initialization: Define an RL agent with appropriate action space (item recommendations) and state space (user and item embeddings).
• Reward Function Design: Develop an explainable reward function considering factors like recommendation accuracy, user satisfaction, and diversity.
• RL Training: Train the RL agent to optimize the reward function through interaction with the environment (users).

Stage 4: Recommendation Generation and Delivery
Recommendation Generation: Use the trained GNN and RL agent to generate personalized recommendations for users.
• Recommendation Delivery: Provide users with extra explanations for your recommendations that were produced by the explainability module.

Stage 5: Explainability and Feedback Integration
• Explanation Generation: Utilize explainability techniques to shed light on the recommendation process (e.g., attention mechanisms, feature importance).
• User Feedback Collection: Gather user feedback on recommendations (e.g., ratings, likes, dislikes, comments).
• Model Refinement: Incorporate user feedback to improve the GNN and RL models.

Stage 6: Continuous Learning and Adaptation
• Iterative Improvement: Continuously update the recommendation system based on new user data and feedback. Model Retraining: Periodically retrain the GNN and RL models to capture evolving user preferences and item characteristics.

The reviewed literature highlights the potential of GNNs and RL for enhancing user-item relationship modeling and real-time adaptation in ARS. However, challenges remain in explainability, limiting user trust and hindering the widespread adoption of these advanced techniques. This proposed approach aims to bridge this gap by combining GNNs and RL for ARS with a strong emphasis on explainability. This approach leverages the strengths of both techniques:
• GNNs capture complex user-item relationships, providing a rich model for recommendation generation.
• RL enables real-time adaptation based on user feedback, ensuring recommendations stay relevant to evolving preferences.
By incorporating XAI principles, our approach strives to:
• Explain how user interactions and contextual factors contribute to recommendations made by the ARS.

• Identify and mitigate potential biases within the GNN and RL models for fair and trustworthy recommendations.
This focus on explainability alongside the combination of GNNs and RL positions this proposed approach to solve the shortcomings of existing ARS and contribute significantly to the field.

B. Emerging Techniques for Agile Recommendation Systems (ARS)

1) Graph Neural Networks (GNNs) for Overcoming Limitations in Modeling User-Item Relationships

Traditional recommender systems often struggle to record intricate connections between between users and items. Collaborative filtering techniques, while effective, can miss implicit connections and struggle with new users or items. GNNs offer a powerful alternative, addressing these limitations:

Detailed Explanation: GNNs work with graphs in which users and items are represented by nodes, and interactions between them—such as views, ratings, and purchases—are represented by edges. To discover the underlying linkages, GNNs repeatedly analyze data from a user's neighborhood, which includes related objects and persons. Each layer aggregates information from neighboring nodes, allowing the GNN to capture complex user-item relationships and collaborative filtering effects.

Real-World Scenario: Consider a social networking site where users follow friends and receive movie recommendations. A GNN-based ARS can process the user's movie viewing history and the viewing habits of their social network (friends they follow). By analyzing this graph of user connections and movie preferences, the GNN can recommend movies that not only align with the user's taste but also factor in the preferences of their trusted connections (friends). For instance, if a user enjoys action movies and follows a friend who frequently watches sci-fi films, the GNN might recommend a well-rated sci-fi action movie, leveraging the combined influence of the user's own preferences and their social network [26].

Limitations: GNNs can be computationally expensive to train, especially with large and complex graphs. Additionally, interpreting the inner workings of GNNs to understand why specific recommendations are made can be challenging. Recent work in Explainable AI (XAI) for GNNs is exploring methods to address this limitation [27].Building upon the proposed ARS framework outlined in Figure 3, this section explores the specific techniques and approaches that underpin the system.

2) Reinforcement Learning (RL) for Dynamic Adaptation in ARS

While traditional ARS can adapt to user preferences based on historical data, they might not react effectively to real-time user feedback. Reinforcement Learning (RL) offers a solution for this by allowing the ARS to learn and adapt continuously:

Detailed Explanation: RL involves an agent (the ARS) interacting with an environment (users) and learning through trial and error. The agent receives rewards (positive feedback from users) or penalties (negative feedback) for its recommendations. Over time, the RL agent learns to choose actions (recommendations) that maximize its reward.

Real-World Scenario: Consider a music streaming service with an ARS that recommends workout playlists. An RL-based ARS can learn from user interactions (e.g., skips, thumbs up/down). If a user skips energetic songs early in their workout, the RL agent understands the preference for slower music during warm-up. As the workout progresses and the user listens to more upbeat songs without skipping, the RL agent adjusts recommendations to progressively suggest faster and more intense music. This creates a personalized workout experience that dynamically adapts to the user's preferences in real-time [28].

Limitations: Defining an effective reward function that accurately captures user satisfaction is crucial for RL-based ARS. Additionally, exploration vs exploitation remains a challenge. The ARS should balance recommending familiar items the user enjoys (exploitation) with exploring new content to discover user preferences (exploration).

While reinforcement learning offers a powerful mechanism for adapting recommendations to dynamic user preferences, the black-box nature of RL models can hinder trust and transparency. To address this, integrating explainability techniques into the RL component is crucial. The following table presents a comparison of explainability techniques applicable to reinforcement learning within the context of ARS.

C. Contribution of This Work to ARS Research

This survey paper contributes to the advancement of ARS research in several key ways: Focus on Emerging Techniques: While existing surveys cover traditional ARS approaches, this work places a strong emphasis on recent advancements like Graph Neural Networks (GNNs) and Reinforcement Learning (RL). It delves into their potential to address limitations in user-item relationship modeling and real-time adaptation, respectively.

Detailed Explanations and Real-World Scenarios: Beyond simply mentioning these techniques, the paper provides in-depth explanations of GNNs and RL in the context of ARS. It includes real-world situations that demonstrate how these techniques can be applied in practical recommendation systems, making the concepts more tangible and relevant.

Explainability, Fairness, and Privacy Considerations: This survey acknowledges that advancements are needed in explainability, fairness, and privacy for robust and trustworthy ARS. It highlights the limitations of current methods and discusses ongoing research efforts in Explainable AI (XAI), fair recommendation systems, and privacy-

TABLE III. Comparison of Explainability Techniques for Agile Recommendation Systems

Feature	Existing Work	Proposed Approach
Explainability Technique (GNN)	User-centric Attention with Counterfactuals: This approach uses attention mechanisms to pinpoint key factors in user profiles and items for recommendations. It then generates explanations showing how recommendations change if user attributes are modified, promoting user understanding of the recommendation rationale and potential biases. [29]	Not well-explored for GNN-RL ARS. Might involve attention mechanisms or graph-based explanations applied to the GNN component
Explainability Technique (RL)	User-Driven Reward Shaping for Recommendations: This personalized recommendations based on user preferences and allows us to explain how user input shapes the suggestions [30]	Relies on general RL explainability techniques like action trajectories or reward decomposition, which might not be specific to ARS.
Focus on User Explainability	Natural Language Explanations: Our approach generates explanations in natural language that highlight the key factors influencing a recommendation. This user-friendly format makes it easier for users to understand the reasoning behind the suggestions and fosters trust in the system [31]	Limited focus on explaining recommendations to users in an understandable way.
Integration with ARS Context	Real-time Explainability for ARS: We consider the dynamic nature of ARS by incorporating real-time user interactions and feedback into the explanation process. This ensures that explanations are always up-to-date and reflect the latest user preferences [32]	Existing work might not explicitly consider ARS context (e.g., real-time updates, user feedback) in explanations.

preserving techniques. This comprehensive approach positions the paper for broader impact in the field.

By incorporating the explainability techniques outlined in Table 3, the proposed ARS framework can significantly enhance user trust and understanding of the recommendation process. Building upon this foundation, the following section highlights the key contributions of this research to the advancement of ARS.

D. *Unique Aspects of the Proposed Approach in Agile Recommendation Systems (ARS)*

Leveraging upon the foundation established in Section 3.3, which highlighted the importance of explainability in ARS, this section explores in more detail the specific challenges and opportunities associated with achieving explainability in GNN and RL-based systems.While combining GNNs and RL for ARS shows promise, our proposed approach offers several unique aspects that address limitations of existing work, particularly concerning explainability:

1) *Explainable GNN-based User Representation*

Existing work primarily focuses on GNNs for user modeling, but explainability within GNNs for ARS remains a challenge. Our approach proposes a novel GNN architecture with interpretable layers. This architecture will not only capture complex user-item relationships but also allow us to understand how specific interactions contribute to the user's embedded representation in the graph. Techniques like attention mechanisms or layer-wise decomposition can be included in order to accomplish this interpretability.

2) *Explainable Reward Shaping for RL*

Current RL-based ARS often struggle with defining effective reward functions and lack transparency in how these rewards influence recommendations. We propose a modular reward function design with interpretable components. This will involve decomposing the reward function into smaller, more understandable parts that capture specific aspects of user satisfaction (e.g., relevance, novelty, exploration). By analyzing which components contribute most to the final reward, we can gain insights into the rationale behind the recommendations.

3) *Integrated Explainability Framework*

Existing research often treats explainability for GNNs and RL in ARS as separate problems. Our approach proposes a unified explainability framework that leverages insights from both models. This framework will combine the user representation interpretability from the GNN with the explainable reward components from the RL to provide a comprehensive explanation of the recommended items. It will not only explain which items are recommended but also why they are relevant to the user based on their interaction history and current preferences.

4) *User-Centric Feedback Loop for Explainability Improvement*

Limited work explores how user feedback can be integrated to improve explainability in ARS. Our approach incorporates a user-centric feedback loop where users can provide feedback on the explainability itself. This feedback can be in the form of ratings on clarity, usefulness, or level of detail.

The feedback loop will be used to refine the explainability framework over time, ensuring it remains aligned with user needs and expectations.By addressing these limitations in explainability, our proposed approach offers a more transparent and user-centric ARS that builds trust and allows users to understand the reasoning behind the recommendations they receive.

To illustrate the potential of combining various explainability techniques with GNN and RL components, Table 4 presents a comparative overview of different approaches. This Table 4 highlights the diverse range of techniques that can be employed to enhance transparency and interpretability in ARS.

TABLE IV. Existing vs. Proposed Approaches in Agile Recommendation Systems (ARS)

Factor	Approach 1	Approach 2	Approach 3	Our Approach
Target Domain	Educational Content	Educational Content	Educational Content	Educational Content
GNN Architecture	R-GCN (Relational GCN) [33]	GAT (Graph Attention Network) [34]	Knowledge-aware GCN (KG-GCN) [35]	User-Preference aware GCN (UP-GCN) [36]
RL Technique	DQN (Deep Q-Network) [37]	SARSA [38]	Actor-Critic	DQN with Curriculum Learning [39]
Explainability	Not addressed	Not addressed	Layer-wise Relevance Propagation [40]	Attention Mechanism with Knowledge Tracing [1]
Bias Mitigation	-	-	User demographics based debiasing [41]	User knowledge and learning style based [42]

E. *Mitigate bias within GNN and RL models in Agile Recommendation Systems (ARS) using explainability*

Expanding on the discussion of explainability techniques in Table 4, this section explores how these methods can be leveraged to identify and mitigate biases within GNN and RL models. By understanding the factors influencing recommendations, we can work towards developing fairer and more equitable recommendation systems.

Identifying Bias in GNNs Attention Mechanisms: By analyzing the attention weights assigned by the GNN layers, we can identify if the model is disproportionately focusing on specific features of users or items. This can reveal biases based on factors like demographics (age, gender) or purchase history (expensive vs. cheap items).

Community Analysis: Explainability techniques can help visualize how information propagates through the user-item graph. This can expose potential biases if the model reinforces connections within specific communities (e.g., recommending similar music to users with similar demographics) while neglecting connections across communities (e.g., recommending new genres to users).

Mitigating Bias in GNNs Regularization Techniques: Techniques like dropout or adding noise to features can be used to prevent the GNN from overfitting to biased patterns in the data. This can help the model learn more generalizable representations of users and items. Fairness-aware GNN Architectures: Recent research explores GNN architectures that incorporate fairness constraints during the learning process. These models can be designed to penalize recommendations that perpetuate biases based on sensitive user attributes.

Identifying Bias in RL Reward Function Analysis: By examining which components of the explainable reward function (e.g., novelty vs. relevance) contribute most to the final recommendation, we can identify if the model is biased towards certain types of items or users. For example, a high emphasis on novelty might lead to biased recommendations for users who frequently explore new items, neglecting users who prefer familiar choices. Action Trajectories: Visualizing the action trajectories of the RL agent can reveal biases in its exploration strategy. If the agent consistently avoids recommending items from certain categories, it suggests a potential bias in the reward function or the underlying user-item graph.

Mitigating Bias in RL Counterfactual Explanations: Explainability methods can be used to generate counterfactual recommendations, which show how recommendations would change if specific user features were altered. This can help identify and address biases based on sensitive attributes. Reward Shaping with Fairness Constraints: The reward function design can be adjusted to incorporate fairness considerations. This could involve penalizing the agent for recommendations that disproportionately favor certain user groups or item categories.

By leveraging explainability techniques in GNNs and RL models, developers of ARS can build more fair and unbiased recommendation systems that cater to the various needs that each user has.

4. Key Characteristics of Agile Recommendation Systems

A. *Real-time Learning*

Unlike traditional systems with static user profiles, ARS continuously learn from user interactions and data streams: Implicit Feedback: Click-through rates, watch history, browsing behavior, and content engagement patterns reveal implicit user preferences [43]. Explicit Feedback: User ratings, likes/dislikes, and reviews provide valuable insights into user preferences for refining recommendations [44]. External Data Sources: Social media activity, search queries, and contextual information (location, time of day) enrich user profiles and personalize recommendations further [45]. This continuous learning loop ensures that user profiles and recommendations remain relevant and reflect evolving user tastes.

B. *Context Awareness*

Contextual factors significantly influence user preferences. ARS leverages these factors to suggest content most relevant to the user's current situation. Examples of context awareness include: Location: Recommending location-based content, such as restaurants near a user's current location or local news updates [46].

Time of Day: Suggesting upbeat music for morning commutes, relaxing playlists for bedtime, or news updates during lunch breaks [47].

Device: Tailoring recommendations based on the device being used (e.g., suggesting downloadable audiobooks for mobile devices). By considering context, ARS can provide highly relevant and timely recommendations that enhance user experience.

C. User Feedback Integration

ARS not only personalize the recommendation experience but also foster active user engagement with the system through user feedback integration: Improved Accuracy: User feedback, such as ratings and reviews, helps ARS refine its understanding of user preferences and adjust recommendations for similar items. This continuous feedback loop leads to more accurate and relevant suggestions over time [48].

Exploration and Serendipity: ARS can leverage explicit feedback (likes/dislikes) on specific content categories to encourage users to explore new areas. By temporarily adjusting the recommendation algorithm based on user exploration behavior, ARS can introduce users to content they might not have discovered otherwise, fostering a sense of serendipity [49].

D. Deep Learning for Agile Recommendation Systems

Recent advancements in deep learning have significantly impacted the field of recommendation systems, particularly in the context of Agile Recommendation Systems (ARS). By effectively capturing complex patterns and relationships within user data, deep learning models enable ARS to deliver more accurate, personalized, and contextually relevant recommendations. This section explores two prominent deep learning approaches: Recurrent Neural Networks (RNNs) for

1) Recurrent Neural Networks (RNNs) for Dynamic User Modeling

RNNs, particularly Long Short-Term Memory (LSTM) networks, excel at modeling sequential data and capturing temporal dependencies. In the realm of ARS, RNNs can effectively learn evolving user preferences by analyzing historical interaction data. By considering a user's interaction history, LSTMs can predict future preferences and generate tailored recommendations. For instance, an LSTM-based ARS can recommend products complementary to a user's recent purchase or suggest items aligned with their evolving interests.

2) Convolutional Neural Networks (CNNs) for Context-Aware Recommendations

CNNs, traditionally used for image processing, have found applications in various domains, including natural language processing and recommendation systems. In the context of ARS, CNNs can effectively extract features from diverse data formats, such as text, images, and audio. By analyzing user interactions with different content modalities, CNNs can provide valuable insights into user preferences and contextual factors. For example, a CNN can analyze click-through patterns on product images to identify visual preferences or extract relevant information from product descriptions to understand user needs.

More advanced and customized recommendation systems that adjust to shifting user behaviors and contextual variables can be made by combining CNNs and RNNs into the ARS framework.

E. Attention Mechanisms for Enhanced Recommendation Focus

To further refine the recommendation process, attention mechanisms can be integrated into ARS. By assigning weights to different components of the input data, attention mechanisms enable the model to focus on the most relevant information for a given recommendation. This approach enhances the precision and relevance of recommendations. For instance, in an RNN-based ARS, attention mechanisms can dynamically adjust the importance of past user interactions, allowing the model to prioritize recent behavior over historical data. Similarly, in a context-aware setting, attention can be used to emphasize specific contextual factors, such as location or time of day, when generating recommendations. By incorporating attention mechanisms into deep learning architectures, ARS can deliver more focused and personalized recommendations that align closely with users' evolving preferences and situational context.

5. Conclusion and Future Work

This survey has explored the evolving landscape of Agile Recommendation Systems (ARS), highlighting the limitations of traditional approaches and the potential of emerging techniques like Graph Neural Networks (GNNs) and Reinforcement Learning (RL). By incorporating explainability, fairness, and privacy considerations, we have emphasized the importance of developing trustworthy and user-centric recommendation systems. The proposed ARS framework, combining GNNs, RL, and explainability, offers a promising approach to address the challenges faced by traditional RS. By capturing complex user-item relationships, adapting to real-time user feedback, and providing transparent recommendations, this framework has the potential to revolutionize the recommendation landscape across various domains. Future research should prioritize the development of privacy-preserving ARS and the creation of advanced explainability techniques tailored to GNN-RL based systems.

References

[1] Z. Li, M. Bilal, X. Xu, J. Jiang, and Y. Cui, "Federated learning-based cross-enterprise recommendation with graph neural networks," *IEEE Transactions on Industrial Informatics*, pp. 1–10, 2022.

[2] W. Fan and et al., "A graph neural network framework for social recommendations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2033–2047, 2022.

[3] A. Bahalul Haque, A. N. Islam, and P. Mikalef, "Explainable artificial intelligence (xai) from a user perspective: A synthesis of prior literature and problematizing avenues for future research,"

Technological Forecasting and Social Change, vol. 186, p. 122120, 2023.

[4] K. Sadeghi R., D. Ojha, P. Kaur, R. V. Mahto, and A. Dhir, "Explainable artificial intelligence and agile decision-making in supply chain cyber resilience," *Decision Support Systems*, vol. 180, p. 114194, 2024.

[5] V. Božić, "Explainable artificial intelligence (xai): Enhancing transparency and trust in ai systems," 2023.

[6] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, and B. Natarajan, "Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[7] U. Javed, K. Shaukat, I. Hameed, F. Iqbal, T. Mahboob Alam, and S. Luo, "A review of content-based and context-based recommendation systems," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, 2021.

[8] X. Xin, X. Liu, H. Wang, P. Ren, Z. Chen, J. Lei, X. Shi, H. Luo, J. Jose, M. Rijke, and Z. Ren, "Improving implicit feedback-based recommendation through multi-behavior alignment," in *Proceedings of the ACM Conference*, 2023, pp. 932–941.

[9] S. Das, A. Tariq, T. Santos, S. Kantareddy, and I. Banerjee, "Recurrent neural networks (rnns): Architectures, training tricks, and introduction to influential research," in *Machine Learning for Brain Disorders*, O. Colliot, Ed. Humana, New York, NY, 2023, vol. 197.

[10] S. Dharmichand and P. Supraja, "Heterogeneous convolutional neural networks for emotion recognition combined with multimodal factorised bilinear pooling and mobile application recommendation," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, pp. 129–142, 2023.

[11] A. Panteli and B. Boutsinas, "Addressing the cold-start problem in recommender systems based on frequent patterns," *Algorithms*, vol. 16, no. 4, p. 182, 2023.

[12] N. Guo, "Recommendation system model based on graph neural network combined with support vector machine and comparative learning for filtering," *Applied and Computational Engineering*, vol. 73, pp. 246–254, 2024.

[13] C. A. Abhijith, S. R. Satadal, S. Shaktimatta, B. Sara, and C. Balaji, "Static model building using artificial intelligence and machine learning techniques," in *14th Biennial International Conference and Exposition SPG 2023*. Singapore: Telesto Energy Pte Limited, 2023.

[14] N. Mingshuo, D. Chen, and W. Dongqi, "Reinforcement learning on graphs: A survey," 2022, arXiv:2204.06127. [Online]. Available: https://doi.org/10.48550/arXiv.2204.06127

[15] N. Balasubramaniam, M. Kauppinen, A. Rannisto, K. Hiekkanen, and S. Kujala, "Transparency and explainability of ai systems: From ethical guidelines to requirements," *Information and Software Technology*, vol. 159, p. 107197, 2023.

[16] A. Klimashevskaia, D. Jannach, M. Elahi *et al.*, "A survey on popularity bias in recommender systems," *User Modeling and User-Adapted Interaction*, 2024.

[17] Y. Ji, A. Sun, J. Zhang, and C. Li, "A critical study on data leakage in recommender system offline evaluation," *ACM Transactions on Information Systems*, vol. 41, no. 3, p. 27, 2023.

[18] B. Khemani, S. Patil, K. Kotecha *et al.*, "A review of graph neural networks: Concepts, architectures, techniques, challenges, datasets, applications, and future directions," *Journal of Big Data*, vol. 11, 2024.

[19] A. Khamaj and A. M. Ali, "Adapting user experience with reinforcement learning: Personalizing interfaces based on user behavior analysis in real-time," *Alexandria Engineering Journal*, vol. 95, pp. 164–173, 2024.

[20] R. Esmeli, M. Bader-El-Den, H. Abdullahi *et al.*, "Implicit feedback awareness for session based recommendation in e-commerce," *SN Computer Science*, vol. 4, 2023.

[21] K. Katahira, "The relation between reinforcement learning parameters and the influence of reinforcement history on choice behavior," *Journal of Mathematical Psychology*, vol. 66, pp. 59–69, 2015.

[22] R. Shimizu, M. Matsutani, and M. Goto, "An explainable recommendation framework based on an improved knowledge graph attention network with massive volumes of side information," *Knowledge-Based Systems*, vol. 239, p. 107970, 2022.

[23] A. K. Shakya, G. Pillai, and S. Chakrabarty, "Reinforcement learning algorithms: A brief survey," *Expert Systems with Applications*, vol. 231, p. 120495, 2023.

[24] Z. Huang, Z. Sun, J. Liu, and Y. Ye, "Group-aware graph neural networks for sequential recommendation," *Information Sciences*, vol. 670, p. 120623, 2024.

[25] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin, "Reinforcement learning to optimize long-term user engagement in recommender systems," in *Proceedings of the ACM Conference*, 2019, pp. 2810–2818.

[26] D. Tallapally, J. Wang, K. Potika, and M. Eirinaki, "Using graph neural networks for social recommendations," *Algorithms*, vol. 16, p. 515, 2023.

[27] J. Kakkad, J. Jannu, K. Sharma, C. Aggarwal, and S. Medya, "A survey on explainability of graph neural networks," *Preprint*, 2023.

[28] Z. Zhu and B. Roy, "Deep exploration for recommendation systems," *Preprint*, 2021.

[29] Z. Wang, "Enhancing user-item interaction through counterfactual classifier for sequential recommendation," *Applied Mathematics and Nonlinear Sciences*, vol. 9, 2024.

[30] H. Sami, J. Bentahar, A. Mourad, H. Otrok, and E. Damiani, "Graph convolutional recurrent networks for reward shaping in reinforcement learning," *Information Sciences*, vol. 608, pp. 63–80, 2022.

[31] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, "Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence," *Information Fusion*, vol. 99, p. 101805, 2023.

[32] Y. Bekkemoen, "Explainable reinforcement learning (xrl): A sys-

tematic literature review and taxonomy," *Machine Learning*, vol. 113, pp. 355–441, 2024.

[33] X. Xu, X. Zhao, M. Wei, and Z. Li, "A comprehensive review of graph convolutional networks: Approaches and applications," *Electronic Research Archive*, vol. 31, pp. 4185–4215, 2023.

[34] G. Su, H. Wang, Y. Zhang, W. Zhang, and X. Lin, "Simple and deep graph attention networks," *Knowledge-Based Systems*, vol. 293, p. 111649, 2024.

[35] G. Li, L. Yang, S. Bai, and et al., "Bikagcn: Knowledge-aware recommendations under bi-layer graph convolutional networks," *Neural Processing Letters*, vol. 56, p. 20, 2024.

[36] H. Xu, G. Wu, E. Zhai, X. Jin, and L. Tu, "Preference-aware light graph convolution network for social recommendation," *Electronics*, vol. 12, no. 11, p. 2397, 2023.

[37] M. Sewak, "Deep q network (dqn), double dqn, and dueling dqn: A step towards general artificial intelligence," in *Advances in Artificial Intelligence*, 2019, pp. 123–134.

[38] M. R. Tousi, S. H. Hosseinian, A. H. Jadidinejad, and M. B. Menhaj, "Application of sarsa learning algorithm for reactive power control in power system," in *2008 IEEE 2nd International Power and Energy Conference*, 2008, pp. 1198–1202.

[39] Y. Kim, K. Lee, B. Nam, and Y. Han, "Application of reinforcement learning based on curriculum learning for the pipe auto-routing of ships," *Journal of Computational Design and Engineering*, vol. 10, no. 1, pp. 318–328, 2023.

[40] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, 2019, pp. 31–42.

[41] L. Sha, D. Gašević, and G. Chen, "Lessons from debiasing data for fair and accurate predictive modeling in education," *Expert Systems with Applications*, vol. 228, p. 120323, 2023.

[42] Z. Mehenaoui, Y. Lafifi, and L. Zemmouri, "Learning behavior analysis to identify learner's learning style based on machine learning techniques," *JUCS - Journal of Universal Computer Science*, vol. 28, pp. 1193–1220, 2022.

[43] H. Wen, L. Yang, and D. Estrin, "Leveraging post-click feedback for content recommendations," in *Proceedings of the ACM Conference*, 2019, pp. 278–286.

[44] P. Kostkova, G. Jawaheer, and P. Weller, "Modeling user preferences in recommender systems," *ACM Transactions on Interactive Intelligent Systems*, vol. 4, pp. 1–26, 2014.

[45] K. Bok, J. Song, J. Lim, and J. Yoo, "Personalized search using user preferences on social media," *Electronics*, vol. 11, no. 19, p. 3049, 2022.

[46] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: A survey," *GeoInformatica*, vol. 19, 2015.

[47] V. Verma, N. Marathe, P. Sanghavi, and D. Nitnaware, "Music recommendation system using machine learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 80–88, 2021.

[48] L. Chen, G. Chen, and F. Wang, "Recommender systems based on user reviews: The state of the art," *User Modeling and User-Adapted Interaction*, vol. 25, 2015.

[49] B. Koç, "The role of user interactions in social media on recommendation algorithms: Evaluation of tiktok's personalization practices from user's perspective," 2023.