International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue5
Sep-Oct- 2026, PP 01-06

Graph Neural Networks For Recommendation Systems: A Review Of Architectures, Training Objectives And Open Challenges

Alisha Raju Shaikh[1]; Prof. S. K. Said[2]; Prof. M. C. Kolse[3]
Department Of Artificial Intelligence And Data Science Jaihind College Of Engineering Pune, Maharashtra, India;

Abstract—Recommendation systems help users discover products, films and articles on digital platforms. Classical collaborative and content-based filtering struggle to capture non-linear relationships and high-order connectivity between users and items, especially when interaction data are sparse. Graph Neural Networks (GNNs) address this by modelling interactions as a graph in which users and items are nodes and interactions or attributes are edges, and by learning embeddings through message passing over neighbourhoods. This paper reviews GNN-based recommendation. It introduces graph convolution, graph attention and inductive sampling, then surveys landmark recommenders including graph convolutional matrix completion, PinSage, Neural Graph Collaborative Filtering, LightGCN, knowledge-graph attention networks and session-based GNNs. Training objectives such as Bayesian personalized ranking and evaluation with recall and normalized discounted cumulative gain are explained. A layered architecture covering graph construction, embedding propagation, prediction and serving is proposed, together with research gaps, objectives and a comparison of models. Challenges of over-smoothing, scalability, cold start, fairness and explainability are discussed. The review concludes that GNNs provide a principled way to exploit relational structure in recommendation, with simpler propagation often outperforming heavier designs.

Index Terms—Graph neural networks, recommender systems, collaborative filtering, LightGCN, graph convolution, knowledge graph, embedding, cold start

I. Introduction

Online platforms offer far more items than any user can browse, and recommendation systems decide what each person sees. Matrix factorization [1] became the workhorse of collaborative filtering by representing users and items as vectors whose dot product estimates preference. Neural collaborative filtering [2] replaced the dot product with learned interaction functions.These methods treat each user–item pair largely in isolation. In reality, interactions form a network: two users who liked the same items are related, and an item liked by a user's similar neighbours is a good candidate even if the user has never seen it. Graph Neural Networks make this structure explicit by passing information along edges, so that each embedding reflects multi-hop neighbourhoods.This paper reviews GNN-based recommendation, proposes a reference architecture and discusses the open problems that remain.

II. Background and Related Work

2.1 Graph Neural Networks

A GNN updates a node's representation by aggregating its neighbours' representations. Graph Convolutional Networks [3] use normalized neighbour averaging, Graph Attention Networks [4] learn attention weights over neighbours, and GraphSAGE [5] samples neighbourhoods to scale to large graphs and to embed unseen nodes.

International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue5
Sep-Oct- 2026, PP 01-06

2.2 Graphs in Recommendation

Fig. 1 shows the main graph types: bipartite user–item graphs, social graphs, knowledge graphs linking items to attributes, and session graphs built from short click sequences.

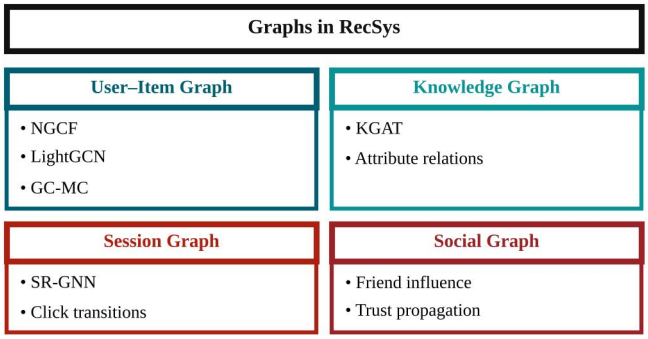


Fig. 1. Graph types used in GNN-based recommendation.

2.3 Review of Related Studies

Van den Berg *et al.* [6] framed rating prediction as link prediction on a bipartite graph with graph convolutional matrix completion, showing that side information can be incorporated naturally. At industrial scale, PinSage [7] combined random-walk-based neighbourhood sampling with graph convolutions to recommend among billions of items at Pinterest.

Wang *et al.* proposed Neural Graph Collaborative Filtering [8], which explicitly propagates embeddings over the user–item graph to encode collaborative signals from higher-order connectivity. He *et al.* [9] then showed with LightGCN that feature transformation and non-linear activation contribute little in collaborative filtering; keeping only neighbourhood aggregation improved accuracy and simplified training.

Beyond interaction graphs, KGAT [10] merged the user–item graph with an item knowledge graph and used attention to weigh relations, improving both accuracy and explanation. SR-GNN [11] modelled each session as a graph of clicked items to capture complex transitions for anonymous users. Wu *et al.* [12] surveyed the field and organized methods by graph type and recommendation task. Table I summarizes these works.

Sr. No.	Ref.	Year	Methodology	Findings	Research Gap
1	[6]	2017	Graph convolutional matrix completion	Rating prediction as link prediction	Shallow propagation
2	[5]	2017	GraphSAGE inductive sampling	Embeds unseen nodes; scalable	Sampling variance
3	[7]	2018	PinSage random-walk GCN	Web-scale recommendation	Engineering-heavy pipeline
4	[8]	2019	Neural graph collaborative filtering	Encodes high-order connectivity	Redundant transformations
5	[10]	2019	Knowledge-graph attention network	Uses item knowledge; explainable	Needs quality knowledge graph

International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue5
Sep-Oct- 2026, PP 01-06

Sr. No.	Ref.	Year	Methodology	Findings	Research Gap
6	[11]	2019	Session graphs with GNN	Handles anonymous sessions	Short-term behaviour only
7	[9]	2020	LightGCN simplified propagation	Better accuracy, simpler model	Collaborative signal only
8	[12]	2022	Survey of GNN recommenders	Taxonomy of methods and tasks	Survey only

TABLE I Summary of Literature Survey

III. Research Gap and Objectives

3.1 Identified Research Gaps

- Deep GNN stacks suffer from over-smoothing, so embeddings of different nodes become too similar.
- Training full-graph models on billions of edges remains expensive in memory and time.
- New users and items with few edges (cold start) are still poorly served.
- Popularity bias can be amplified by message passing, harming fairness and diversity.
- Reported gains are sensitive to evaluation protocol and baseline tuning.

3.2 Objectives of the Review

- 1. To explain GNN fundamentals relevant to recommendation.
- 2. To review major GNN recommenders and their design choices.
- 3. To propose a layered architecture for a GNN-based recommender.
- 4. To compare models in accuracy, complexity and applicability.
- 5. To identify open challenges and future directions.

I. Design of the Proposed System

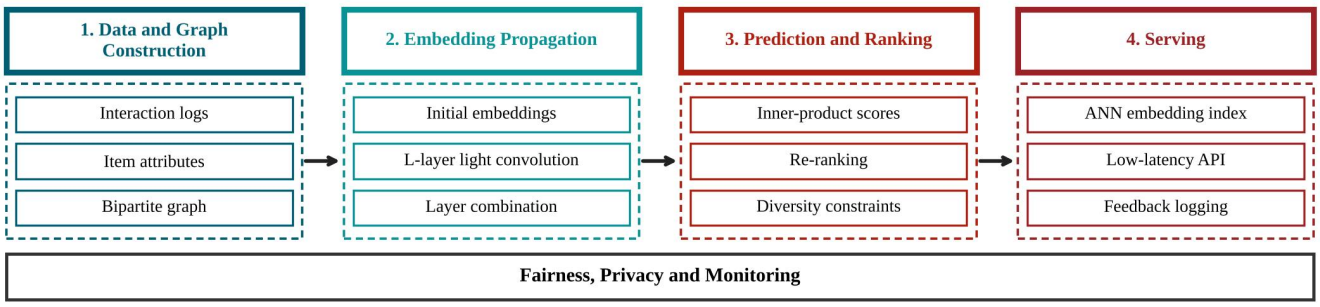


Fig. 2 presents a layered GNN recommender suitable for an e-commerce or media platform.

Fig. 2. Proposed layered architecture of a GNN-based recommender.

International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue5
Sep-Oct- 2026, PP 01-06

3.3 Data and Graph Construction Layer

Clicks, purchases and ratings form a bipartite graph; item attributes and social links are added as extra node and edge types where available.

3.4 Embedding Propagation Layer

Initial embeddings are refined by L layers of light graph convolution, and the final representation is a weighted sum across layers.

3.5 Prediction and Ranking Layer

Scores are computed as the inner product of user and item embeddings; candidates are re-ranked with business rules and diversity constraints.

3.6 Serving Layer

Item embeddings are indexed for approximate nearest-neighbour search so that recommendations are returned within milliseconds.

3.7 Methodology

In LightGCN the embedding of user u at layer k+1 is

$$e_u^{(k+1)} = \Sigma_i \in N(u)\ e_i^{(k)} / \sqrt{(|N(u)| \cdot |N(i)|)} \quad (1)$$

and models are trained with the Bayesian personalized ranking loss [13], which prefers observed items i over unobserved items j:

$$L = -\Sigma \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|\Theta\|^2 \quad (2)$$

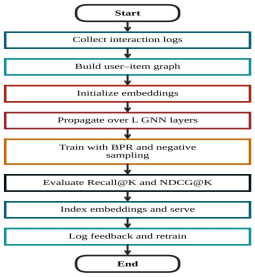


Fig. 3. Training and serving workflow of a GNN recommender.

Performance is reported with Recall@K and NDCG@K on held-out interactions, and training uses negative sampling to keep cost manageable.

IV. Discussion and Comparison

Table II compares representative models. LightGCN offers the best accuracy-to-complexity ratio for pure collaborative filtering, PinSage scales to very large graphs, KGAT exploits side knowledge, and SR-GNN suits session data.

International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue5
Sep-Oct- 2026, PP 01-06

TABLE II Comparison of GNN Recommenders

Model	Graph Type	Complexity	Best Use
GC-MC	User–item	Low	Explicit ratings
NGCF	User–item	Medium	Implicit feedback
LightGCN	User–item	Low	Collaborative filtering
PinSage	Item graph	High	Billion-scale catalogues
KGAT	User–item + KG	High	Side knowledge, explanation
SR-GNN	Session	Medium	Anonymous sessions

V. Challenges and Future Directions

Table III lists the main challenges. Promising directions include self-supervised contrastive learning on graphs, dynamic graphs that update in real time, combining GNNs with large language models for richer item understanding, and fairness-aware propagation.

TABLE III Challenges and Mitigation

Challenge	Mitigation
Over-smoothing	Few layers; layer combination; residuals
Scalability	Neighbour sampling; mini-batch training
Cold start	Attribute and knowledge-graph features
Popularity bias	Re-weighting; diversity-aware ranking
Reproducibility	Standard splits and tuned baselines

VI. Conclusions

GNNs turn the relational structure of user behaviour into better recommendations. The reviewed literature shows that propagating embeddings over interaction graphs captures high-order collaborative signals, that simpler designs such as LightGCN often outperform heavier ones, and that knowledge and session graphs extend the approach to new settings. Scalability, cold start, over-smoothing and fairness remain the key problems for future work.

6.1 Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

References

1. Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, 2009.
2. X. He et al., “Neural collaborative filtering,” in Proc. 26th Int. Conf. World Wide Web (WWW), 2017, pp. 173–182.
3. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. ICLR, 2017.
4. P. Veličković et al., “Graph attention networks,” in Proc. ICLR, 2018.

5. W. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in Proc. NeurIPS, vol. 30, 2017.
6. R. van den Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” arXiv:1706.02263, 2017.
7. R. Ying et al., “Graph convolutional neural networks for web-scale recommender systems,” in Proc. 24th ACM SIGKDD, 2018, pp. 974–983.
8. X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, “Neural graph collaborative filtering,” in Proc. 42nd ACM SIGIR, 2019, pp. 165–174.
9. X. He et al., “LightGCN: Simplifying and powering graph convolution network for recommendation,” in Proc. 43rd ACM SIGIR, 2020, pp. 639–648.
10. X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, “KGAT: Knowledge graph attention network for recommendation,” in Proc. 25th ACM SIGKDD, 2019, pp. 950–958.
11. S. Wu et al., “Session-based recommendation with graph neural networks,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 346–353.
12. S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, “Graph neural networks in recommender systems: A survey,” ACM Comput. Surv., vol. 55, no. 5, Art. no. 97, 2022.
13. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” in Proc. 25th Conf. Uncertainty Artif. Intell. (UAI), 2009, pp. 452–461.