OPEN ACCESS 111

ISSN 2356-5462
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 11, No. 2, Dec 2025. pp. 111-117
doi: 10.21108/ijoict.v11i2.9878

Evaluating The Performance of Graph-Based Recommendation Systems: A Case Study on Amazon Data

Zaid Mundher [1*], Manar Talat Ahmad [2]

[1, 2] *Department of Computer Science, University of Mosul, Mosul, Iraq*
Mosul, Iraq
*zaidabdulelah@uomosul.edu.iq

Abstract

Today, recommendation systems are considered a main component of social media platforms and many other online websites. Recommendation systems can be defined as tools that aim to introduce and suggest products to users. The suggestion process depends on many factors, such as user behavior and product similarity. In recent years, many research papers have discussed recommendation systems and introduced new solutions and methods to build them. On the other hand, in the last few years, data representation has also become an important issue. Because of the massive increase in data, new methods to represent data have been introduced and adopted, such as graph-based data representation. In this work, the efficiency of employing graph-based databases in building recommendation systems was evaluated and compared to traditional approaches. Specifically, Amazon Product Reviews dataset was used to build a recommendation system using traditional methods. This data was then transformed to a graph format and used to generate recommendations. The results indicate that the graph-based approach achieves higher precision (0.89 versus 0.65) and a better F1 score (0.76 versus 0.72), highlighting the advantage of capturing complex relationships in graph structures.

Keywords: Recommendation Systems, Graph-Based Databases, Amazon Product Reviews

I. INTRODUCTION

WITH the widespread of e-commerce, social media platforms, and video streaming sites, recommendation tools has become an essential feature of these platforms. These tools, in short, recommend products offered by the platform to users, achieving a double benefit: both for the site, which increases its sales and views, and for the user, who can find what they want without the effort of searching. The process of presenting products that truly interest the user is increasingly complex as we increase its accuracy. This process has undergone several improvements and developments, all of which aim to increase the accuracy of the resulting recommendations so that they truly meet the user's interests. The process of building recommendation systems initially relies on historical data that includes user behavior and product details.

Typically, as with most machine learning projects, data is stored in a dataset format like CSV files. With the increasing need and use of graph-based data representation, this research attempts to study the feasibility of developing a recommendation system based on graph data instead of traditional data formats. The main focus of this paper is to evaluate the efficiency of using graph databases to build recommendation systems compared to traditional database approaches. Although several studies have suggested that graph databases offer better performance in analyzing complex relationships, there is still a gap in detailed practical evaluations comparing graph-based approaches with traditional relational models to determine the feasibility

Received on 1 Sep 2025, Revised on 12 Sep 2025, Accepted and Published on 1 Dec 2025

of using different data representations to obtain more accurate recommendations. The goal of this work is to fill this gap by introducing the following contributions:

1) We identify the limitations of traditional data representations in recommendation systems and encourage the use of graph-based approaches.

2) We design and implement a recommendation system using a graph database (Neo4j and the Cypher query language) and compare it with a traditional approach.

3) We provide an experimental evaluation and discussion that illustrates the conditions under which graph databases can improve recommendation quality.

II. LITERATURE REVIEW

Although using graph databases to introduce a new method for building recommendation systems is relatively a new topic, there are already some papers that have addressed this area in different ways. For instance, [17] presented a survey of knowledge graph-based recommender systems, illustrating how semantic relationships between entities can be used to improve recommendation accuracy. Another study by [18] developed MovieQueue, a prototype recommender system that utilizes a graph database to model movie-user interactions. In [19], an enhanced recommendation method was proposed, combining knowledge graph embedding with neural collaborative filtering. Moreover, in [20], graph data representation was used to address the problem of integrating heterogeneous data sources. The work demonstrated the flexibility and scalability of graph databases in unifying diverse information, which is particularly useful for enhancing the integration of recommender systems. On the other hand, in [21], the graph data representation was adopted to model tag similarity and user interests, replacing traditional synonym-based approaches. By forming tag communities through graph community detection, the proposed method achieved more accurate, clearer, and memorable recommendations.

In contrast, this work aims to implement a recommendation system on the same dataset—first using traditionally stored data in CSV format, and then using the same data modeled and stored as a graph. To the best of the researcher's knowledge, no previous work has provided a clear and practical study that highlights both the advantages and limitations of using graph databases to build recommendation systems.

A. Recommendation Systems

In general, recommendation systems are tools designed to suggest items of interest to users based on past interactions, ratings, or behavior [1]. Over the past decade, several studies and methods have been introduced to develop recommender systems and improve their accuracy and user satisfaction. A variety of mathematical models and algorithms have been employed to design these systems [2][3]. Although there are several types of recommender systems, the most common ones are:

- Collaborative Filtering (CF): relies on user behavior and preferences.
- Content-Based Filtering: depends on item features.
- Hybrid Approaches: combine multiple techniques to enhance performance.

More details about each type can be found in various papers, such as [4][5][6][7].

B. Graph Databases

Graph databases can be considered a type of NoSQL database that uses nodes to represent entities and edges to represent relationships. In the case of social networks, users and items are represented as nodes, while the relationships between them—such as likes or follows—are represented as edges [8][9]. To work with graph databases in practice, Neo4j—a popular open-source graph database management system—is commonly used. Neo4j allows developers to interact with data and execute queries using the Cypher Query Language [10][11][12]. Various operations can be performed using Cypher, such as adding relationships, searching for patterns, and updating nodes [13][14][15][16].

III. RESEARCH METHOD

A. Dataset Overview and Statistics

In this work, a public dataset named Amazon Product Reviews, which is freely available, was used [22]. The dataset contains millions of real interactions between users and products, making it highly suitable for building recommendation systems. Each record in the dataset consists of three key attributes: User ID, Item ID, and Rating. Fig. 1 shows a sample from the dataset.

	user_id	item_id	rating
0	A2NGLQZKRHT4J3	B000VX6XL6	5.0
1	A20FEGDSMTWCNK	B00531E08O	1.0
2	A1X5EZN73CXAEA	B001PUSERA	4.0
3	AX7317G153T8W	B000MGGTY8	5.0
4	A2XMQU9JIPF2B	B00004Z0C7	5.0

Fig. 1. A Sample from The Amazon Product Reviews Dataset

B. Data Reduction and Processing Challenges

One of the main challenges faced in this work was handling the large-scale Amazon Product Reviews dataset, which contained 2,109,869 records. Processing the entire dataset on a personal computer—especially when using a graph database—was not feasible, based on experimental observations.

Therefore, to overcome the computational limitations, a data reduction technique was applied. Users and products were filtered to reduce the total dataset size: users who rated fewer than 100 products were excluded, and products that received fewer than 50 ratings were also removed. After reducing the number of users and products, a total of 10,000 samples were randomly selected. This reduction was essential to implement a graph-based recommendation system, which typically requires significant memory and processing power.

C. Methodology

In this paper, two different methods are used to build the proposed recommendation systems. In the first method, the K-Nearest Neighbors (KNN) algorithm was applied to develop a traditional recommendation system. In contrast, a graph-based recommendation system was implemented in the second method. More details about each method are provided in the following sections.

1) Traditional Recommendation Systems: The traditional method relies on a collaborative filtering model using the K-Nearest Neighbors (KNN) algorithm. The steps are as follows:

1. Create a user-item rating matrix, where each row represents a user, each column represents an item, and each cell contains a rating value.
2. Compute a similarity matrix using the cosine similarity measure to find the k most similar users (neighbors) to a given target user.
3. Recommend items that the target user has not yet interacted with but are common among similar users.
4. Split the dataset into 80% training and 20% testing to evaluate the model's performance on unseen interactions.

This approach enables traditional recommendation methods to be evaluated on the same dataset for direct comparison with the graph-based model. Fig. 2 summarizes the steps, while Fig. 3 shows a sample output.

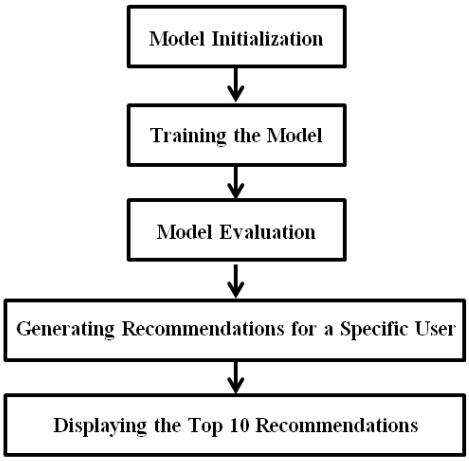

Fig. 2. Steps of The Traditional Method

```
Top 10 Recommendations for the User:
Product: B0019EHU8G | Predicted Rating: 4.69
Product: B00AJHDZSI | Predicted Rating: 4.55
Product: B005DSPLCQ | Predicted Rating: 4.53
Product: B003ES5ZUU | Predicted Rating: 4.50
Product: B00G2TK76A | Predicted Rating: 4.46
Product: B000QUUFRW | Predicted Rating: 4.46
Product: B00DTZYHX4 | Predicted Rating: 4.45
Product: B005FVNGRI | Predicted Rating: 4.45
Product: B001TH7GSW | Predicted Rating: 4.45
Product: B003VAGXWK | Predicted Rating: 4.45
```

Fig. 3. A Sample Output

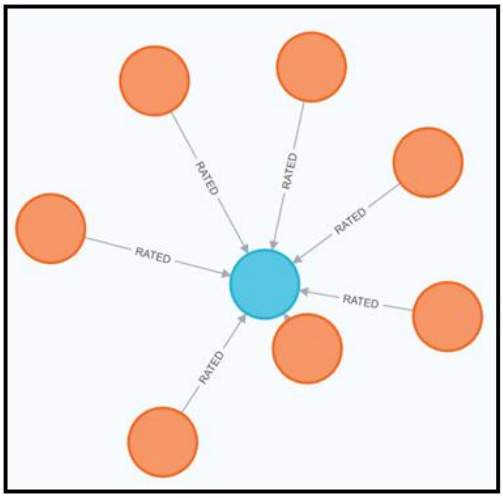

Fig 4. General Graph View Showing Nodes (Users and Items) and Their Relationships

2. Graph-based Recommendation Systems: In the graph-based approach, the Amazon product reviews dataset is transformed into a graph structure, where users and items represent nodes, and interactions (ratings) represent edges.

1. Parse the CSV data to create user and item nodes.

2. Link each user node to the item node using a RATED relationship, which stores the rating value as an edge property.
3. Use Neo4j and the Cypher query language to execute queries that generate recommendations based on graph traversal and adjacency analysis.

Visualization was used to analyze the dataset:

- Fig. 4 shows an overview of the graph database, including nodes and edges.
- Fig. 5 highlights the most active users based on the number of ratings submitted.
- Fig. 6 highlights the most highly rated items.

This methodology enables a practical evaluation of graph-based recommendation systems. Compared with traditional KNN-based collaborative filtering, the graph-based approach can model complex relationships and multi-hop connections more naturally, potentially improving recommendation quality.

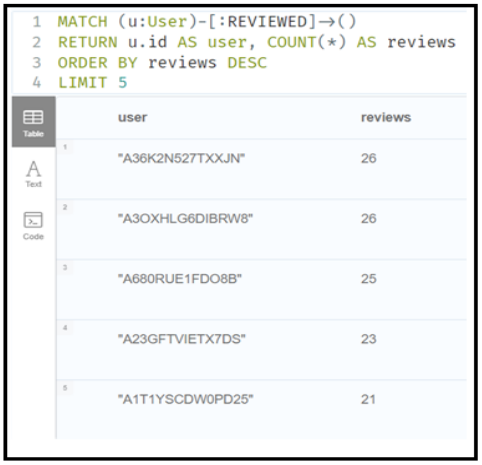

Fig. 5. Visualization of The Most Active Users Based on The Number of Ratings Submitted

Fig. 6. Visualization of The Most Rated Items in The Dataset

D. Evaluation Metrics

To compare the two approaches, standard metrics were adopted:

- Precision: the fraction of recommended items that are relevant.
- Recall: the fraction of relevant items that are successfully recommended.
- F1-Score: balances Precision and Recall.

IV. RESULTS AND DISCUSSION

To evaluate the performance of the proposed methods, Precision, Recall, and F1-Score were calculated, as shown in Table 1. The traditional method achieved a Recall of 0.80, indicating its strength in retrieving relevant items. However, it had a lower Precision of 0.65, suggesting a higher rate of false positives in the recommendations. On the other hand, the graph-based method performed much better in terms of Precision, reaching 0.89, indicating more accurate and relevant suggestions. However, its Recall was slightly lower, at 0.67. When considering the balance between Precision and Recall using the F1-Score, the graph-based method still outperforms the traditional method with a score of 0.76 compared to 0.72. These improved results likely stem from the graph model's ability to naturally capture complex and indirect relationships between items, which traditional table-based systems may struggle to handle.

TABLE I
PERFORMANCE COMPARISON: TRADITIONAL VS. GRAPH-BASED METHODS

	Traditional Method	**Graph-Based Method**
Precision	0.65	0.89
Recall	0.80	0.67
F1-Score	0.72	0.76

Overall, these results demonstrate the practical utility of using graph databases for recommendation tasks and provide researchers with insights for choosing the most appropriate data representation approach based on dataset characteristics and performance requirements.

V. CONCLUSION

With the increasing adoption of recommendation systems, and the increasing need to employ graph-based data representation to improve efficiency, this research presents a model for practically integrating these two topics by presenting a recommendation system based on graph-structured data. The research relied on Amazon data, originally stored in CSV format, which was programmatically converted into a graph format to be later used to generate the desired recommendations. The experimental results highlighted several key insights:

- Graph-based approaches outperformed traditional methods in Precision and F1-Score, demonstrating their ability to deliver more accurate and relevant recommendations.
- Traditional methods exhibited higher Recall, indicating a trade-off between retrieval rate and recommendation precision.
- Graph-based systems require higher computational resources, emphasizing the need for careful consideration of infrastructure when handling large-scale graph data.

The results indicate that graph databases offer practical advantages for recommender systems, particularly when capturing complex and indirect user-item relationships. For future research, it would be valuable to examine the feasibility of employing techniques such as parallel processing or the Apache Spark environments to improve the efficiency of handling of graph-based data.

REFERENCES

[1] A. Al-Badarenah and J. Alsakran, "An Automated Recommender System for Course Selection," 2016. [Online]. Available: www.ijacsa.thesai.org

[2] B. Alhijawi and Y. Kilani, "The Recommender System: A Survey," Int. J. Adv. Intell. Paradigms, vol. 10, no. 1, p. 1, 2018, doi: 10.1504/ijaip.2018.10010164.

[3] P. M. Chawan, "Recommendation System using Machine Learning Techniques," 2022. [Online]. Available: www.irjet.net

[4] M. Sharma, D. Saxena, and A. K. Singh, "A Survey and Classification on Recommendation Systems," [Online]. Available: https://www.researchgate.net/publication/357562406

[5] C. Mediani, S. Harous, and M. Djoudi, Content-Based Recommender System using Word Embeddings for Pedagogical Resources, 2023, doi: 10.1109/PAIS60821.2023.10321989.

[6] Z. Mundher and M. T. Ahmad, "A Recommendation System for University Discussion Committees," SISTEMASI, vol. 13, no. 6, pp. 2722–2731, Nov. 2024, doi: 10.32520/stmsi.v13i6.4726.
[7] Z. Mundher, "Recommendation systems: A Review," International Journal of Electrical Engineering and Intelligent Computing, vol. 2, pp. 43–47, 2020, doi: 10.33387/ijeeic.v2i2.9956.
[8] H. Baqal and M. Sidiq, "Graph Databases: Revolutionizing Database Design and Data Analysis," Curr. J. Appl. Sci. Technol., vol. 43, pp. 45–56, 2024, doi: 10.9734/cjast/2024/v43i114443.
[9] V. Santos and B. Cuconato, "NoSQL Graph Databases: an overview," arXiv preprint, arXiv:2412.18143, 2024, doi: 10.48550/arXiv.2412.18143.
[10] L. Pop and L. Miclea, "Implementation of network data analysis using the Neo4j graph database," Acta Univ. Sapientiae, Informatica, vol. 13, no. 2, pp. 123–134, 2021. [Online]. Available: https://acta.sapientia.ro/content/docs/14-357886.pdf
[11] A. Kumar and R. Singh, "Issues and concepts of graph database and a comparative analysis on list of graph database tools," Int. J. Sci. Technol. Res., vol. 9, no. 5, pp. 1234–1240, 2020. [Online]. Available: https://www.researchgate.net/publication/341812166
[12] Y. He, R. Fang, I. Dillig, and Y. Wang, "Graphiti: Bridging Graph and Relational Database Queries," arXiv preprint, arXiv:2504.03182, 2025, doi: 10.48550/arXiv.2504.03182.
[13] N. Francis et al., "Cypher: An evolving query language for property graphs," Proc. 2018 Int. Conf. Management of Data (SIGMOD '18), pp. 1433–1445, 2018, doi: 10.1145/3183713.3190657.
[14] N. Francis et al., "Formal Semantics of the Language Cypher," arXiv preprint, arXiv:1802.09984, 2018, doi: 10.48550/arXiv.1802.09984.
[15] N. Francis et al., "Cypher: An Evolving Query Language for Property Graphs," Proc. 2018 ACM SIGMOD Int. Conf. Management of Data, pp. 1433–1445, 2018, doi: 10.1145/3183713.3190657.
[16] A. Green et al., "Updating graph databases with Cypher," Proc. VLDB Endowment, vol. 12, no. 11, pp. 2242–2254, 2019, doi: 10.14778/3352063.3352139.
[17] Q. Guo et al., "A survey on knowledge graph-based recommender systems," IEEE Trans. Knowl. Data Eng., pp. 1–1, 2020, doi: 10.1109/TKDE.2020.3028705.
[18] R. C. Juricic, "MovieQueue: Leveraging graph databases to build recommender systems," SPARK Symposium Presentations, no. 114, 2025. [Online]. Available: https://spark.parkland.edu/spark/114
[19] Z. Shokrzadeh, M. R. Feizi Derakhshi, M. Balafar, and J. Bagherzadeh, "Knowledge graph-based recommendation system enhanced by neural collaborative filtering and knowledge graph embedding," Ain Shams Eng. J., vol. 15, p. 102263, 2023, doi: 10.1016/j.asej.2023.102263.
[20] Kwon, Junhyuk & Ahn, Seokho & Young Duk, Seo. (2025). RecKG: Knowledge Graph for Recommender Systems. 10.48550/arXiv.2501.03598.
[21] Shokrzadeh, Zeinab & Feizi Derakhshi, Mohammad Reza & Balafar, Mohammad & Bagherzadeh, J. (2023). Graph-Based Recommendation System Enhanced by Community Detection. Scientific Programming. 2023. 1-12. 10.1155/2023/5073769.
[22] Saurav, "Amazon Product Reviews," Kaggle Dataset, 2021. [Online]. Available: https://www.kaggle.com/datasets/saurav9786/amazon-product-reviews. [Accessed: 27-May-2025].