STM JOURNALS

Journal of
Advanced Database
Management & Systems

ISSN: 2393-8730
Volume 12, Issue 3, 2025
September–December
DOI (Journal): 10.37591/JoADMS

https://journals.stmjournals.com/joadms

Research JoADMS

An Analysis of Graph Database in Data Modelling and Analysis for a Recommendation System

Tushar Samaniya[1,*], Sankeeta Jha[2]

Abstract

This research work focuses on graph databases, mainly Neo4j databases, in recommendation systems for e-commerce websites. The importance of research is that it explains how graph databases efficiently handle the complex relationship between user-items, which is difficult for traditional databases. Sparsity, limited diversity, and high setup costs are the challenges traditional databases face. This research work overcomes these problems using Ne04j with Cypher query language and graph algorithms (PageRank, Shortest Path, Community Detection), which give real-time and personalized suggestions. There is use of data modelling (nodes, edges, properties), collaborative and content-based filtering, a hybrid approach, which includes Neo4j along with Deep learning (Graph Neural Networks) as a Methodology. The result of this research work is that these systems are fast, scalable, and accurate, like the "frequently bought together" suggestion in Amazon. These systems increase the user experience and business revenue, but privacy and scalability is still the challenging part.

Keywords: Graph database, algorithm, data modelling, deep learning, MySQL databases, E-commerce, amazon, Flipkart

INTRODUCTION

While using online shopping platforms like Amazon, Flipkart or Movie streaming platforms like Netflix, it has been observed that when the user clicks on some product or movie, then suddenly a series of suggestions arise that "Also try this"; this happens due to the recommendation system. This system observes what the user likes, the products, moves, and according to it suggests the specific products or movies from the bundle. To make the recommendation system, it is not preferred to use older databases like Relational databases which are MySQL databases. It becomes very complex to establish the relationships between the products and products and between the user and the product [1]. As the connection between the user and the products is large, then these systems which are made up of older databases become slow. In this research work, uses of Neo4j graph databases are provided. How these Neo4j databases are able to connect so many relationships between the user and the products. It stores the data in nodes and edges, which clearly shows the connection between users and the products. By using these databases, users get the suggestion very fast, efficiently and accurately like "frequently bought together" in Amazon [2]. In this research work, there are three ways of recommendation systems: collaborative filtering, content-based filtering and hybrid-based filtering. In combination with all these things, a real-time suggestion for the e-commerce website is created which is profitable for the user and the business of the company [3]. This system also faces challenges like it requires high setup cost, data is also very

***Author for Correspondence**
Tushar Samaniya
E-mail: 23cse139tushar@eitfaridabad.co.in

[1]Student, Department of Computer Science and Engineering. Echelon Institute of Technology, Faridabad, Haryana, India
[2]Assistant Professor, Department of Computer Science and Engineering Echelon Institute of Technology, Faridabad, Haryana, India

Received Date: June 28, 2025
Accepted Date: October 06, 2025
Published Date: October 15, 2025

Citation: Tushar Samaniya, Sankeeta Jha. An Analysis of Graph Database in Data Modelling and Analysis for a Recommendation System. Journal of Advanced Database Management & Systems. 2025; 12(3): 33–39p.

limited for the new users, privacy is also an issue; however Neo4j databases is the future of databases for recommendation systems [4].

BACKGROUND

Graph Database

A graph database is a type of NoSQL database that uses graph theory to represent and store data in the form of nodes, edges and properties. Nodes represent entities, Edges represent relationships between the nodes, properties are attributes that describe nodes and edges [5].

Graph database is optimized for query which means it has high performance. It is highly scalable means it can store a large amount of data. Graph databases are schema-less which means data models are involved over time, this makes it more flexible and also adaptable to changes in business model. These are the features of a graph database which make it different from other databases [6].

When a query is executed in a graph database, the database traverses the graph to find the relevant data. This traversing is often done with the graph algorithm, which can easily find patterns and relationships with the data. As a result, graph databases are particularly well suitable for the queries that involve complex relationships and patterns; for example, social media network systems under the graph database often use a combination of indexing and caching techniques to optimize the query performance [7].

To achieve optimal performance, we use graph databases. Neo4j, Amazon Neptune, Microsoft Azure Cosmos DB and etc., are the most popular graph databases. To make a recommendation system, Neo4j graph database is used (Figure 1).

Neo4j

Neo4j is an open-source NoSQL graph database, which stores data in the form of nodes and edges. It is different from traditional databases like MySQL because Neo4j databases focus on the relationships or connections [8].

Nodes: These are entities like movies or products.

Edges: These are the relationships between the nodes.

Properties: It is the attribute of Nodes and edges, for example, name of the user or name of the movie.

To understand Neo4j in a better way, it is required to understand Cypher. Cypher is Neo4j's declarative query language. Cypher is similar to SQL, but optimized for graphs. It was created to allow users to focus on what data to retrieve from a gram rather than how to retrieve it [9].

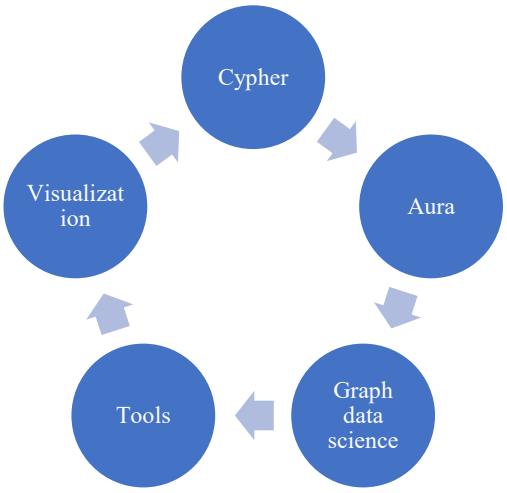

Figure 1. Flowchart of Neo4j.

To write the query to search the data, cypher programming language is used. Due to index-free adjacency, Neo4j is fast.

RECOMMENDATION SYSTEM

A recommendation system algorithm is a type of software that gives personalized suggestions to the user, based on their preference, behavior and past interaction [10].

For example,

- *Netflix:* If liked "Inception" then watch "Matrix".
- *Amazon:* If you are looking for the "Headphone" then you also see "Laptop".
- *Spotify:* If you listen to that song, then try this playlist.

The goal of the Recommendation system is to improve user experience, increase user engagement, and generate revenue for the business.

Key components of the recommendation system are:

1. *User:* Person who retrieves suggestions.
2. *Items:* Products, movies, songs, i.e., type of content which are suggested to the user.
3. *Interaction:* It includes actions like rating, purchases, clicks, views etc. in between the users.
4. *Algorithm:* It creates or decides the logic to suggest the items.

What is the Recommendation System?

Along with it, it is important to have a recommendation system in our platforms, as it serves the user's perspective by saving time and showing relevant content. It also serves a business perspective as it increases customer retention, sales and loyalty. In today's data-driven world, where big data plays a crucial role, recommendation systems provide a competitive edge to the companies.

Types or Methods of Recommendation System

This research work primarily focuses on the E-commerce recommendation system. There are mainly three types of E-commerce recommendation systems.

1. *Collaborative Filtering:* This recommendation system is the most popular recommendation system. It suggests the recommendation based on interactions between the user and items like ratings, purchases etc. It works on the principle that if two users have the similar taste then it suggests the items liked or purchased by the first user are likely to be preferred by the other user. Collaborative filtering is further classified into two parts:
   a. User-Based Collaborative Filtering and
   b. Item-Based Collaborative Filtering.

   In User-based collaborative filtering, it finds the similar users which is based on rating and behavior.

   In Item-based collaborative Filtering, it finds similar items which are based on users who like both the items.

   The advantages of Collaborative filtering: Scalability is better and a lower Cold play problem, as it is enough for the new user to have to existing similar items.

   The main problem of Collaborative Filtering is Limited diversity which means it is limited to suggest only similar items.
2. *Content-based Filtering:* This recommendation system uses the approach to suggest the items based on item features and user preferences. It works on the principle that it suggests those items to the user, which are similar to the past items. For example, Headphones and Laptop are mainly purchased in combination, the system would suggest a laptop to a user who has purchased headphones.

   The advantages of Content-based filtering are the same as that of Collaborative filtering which is Scalability is better and Cold start problem is low.

Along with it the challenge in content filtering is same as that of Collaborative filtering which is Limited diversity.

The key component of Content-based Filtering is:

a. *Item Profile:* It is the vector of the product features. For example, {brand: Sony, category: Electronics, price: 2000, color: Black}.
b. *User Profile:* It is the vector of the User's preferences. For Example, {prefers: Electronics, brand: Sony, price range: 1000–3000}.
c. *Matching:* Cousin and TF-IDE have similarity (Cousin and TF-IDE are mathematical tools or formulas in which Recommendation system logic is created).

3. *Hybrid Recommendation System:* This type of recommendation system combines both collaborative and content-based recommendation systems which is due to it covers other two weaknesses. It works on the method that is:
   a. *Weighted Hybrid:* Combines the scores of both approaches by giving them weights.
   b. *Switching Hybrid:* It chooses one approach on the basis of context. For example, for new users it chooses content-based and for old users it chooses collaborative.
   c. Initial recommendations from one approach, then refinement using the other.

   This Hybrid recommendation system addresses the problems like Cold start, sparsity and diversity. Along with it, this recommendation system gives more accurate recommendations.

Graph database in recommendation system

Till now it is clear what is meant by Graph database and what actually is a recommendation system. Before understanding how a graph database works in a recommendation system lets first understand why graph databases are used for the Recommendation system.

Graph databases are perfect for recommendation system because:

a. *Natural Relationship Modelling:* It stores nodes and edges in the form of User-item interaction.
b. *Fast Traversal:* Neo4j uses index-free adjacency, in which the relationship is directly traversed; it works without being costly.
c. *Real-Time Recommendation:* In E-Commerce websites it is necessary to have real-time suggestions, due to this the user can get the related products when the user gets in the "Add to cart" section. Neo4j Cypher queries give real-time results.
   - *Scalability:* It can handle billions of nodes and edges, like the Amazon platform.
   - *Rich Queries:* Complex Queries like "Users which are similar to Rahul and purchase products under the Electronics category."

HOW GRAPH DATABASE WORKS IN RECOMMENDATION SYSTEM

This section is the core part of this research work; this section can deeply be understood by the help of taking the example of Neo4j.

Data Modelling

- *Nodes:* It includes Users, Products, Categories, Brands, Genres. For example, Rahul is users, headphones is product and electronics is category, Sony is brand.
- *Edges:* It includes relationships like "purchased", "viewed", "liked", "co-purchased", "belongs to". For example, Rahul purchased headphones on 02-05-2025, The headphones belong to electronics, headphones have brand Sony and Rahul viewed it from his laptop.
- *Properties:* It includes the name of the user, price of the product, purchase date, and rating. For example, Headphones are co-purchased to a laptop because most of the time they are purchased together.

Querying with Cypher

```
MATCH (p: Product {name: "Hearphones"})<- [:PURCHASED]-(u: User)-[PURCHASED] ->(recommendation: Product)
```



```
RETURN recommended name, count(*) AS score ORDER BY score DESC KIMIT 5
```

In Neo4j, Cypher query language is used, which makes simple and expressive graph-based queries.

Collaborative Filtering Query

```
MATCH (u:User {name: "Rahul"})-[:PURCHASED]->(p:Product)-[:BELONGS_TO]-
>(c:Category {name: "Electronics"})
MATCH (recommended:Product)-[:BELONGS_TO]->(c)
WHERE recommended.brand = "Sony" AND recommended.price BETWEEN 1000 AND 3000
RETURN recommended.name, recommended.price
LIMIT 5
```

These types of query used Recommendation system like “Frequently bought together” for example Headphones with Laptops.

Content-Based Filtering Query

This query suggests products to Rahul having preferences like Electronics and Sony etc.

Hybrid Query

```
MATCH (u:User {name: "Rahul"})-[:PURCHASED]->(p:Product)<-[:PURCHASED]-
(other:User)-[:PURCHASED]->(recommended:Product)

WHERE recommended.brand = "Sony"
RETURN recommended.name, count(*) AS score
ORDER BY score DESC
LIMIT 5
```

These are the hybrid queries which include the collaborative query which is similar to users and content-based which is the Sony brand.

Graph Algorithm: Neo4j enhances recommendation system.

1. *PageRank Algorithm:* This Algorithm identifies the important products based on frequent purchases and connections. For example, if Headphones are purchased by more customers, then their PageRank score is high.
2. *Shortest Path Algorithm:* It finds the relevant path from user to product. For example, Rahul → purchased → headphones → co-purchased → Laptop.
3. *Community Detection Algorithm:* This algorithm helps in finding the clusters of similar users and products. For example, it creates a community for electronics lovers. With the help of Community Detection algorithm, the real-time recommendation system is fast and can easily handle complex relationships.
4. *Deep Learning Algorithm:* This algorithm captures the advanced patterns and complex relationship, which is difficult from the older algorithms.

Using Graph databases (Neo4j) along with deep learning algorithms is more powerful because graphs give structured data and deep learning makes the recommendation system with the help of this high-quality data. There are two deep learning algorithms:

1. *Neural Collaborative Filtering (NCF):* This algorithm uses Multi-layer Perceptron’s to capture the non-linear relationships; and
2. *Graph Neural Networks (GNNs):* GNNs generate embeddings (numerical representations) for nodes by aggregating information from their neighbors. Amazon uses GNNs with graph databases to suggest co-purchased products (Figure 2).

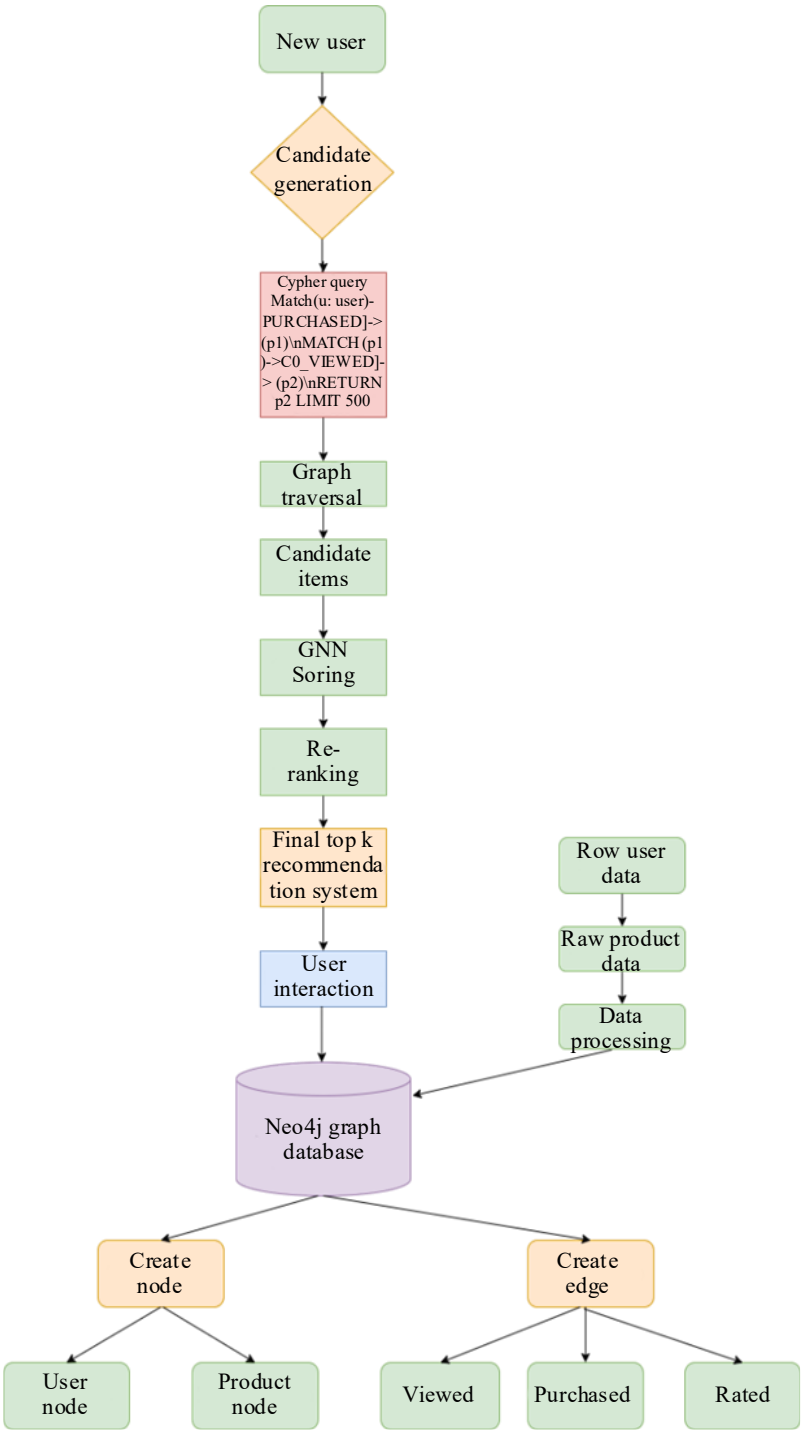

Figure 2. Using graph databases (Neo4j).

Real-World E-commerce Examples:

- *Amazon:* Its recommendation system is based on collaborative filtering.
- *Alibaba:* Its recommendation system is based on GNN, Deep learning, collaborative filtering.
- *Flipkart:* Its recommendation system is based on Hybrid.

Challenges and Limitations

Although there are benefits to using Neo4j graph databases to make the recommendation system, it also faces Challenges like High Setup and Maintenance costs. It is very costly and complex to set up

the recommendation system. To maintain the recommendation system, expertise is required, which means it is required to have skills on graph modelling and Cypher. For the new users, the data is very limited, therefore it affects the recommendation system, this problem is known as Data Sparsity. Scalability Limited, to run the recommendation system, there might be a large number of graphs used which directly affects the performance of the recommendation system. Privacy concerns are also one of the major challenges faced by the recommendation system, which means it is very risky for the user to share or store the data over the internet or the platform. It also creates a low Diversity issue in recommendation system, that is, graphs-based systems focus only on the popular items, for which the diversity of the items is low.

CONCLUSION

In this research work, it is seen how graph databases like Neo4j make a better recommendation system. This database stores users and items (like products and movies) in their connections with nodes and edges. By using this, it is very easy to give and receive the suggestions. For example, when we buy headphones from Amazon then it gives suggestions to buy a laptop also. It happens with the help of Neo4j graph database. Movie streaming platforms like Netflix also use this technology to give the suggestions to the users what they like, and along with this, the company's business also grows. Graph databases are faster and more flexible compared to old databases as they directly handle the connections.

Using Neo4j is challenging also because it requires more money and hard work for the setup of Neo4j, data is also very limited for the new users. However, the future of these databases is also very bright. Taking suggestions from Graph neural Networks (GNNs) is more accurate in future, because it has the ability to understand deeper patterns. It can also show the real time suggestions enabling users to buy new products quickly. In addition, emerging methods now emphasize user data privacy to ensure information security. Furthermore, Explainable AI also clarifies why the suggestions are presented.

REFERENCES

1. Kumar A. Implementing real time recommendation systems using graph algorithms & exploring graph analytics in a graph database platform (neo4j). Doctoral dissertation. Dublin: Dublin Business School; 2019.
2. Dai J, Jia Z, Gao X, Chen G. A Hierarchical Optimizer for Recommendation System Based on Shortest Path Algorithm. arXiv preprint arXiv:1911.08994. 2019 Nov 7.
3. Giabelli A, Malandri L, Mercorio F, Mezzanzanica M, Seveso A. Skills2Job: A recommender system that encodes job offer embeddings on graph databases. Appl Soft Comput. 2021 Mar 1; 101: 107049.
4. Stanescu L. A Comparison between a Relational and a Graph Database in the Context of a Recommendation System. In Fed CSIS (Position Papers). 2021 Sep; 133–139.
5. Mohammedali N. Recommendation System Based on Graph Database Techniques. Int Res J Eng Technol. 2019; 6(10): 754–63.
6. Fayyaz Z, Ebrahimian M, Nawara D, Ibrahim A, Kashef R. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. Appl Sci. 2020 Nov 2; 10(21): 7748.
7. Morales A, Gonzalez J, Alquerque K, Reyes J, Sanchez S. Recommendation system with graph-oriented databases for repository of open educational resources. In IOP Conf Ser: Mater Sci Eng (IOP Publishing). 2021 Jun 1; 1154(1): 012021.
8. Timón-Reina S, Rincón M, Martínez-Tomás R. An overview of graph databases and their applications in the biomedical domain. Database. 2021 Jan 1; 2021: baab026.
9. Xu J, Chen Z, Yang S, Li J, Wang W, Hu X, Hoi S, Ngai E. A Survey on Multimodal Recommender Systems: Recent Advances and Future Directions. arXiv preprint arXiv:2502.15711. 2025 Jan 22.
10. Patel AA, Dharwa JN. An integrated hybrid recommendation model using graph database. In 2016 IEEE International Conference on ICT in Business Industry & Government (ICTBIG). 2016 Nov 18; 1–5.

