GRAPH NEURAL NETWORKS IN RECOMMENDER SYSTEMS: AN OVERVIEW AND COMPARATIVE ANALYSIS

Nikolaos Kakonas

Athens University of Economics and Business

Abstract

This thesis explores the integration of GNNs into recommender systems to overcome the limitations of traditional models, such as data sparsity, the cold-start problem, and scalability issues. GNNs, which can simulate intricate, non-Euclidean data structures like user-item interactions, present a promising way to improve the precision, resiliency, and scalability of recommender systems. In addition to technical advancement, this improved performance has important implications for greater commercial value by making recommendations that are more accurate and tailored, increasing user engagement, revenue growth, and customer loyalty. In this work, a thorough introduction to recommender systems and GNNs is first provided, followed by a detailed investigation of GNN-based recommender systems. The state-of-the-art GNN architectures ConsisRec, DMGCF (Dynamic Multi-Graph Collaborative Filtering), and KGNN-LS (Knowledge-aware Graph Neural Networks with Label Smoothness Regularization), which are all utilized in recommender systems, are also compared. The objective is to comprehend how the performance of recommender systems may be improved by using GNNs while also appreciating the associated business value. As the body of current knowledge is concurrently expanded and the development of more advanced, trustworthy, and user-centric recommender systems is furthered, research problems and future directions in this dynamic subject are also highlighted.

1 Introduction

Massive amounts of data have been produced by the quickly developing digital environment, providing organizations with a variety of opportunities and difficulties to better understand and engage their people. Among these, recommender systems have come to be recognized as a key instrument for creating tailored experiences, assisting user decision-making, and promoting business success in a variety of industries, from e-commerce to entertainment and beyond. The classic models of recommender systems do, however, have some drawbacks, such as data scarcity, the cold-start problem, and scalability problems, which frequently lower the quality of recommendations.

In response to these difficulties, the use of GNNs in recommender systems has become a potential field of study. By utilizing the modeling capabilities of GNNs to represent complicated, non-Euclidean data in the form of graphs, such as user-item interactions, this method enables more reliable and precise recommendation systems.

This thesis gives a thorough investigation of GNN-based recommender systems. It begins with a thorough explanation of recommender systems and GNNs, then delves deeply into how these ideas are combined. Additionally, it contrasts a number of cutting-edge models for recommender systems that use GNNs, such as ConsisRec, Dynamic Multi-Graph Collaborative Filtering, and Knowledge-Aware Graph Neural Networks with Label Smoothness Regularization (KGNN-LS).

The main goals of this study are to understand how GNNs might improve the performance of recommender systems as well as to identify the difficulties and potential future paths in this field of study. By doing this, this thesis hopes to add to the body of information already available and open the door to new developments in the development of sophisticated, trustworthy, and user-centric recommender systems.

2 GNN-based Recommender Systems: An Overview

A thorough introduction of Graph Neural Network (GNN)-based Recommender Systems will be given in this section. We'll start by thoroughly introducing both recommender systems and GNNs. The complexity of GNN-based recommender systems will then be explained in more detail.

2.1 Introduction to Recommender Systems

It is essential to first comprehend the concept of recommender systems, the different types of recommender systems, how they work, how important they are to users and businesses, as well as the limitations of conventional recommender system models, in order to fully comprehend GNN-based recommender systems.

2.1.1 Definition

Recommender systems are a type of filtering system that attempt to give users individualized information, hence boosting the user experience and promoting business profitability [1]. According to a further study by M. Mollanoroozi in 2022 [2], recommender systems have become an important tool for handling the sizable amount of data and meeting needs. Additionally, according to Mollanoroozi, these systems capture both explicit and implicit information about users in order to present users with a carefully curated list of suggested products. The study of recommendation problems that explicitly depend on the ratings structure gave rise to the field of recommender systems in the mid-1990s [3]. The suggestion challenge is simplified to the issue of estimating ratings for the things that a user hasn't viewed, according to the same study, in its most typical version. The authors of this work further support the notion that, when ratings for the items that have not yet been rated can be approximated, the item(s) that have the highest estimated rating(s) can be recommended to the user.

2.1.2 Types of recommender systems

In her book [4], Charu C. Aggarwal lists several well-known varieties of recommender systems, including:

• **Collaborative Filtering:** Using the tastes and actions of comparable users, this method suggests products. To find patterns and create tailored suggestions, it makes advantage of user-item evaluations or interactions.

• **Content-Based Filtering:** This method makes suggestions for products based on the traits or contents of the products themselves. It evaluates the characteristics or properties of an item and proposes comparable goods to those in which a user has expressed interest.

• **Knowledge-Based Recommender Systems:** These systems base their recommendations on explicit knowledge of user preferences and item attributes. To determine user preferences and offer tailored recommendations, they frequently rely on rule-based or expert systems.

• **Hybrid Recommender Systems:** Hybrid systems combine multiple recommendation approaches, such as collaborative filtering and content-based filtering, to provide more accurate and diverse recommendations. They aim to leverage the strengths of different techniques and overcome their individual limitations.

• **Context-Aware Recommender Systems:** Context-aware systems deliver recommendations by considering additional contextual elements like time, location, and user context. Depending on the precise situational setting in which the user is making decisions, they modify their recommendations.

• **Knowledge Graph-Based Recommender Systems:** These systems leverage structured knowledge graphs to represent relationships between users, items, and other relevant entities. They utilize graph-based algorithms to generate recommendations by traversing and analyzing the connections within the knowledge graph.

• **Demographic:** When specific information about a user's preferences is not accessible, demographic data might help provide recommendations that are somewhat tailored. Demographic data may contain things like age, gender, place of residence, level of education, etc. The things associated with this stereotype are suggested once the demographic data is matched to a stereotype. Due to the generalization to a stereotype, personalization for the user is constrained [5].

2.1.3 Importance of Recommender Systems for Users and Business

Due to their ability to create value, recommender systems are crucial for individuals as well as for organizations. The production of value for customers, suppliers, and other key stakeholders is these systems' main goal. In essence, they improve the user experience while also providing the participating businesses with significant value. Users can get recommendations from recommender systems on products they might like to check out or buy. Such systems' recommendations can aid users in navigating through vast amounts of product descriptions, news articles, and other content [6]. They can improve user experience in a number of ways as well. [7] list a few of them as follows:

• **Personalized Recommendations:** • Recommender systems offer customized suggestions based on user preferences, saving time and effort when locating pertinent content.

• **Improved Decision-Making:** Users receive options that have been carefully chosen to fit their tastes, enabling them to make well-informed choices without feeling overloaded with options.

• **Serendipitous Discovery:** Recommender systems expose consumers to fresh, varied information, broadening their horizons and providing fresh experiences.

• **Enhanced User Engagement:** Personalized recommendations keep users interested, which promotes more interaction and user happiness.

• **Time Savings:** By offering users the most pertinent options, recommender systems help users save time and effort.

In relation to business value and drawing insights from the literature sources, namely [8] [9] [10] [11] [12], there exists a consensus that recommender systems can yield positive business impacts across various dimensions [13]. The ideas that these systems have the capacity to provide organizations with vital measures that enhance their worth and profitability are further supported by the works of Jannach and Jugovac. Sales or income numbers, click-through rates (CTR), increased user engagement, and customer retention rates are examples of these indicators.

2.1.4 Limitations of Traditional Recommender Systems

Several issues have been found that potentially restrict the effectiveness and usability of classic recommender systems, despite the apparent benefits and rising dependence on them in numerous industries. These restrictions typically result from a number of difficulties that classic recommender systems have by nature.

First off, these systems frequently struggle with the "cold-start" problem, which arises when the system makes recommendations to new users who have no prioritizations for any things or does so by suggesting items that no other members of the community have yet seen [14]. To improve the cold start issue in recommendation systems, numerous initiatives have been made. Regrettably, established approaches haven't been able to adequately address this problem or make significant gains. Second, scalability problems plague conventional recommender systems frequently [15]. The authors point out that conventional recommender systems need a lot of training data, especially those that use collaborative filtering techniques. Finding what people are seeking for among a huge number of items in large databases becomes more difficult as both the number of users and the number of objects rises. Due to scalability issues caused by the need for a lot of data, it is challenging to efficiently offer accurate and individualized recommendations. The tendency of classic recommender systems to display a popularity bias is another key drawback [16]. The authors contend that while earlier studies have emphasized the item-centered perspective of popularity bias, it is important to analyze how this prejudice differs for various users. They draw attention to the fact that not all users are equally interested in popular or less popular content, and that popularity bias can have varied degrees of effects on user groups based on their interest in popular goods. Users who had less interest in popular music or movies, for instance, were found to be more adversely affected by the popularity bias.

2.2 Introduction to Graph Neural Networks

A potent method for studying graph-structured data is to use GNNs. By leveraging the inherent connectivity and structure of graphs, GNNs can capture complex relationships, exploit local and global structures, and learn representations of nodes. By combining data from nearby nodes and creating an extensive graph-level representation, they are excellent at managing graph data. Although there is a lot of potential for GNNs in areas like social network analysis, predicting molecular structures, and recommendation systems, there are still difficulties to be solved, including scalability, interpretability, and combining different features. By addressing these issues, future research can reveal GNNs' full potential and open the door to a larger range of real-world applications.

2.2.1 Overview of Graph Neural Networks

Graphs are the fundamental data structure in this context and must first be understood in order to fully understand the notion of Graph Neural Networks. Consequently, graphs have played a significant role in presenting issues for a very long time because of their capacity to represent the real world in a way that is simple to understand. Because there is a wealth of relationship between the data items in graph data, they are also utilised. In a broader sense, a graph is a type of data structure that is employed to represent a collection of objects (nodes) and the connections between them (edges) [17]. Graphs can have features at the node, edge, or graph level and can be weighted or unweighted, directed, or both. In a directed graph, an edge from node i to node j denotes a relationship from i to j but not the other way around. The relationship is bidirectional in an undirected graph. Each edge in a weighted graph is given a weight, which frequently indicates the importance or strength of the link. People might be the nodes in a social network graph, friendships could be the edges, and shared interests could be the edge weights [18].

Over a decade ago, GNNs were developed as a method for dealing with complex data that is best represented as graphs with numerous connections and dependencies between objects. These data consist of biological interactions, genetic information, social networks, and road networks. GNNs are better suited to processing data in the form of graphs with irregular sizes and shapes than Convolutional Neural Networks (CNNs), which are excellent at processing data like images, text, or video. In order to execute prediction or classification tasks at the level of the entire graph, or for each node or edge, GNNs learn to encode information about the local surroundings of each node in a graph [19].

Utilizing a node's neighborhood for feature learning is a fundamental concept in GNNs. In other words, local feature extraction occurs when a node's feature is updated by combining features from its neighbors and its own outdated features. GNNs are able to recognize both local and global structures in the data thanks to this repeated process of local aggregation and transformation that produces a global graph-level representation [20].

The types of graphs used in GNNs are mostly determined by the task and data. For example, simple hierarchical graphs may be necessary for molecular biology whereas simple graph topologies may be sufficient to explain social interactions [21].

2.2.2 Key Components of GNNs

Computational models' efficacy in the dynamic fields of data science and artificial intelligence frequently depends on their capacity to faithfully reflect and capture the intricate patterns seen in data. GNNs are one such potent paradigm that has proved essential in revealing the intricate details concealed within organized data. The essence of these networks'

operation and their capacity to represent interrelated interactions require a thorough examination of their fundamental constituents in order to fully grasp their complexity. These core components, which include nodes, edges, node features, edge features, propagation rules, and the readout function, are covered in this section. Each of these components contributes to the overall resilience and adaptability of GNNs.

1. Nodes and Edges: The structure of a graph is formed from nodes, often referred to as vertices, and edges, which are also known as links or connections. Nodes stand as representations of entities within the graph, with edges acting as illustrations of the relationships or interactions existing between those entities. As an example, in the graph of a social network, each individual is symbolized by a node, whereas an edge symbolizes the existence of a friendship between two individuals. This relationship can be depicted in two ways, either directed or undirected. In a directed graph, edges possess a direction. An example of such a situation is the follower relationships on Twitter. On the other hand, in an undirected graph, edges do not have a direction. An example of such a situation is friendships on Facebook [21].

2. Node Features and Edge Features: Nodes and edges can be enriched with attributes, often referred to as features. Node features might represent attributes or characteristics of the entities, such as a user's age, gender, or interests in a social network. Edge features, on the other hand, could represent the attributes of the relationships, like the strength or type of a relationship, or timestamps of interactions. In many GNN models, these features serve as the input to the model, allowing the GNN to capture complex patterns based on both the structure of the graph and the features of nodes and edges [20].

3. Propagation Rule (or Message Passing): The propagation rule, often implemented as a message-passing mechanism, is what makes GNNs unique. The feature representation of a node is updated based on its own features and the features of its neighboring nodes. This rule is applied iteratively over the entire graph, with the features of nodes updated at each step. The exact way in which features are propagated and updated can vary between different GNN models. For example, Graph Convolutional Networks (GCNs) use a weighted average of the features of a node and its neighbors [18].

4. Readout Function: After multiple rounds of applying the propagation rule, the GNN uses a readout function to aggregate the features of all nodes in the graph into a single vector that represents the entire graph. This graph-level representation can then be used for downstream tasks, such as graph classification or regression. The exact nature of the readout function can vary between different GNN models and tasks. For example, one simple readout function could be to sum or average the features of all nodes (Junhyun, Lee, and Kang, 2019).

2.2.3 Advantages of GNNs in Handling Graph Data

Convolutional and recurrent neural networks, as well as deep learning, have helped to advance artificial intelligence and machine learning to new levels, effectively tackling a variety of challenging issues [22]. But as the scientific and technological community focuses more on organized data, difficulties with non-Euclidean data, like graphs, become apparent. Despite their effectiveness with Euclidean data, conventional neural networks struggle with graph data [23].

GNNs have become an effective method to manage such structured data in this situation. GNNs stand out due to their ability to properly represent interconnected nodes and edges and so capture relationships within the data [24]. The ability of GNNs to process and learn from non-Euclidean data makes them a special tool in modern machine learning [21].

In this section, the considerable benefits of using GNNs to manage graph data are outlined. The advanced learning capabilities of GNNs, their adaptive learning of node attributes, and their key trait of invariance to graph isomorphism are explored [25]. How GNNs scale, which is important when working with large graphs, is also discussed.

1. Capturing Complex Relationships: When processing non-Euclidean data, such as graph structures, GNNs perform noticeably better than conventional neural networks. This is principally caused by the fact that GNNs naturally process entities (nodes) and the connections between them (edges), as was already mentioned above. GNNs are particularly adept at simulating intricate relationships between things represented in a graph. Since traditional neural networks frequently presuppose independence between inputs, they fall short in this regard. GNNs, on the other hand, are made to explicitly capture these dependencies. They allow each linked node to be aware of its surroundings by propagating information between connected nodes. Due to this information flow, a rich representation is produced that includes both the specific characteristics of each node and the larger context of their relationships [20], [19]. Additionally, GNNs have the ability to learn relationships between nodes that are immediately adjacent to one another as well as higher-order interactions by propagating information over a number of steps. Because of this, GNNs are able to efficiently capture long-distance dependencies, which frequently prove to be essential for tasks like community discovery [19], [26],which require a grasp of the overall structure of the graph. GNNs can also manage a variety of associations, including those that can be found in heterogeneous graphs, where nodes and edges can have diverse types, because to their adaptable design. When working with complicated real-world data including several entities and relationships, this is especially helpful [27].

2. Exploitation of Graph Structure: A family of machine learning models called GNNs are created expressly to work with data that is graph-structured. GNNs use the innate structure of graphs to extract both local and global information

from the input, in contrast to typical neural networks that analyze grid-like or sequential data. When working with data that is irregularly structured, such as graphs, this strategy that disregards structure is especially helpful [28]. By adding neighborhood information during learning, GNNs take advantage of the graph structure. Every node in the graph has a corresponding feature vector, which is iteratively updated by compiling data from the node's neighbors. GNNs are able to collect the local dependencies and relational data that are present in the graph thanks to this aggregation process. GNNs are able to learn representations that encode the structure and context of the graph by considering the nearby nodes [26]. Additionally, GNNs are able to gather global data by repeatedly iterating the aggregation process. Due to the iterative nature of GNNs, the model can take advantage of the global structure and relationships contained in the data by propagating information across the whole graph. Understanding complicated interactions and patterns within the network requires the capacity to simultaneously capture local and global information [24]. Numerous research have shown how well GNNs can take advantage of graph structure. In order to extract the structural information from the graph, the GraphSAGE model [26] makes use of a neighborhood aggregation technique. By efficiently utilizing the graph structure, the authors demonstrated that Graphs AGE outperforms conventional approaches on tasks like node classification and connection prediction. The Graph Convolutional Network (GCN), which employs a localized first-order approximation of spectral graph convolutions, was proposed by Kipf and Welling (2016) [18] in a different study. The authors showed that GCN effectively uses the graph structure to attain state-of-the-art performance on node classification tasks. In conclusion, GNNs take advantage of the graph structure to gather both local and global information, allowing them to successfully learn from data with irregularly structured data. GNNs can identify dependencies and patterns in the graph by gathering data from nearby nodes and executing iterative updates. Numerous researches have proven that GNNs are effective at utilizing graph structure.

3. Learning Representations of Nodes: One of the most critical advantages of GNNs lies in their capacity to learn intricate representations of nodes, derived not only from their inherent attributes but also their relative positions within the graph structure. This capability proves to be exceptionally crucial in applications where the unique representation of a node dramatically influences the performance of the assigned task. This includes domains such as social network analysis or molecular structure prediction. In the realm of social network analysis, for example, understanding the nuanced role of a node (individual user) and their interconnectedness within the network structure can provide invaluable insights into the dynamics of information propagation, community structure, and user behavior [29]. This is well demonstrated in the paper "Graph Neural Networks for Social Recommendation" [29] where the researchers successfully utilized GNNs to enhance recommendation systems by capturing complex user-item interactions within social networks. In the field of molecular structure prediction, the chemical properties of a molecule can be largely defined by the spatial arrangement of atoms (nodes) and bonds (edges), forming a natural graph structure. Consequently, GNNs are exceptionally suited for such tasks, enabling the prediction of various chemical properties or potential drug efficacy based on the structural graph data of the molecule. A strong example of this is seen in the work by Gilmer et al., 2017, [30] where GNNs were used to predict molecular properties more accurately than traditional machine learning models. In sum, GNNs' capacity to understand and model complex node attributes and their interconnectedness forms a strong basis for their superiority in handling graph data.

4. Invariance to Graph Isomorphism: GNNs exhibit the intriguing property of being invariant to input graph permutations, where changes in the labeling or ordering of nodes do not affect the output of the GNN. This property makes GNNs robust and consistent in their performance, as they capture and comprehend the underlying structural relationships within the graph, rather than being influenced by superficial variations in node representation [31]. The literature offers both theoretical and empirical studies of this invariance to graph isomorphism. For instance, there has been an investigation [32] into the provable power of graph networks showed that some GNN topologies can discriminate between isomorphic and non-isomorphic graphs. In an other investigation [33] of the expressive capabilities of GNNs for graph isomorphism tasks demonstrated that GNNs are capable of distinguishing between pairs of non-isomorphic graphs with high accuracy. In many graph-based tasks, such as graph classification, link prediction, and node representation learning, where understanding the structural aspects of the graph is crucial, the trait of invariance to graph isomorphism is very helpful.

5. Scalability: A key benefit of GNNs is their extraordinary scalability, which makes them a particularly good option for jobs involving vast amounts of graph data. The capacity to function well even on very large graphs is a unique trait that GNNs take advantage of, mostly through processing the graph data locally. GNNs can infer node-level information thanks to the benefit of localized processing, which eliminates the need to load the complete graph into memory [19]. The fundamental structure of GNNs, which encapsulates the message-passing paradigm between nearby nodes, is to blame for this confined processing. For example, Graph Convolutional Networks (GCNs), a subtype of GNN, employ convolutional layers to enable the network to concentrate on the immediate area surrounding each node [18]. The amount of computational resources needed is greatly decreased by this targeted focus, which also enables efficient processing even when only a portion of the graph is ever brought into

memory. This trait is essential for improving memory efficiency, which is important when working with massive graph data [34]. In addition, methods like mini-batch learning and graph sampling are employed to promote effective and efficient learning on big graphs [26]. These techniques make sure that the learning process only uses a portion of the nodes and edges at each stage, avoiding memory saturation that can happen if the entire graph data is used. The development of GraphSAGE, a highly scalable inductive framework presented in [35], demonstrates an effective way to create node embeddings for extremely large graphs. This work also exemplifies the scalability of Graph Neural Networks.

2.2.4 Research Challenges and Open Questions

Although GNNs have shown significant promise in many areas, including recommender systems, there are still many research obstacles and unanswered problems. Some of these issues are covered in this part, along with some of the crucial areas that demand more study.

Scalability and Efficiency: Despite the inherent scalability of GNNs, as the size and complexity of the graph increase, computational requirements can become prohibitively high [34]. Methods like GraphSAGE [26] have been proposed to make GNNs more scalable, but this is still an area where additional research is needed. Similarly, the efficiency of GNNs on dynamic graphs, where the graph structure changes over time, remains a challenging issue.

Graph Construction: While GNNs can manage data in the form of graphs intuitively, the process of constructing these graphs from raw data is non-trivial and can have a significant impact on the performance of the model [19]. Defining what constitutes a node, an edge, or even the weights of these edges require careful consideration and depends largely on the specific use case.

Interpretability: Like many other deep learning models, GNNs suffer from a lack of interpretability. The decision-making process in these models can often be opaque, which may not be desirable in applications where interpretability is important, such as healthcare or finance [36].

Adversarial Attacks: With the growing use of GNNs, their robustness against adversarial attacks is another critical issue. Recent research has shown that GNNs are vulnerable to attacks where the adversary perturbs the graph structure or node attributes to mislead the learning process [37].

Transfer Learning: GNNs often struggle with transfer learning, where the model is trained on one graph and then applied to another. Since the structures and properties of the graphs can vary significantly, transfer learning becomes challenging for GNNs [38].

Incorporating Node and Graph Features: How to best incorporate node features and graph-level features into GNNs, especially when these features have diverse types and scales, is a significant research question [39].

In summary, while GNNs have demonstrated promising results in numerous applications, addressing these challenges can unlock further potential and facilitate the deployment of GNNs in more diverse and complex real-world scenarios.

2.3 GNN-based Recommender Systems

The evolution of recommender systems has gone through three major phases [40]. Starting with shallow models and moving through neural models to models based on GNNs, these stages, which show the evolution of technology in the field, are sequential. In terms of sophistication, capacity, and performance, each consecutive step of the development of these systems represents a considerable improvement over the one before it. The most recent and complex phase in this evolutionary trajectory, GNN-based models, are the focus of this work. With their capacity for deeper learning and better data interpretation, these models represent the cutting edge of recommender system technology and have the potential to completely change how recommendations are created.

2.3.1 Overview of GNN-based Recommender Systems

In GNN-based recommender systems, which combine the benefits of both paradigms, recommender systems and graph neural networks work together. These systems often use Graph Neural Networks to process the graphed interactions and relationships in the data before attempting to generate recommendations. Utilizing the rich relational information in graph data, such as interactions between people and objects, is the notion behind this technique, which may be effective for improving the quality of suggestions [21].

Because of the structure of the GNN, one distinguishing feature of GNN-based recommender systems is their capacity for high-order connectivity-based reasoning. It means these systems can model complex relationships in the data beyond simple user-item interaction, which can include transitive relationships or user-user, item-item connections [41]. Think about a situation where User A and User B have similar shopping habits and User B and User C like the same movies. Even though there is no direct connection between Users A and C in this scenario, GNN-based systems may nonetheless suggest movies to User A that User C like. It shows how GNNs may extract and infer high-order relationships from a graph.

A GNN-based recommender system can be designed in a variety of ways, depending on the problem at hand, the data characteristics, and the requirements. Recommender systems based on GCN, GAT, R-GCN, and GraphSAGE are examples of common techniques [35]. While these approaches have a common GNN core, the way that node feature propagation and aggregation are done varies, highlighting the adaptability of GNN-based systems.

2.3.2 How GNNs Enhance Recommender Systems

GNNs effectively utilize the topological structure and characteristics of graph-based data representations to significantly improve the capabilities of recommender systems. More particularly, GNNs improve recommender systems in the ways listed below:

Improved Representation Learning: By successfully capturing and utilizing the relational information in the data, GNNs make it possible to learn representations more accurately. By treating user-item interactions as a graph, where users and things are the nodes, and interactions are the edges, they can accommodate the intrinsic structure of the user-item interactions. Simple vector-based representations could miss complicated patterns and dependencies between users and items, but this graph-based method makes them possible to find [41].

High-order Connectivity-Based Reasoning: Traditional recommender systems sometimes prioritize direct connections while underutilizing indirect or higher-order interactions that may be valuable. Contrarily, GNNs may detect high order connectivities in the graph data, enabling the model to infer more intricate relationships. Better personalisation results from doing this, which can also greatly raise the standard of recommendations [21].

Better Handling of Sparse Data: Dealing with sparsity in user-item interactions is a challenge for recommender systems. By propagating and aggregating information across the nodes in the network, GNNs can solve this problem by improving each node's representation by, in essence, borrowing information from nearby nodes. When new users or items have few interactions during a cold start, this GNN functionality can be quite helpful [35].

Contextualized Recommendations: By successfully incorporating new data into the model, GNNs can deliver recommendations that are more individualized and context sensitive. A network structure's relational dependencies between various entities (users, objects, and additional context) might provide this information [42]. Learning Long-range Dependencies: By executing graph convolutions that take neighborhood information into account, GNNs, in contrast to conventional techniques, may understand long-range dependencies in data and produce more nuanced suggestions [43].

Handling Cold-start Problem: The cold-start problem, which occurs when new users or items lack adequate interaction data, is a major obstacle in many recommender systems. By utilizing the connections and properties of nearby nodes and drawing useful conclusions about these novel entities, GNNs assist in resolving this problem [42].

Sequential Recommendation: In tasks where the sequence of user interactions matters (e.g., a series of clicks on a website), GNNs can be particularly powerful. Works such as [44] demonstrate how the use of GNNs can model dynamic user-item interactions over time and offer sequential recommendations effectively.

2.3.3 Common GNN Architectures Used in Recommender Systems

Because they can take advantage of the relational data in graphs, different Graph Neural Network (GNN) architectures have been widely used in recommender systems. Several popular GNN architectures in recommender systems include the ones listed below:

Graph Convolutional Networks (GCNs) for Recommender Systems: The Kipf and Welling (2016) [18] presented GCNs, which offer a potent method of extracting and utilizing the graph structure, have been used in a number of recommender systems. They used a semi-supervised classification environment to show this application, which was later modified for recommender systems. To learn user and item embeddings from the user-item interaction graph, for instance, the Graph Convolutional Matrix Completion (GC-MC) technique [42] uses GCNs. These embeddings are then applied to recommendation.

Graph Attention Networks (GATs) for Recommender Systems: GATs [45] add an attention mechanism to the GCN model, enabling it to assign different importance to different nodes in a neighborhood. This allows GATs to model more nuanced dependencies between nodes and has led to its application in various recommender systems.

GraphSAGE for Recommender Systems: GraphSAGE, proposed by [26], is another significant GNN model used in recommender systems. It generates node embeddings by aggregating and transforming feature information from the node's local neighborhood. Its ability to generate embeddings for unseen data makes it suitable for dynamic scenarios, a common characteristic of recommender systems.

Heterogeneous Graph Neural Networks for Recommender Systems: The interaction network is diverse in many real-world recommendation contexts, consisting of several kinds of nodes and edges. Such heterogeneity can be handled by Heterogeneous Graph Neural Networks (HGNNs). For instance, Heterogeneous Graph Attention Network (HAN) [41] has been used in recommender systems to simulate the various connections in a graph using an attention mechanism.

2.3.4 Current Research and Future Directions

GNNs used in recommender systems are the subject of extensive and quickly developing current research. There have been a number of notable developments, and there are many exciting new research avenues to investigate.

One such avenue is the use of heterogeneous graphs in recommender systems. Heterogeneous graphs can better capture complex relationships between different types of entities, potentially improving recommendation performance. The Graph Heterogeneous Multi-Relational Recommendation [1] presents an innovative method to handle such complexities.

Another focus of current research is on improving the interpretability of GNN-based recommender systems. While GNNs can provide excellent performance, it is often difficult to understand why a particular recommendation was made. Strategies like providing explanation interfaces and integrating attention mechanisms into GNN architectures are being explored to make these models more transparent. For example, the study "Explainability in Graph Neural Networks: A Taxonomic Survey" [46] provides an in-depth examination of different strategies to increase the interpretability of GNNs. Furthermore, researchers are looking into techniques for dealing with the dynamic nature of user-item interaction graphs. Many real-world recommendation scenarios are dynamic, where new users and items continually enter the system, and the preferences of existing users evolve over time. A paper titled "Dynamic Graph Collaborative Filtering" [47] presents an approach that effectively captures these temporal dynamics in user-item interactions. There's also a growing interest in leveraging reinforcement learning with GNNs for recommendation. This approach considers the sequential interactions of users with items, which can lead to a more effective recommendation policy over time. An example is the work "Graph Convolutional Reinforcement Learning" [48], which proposed a novel method for integrating reinforcement learning with graph convolutional networks.

Regarding potential future developments, one exciting prospect is the combination of GNNs with other cutting-edge machine learning methods, including deep generative models, to improve the caliber of suggestions. Given that training large-scale GNNs is computationally intensive and can be a bottleneck for their implementation in big recommender systems, developing more effective training algorithms for GNNs is another potential field. Additionally gaining popularity is the idea of multi-modal GNN-based recommender systems. Such systems might add more user or item details to the graph, such text or image descriptions, which might result in more precise and detailed recommendations.

In conclusion, the field of GNN-based recommender systems is exciting and brimming with potential. While significant advancements have been made, many promising research directions remain to be explored.

3 Comparative Analysis

The contributions and advancements of many studies that suggest recommender systems based on GNNs are compared in this section, along with their advantages over earlier approaches. Each research's essential ideas, approaches, metrics, experimental layout, and data sets are condensed. To assess the performance improvements brought about by GNN-based approaches and identify any limitations, a complete study of experimental results is done. The analysis highlights the value of GNNs in improving recommender systems while outlining the benefits and drawbacks of each strategy. It also offers useful data for further study.

3.1 ConsisRec in Social Recommendation

The authors of the study "ConsisRec: Enhancing GNN for Social Recommendation via Consistent Neighbor Aggregation" [49] put up a brand-new strategy dubbed ConsisRec. This method, which is based on GNNs, was created expressly to deal with the social inconsistency issue that social recommendation systems encounter. The issue of social inconsistency in social recommendation systems is addressed. ConsisRec uses a query layer, relation attention, and neighbor sampling as its main approaches. On the basis of the consistency scores between neighbors and a query embedding, neighbor sampling is used to choose consistent neighbors. As a result, the framework can concentrate on neighbors who offer useful information for recommendations. When assigning important elements to sampled neighbors, relation attention takes into account the connected relations. The query layer generates query embeddings to select consistent neighbors based on the item being recommended.

The Ciao and Epinions real-world datasets were used by the authors for evaluation. These datasets include user-item interactions as well as user social connections. Testing, validation, and training sets were created from the datasets. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) evaluation measures were employed to assess the accuracy of rating prediction. The experimental design comprised early halting to avoid overfitting and hyper-parameter adjustment via grid search. Several conventional recommender systems were compared, including GNN-based methods, non-GNN graph embedding methods, and matrix factorization methods.

On both the Ciao and Epinions datasets, the experimental results demonstrated that ConsisRec outperformed the baseline methods in terms of RMSE and MAE. As seen in Table 1, ConsisRec made considerable advancements, with an average relative improvement of 1.7% when compared to the second-best approach. This indicates how well the GNN-based strategy works to handle the issue of social inconsistency in social recommendation.

Although the results were encouraging, the experiments also revealed some drawbacks and trade-offs. According to the report, the neighbor sampling module was essential and that its removal severely decreased performance. This emphasizes how crucial it is to choose consistent neighbors for aggregation. The research also explored the sensitivity of some hyper-parameters, including learning rate, embedding size, and neighbor percent. To attain the best performance, these hyper-parameters need to be carefully tuned.

The proposed ConsisRec approach successfully addresses the social inconsistency issue in social recommendation systems, in order to sum up. The tests showed that in terms of rating prediction accuracy, ConsisRec outperformed con-

Table 1: Overall comparison. The best and the second-best results are in bold and underlined, respectively [49].

Method	Ciao		Epinions	
	RMSE	MAE	RMSE	MAE
SoRec	1.2024	0.8693	1.3389	1.0618
SoReg	1.0066	0.7595	1.0751	0.8309
SocialMF	1.0013	0.7535	1.0706	0.8264
GCMC+SN	1.0301	0.7970	1.1070	0.8480
GraphRec	1.0040	0.7591	1.0799	0.8219
CUNE	1.0002	0.7591	1.0681	0.8284
ConsisRec	**0.9722**	**0.7394**	**1.0495**	**0.8046**
Improvement	2.79%	1.87%	1.74%	2.1%

ventional recommender systems. The GNN-based method offered appreciable performance gains because to its main techniques of neighbor sampling, relation attention, and query layer. The experiments, however, also emphasized the value of hyper-parameter adjustment and the limitations of the strategy. Neighbor selection should be improved in future study, along with other inconsistency issues in related graph-based research topics.

3.2 DMGCF

The authors of paper "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems" [50] introduce a unique framework known as DMGCF (Dynamic Multi-Graph Collaborative Filtering). This method uses numerous graphs and a dynamic evolution mechanism to bring a novel approach to recommendation systems. The model enhances the performance and accuracy of recommendation algorithms by combining the capabilities of GNNs with collaborative filtering approaches. In order to overcome the drawbacks of conventional recommender systems, DMGCF mines relations and creates several graphs that simulate side information. It combines dynamically constructed and updated user and item graphs that are updated at various points during the training process. The model also introduces a dual-path GNN architecture that effectively captures low-dimensional and high-dimensional features. By utilizing these key techniques, DMGCF aims to enhance recommendation accuracy by exploiting graph-based information and dynamic evolution.

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE), two common evaluation metrics for recommendation systems, are used to evaluate the effectiveness of DMGCF in the experimental assessment. Comparisons are made between the model's performance and that of conventional recommender systems, such as SVD (Singular Value Decomposition), and deep learning techniques, such as NCF (Neural Collaborative Filtering) and DeepFM (Deep Factorization Machines). The experiments are run on a variety of datasets, including Yelp, ML-100K, ML-1M, YahooMusic, Flixster, and ML-100K. These datasets exhibit diverse degrees of sparsity and comprise a range of rating scales, including 5-point and 100-point scales.

The experimental findings show that DMGCF outperforms conventional recommender systems in terms of performance. In terms of RMSE and MAE, the conventional matrix factorization approach, SVD, performs poorly. NCF and DeepFM, two deep learning techniques, outperform SVD, demonstrating the benefits of deep learning for recommendation tasks. But deep learning techniques that use graphs, such as sRMGCNN, NGCF, and DMGCF, consistently beat those that don't. This comparison highlights how crucial it is for recommendation systems to incorporate graphs and graph convolution networks. Notably, as shown in Table 2, DMGCF regularly outperforms NGCF, which is regarded as the most advanced graph-based collaborative filtering method, to produce the best results across all datasets.

Across the tested datasets, DMGCF shows considerable performance gains above NGCF, the second-best approach. The reductions in RMSE, which range from 0.74% to 4.37%, illustrate how successful the suggested GNN-based method is. The results show that DMGCF works better than other approaches, notably on datasets like YahooMusic and Flixster, where the gains are particularly notable. The outcomes demonstrate the benefits of using multiple graphs, dynamic evolution processes, and dual-path GNNs to capture both low-order and high-order feature interactions. The DMGCF performance improvements highlight the development and potential of the GNN-based recommendation system approach.

Although the experimental findings show that DMGCF is beneficial, there are several restrictions and trade-offs to take into account. According to the article, performance gains vary depending on the size of the dataset, with smaller datasets exhibiting more steady advances than bigger ones. This shows that the dataset's properties and amount of sparsity may have a greater impact on the model's performance. The impact of other hyperparameters, including the embedding size, the number of graph linkages, and the dynamic evolution mechanism's start time and update frequency, is also covered in the study. The study offers insights into the impact of these hyperparameters, which need to be carefully tuned to produce the best outcomes. A deeper investigation and analysis of the model's sensitivity to these hyperparameters, however, might lead to a better understanding of the system overall. Additionally, the experiments focus on offline evaluation metrics, and the paper suggests that online evaluations, such as A/B testing, could be conducted in future work to validate the model's performance in real-world recommendation scenarios.

DMGCF, which makes use of GNNs, many graphs, and

Table 2: Performance comparison [50]

	Flixster		YahooMusic		ML-100K		ML-1M		Yelp	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
SVD [14]	0.9476	0.7486	54.71	47.69	0.9856	0.7922	0.9850	0.8014	1.1250	0.9064
NCF [22]	0.9150	0.7076	26.44	21.17	0.9390	0.7415	0.8692	0.6820	1.0889	0.8630
DeepFM [60]	0.9109	0.7001	25.47	20.33	0.9489	0.7531	0.8834	0.6943	1.0926	0.8610
sRMGCNN [54]	0.9261	0.7139	22.31	18.76	0.9292	0.7378	0.8776	0.6920	1.0881	0.8719
NGCF [10]	0.9098	0.6888	21.49	17.41	0.9319	0.7321	0.8670	0.6802	1.0931	0.8567
DMGCF(ours)	**0.8928**	**0.6691**	**20.55**	**16.10**	**0.9234**	**0.7254**	**0.8586**	**0.6724**	**1.0801**	**0.8519**
Improv.	1.87%	2.86%	4.37%	7.52%	0.62%	0.92%	0.97%	1.15%	0.74%	0.56%
p-value	8.22E-06	2.84E-06	2.10E-06	2.24E-06	1.05E-03	3.30E-04	2.81E-05	5.00E-05	5.41E-03	1.18E-01

dynamic evolution mechanisms, offers a viable method for recommendation systems. The experimental findings show how it outperforms more established recommender systems and provide considerable performance gains. There are still opportunities for further investigation and improvement, such as honing hyperparameters and performing online reviews. However, DMGCF's improvements in performance and accuracy contribute to the changing environment of recommendation systems by offering insightful information and methods for graph-based collaborative filtering.

3.3 KGNN-LS

The paper outlined in [41] proposes a method known as Knowledge-Aware Graph Neural Network with Label Smoothness regularization (KGNN-LS) for recommendation systems. By employing Graph Neural Networks (GNNs), the method harnesses the potential of knowledge graphs (KGs) to enhance the performance of recommender systems. User-specific relation scoring functions, neighborhood information aggregation with various weights, and the addition of label smoothness regularization to learn the edge weights in KGs are the main strategies applied in this methodology.

The authors do trials on numerous datasets and use a variety of evaluation measures to gauge how effective the suggested technique is. Recall@K for top-K recommendation and AUC for click-through rate (CTR) prediction are two of the evaluation metrics that are employed. In the experimental setting, the performance of the suggested technique, KGNN-LS, is compared to that of a number of well-known recommender systems, including SVD, LibFM, PER, CKE, RippleNet, and KGNN-LS. MovieLens-20M, Book-Crossing, Last.FM, and Dianping-Food are some of the datasets that were used for evaluation and each one represents a different domain.

In diverse evaluation circumstances, the comparative analysis of the experimental findings shows that KGNN-LS performs significantly better than the conventional recommender systems. The findings demonstrate that KGNN-LS outperforms the baseline models in terms of recall rates, AUC scores, and recommendation performance. These results show the advantages of utilizing knowledge graphs for better suggestions and validate the efficacy of KGs and GNNs in recommendation systems.

In comparison to conventional recommender systems, the GNN-based technique, KGNN-LS, produces considerable performance gains in terms of suggestion accuracy and CTR prediction. According to the experimental findings, KGNN-LS performs better on various datasets and recommendation tasks in terms of recall rates and AUC scores. The benefits are especially noteworthy in cases involving cold starts with few user-item interactions. This shows that the addition of KGs and GNNs helps resolve the sparsity issue and improves the recommendation system's overall predictive performance.

Despite the positive outcomes, the experiments also point up some drawbacks and trade-offs. The importance of thoughtfully selecting the depth of the GNN architecture is highlighted by the paper's acknowledgement that using more GNN layers may have a negative impact on performance. In order to balance model capacity and overfitting, the dimension of the hidden layers in the GNNs should also be carefully chosen. Even though KGNN-LS is scalable in terms of KG size, other variables like the size of the mini-batch and the number of epochs might affect how long it takes to run. These restrictions and compromises point out places for additional analysis and study optimization in the future.

The paper concludes by presenting a knowledge-aware graph neural network for recommendation systems that has label smoothness regularization. To improve recommendation accuracy and address the sparsity issue in user-item interactions, the suggested approach makes use of KGs and GNNs. According to experimental findings, KGNN-LS outperforms conventional recommender systems in terms of recall rates,

Table 3: The results of Recall@K in top-K recommendation [41].

Model	MovieLens-20M				Book-Crossing				Last.FM				Dianping-Food			
	R@2	R@10	R@50	R@100	R@2	R@10	R@50	R@100	R@2	R@10	R@50	R@100	R@2	R@10	R@50	R@100
SVD	0.036	0.124	0.277	0.401	0.027	0.046	0.077	0.109	0.029	0.098	0.240	0.332	0.039	0.152	0.329	0.451
LibFM	0.039	0.121	0.271	0.388	0.033	0.062	0.092	0.124	0.030	0.103	0.263	0.330	0.043	0.156	0.332	0.448
LibFM + TransE	0.041	0.125	0.280	0.396	0.037	0.064	0.097	0.130	0.032	0.102	0.259	0.326	0.044	0.161	0.343	0.455
PER	0.022	0.077	0.160	0.243	0.022	0.041	0.064	0.070	0.014	0.052	0.116	0.176	0.023	0.102	0.256	0.354
CKE	0.034	0.107	0.244	0.322	0.028	0.051	0.079	0.112	0.023	0.070	0.180	0.296	0.034	0.138	0.305	0.437
RippleNet	0.045	0.130	0.278	0.447	0.036	0.074	0.107	0.127	0.032	0.101	0.242	0.336	0.040	0.155	0.328	0.440
KGNN-LS	0.043	**0.155**	**0.321**	**0.458**	**0.045**	**0.082**	**0.117**	**0.149**	**0.044**	**0.122**	**0.277**	**0.370**	**0.047**	**0.170**	**0.340**	**0.487**

Table 4: The results of AUC in CTR prediction [41].

Model	Movie	Book	Music	Restaurant
SVD	0.963	0.672	0.769	0.838
LibFM	0.959	0.691	0.778	0.837
LibFM + TransE	0.966	0.698	0.777	0.839
PER	0.832	0.617	0.633	0.746
CKE	0.924	0.677	0.744	0.802
RippleNet	0.960	0.727	0.770	0.833
KGNN-LS	**0.979**	**0.744**	**0.803**	**0.850**

AUC ratings, and suggestion effectiveness. The report also admits some limitations and trade-offs in the trials and emphasizes the need of selecting proper hyperparameters.

4 Discussion, Conclusions and Future Work

The use of GNNs in recommender systems has been examined throughout this paper as a potential means of getting around the drawbacks of conventional models. The capabilities of GNNs have been carefully examined, especially their capacity to model complicated, non-Euclidean data like user-item interactions in graph form. This study demonstrates how GNNs can improve the reliability, scalability, and accuracy of recommender systems.

The GNN-based recommender systems that were investigated also showed a lot of promise for increasing economic value. They may boost user engagement, increase income, and gain more devoted clients by providing more precise and customized recommendations. Numerous models, including ConsisRec, DMGCF, and KGNN-LS, were compared, and the results provided important insights into the performance variations and practical applications of different GNN architectures in the field of recommender systems.

However, despite these encouraging developments, there are still lots of problems and research gaps. One such difficulty is the requirement for increased explainability and transparency in recommender systems based on GNNs. Future studies can concentrate on creating models that offer not just excellent recommendations but also a clear justification for them, enhancing user confidence. Handling dynamic networks, where relationships and nodes might alter over time, is another important subject for future research. Even though considerable work has been made in this area, creating appropriate algorithms for dynamic graph data is still a burgeoning area of study with enormous promise.

In conclusion, this study advances knowledge of GNN-based recommender systems and the implications they have for corporate growth and technological advancement. It offers opportunities for additional research into the development of more intricate, dependable, and user-centric recommender systems, thereby promising to deepen our grasp of this intriguing subject.

5 References

[1] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma, "Graph heterogeneous multi-relational recommendation," *In Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[2] Mehrdad Mollanoroozi, "Review on recommender system and architecture," *Majlesi Journal of Telecommunication Devices*, vol. 11, pp. 177–185, 2022.

[3] Gediminas Adomavicius and Alexander Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, 2005.

[4] Charu C Aggarwal, *Recommender Systems*, 2016.

[5] Arjan Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald Lagendijk, and Qiang Tang, "Privacy in recommender systems," *Social media retrieval*, 2013.

[6] Robin Burke, "Knowledge-based recommender systems," *Encyclopedia of library and information systems, 69(Supplement 32)*, 2000.

[7] Jannach and Dietmar, *Recommender Systems: An Introduction*, 2010.

[8] Xavier Amatriain and Justin Basilico, "Netflix recommendations: Beyond the 5 stars (part 1)," 4 2012.

[9] Davidson James, Liebald Benjamin, Liu Junning, Nandy Palash, and Vleet Taylor Van, "The youtube video recommendation system," *In Proceedings of the fourth ACM conference on Recommender systems*, p. 386, 2010.

[10] Carlos A. Gomez-Uribe and Neil Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems*, vol. 6, 12 2015.

[11] Jayasimha Katukuri, Tolga Könik, Rajyashree Mukherjee, and Santanu Kolay, "Recommending similar items in large-scale online marketplaces," *IEEE International Conference on Big Data (Big Data)*, 2014.

[12] Jayasimha Katukur, Tolga Konik, Rajyashree Mukherjee, and Santanu Kolay, "Post-purchase recommendations in large-scale online marketplaces," *IEEE International Conference on Big Data (Big Data)*, 2015.

[13] Dietmar Jannach and Michael Jugovac, "Measuring the business value of recommender systems," *ACM Transactions on Management Information Systems*, vol. 10, 12 2019.

[14] Xuan Nhat Lam, Ho Chi Minh, Thuc Vu, Trong Duc Le, and Anh Duc Duong, "Addressing cold-start problem in recommendation systems," *In Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 2008.

[15] Siavash Ghodsi Moghaddam and Ali Selamat, "A scalable collaborative recommender algorithm based on user density-based clustering," *In The 3rd International Conference on Data Mining and Intelligent Information Technology Applications*, 2011.

[16] Himan Abdollahpouri, Masoud Mansoury, Robin Burke, Bamshad Mobasher, and Edward Malthouse, "User-centered evaluation of popularity bias in recommender systems," *UMAP 2021 - Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 119–129, 6 2021.

[17] Atika Gupta, Priya Matta, and Bhasker Pant, "Graph neural network: Current state of art, challenges and applications," 2021, vol. 46, pp. 10927–10932, Elsevier Ltd.

[18] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint*, 9 2016.

[19] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 4–24, 1 2021.

[20] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint*, 6 2018.

[21] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 1 2020.

[22] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 5 2015.

[23] Michael M. Bronstein, Joan Bruna, Yann Lecun, Arthur Szlam, and Pierre Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, pp. 18–42, 7 2017.

[24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, pp. 61–80, 1 2009.

[25] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, "How powerful are graph neural networks?," *arXiv preprint*, 2018.

[26] William L Hamilton, Rex Ying, and Jure Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, 2017.

[27] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun, "Heterogeneous graph transformer," 4 2020, pp. 2704–2710, Association for Computing Machinery, Inc.

[28] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, 12 2019.

[29] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin, "Graph neural networks for social recommendation," *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pp. 417–426, 5 2019.

[30] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl, "Neural message passing for quantum chemistry," *In International conference on machine learning*, 2017.

[31] Hoang NT and Takanori Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv preprint*, 5 2019.

[32] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman, "Provably powerful graph networks," *Provably powerful graph networks. Advances in neural information processing systems*, 2019.

[33] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," *In Proceedings of the AAAI conference on artificial intelligence*, p. 19, 2019.

[34] Jie Chen, Tengfei Ma, and Cao Xiao, "Fastgcn: Fast learning with graph convolutional networks via importance sampling," *arXiv preprint*, 1 2018.

[35] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec, "Graph convolutional neural networks for web-scale recommender systems," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 974–983, 7 2018.

[36] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, and Uc San Diego, "Aggregated residual transformations for deep neural networks," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[37] Daniel Zügner, Oliver Borchert, Amir Akbarnejad, and Stephan Günnemann, "Adversarial attacks on graph neural networks: Perturbations and their paterns," *ACM Transactions on Knowledge Discovery from Data*, vol. 14, 8 2020.

[38] Qi Liu, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L Gaunt, "Constrained graph variational autoencoders for molecule design," *Advances in neural information processing systems*, 2018.

[39] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint*, 5 2019.

[40] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, and Yong Li, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Transactions on Recommender Systems*, vol. 1, pp. 1–51, 3 2023.

[41] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat Seng Chua, "Neural graph collaborative filtering," 7 2019, pp. 165–174, Association for Computing Machinery, Inc.

[42] Rianne van den Berg, Thomas N. Kipf, and Max Welling, "Graph convolutional matrix completion," *arXiv preprint*, 6 2017.

[43] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodoì, Jan Svoboda, and Michael M Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[44] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan, "Session-based recommendation with graph neural networks," *In Proceedings of the AAAI conference on artificial intelligence*, p. 19, 2019.

[45] Petar Veličkovi, Guillem Cucurull, Arantxa Casanova, Adriana Romero, and Yoshua Bengio, "Graph attention networks," *arXiv preprint*, 2017.

[46] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji, "Explainability in graph neural networks: A taxonomic survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5 2022.

[47] Xiaohan Li, Mengqi Zhang, Shu Wu, Zheng Liu, Liang Wang, and Philip S. Yu, "Dynamic graph collaborative filtering," 11 2020, vol. 2020-November, pp. 322–331, Institute of Electrical and Electronics Engineers Inc.

[48] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu, "Graph convolutional reinforcement learning," *arXiv preprint*, 10 2018.

[49] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S. Yu, "Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation," 7 2021, pp. 2141–2145, Association for Computing Machinery, Inc.

[50] Hao Tang, Guoshuai Zhao, Xuxiao Bu, and Xueming Qian, "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems," *Knowledge-Based Systems*, vol. 228, 9 2021.