electronics

Article

A Time-Sensitive Graph Neural Network for Session-Based New Item Recommendation

Luzhi Wang * and Di Jin

Collage of Intelligence and Computing, Tianjin University, Tiainjin 300350, China; jindi@tju.edu.cn
* Correspondence: wangluzhi@tju.edu.cn

Abstract: Session-based recommendation plays an important role in daily life and exists in many scenarios, such as online shopping websites and streaming media platforms. Recently, some works have focused on using graph neural networks (GNNs) to recommend new items in session-based scenarios. However, these methods have encountered several limitations. First, existing methods typically ignore the impact of items' visited time in constructing session graphs, resulting in a departure from real-world recommendation dynamics. Second, sessions are often sparse, making it challenging for GNNs to learn valuable item embedding and user preferences. Third, the existing methods usually overemphasize the impact of the last item on user preferences, neglecting their interest in multiple items in a session. To address these issues, we introduce a time-sensitive graph neural network for new item recommendation in session-based scenarios, namely, TSGNN. Specifically, TSGNN provides a novel time-sensitive session graph constructing technique to solve the first problem. For the second problem, TSGNN introduces graph augmentation and contrastive learning into it. To solve the third problem, TSGNN designs a time-aware attention mechanism to accurately discern user preferences. By evaluating the compatibility between user preferences and candidate new item embeddings, our method recommends items with high relevance scores for users. Comparative experiments demonstrate the superiority of TSGNN over state-of-the-art (SOTA) methods.

Keywords: graph neural networks; new item recommendation; session-based recommendation

Citation: Wang, L.; Jin, D. A Time-Sensitive Graph Neural Network for Session-Based New Item Recommendation. *Electronics* **2024**, *13*, 223. https://doi.org/10.3390/electronics13010223

Academic Editor: Shinichi Yamagiwa

Received: 29 November 2023
Revised: 24 December 2023
Accepted: 29 December 2023
Published: 3 January 2024

1. Introduction

Recommender systems significantly impact people's daily lives; they filter information for people and recommend things that people may be interested in [1]. Recommendation systems widely exist in different fields, such as e-commerce [2], job markets [3], and streaming media [4]. A key scenario in recommendation systems is session-based recommendations. A session refers to a sequence of actions carried out by a user, such as clicking buttons, viewing pages, or making purchases [5]. These actions occur within a short time and are considered a single visit [6]. This type of recommender system can provide personalized suggestions to users within a specific session, typically without requiring long-term user profiles or histories [7]. These systems are widely used in content streaming [8], news websites [9], and other online platforms [10].

Numerous methods have been developed to design session-based recommender systems. Attention-based methods are crucial in the field of deep learning [11]. It is also crucial in session-based recommendations. For instance, in the work of Zhang et al. [12], a vanilla attention mechanism is utilized to combine heterogeneous features for extracting the transition patterns between items. Yuan et al. [13] provide a dual sparse attention network to mitigate the effect of irrelevant items on user preference. Additionally, topic models have emerged as a recent area of focus. Sottocornola et al. [14] train topic models to track changes in users' interest in news over time; ref. [15] is a music recommender system where the music consists of a session. This work regards the music list as a set of topics, helping to uncover latent user patterns. In addition to the above two types of methods, there are many other methods, such as collaborative filtering-based methods [16], content-based methods [17], and matrix factorization-based methods [8].

Recently, graph neural networks (GNNs) have been the most popular methods [18–20]. Many GNN methods for session-based recommender systems have emerged, such as TMI-GNN [21], RN-GNN [22], and SR-GNN [23]. Compared to other recommender systems, GNN-based recommender systems can learn intricate item conversions and effectively model user behavior within a specific session graph, which is ignored by other methods. Specifically, GNN-based recommender systems model the input session as a graph, in which nodes refer to various items within the session, and edges refer to transitions or relations between items. GNNs are capable of effective feature learning for the items in the session graph. By fully considering and modeling complex inter-relationships between items, these GNN-based methods can understand user behaviors and preferences during recommendations.

However, the methods mentioned above are designed for next item recommendation and might not be effective in recommending new items. This ineffectiveness arises from the lack of user interaction with new items, leading to new items independent of the session graph that are hard to learn by GNNs. To tackle this issue, a GNN for the session-based new item recommendation method (NirGNN) has been proposed recently [24]. NirGNN incorporates a dual-intent network that mimics user decision-making processes through a soft-attention mechanism and a β distribution mechanism. To address the challenge of new items that are hard to learn by using GNNs, NirGNN leverages inspiration from zero-shot learning and infers the embedding of new items based on their attributes. Consequently, NirGNN calculates recommendation scores and prioritizes new items with higher scores for user recommendations. Nonetheless, this method may encounter three limitations:

- Lack of Time Sensitivity: The previous research models session graphs simply as directed graphs, failing to consider the temporal aspect of item interactions. This omission is a critical flaw as the timing of an item's appearance in a session can greatly influence its relevance and importance. Without considering this temporal dimension, the session graph falls short of mirroring the dynamic nature of actual user interactions and preferences.
- Sparsity of Sessions: This scarcity of data within individual sessions poses a significant challenge for accurately learning user intent in graph neural networks. The sparsity sessions can lead to incomplete or biased interpretations of user preferences as the graph neural network has limited interaction data to analyze and learn from.
- Flawed Attention Mechanism: Previous works usually use attention mechanisms to learn users' preferences. However, the existing attention mechanism disproportionately increases the preference weight of the last item visited by the user. This approach can lead to a skewed understanding of user preferences as it assumes the most recent interaction is the most significant for users. Such a mechanism neglects the possibility that earlier items in the session might hold equal or greater relevance to the user's preferences. Consequently, this results in recommendations that do not accurately reflect the user's overall preferences, focusing narrowly on their most recent activity.

As a remedy for the shortcomings, we introduce a time-sensitive method for a session-based new item recommendation, called the **T**ime-**S**ensitive **G**raph **N**eural **N**etwork (TSGNN). (1) To address the limitation on time sensitivity, we propose an innovative modeling technique for constructing session graphs that incorporates a temporal element. Specifically, we model the session graph according to the sequence of node visits. To fully express the temporal relationship, we apply a time-sensitive weight to each edge. Consequently, GNN-based models are enabled to learn the temporal information of nodes through these edge weights. With this graph modeling technique, the model considers the temporal sequence of item interactions, ensuring that the appearance of each item is taken into account, and reflects user interactions and preferences over time. (2) To tackle the challenge of sparse sessions, we incorporate a graph augmentation technique into the session graph. This technique involves altering the original graphs, allowing the TSGNN to generate a multitude of augmented graphs. This improvement significantly enriches the session data by providing a set of informative graphs for the GNN encoder. This strategy effectively mitigates the impact of graph sparsity, enhancing the overall performance of the model. (3) To provide a comprehensive attention mechanism, we propose a new attention mechanism, called the time-aware attention network. The time-aware attention mechanism emphasizes the temporal aspects' influence on the user's preference learning process. By incorporating this approach, the time-aware attention mechanism mitigates the excessive emphasis often placed on the most recently visited item. Instead, this attention network amplifies the temporal impact on attention allocation, ensuring a more accurate and nuanced interpretation of user preferences.

By focusing on the temporal aspect, TSGNN aims to capture more accurate and comprehensive user preferences than previous works, leading to more relevant recommendations. Our work's enhancements in session-based new items recommendation can be encapsulated in the following key contributions:

- We highlight and address the problem of previous research that ignores the aspect of time influence in session graph modeling. This inclusion of time sensitivity ensures a dynamic representation of user interactions, aligning the model closely with the actual user behavior and preference.
- We incorporate graph augmentation technology into the session-based new item recommendation process. This innovation significantly reduces the sparsity of session graphs, leading to a situation in which the session graph is easily learned by GNNs.
- We propose a novel attention mechanism specifically designed for learning user preference with a time-aware perspective. This method adjusts the focus of the attention mechanism, ensuring that it accounts for the temporal aspects of user interactions. By doing so, a more accurate understanding of user preferences over time reduces the overemphasis on the most recent interactions, which has been a drawback of previous models.

The structure of this paper is outlined as follows: Section 2 presents the related work relevant to our study. In Section 3, we detail our method, TSGNN, highlighting several innovative techniques, including the time-sensitive weight method, session graph augmentation, and time-aware attention networks. Section 4 describes a series of experiments conducted to validate the efficacy of our proposed methods. Finally, in Section 5, we conclude this paper and offer directions for future research.

2. Related Works

In this section, we will explore the related works in the field of session-based recommendation systems, with a particular focus on session-based recommendation, GNNs for session-based recommendation, and graph augmentation methods.

2.1. Session-Based Recommendation

Session-based recommender systems are designed to provide personalized suggestions to users during a single session [25]. Unlike traditional recommender systems that rely heavily on long-term user profiles and historical data, the session-based recommender system focuses on the short-term, often within the context of a single interaction or visit [26]. Some works use the Markov Chain for session-based recommendation to predict user preferences and actions within a specific session [27]. According to the memoryless property, Markov Chains predict the future state of a recommender system based on its previous state [28]. Each state in a Markov Chain represents a specific user interaction, such as viewing a product or clicking on a link. The transitions between these states, governed by a set of probabilities, reflect the likelihood of a user moving from one action to another. By analyzing these transitions and states, the Markov-Chain-based recommender system can predict a user's next likely action within a session. User interactions are captured sequentially and temporally using this approach. However, it does face challenges, notably the potential limitation in capturing long-term user preferences since it focuses primarily on short-term session data.

Recently, some researchers have designed recurrent neural network (RNN)-based methods for session-based recommender systems to understand and predict user preferences based on their current sessions [29]. In these systems, RNNs, known for their effectiveness in handling sequential data, are utilized to analyze the sequence of actions a user takes during a session, such as clicking on items, viewing pages, or adding products to a cart [30]. As users interact with various items during a session, the RNN-based model continuously updates its understanding of the user's current interests. This is achieved through the network's ability to maintain a form of memory, which helps in capturing the temporal dynamics of user behavior within a session. The model processes each action in the sequence, updating its internal state to reflect the latest interactions, thus providing real-time, dynamic recommendations that evolve as the session progresses [31].

However, despite their effectiveness, these systems also face challenges, such as handling the complexity of RNN-based models and ensuring they are efficient enough to provide real-time recommendations. Moreover, the system's dependency on session data means that it might not perform as well when dealing with new or infrequent users with limited interaction history.

2.2. Graph Neural Networks for Session-Based Recommendation

Graphs are an important data structure [32–35], and GNNs are a type of method for solving graph problems that has attracted much attention recently [36,37]. Their applications span a wide range of fields, including federated learning [38], information security [39–41], anomaly detection [42–44], and the financial sector [45]. Owing to the capability of GNNs to model complex relationship nodes [46], GNN session-based recommender systems have recently become popular [47]. In these systems, GNNs are used to model session data as a graph structure. In this representation, nodes represent items, and edges signify the interactions between users and these items. This graph-based method captures complex relationships and dependencies between items within a session [46]. The strength of GNNs lies in their ability to propagate and aggregate information across the graph, effectively learning representations of items based on their context within the user's session. For instance, if a user browses through a series of related products, the GNN can understand the underlying pattern or theme of the session, leading to more accurate and relevant recommendations.

GNN-based recommender systems require careful design to efficiently handle the complexity of graph structures. For example, SR-GNN [23] combines global preferences and current session interests using an attention network. MGU-GNN [48] integrates a soft-attention network with a target-based interest-aware network. This model is adept at adjusting to the evolving interests of users while effectively balancing long-term preferences and immediate session-based interests. Its soft-attention network module is specifically designed to harmonize these long-term and current session interests, enhancing the accuracy of predictions regarding the user's next item. BA-GNN [49] presents an application-specific behavioral GNN tailored for session-based recommendation systems. To enhance the quality of these representations, a sparse self-attention module is implemented, designed to filter out noise in behavior sequences. Additionally, a gating mechanism is employed to form comprehensive session representations.

NirGNN [24] proposes a dual-intent network to understand user preference through two key components: analyzing historical data distribution and establishing an attention mechanism. This network aims to mimic the user's decision-making process when interacting with new items. NirGNN infers the new item embeddings within the GNN embedding space by utilizing item attributes. This method allows for more accurate predictions and recommendations involving new items.

2.3. Graph Augmentation and Contrastive Learning

Graph augmentation refers to modifying and enhancing existing graphs to improve their quality in various graph learning methods. This can involve several techniques and methods, depending on the specific goals and the nature of the graph. Node augmentation and edge augmentation are common methods. Node augmentation generally modifies the attributes of nodes. This might involve updating weights, labels, or other properties based on new data [50]. Edge augmentation introduces new edges or reduces edges between existing nodes to create an augmented graph [51]. These methods involve changing connections between nodes, helping to explore alternative network structures.

To optimize augmented graphs, we introduce contrastive learning. Graph contrastive learning is an advanced machine learning technique used to learn informative representations of graph-structured data [52,53]. Graph contrastive learning is part of contrastive learning methods [54], which focus on learning to differentiate between similar and dissimilar pairs of data points. Graph contrastive learning specifically focuses on generating node embeddings that bring similar nodes closer together in the representation space while distancing dissimilar nodes. This objective is achieved through the development of a contrastive loss function, which serves as the target for optimization by the learning algorithm. Creating variations of input graphs through graph augmentation helps to generate positive pairs for training. In this approach, the graph data are used to capture their intrinsic properties and relationships. Graph contrastive learning methods have diverse applications across various domains, such as social network analysis [55], bioinformatics [53], and recommendation systems [56]. In these applications, they enable the effective learning of graph embeddings that reflect the underlying structure and relationships, which can then be used for tasks such as node classification [57], link prediction [58], and graph classification [59].

3. Method

This section introduces the preliminary concepts for this paper; provides an overview of our model, TSGNN; and illustrates each component of TSGNN.

3.1. Preliminary

A user's session consists of items they have visited, e.g., $S = \{v_1, v_2, ..., v_n\}$. Each v_i represents one item, and n represents how many items there are. There is an order for every two items in a session, which represents the user's actions. The session can be constructed as a directed graph $G = (V, X, A)$, where $V \in \mathbb{R}^n$ denotes a set of nodes, and $X \in \mathbb{R}^{n \times d}$ represents the features of nodes with dimension d. A is the adjacent matrix, and $A_{i,j} = (v_i, v_j)$. An edge in a directed graph is the order of items in a session.

The GNN session-based new item recommendation aims to design a GNN-based method for recommending new items to users. Specifically, the GNN-based methods for new item recommendation often learn user preference by using session graphs. As the new items are not directly learnable by GNNs, the GNN session-based methods often introduce external knowledge, such as attribute to create simulated embeddings for these new items.

3.2. Overview

Figure 1 shows the overview of TSGNN. TSGNN obtains input from user sessions. In step (a), a time-sensitive session graph is constructed from the input session. Subsequently, in step (b), the session graph undergoes augmentation through a graph augmentation function, resulting in two augmented graphs. In step (c), the two augmented graphs are fed into two weights-shared GNN encoders, ensuring that all learned embeddings are located within the same space. The encoding process generates representations for both graphs, which are subsequently fed into a time-aware attention mechanism. This network is tailored to comprehend and capture user preferences, taking into account the impact of time on the session graph. In step (d), TSGNN calculates the compatibility scores between user preferences and new items. It then identifies and recommends the most appropriate item for the user based on these scores. A detailed explanation of each component will be presented in the subsequent sections.

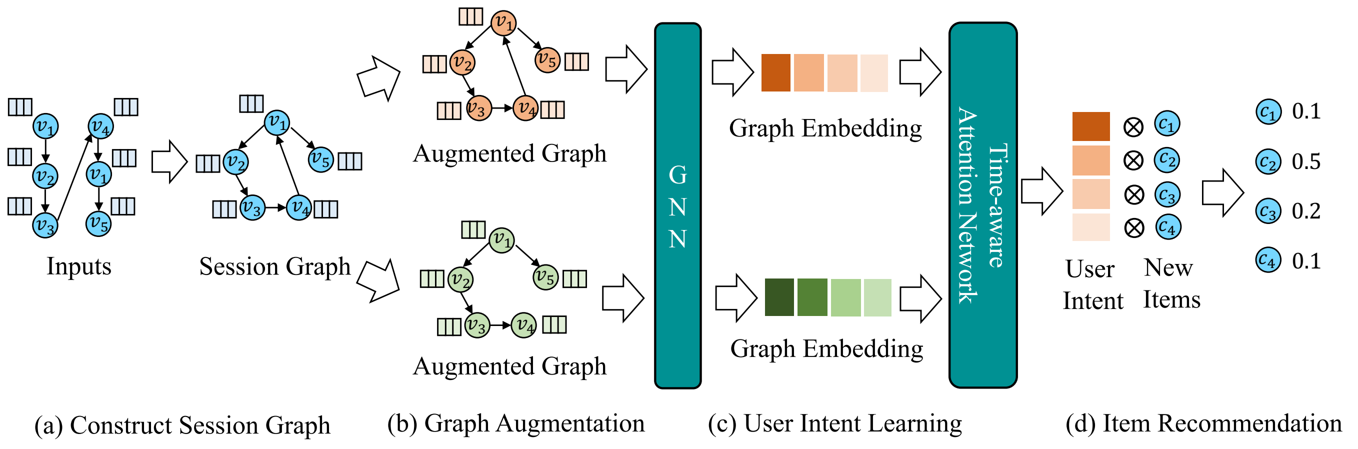


Figure 1. Overview of TSGNN.

3.3. Time-Sensitive Session Graph Construction

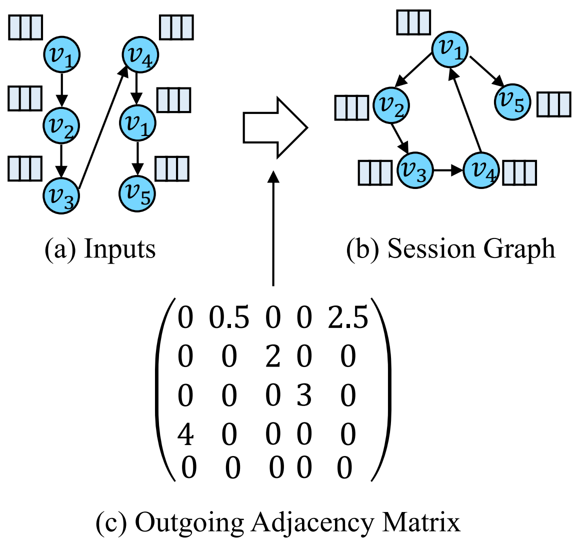
Given the limited consideration of the impact of time on session graph modeling in previous research, we reconsider this aspect in session graph construction. A session is a directed sequence, consisting of various items (nodes) and directed edges between items. The order of directed edges represents the time sequence in which these nodes are visited. The strategy utilized by SR-GNN [23] was followed, which involves transforming a session into a directed session graph. If a node is visited multiple times, we do not create new vertices for each visit. Instead, we directly connect the node, which has been visited multiple times, to its subsequent node (or the preceding node). If a node in the graph is visited multiple times, we do not create new vertices for each visit. Instead, we directly connect the node, which has been visited multiple times, to its subsequent node (or the preceding node). There are two adjacency matrices in a directed session graph: one for an incoming adjacency matrix and one for an outgoing adjacency matrix [23]. Suppose we have an edge A_{ij} with nodes v_i and v_j, where v_i starts the edge and v_j ends it. The weight of each edge can be defined as:

$$weight_{ij} = \frac{D_{in}(v_i)}{D_{out}(v_i)}, \tag{1}$$

$D_{in}(\cdot)$ calculates the incoming degree of node v_i, while $D_{out}(\cdot)$ calculates the outgoing degree of node v_i. Building upon the weighting method of SR-GNN, we incorporate a temporal influence by assigning time weights to each edge and constructing a time-sensitive adjacency matrix. Specifically, edges are sorted by their appearance timestamps, and we assign a rank score $\mathcal{S}(A_{ij})$ as a weight for the edge A_{ij}. Specifically, the $\mathcal{S}(\cdot)$ calculates the order in which the edge A_{ij} appears. For example, the first edge encountered in the session is assigned a weight of 1, the second edge is assigned a weight of 2, and so forth, with the n-th visited edge receiving a weight of n. The item last visited may reflect the user's preference over the early visited items. By assigning weights in this manner, different edges are given different degrees of importance based on their time of visit. Recently visited edges are given a higher weight. Consequently, the rank score $\mathcal{S}(A_{ij})$ effectively captures the temporal influence. The time-sensitive weighting method can be defined as:

$$weight_{ij} = \frac{D_{in}(v_i) * \mathcal{S}(A_{ij})}{D_{out}(v_i)}, \tag{2}$$

where $\mathcal{S}$ is a ranking method. We provide an example of how to calculate an outgoing time-sensitive adjacency matrix. As illustrated in Figure 2a, considering a session with 5 items, $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_1 \rightarrow v_5$, our objective is to construct it as a time-sensitive session graph (b), achieved by calculating outgoing and incoming adjacency matrices (c). For edge A_{12}, the in-degree of v_1 is 1, the out-degree of v_1 is 2, and the rank of A_{12} is 1. Consequently, the weight assigned to this edge is 0.5. The calculation of the incoming adjacent matrix is the same.



$$\begin{pmatrix} 0 & 0.5 & 0 & 0 & 2.5 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 2. In this figure, part (**a**) illustrates an input session. By connecting nodes and assigning weights to each edge, the session graph depicted in part (**b**) can be constructed. Part (**c**) showcases the outgoing adjacency matrices of the session graph.

3.4. Session Graph Augmentation and Embedding

Using the mechanism mentioned above, a time-sensitive session graph can be constructed. To address the issue of the sparsity of session graphs, we adapt two commonly used methods for session graph augmentation, specifically for the time-sensitive graph. Drawing on established methods [51], we implement ‘drop feature’ and ‘drop edge’ techniques. For node features, we employ a random probability, denoted as R_f, to selectively drop features [60], e.g., $X = R_f \times X$. Similarly, we apply this random function R_e to the adjacency matrix, such that $A = R_e \times A$. To learn the embedding h_i for the vertex v_i, we use a GNN as an encoder, such as the gated graph neural network (GGNN) [61]. In the case of a node v_i, the embedding h_i takes the following form [62]:

$$\begin{aligned} h_i^{(1)} &= [x_i^T, 0]^\top \\ a_i^{(t)} &= \left[A_{i:}^{In}(h_1^{(t-1)}, \ldots, h_n^{(t-1)}) \oplus A_{i:}^{Out}(h_1^{(t-1)}, \ldots, h_n^{(t-1)})\right]^\top + b, \\ z_i^{(t)} &= \sigma\left(W_z a_i^{(t)} + U_z h_i^{(t-1)}\right), \\ r_i^{(t)} &= \sigma\left(W_r a_i^{(t)} + U_r h_i^{(t-1)}\right), \\ \widetilde{h_i^{(t)}} &= \tanh\left(W_o a_i^{(t)} + U_o\left(r_i^{(t)} \odot h_i^{(t-1)}\right)\right), \\ h_i^{(t)} &= \left(1 - z_i^{(t)}\right) \odot h_i^{(t-1)} + z_i^{(t)} \odot \widetilde{h_i^{(t)}}, \end{aligned} \tag{3}$$

where $A_{i:}^{In}$ and $A_{i:}^{Out}$ denote the incoming and outgoing adjacency matrices of the i-th row in the matrix, respectively. The variable t symbolizes the training step, $\sigma(\cdot)$ signifies the sigmoid function, and $\odot$ indicates element-wise multiplication. W_z, W_r, W_o and U_z, U_r, U_o

are parameters that can be adjusted during the learning process. A GGNN acquires the embeddings of items within a session graph G by propagating information among neighboring nodes.

3.5. Time-Aware Attention Network

This section introduces a temporal attention network for learning user preferences. It has been found that some previous studies overemphasize users' preferences based on the last item they visited [23], overlooking the influence of timestamps on user preferences. To address this limitation, we introduce a time-aware attention network to learn user preferences under the time influence. Given that each node's representation includes temporal information, we utilize the cosine similarity between h_i and h_j to assess the time influence of h_j on h_i. By splicing the similarity of the current nodes and other nodes, the preferences of users can be tracked over time. By concatenating these time influences from other nodes, node h_i embodies the information from other items, implicitly reflecting user preferences. The significance of each node is then determined using the attention mechanism. Thus, the overall time-aware attention α is as follows:

$$\alpha = \sigma(W * (h_i \oplus \sum_{j \in V \backslash i} \texttt{cos}(h_i, h_j)h_i)), \tag{4}$$

where W represents the learnable weights of item embedding vectors, and $cos(\cdot)$ is the cosine similarity function. User preference I influenced by time can be defined as:

$$I = \sum_{i}^{n} \alpha * h_i. \tag{5}$$

Time-aware attention networks take into account the temporal order of items, recognizing that the items in which users are viewed are crucial to learning preferences.

3.6. Optimization

After learning the users preference, we employ the following function to compute the recommendation score [24]:

$$\hat{z_i} = \text{softmax}(I^\top * c_i), \tag{6}$$

where c_i is the candidate new item embedding.

To optimize the entire model effectively, we integrate a variety of loss functions for the overall loss. Particularly, since we have added graph augmentation in the model, we employ the InfoNCE loss as a foundational component for model optimization. The InfoNCE loss is formulated as follows [63]:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\cos(h_i, h_i^+))}{\exp(\cos(h_i, h_i^+)) + \sum_{h_j^-} \exp(\cos(h_i, h_j^-))}. \tag{7}$$

Here, $cos(h_i, h_j)$ denotes the similarity between nodes h_i and h_j, h_i^+ is a corresponding node in the augmented graph for h_i, and h_j^- represents non-corresponding nodes. The essence of this loss is to pull corresponding nodes (those that are augmented versions of the same node) closer together and push non-corresponding nodes apart in the embedding space. Following the augmented method described in Section 3.4, we obtain two augmented graphs G_1 and G_2. In G_1, we select node h_i as the anchor node and h_j as the corresponding node in G_2. In both graphs G_1 and G_2, the node h_j is treated as a positive sample of h_i whereas the other nodes (e.g., h_k) are treated as negative samples [60]. Building on this, we can express our contrastive loss function as follows:

$$L_c(h_i, h_j, h_k) = -\log \frac{\exp(\cos(h_i, h_j)/\tau)}{\exp(\cos(h_i, h_j)/\tau) + \sum_{k=1}^{N} \exp(\cos(h_i, h_k)/\tau)}, \tag{8}$$

where $\exp(\cdot)$ represents a exponential function and τ represents a temperature coefficient. This loss optimizes the graph encoder by making the corresponding nodes more similar and non-corresponding nodes more dissimilar.

In addition, we incorporate two essential loss functions in the recommendation system, including a cross-entropy loss L_{ce} to optimize the recommendation process, and a new item learning loss $L_t = \theta(h_i, c_i)$ to learn new items embeddings. $\theta(\cdot)$ is a distance function. As a result, the final loss function can be formulated as follows:

$$L = L_{ce}(z_i, \hat{z_i}) + \gamma L_c(h_i, h_j, h_k) + (1 - \gamma) L_t(h_i, c_i), \tag{9}$$

where $\hat{z}$ represents the ground truth item embedding, and γ is employed to balance the new item embedding learning loss and the contrastive loss for augmented graph learning. Given that L_{ce} is the primary loss for recommendation, we do not assign a trade-off parameter to it.

4. Experiments

In this section, we conduct a comparative analysis of our method, TSGNN, against other SOTA methods and highlight the benefits and enhanced performance of our approach. We execute our method on an NVIDIA GeForce GTX TITAN X 24G GPU. The outcomes of our experiments, along with an ablation study and an analysis of parameter sensitivity, are reported based on trials conducted using general datasets.

4.1. Datasets

We utilize the dataset provided by the NirGNN [24], which includes two datasets: one dataset is Amazon Grocery and Gourmet Food (Amazon G&GF) and the other is Yelpsmall. Both datasets encompass a comprehensive range of information, including items, users, user–item interaction sessions, timestamps, and user reviews. This array of datasets adequately satisfies the requirements for our experimental conditions, offering a rich source of information to analyze user behavior, preferences, and interaction patterns within session-based recommendation frameworks. Table 1 shows the details of datasets. Following the setting of NirGNN, the Amazon G&GF dataset includes a total of 51,958 train sessions and 37,813 test sessions. Meanwhile, the Yelpsmall dataset includes 19,035 train sessions and 2311 test sessions.

Table 1. Description on datasets.

Dataset	Items	Description
Amazon G&GF	18,889	Including user reviews and item-to-item timestamps.
Yelpsmall	14,726	Including user data, interactions, and reviews.

4.2. Baselines

We compare our method with seven state-of-the-art methods, including SR-GNN [23], SR-GNN-ATT, GC-SAN [64], GCE-GNN [65], DHCN [66], COTREC [67], and NirGNN [24]. Table 2 shows the details of baselines. SR-GNN, SR-GNN-ATT, GC-SAN, and GCE-GNN are recognized as classic GNN session-based recommendation systems. Most of them use attention mechanisms that ignore the time influence. By comparing our method with these established approaches, we aim to demonstrate the effectiveness and superiority of the time-aware attention mechanism that we have developed. DHCN and COTREC are self-supervised GNN session-based recommender systems. An analysis contrasting our method with these two highlights the advantages of our self-supervised approach. The NirGNN is the GNN-based recommender system for new item recommendation in session scenarios. It is the first GNN-based approach specifically tailored for session-based new item recommendations, which use dual-intent to learn users' preferences.

Table 2. Description of baselines.

Baselines	**Description**
SR-GNN	The first session-based recommendation method based on GNNs.
SR-GNN-ATT	An attention enhanced version of SR-GNN.
GC-SAN	A multi-layer self-attention based method.
GCE-GNN	A position-aware attention based method.
DHCN	A self-supervised based method with hypergraphs.
COTREC	A co-training self-supervised based method.
NirGNN	A GNN model for session-based new item recommendaiton.

Following the previous works, we use $P@k$ and $MRR@k$ to measure our work's performance. $P@k$ measures the accuracy of the top k items in a list of predictions. It can be defined as [68]:

$$P@k = \frac{p}{k}, \tag{10}$$

where p indicates how many predictions are correct. $MRR@k$ is a metric used to evaluate the average of the reciprocal ranks of the query items in the recommendation list. It can be defined as:

$$MRR@k = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{q_i}, \tag{11}$$

where q_i is the position in which v_i appears in the recommendation list [68]. Following the previous works, we select $k = \{10, 20\}$ as the range in this paper.

4.3. Performance

In Tables 3 and 4, the experimental results are detailed. The results of baselines come from NirGNN [24]. From the tables, we can find that the NirGNN is the SOTA baseline. The experimental results demonstrate significant enhancements accomplished by the TSGNN method when compared to existing state-of-the-art methods. For instance, in the Amazon G&GF dataset, our method shows a 10.29% improvement at P@20 compared with the NirGNN method. At MRR@20 metric, TSGNN achieves a 36.56% enhancement. Additionally, at P@10, TSGNN outperforms a 22.81% increase compared to NirGNN. Most notably, TSGNN demonstrates a remarkable 39.96% enhancement in MRR@10. Furthermore, in the Yelpsmall dataset, TSGNN exhibits an 8.37% improvement on the P@20 metric and a 0.46% increase on MRR@20. These results demonstrate the superiority of TSGNN. We observed anomalies in DHCN's performance in P@10 and MRR@10. This may also be due to the short session of the Yelpsmall data set, which makes it difficult for DHNC to accurately construct the hypergraph, thereby impacting its recommendation invalidly. Although TSGNN performs poorly on the two metrics on Yelpsmall, it performs optimally on other datasets, which proves its effectiveness. In addition, TSGNN failed on the P@10 and MRR@10 metrics. This underperformance may be due to the fact that sessions in Yelpsmall have a lower average session length (4.66) than sessions in Amazon G&GF (8.39). Short sessions may limit the learning ability of graph neural networks, resulting in a decline in various evaluation metrics. Therefore, short sessions may not provide sufficient interaction data for the TSGNN to effectively capture complex user preferences.

Table 3. Experiments on Amazon G&GF dataset.

Methods	P@20	MRR@20	P@10	MRR@10
SR-GNN	1.438 ± 0.134	0.336 ± 0.025	0.086 ± 0.078	0.297 ± 0.021
SR-GNN-ATT	0.678 ± 0.202	0.157 ± 0.038	0.375 ± 0.144	0.136 ± 0.040
GC-SAN	2.028 ± 0.108	0.580 ± 0.093	1.288 ± 0.059	0.529 ± 0.090
GCE-GNN	1.650 ± 0.291	0.478 ± 0.082	1.053 ± 0.200	0.441 ± 0.076
DHCN	0.403 ± 0.002	0.058 ± 0.001	0.206 ± 0.001	0.047 ± 0.001
COTREC	1.681 ± 0.837	0.434 ± 0.229	1.078 ± 0.583	0.393 ± 0.213
NirGNN	2.420 ± 0.039	0.599 ± 0.017	1.578 ± 0.056	0.543 ± 0.018
TSGNN	**2.669 ± 0.013**	**0.818 ± 0.024**	**1.938 ± 0.020**	**0.760 ± 0.031**
Improve (%)	**10.29%**	**36.56%**	**22.81%**	**39.96%**

Table 4. Experiments on Yelpsmall dataset.

Methods	P@20	MRR@20	P@10	MRR@10
SR-GNN	0.764 ± 0.534	0.192 ± 0.118	0.476 ± 0.346	0.180 ± 0.111
SR-GNN-ATT	1.631 ± 0.353	0.410 ± 0.089	0.989 ± 0.183	0.366 ± 0.078
GC-SAN	0.793 ± 0.115	0.174 ± 0.094	0.418 ± 0.144	0.165 ± 0.103
GCE-GNN	1.702 ± 0.462	0.789 ± 0.224	**1.428 ± 0.563**	**0.776 ± 0.229**
DHCN	0.396 ± 0.021	0.026 ± 0.001	0.000 ± 0.000	0.000 ± 0.000
COTREC	0.833 ± 0.417	0.256 ± 0.199	1.021 ± 0.583	0.269 ± 0.222
NirGNN	1.817 ± 0.087	1.092 ± 0.017	0.779 ± 0.043	0.299 ± 0.049
TSGNN	**1.969 ± 0.541**	**1.097 ± 0.413**	0.822 ± 0.043	0.355 ± 0.068
Improve (%)	**8.37%**	**0.46%**	-	-

4.4. Ablation Study

To confirm the effectiveness of each component in our proposed model, we conducted ablation experiments. The outcomes of these experiments are detailed in Tables 5 and 6. TSGNN represents the complete time-sensitive graph neural network. 'TSGNN w/o weight' refers to a TSGNN version that replaces the time-sensitive session graph construction with ordinary modeling methods. 'TSGNN w/o attention' denotes the TSGNN without the time-aware attention mechanism and uses the general soft-attention mechanism. 'TSGNN w/o L_c' indicates the TSGNN without graph augmentation and the optimal loss. The results presented in both tables clearly illustrate that each component plays a vital role in enhancing the overall performance of the TSGNN. In Table 5, TSGNN achieves optimal effectiveness when all components are integrated. The 'TSGNN w/o weight' method results in a general decline in TSGNN's performance, underscoring the significance of the temporal weighting modeling approach for this model. Likewise, graph augmentation and contrastive loss hold substantial significance. However, TSGNN's performance on P@20 is lower than that of TSGNN w/o weight. This indicates that while graph augmentation can enhance the richness of graph data, it might also disrupt inherent structures on session graphs. Such an alteration in the graph's structure could lead to a degradation in the model's ability to accurately capture and represent the underlying patterns. Eliminating the contrastive loss and graph augmentation results in a decline across all metrics, indicating that this loss component effectively enhances the model's learning capabilities. 'TSGNN w/o attention' leads to a significant decrease in its performance, thereby validating the importance of the attention method. In Table 6, although TSGNN does not secure the top position in the P@10 metric, it attains the second-best performance. By comparing with the model without other key components, TSGNN still has advantages, proving that integrating each component into the model is beneficial.

Table 5. An ablation study on Amazon G&GF.

DataSets	P@20	MRR@20	P@10	MRR@10
TSGNN	2.669 ± 0.013	**0.818 ± 0.024**	**1.938 ± 0.020**	**0.760 ± 0.031**
TSGNN w/o weight	2.673 ± 0.063	0.762 ± 0.028	1.829 ± 0.101	0.679 ± 0.097
TSGNN w/o L_c	**2.685 ± 0.109**	0.784 ± 0.002	1.920 ± 0.103	0.753 ± 0.011
TSGNN w/o attention	2.508 ± 0.259	0.811 ± 0.076	1.911 ± 0.980	0.760 ± 0.039

Table 6. An ablation study on Yelpsmall.

DataSets	P@20	MRR@20	P@10	MRR@10
TSGNN	**1.969 ± 0.541**	**1.097 ± 0.413**	0.822 ± 0.043	**0.355 ± 0.068**
TSGNN w/o weight	1.731 ± 0.606	1.010 ± 0.277	0.736 ± 0.923	0.186 ± 0.343
TSGNN w/o L_c	1.644 ± 0.476	1.077 ± 0.133	**1.010 ± 0.447**	0.340 ± 0.148
TSGNN w/o attention	1.774 ± 0.346	0.970 ± 0.064	0.736 ± 0.476	0.343 ± 0.031

4.5. Parameter Sensitivity Analysis

4.5.1. Analysis for Random Parameters R_f and R_e

We conduct sensitivity analysis for the two parameters R_f and R_e on the Amazon G&GF and Yelpsmall datasets, respectively. The results of this analysis are depicted in Figures 3 and 4. In the Amazon G&GF dataset, we observe a correlation between increasing values of R_f and improved model performance. As R_f rises, indicating a gradual reduction in features, the effect of the model becomes better. This suggests that the performance of the model is increasingly dependent on the graph structure. Compared to node features, the structural composition of the data plays an effective role in learning user preferences. For R_e, we observe that the influence of varying R_e values on the model's performance is inconsistent. As R_e increases, there is a noticeable improvement in the model's effectiveness, reaching its zenith at a value of 0.3. However, as R_e continues to rise, the model's performance starts to fluctuate and deteriorate. This trend indicates that an appropriate R_e value can enhance the model's performance, but exceeding this optimal point leads to diminishing returns. In the Yelpsmall dataset, both R_f and R_e show optimal performance at a value of 0.5. Different from R_f in Amazon G&GF, as R_f increases, the performance of the model initially increases but then starts to decline, peaking at 0.5. This shows that for the Yelpsmall data set, the features are as important as the structure. A certain R_f will promote the improvement of the model effect.

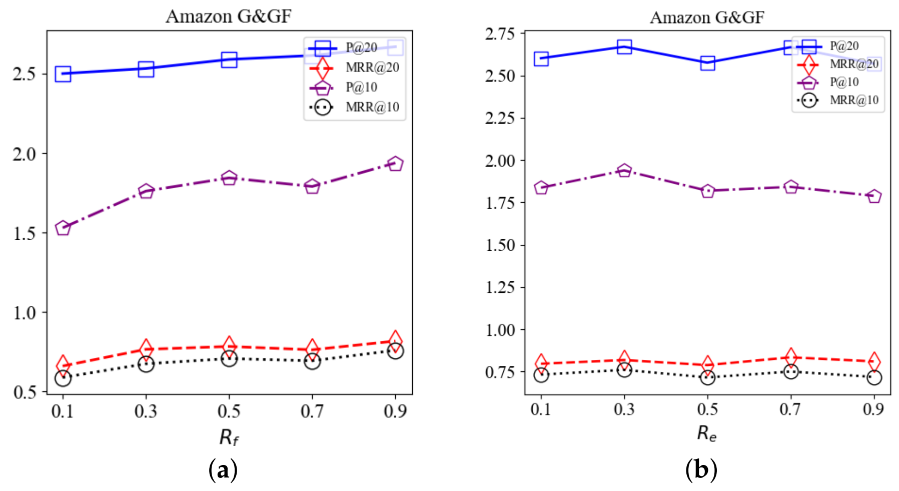


Figure 3. Parameter analysis of R_f and R_e on Amazon G&GF dataset. (**a**) The influence of R_f on Amazon G&GF dataset. (**b**) The influence of R_e on Amazon G&GF dataset.

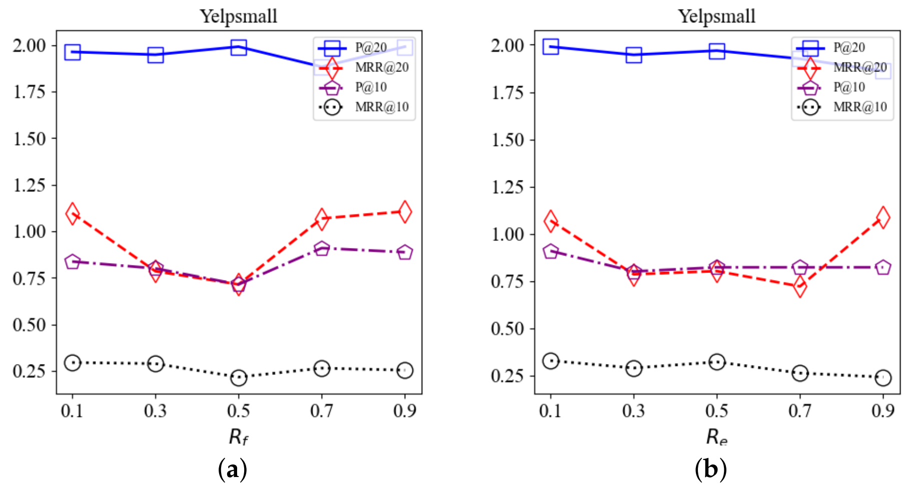


Figure 4. Parameter analysis of R_f and R_e on Yelpsmall dataset. (**a**) Influence of the parameter R_f. (**b**) Influence of the parameter R_e.

4.5.2. Analysis for Loss Parameter γ

To determine the contribution of each loss, we analyze the parameters of each loss. Figure 5 illustrates the results. Figure 5a reveals that as the γ parameter increases, there is a general upward trend in performance, highlighting the efficacy of our augmented method. However, in Figure 5b, we observed fluctuating behavior in the model's performance. Such fluctuation suggests that the impact of loss proportions on the model is not uniform, indicating a more intricate interplay between the loss parameters and the model's overall performance.

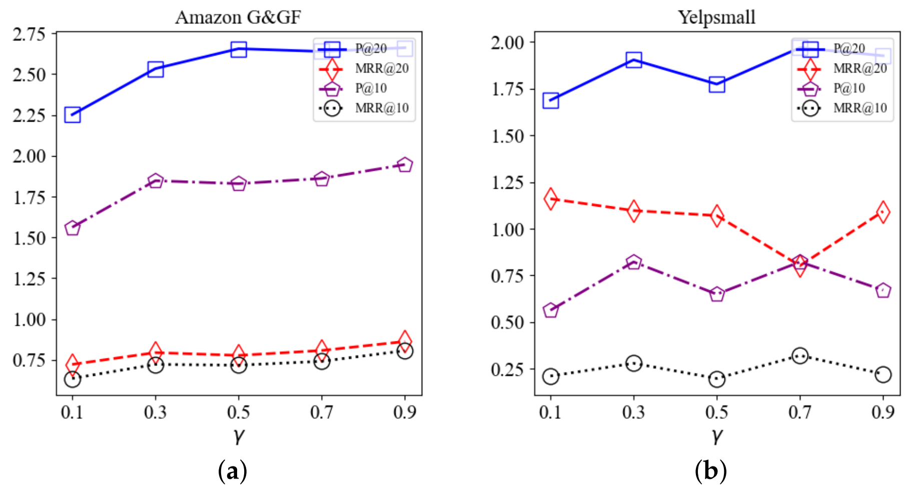


Figure 5. Parameter sensitivity analysis on γ. (**a**) γ on Amazon G&GF dataset. (**b**) γ on Yelpsmall dataset.

4.5.3. Analysis for Learning Rate Parameters

We show the learning rate (lr) sensitivity analysis experiment in Figure 6. In the Amazon G&GF dataset, we experiment with various learning rates, including {0.05, 0.01, 0.005, 0.001, and 0.0005}. As illustrated in Figure 6a, the model demonstrates optimal performance at a learning rate of 0.005. The effectiveness of TSGNN is reduced when the learning rate is too high, such as 0.05, or too low, such as 0.0005. As shown in Figure 6b, the model demonstrates its best performance in the Yelpsmall dataset at a learning rate of 0.001. This variation indicates that different datasets require different learning rates for optimal results, and it is essential to tailor the learning rate to the specific requirements of each dataset.

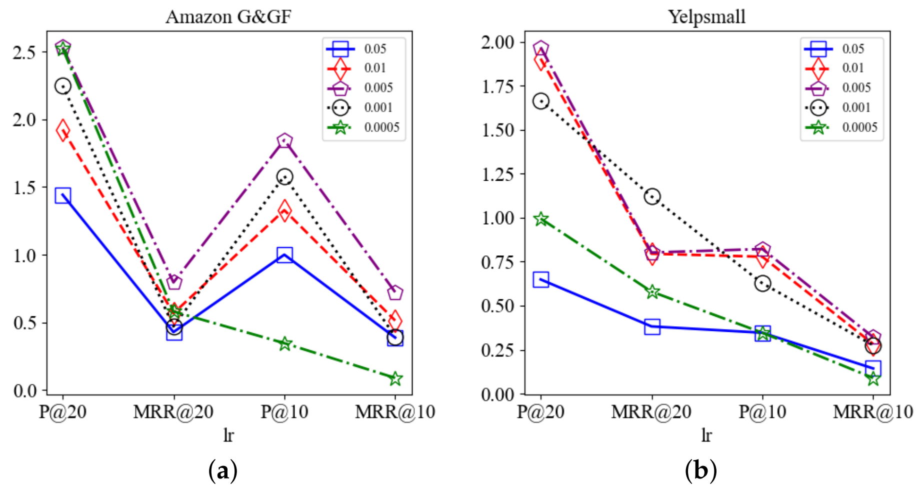


Figure 6. Parameter sensitivity analysis on learning rate. (**a**) Learning rate on Amazon G&GF dataset. (**b**) Learning rate on Yelpsmall dataset.

5. Conclusions

In this paper, we provide a time-sensitive graph neural network (TSGNN) specifically designed for session-based new item recommendations. By applying time-sensitive weights to each edge, TSGNN constructs time-sensitive session graphs, which solves the problem of ignoring user preference learning over time. The ablation experiments demonstrate that the time-sensitive session graph construction method outperforms the approach that does not incorporate time-sensitive weights. TSGNN incorporates graph augmentation technology and contrastive learning into its training process, effectively reducing the limitations on graph sparsity. Moreover, TSGNN adopts a novel strategy to mitigate the influence of the last visited item on user preferences by time-aware attention. This ensures a balanced assessment of each item's impact on user preference. The results from the ablation study affirm that the time-sensitive attention network substantially enhances the model's effectiveness. By integrating these three proposed methods, TSGNN surpasses existing SOTA models in a series of experiments, achieving an impressive enhancement of 36.56%. In the future, we aim to extend the application of time-sensitive methods across various recommendation fields and implement our proposed techniques in real-world scenarios.

Author Contributions: Methodology, L.W.; writing—original draft, L.W.; supervision, D.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data come from public datasets.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open-access journals
TLA	Three-letter acronym
LD	Linear dichroism

References

1. Jin, D.; Wang, L.; Zhang, H.; Zheng, Y.; Ding, W.; Xia, F.; Pan, S. A survey on fairness-aware recommender systems. *Inf. Fusion* **2023**, *100*, 101906. [CrossRef]
2. Loukili, M.; Messaoudi, F.; El Ghazi, M. Machine learning based recommender system for e-commerce. *Iaes Int. J. Artif. Intell.* **2023**, *12*, 1803–1811. [CrossRef]
3. Kokkodis, M.; Ipeirotis, P.G. The good, the bad, and the unhirable: Recommending job applicants in online labor markets. *Manag. Sci.* **2023**, *69*, 11. [CrossRef]
4. Gaw, F. Algorithmic logics and the construction of cultural taste of the Netflix Recommender System. *Media Cult. Soc.* **2022**, *44*, 706–725. [CrossRef]
5. Li, X.; Grahl, J.; Hinz, O. How do recommender systems lead to consumer purchases? A causal mediation analysis of a field experiment. *Inf. Syst. Res.* **2022**, *33*, 620–637. [CrossRef]
6. Piccardi, T.; Gerlach, M.; Arora, A.; West, R. A large-scale characterization of how readers browse Wikipedia. *ACM Trans. Web* **2023**, *17*, 1–22. [CrossRef]
7. Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A survey on session-based recommender systems. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–38. [CrossRef]
8. Guo, L.; Yin, H.; Wang, Q.; Chen, T.; Zhou, A.; Quoc Viet Hung, N. Streaming session-based recommendation. In Proceedings of the Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1569–1577.
9. Zihayat, M.; Ayanso, A.; Zhao, X.; Davoudi, H.; An, A. A utility-based news recommendation system. *Decis. Support Syst.* **2019**, *117*, 14–27. [CrossRef]
10. Lv, F.; Jin, T.; Yu, C.; Sun, F.; Lin, Q.; Yang, K.; Ng, W. SDM: Sequential deep matching model for online large-scale recommender system. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2635–2643.
11. Jin, D.; Wang, L.; Zheng, Y.; Li, X.; Jiang, F.; Lin, W.; Pan, S. CGMN: A Contrastive Graph Matching Network for Self-Supervised Graph Similarity Learning. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI, Vienna, Austria, 23–29 July 2022.
12. Zhang, T.; Zhao, P.; Liu, Y.; Sheng, V.S.; Xu, J.; Wang, D.; Liu, G.; Zhou, X. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, Macao, China, 10–16 August 2019; pp. 4320–4326.
13. Yuan, J.; Song, Z.; Sun, M.; Wang, X.; Zhao, W.X. Dual sparse attention network for session-based recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 4635–4643.
14. Sottocornola, G.; Symeonidis, P.; Zanker, M. Session-based news recommendations. In Proceedings of the Companion Proceedings of the The Web Conference, Lyon, France, 23–27 April 2018; pp. 1395–1399.
15. Hariri, N.; Mobasher, B.; Burke, R. Context-aware music recommendation based on latenttopic sequential patterns. In Proceedings of the Sixth ACM Conference on Recommender Systems, Dublin, Ireland, 9–13 September 2012; pp. 131–138.
16. Park, S.E.; Lee, S.; Lee, S.g. Session-based collaborative filtering for predicting the next song. In Proceedings of the 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering, Jeju, Korea, 23–25 May 2011; pp. 353–358.
17. Bhaskaran, S.; Marappan, R. Analysis of collaborative, content & session based and multi-criteria recommendation systems. *Educ. Rev. USA* **2022**, *6*, 387–390.
18. Liu, Y.; Zheng, Y.; Zhang, D.; Lee, V.C.; Pan, S. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 4516–4524.
19. Zhang, H.; Wu, B.; Yang, X.; Zhou, C.; Wang, S.; Yuan, X.; Pan, S. Projective Ranking: A Transferable Evasion Attack Method on Graph Neural Networks. In Proceedings of the CIKM, Queensland, Australia, 1–5 November 2021; pp. 3617–3621.
20. Zheng, Y.; Zhang, H.; Lee, V.C.; Zheng, Y.; Wang, X.; Pan, S. Finding the Missing-half: Graph Complementary Learning for Homophily-prone and Heterophily-prone Graphs. In Proceedings of the ICML, Honolulu, HI, USA, 23–29 July 2023.
21. Shen, Q.; Zhu, S.; Pang, Y.; Zhang, Y.; Wei, Z. Temporal aware multi-interest graph neural network for session-based recommendation. In Proceedings of the Asian Conference on Machine Learning, Hyderabad, India, 12–14 December 2023.
22. Wang, J.; Xie, H.; Wang, F.L.; Lee, L.K.; Wei, M. Jointly modeling intra-and inter-session dependencies with graph neural networks for session-based recommendations. *Inf. Process. Manag.* **2023**, *60*, 103209. [CrossRef]
23. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. Session-based recommendation with graph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 346–353.
24. Jin, D.; Wang, L.; Zheng, Y.; Song, G.; Jiang, F.; Li, X.; Lin, W.; Pan, S. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In Proceedings of the ACM Web Conference, Austin, TX, USA, 30 April–4 May 2023; pp. 684–693.
25. Wang, S.; Zhang, Q.; Hu, L.; Zhang, X.; Wang, Y.; Aggarwal, C. Sequential/Session-based Recommendations: Challenges, Approaches, Applications and Opportunities. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, 11–15 July 2022; pp. 3425–3428.

26. Ninichuk, M.; Namiot, D. Survey On Methods For Building Session-Based Recommender Systems. *Int. J. Open Inf. Technol.* **2023**, *11*, 22–32.
27. Ludewig, M.; Jannach, D. Evaluation of session-based recommendation algorithms. *User Model.-User-Adapt. Interact.* **2018**, *28*, 331–390. [CrossRef]
28. Norris, J.R. *Markov Chains*; Number 2; Cambridge University Press: Cambridge, UK, 1998.
29. Quadrana, M.; Karatzoglou, A.; Hidasi, B.; Cremonesi, P. Personalizing session-based recommendations with hierarchical recurrent neural networks. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 130–137.
30. Tan, Y.K.; Xu, X.; Liu, Y. Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 17–22.
31. Wang, Z.; Chen, C.; Zhang, K.; Lei, Y.; Li, W. Variational recurrent model for session-based recommendation. In Proceedings of the Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 1839–1842.
32. Wu, J.; Li, C.M.; Wang, L.; Hu, S.; Zhao, P.; Yin, M. On solving simplified diversified top-k s-plex problem. *Comput. Oper. Res.* **2023**, *153*, 106187. [CrossRef]
33. Wu, J.; Li, C.M.; Zhou, Y.; Yin, M.; Xu, X.; Niu, D. HEA-D: A Hybrid Evolutionary Algorithm for Diversified Top-k Weight Clique Search Problem. In Proceedings of the IJCAI, Vienna, Austria, 23–29 July 2022; p. 7.
34. Wang, L.; Li, C.; Zhou, J.; Jin, B.; Yin, M. An Exact Algorithm for Minimum Weight Vertex Cover Problem in Large Graphs. *arXiv* **2019**, arXiv:1903.05948.
35. Wang, L.; Hu, S.; Li, M.; Zhou, J. An exact algorithm for minimum vertex cover problem. *Mathematics* **2019**, *7*, 603. [CrossRef]
36. Zhang, H.; Wu, B.; Yuan, X.; Pan, S.; Tong, H.; Pei, J. Trustworthy Graph Neural Networks: Aspects, Methods and Trends. *arXiv* **2022**, arXiv:2205.07424.
37. Zheng, Y.; Koh, H.Y.; Ju, J.; Nguyen, A.T.; May, L.T.; Webb, G.I.; Pan, S. Large language models for scientific synthesis, inference and explanation. arXiv **2023**, arXiv:2310.07984.
38. Tan, Y.; Liu, Y.; Long, G.; Jiang, J.; Lu, Q.; Zhang, C. Federated learning on non-iid graphs via structural knowledge sharing. In Proceedings of the AAAI, Washington, DC, USA, 7–14 February 2023.
39. Zhang, H.; Yuan, X.; Nguyen, Q.V.H.; Pan, S. On the Interaction between Node Fairness and Edge Privacy in Graph Neural Networks. *arXiv* **2023**, arXiv:2301.12951.
40. Zhang, H.; Wu, B.; Wang, S.; Yang, X.; Xue, M.; Pan, S.; Yuan, X. Demystifying Uneven Vulnerability of Link Stealing Attacks against Graph Neural Networks. In Proceedings of the ICML, PMLR. *Proc. Mach. Learn. Res.* **2023**, *202*, 41737–41752.
41. Zhang, H.; Yuan, X.; Zhou, C.; Pan, S. Projective Ranking-Based GNN Evasion Attacks. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 8402–8416. [CrossRef]
42. Liu, Y.; Ding, K.; Lu, Q.; Li, F.; Zhang, L.Y.; Pan, S. Towards Self-Interpretable Graph-Level Anomaly Detection. In Proceedings of the NeurIPS 2023, New Orleans, LA, USA, 10–16 December 2023.
43. Liu, Y.; Ding, K.; Liu, H.; Pan, S. Good-d: On unsupervised graph out-of-distribution detection. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 339–347.
44. Wu, B.; Zhang, H.; Yang, X.; Wang, S.; Xue, M.; Pan, S.; Yuan, X. GraphGuard: Detecting and Counteracting Training Data Misuse in Graph Neural Networks. In Proceedings of the NDSS, San Francisco, CA, USA, 26 February–1 March 2024.
45. Zheng, Y.; Lee, V.C.; Wu, Z.; Pan, S. Heterogeneous graph attention network for small and medium-sized enterprises bankruptcy prediction. In Proceedings of the PAKDD, Virtual Event, 11–14 May 2021.
46. Liu, Y.; Ding, K.; Wang, J.; Lee, V.; Liu, H.; Pan, S. Learning Strong Graph Neural Networks with Weak Information. In Proceedings of the KDD, Long Beach, CA, USA, 6–10 August 2023.
47. Zhu, X.; Tang, G.; Wang, P.; Li, C.; Guo, J.; Dietze, S. Dynamic global structure enhanced multi-channel graph neural network for session-based recommendation. *Inf. Sci.* **2023**, *624*, 324–343. [CrossRef]
48. Kumar, C.; Abuzar, M.; Kumar, M. Mgu-gnn: Minimal gated unit based graph neural network for session-based recommendation. *Appl. Intell.* **2023**, *53*, 23147–23165. [CrossRef]
49. Chen, Z.; Xiao, T.; Kuang, K. Ba-gnn: On learning bias-aware graph neural network. In Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 3012–3024.
50. Zhang, Y.; Zhu, H.; Song, Z.; Koniusz, P.; King, I. Spectral feature augmentation for graph contrastive learning and beyond. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 11289–11297.
51. You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; Shen, Y. Graph contrastive learning with augmentations. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5812–5823.
52. You, Y.; Chen, T.; Shen, Y.; Wang, Z. Graph contrastive learning automated. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 12121–12132.
53. Wang, L.; Zheng, Y.; Jin, D.; Li, F.; Qiao, Y.; Pan, S. *Contrastive Graph Similarity Networks*; Association for Computing Machinery: New York, NY, USA, 2023 [CrossRef]
54. Zheng, Y.; Pan, S.; Lee, V.; Zheng, Y.; Yu, P.S. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 10809–10820.

55. Ju, W.; Gu, Y.; Luo, X.; Wang, Y.; Yuan, H.; Zhong, H.; Zhang, M. Unsupervised graph-level representation learning with hierarchical contrasts. *Neural Netw.* **2023**, *158*, 359–368. [CrossRef]
56. Chen, M.; Huang, C.; Xia, L.; Wei, W.; Xu, Y.; Luo, R. Heterogeneous graph contrastive learning for recommendation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, Singapore, 27 February–3 March 2023; pp. 544–552.
57. Xu, D.; Cheng, W.; Luo, D.; Chen, H.; Zhang, X. Infogcl: Information-aware graph contrastive learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 30414–30425.
58. Zhang, Z.; Sun, S.; Ma, G.; Zhong, C. Line graph contrastive learning for link prediction. *Pattern Recognit.* **2023**, *140*, 109537. [CrossRef]
59. Luo, X.; Ju, W.; Qu, M.; Chen, C.; Deng, M.; Hua, X.S.; Zhang, M. Dualgraph: Improving semi-supervised graph classification via dual contrastive learning. In Proceedings of the IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, 9–12 May 2022; pp. 699–712.
60. Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; Wang, L. Deep Graph Contrastive Representation Learning. In Proceedings of the ICML, Virtual Event, 13–18 July 2020.
61. Li, Y.; Tarlow, D.; Brockschmidt, M.; Zemel, R.S. Gated Graph Sequence Neural Networks. In Proceedings of the 4th International Conference on Learning Representations, ICLR, San Juan, Puerto Rico, 2–4 May 2016.
62. Wang, J.; Xu, Q.; Lei, J.; Lin, C.; Xiao, B. PA-GGAN: Session-based recommendation with position-aware gated graph attention network. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
63. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
64. Xu, C.; Zhao, P.; Liu, Y.; Sheng, V.S.; Xu, J.; Zhuang, F.; Fang, J.; Zhou, X. Graph contextualized self-attention network for session-based recommendation. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; Volume 19, pp. 3940–3946.
65. Wang, Z.; Wei, W.; Cong, G.; Li, X.L.; Mao, X.L.; Qiu, M. Global context enhanced graph neural networks for session-based recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 169–178.
66. Jiu, M.; Sahbi, H. DHCN: Deep hierarchical context networks for image annotation. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 3810–3814.
67. Xia, X.; Yin, H.; Yu, J.; Shao, Y.; Cui, L. Self-supervised graph co-training for session-based recommendation. In Proceedings of the Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual Event, 1–5 November 2021; pp. 2180–2190.
68. Chen, Y.; Xiong, Q.; Guo, Y. Session-based recommendation: Learning multi-dimension interests via a multi-head attention graph neural network. *Appl. Soft Comput.* **2022**, *131*, 109744. [CrossRef]