On the Evaluation of Session-based Recommendation Unlearning

Liu Yang
Shandong University
Qingdao, China
yangliushirry@gmail.com

Zhaochun Ren
z.ren@liacs.leidenuniv.nl
Leiden University
Leiden, The Netherlands

Ziqi Zhao
Shandong University
Qingdao, China
ziqizhao.work@gmail.com

Pengjie Ren
Shandong University
Qingdao, China
jay.ren@outlook.com

Zhumin Chen
Shandong University
Qingdao, China
chenzhumin@sdu.edu.cn

Jun Ma
Shandong University
Qingdao, China
majun@sdu.edu.cn

Xin Xin[†]
Shandong University
Qingdao, China
xinxin@sdu.edu.cn

ABSTRACT

Session-based recommendation predicts users' future interests from previous interactions in a session. Despite the memorizing of historical samples, the request of unlearning, i.e., to remove the effect of certain training samples, also occurs for reasons such as user privacy or model fidelity. However, existing studies on unlearning are not tailored for the session-based recommendation and seldom work has conducted the research to evaluate the unlearning effectiveness in the session-based recommendation scenario.

In this paper, we firstly propose SRU, a **s**ession-based **r**ecommendation **u**nlearning framework, which enables high unlearning efficiency, accurate recommendation performance, and improved unlearning effectiveness in session-based recommendation. To improve the unlearning effectiveness, we further propose three extra data deletion strategies. Besides, we set different methods to explore the evaluation of unlearning effectiveness in session-based recommendation: For item-level unlearning we propose an evaluation metric that measures whether the unlearning sample can be inferred after the data deletion to verify the unlearning effectiveness. And for session-level unlearning, we apply the membership inference attack to validate the unlearning effectiveness. We implement SRU with three representative session-based recommendation models and conduct experiments on three benchmark datasets. Experimental results demonstrate the effectiveness of our methods. Codes and data are available at https://github.com/shirryliu/SRU-code.

[*]Corresponding author.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Security and privacy** → **Web application security**.

KEYWORDS

Unlearning, Session-based Recommendation, Privacy Protection, Recommender System, Information Security

ACM Reference Format:
Xin Xin, Liu Yang, Ziqi Zhao, Pengjie Ren, Zhumin Chen, Jun Ma, Zhaochun Ren. 2018. On the Evaluation of Session-based Recommendation Unlearning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Session-based recommendation models have shown their effectiveness in predicting users' future interests from memorized sequential interactions [16, 41]. However, the ability to eliminate the influence of specific training samples, known as unlearning, also holds crucial significance. From a legitimacy perspective, several data protection regulations have been proposed, such as the General Data Protection Regulation (GDPR) [26] and the California Consumer Privacy Act (CCPA) [17]. These legislative regulations emphasize individuals' right to have their private information removed from trained machine learning models. As for user perspective, there has been a surge in research which proved that various user privacy information such as gender, age, and even political orientation could be inferred from historical interactions with a recommender system [4, 6, 44]. Addressing privacy concerns, users might find it imperative to request the expunction of specific historical interactions. Besides, a proficient recommendation model possesses the capacity to eliminate the impact of noisy training interactions to gain better performance.

Machine unlearning. Machine unlearning enables a model to forget certain data or patterns that it has previously learned. *Exact unlearning* targets on completely eradicating the impact of the data

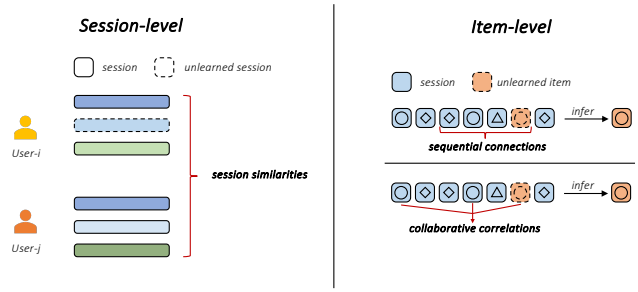

Figure 1: Exact unlearning is hard to achieve. Due to the similarity between sessions, user's intend could still be inferred from the rest of the sessions. And the unlearned item could still be inferred due to collaborative correlations and sequential connections across items in the session.

to be forgotten as if they never occurred in the training process. A straightforward exact unlearning method is to remove the targeted samples from the training dataset and then retrain the entire model from scratch. Unfortunately, this approach is hindered by its time-consuming and resource-intensive nature. To address this issue, existing methods [1, 2] focus on enhancing the efficiency of unlearning. One of the most representative unlearning methods is SISA [1]. The SISA initially divides the training dataset into disjoint shards and sub-models are trained on each shard independently. The final model prediction is by aggregating the predictions from every sub-model through majority voting or average. In the event of an unlearning request, sorely the sub-model that was trained on the shard containing the unlearning data point is retrained, rather than the entire model. SISA achieves significant improvement in unlearning efficiency compared with the whole retraining.

Challenges of unlearning in session-based recommendation. In the recommendation field, RecEraser [5] applies the SISA framework to non-sequential collaborative filtering. Nevertheless, we argue that there still exists the following key challenges for unlearning in session-based recommendation:

i). Exact unlearning is hard to achieve. Existing exact unlearning methods hold the assumption that the effect of unlearning samples would be completely removed if such samples do not exist in the retrained models. However, the assumption does not hold for session-based recommendation. Different from other domains such as image classification, where the correlations between training samples are sparse, there exists plenty of *collaborative correlations* and *sequential connections* across the interacted items in session-based recommendation. Consequently, simply removing the unlearning samples cannot achieve the exact unlearning effect, i.e., the unlearned item could still be inferred from the remaining items in the session, as shown in Figure 1.

ii). Existing recommendation unlearning methods do not evaluate the unlearning effectiveness. Existing methods [5, 23] mainly focus on the trade-off between recommendation performance and unlearning efficiency. However, seldom work conducted the evaluation regarding the unlearning effectiveness, i.e., to which extent the effect of unlearned samples is eliminated. The evaluation is especially important to verify the unlearning effectiveness of session-based recommendation, given the case that exact unlearning cannot be simply achieved.

The proposed method. In this paper, we propose **s**ession-based **r**ecommendation **u**nlearning (SRU), an unlearning framework tailored to session-based recommendation, achieving high unlearning efficiency, accurate recommendation, and improved unlearning effectiveness. Concretely, we first partition the training sessions into separate shards according to the similarity of the sessions and then a corresponding sub-model is trained upon each data shard. Such a data division strategy attempts to make similar sessions fall into the same shard, and thus each sub-model tends to learn a clustering of similar sequential patterns, resulting in improved recommendation performance. Utilizing the trained sub-models, we can obtain the hidden states which represent the session from the perspective of each sub-model. Then, an attention-based aggregation layer is trained to fuse the hidden states based on the correlations between the session and the centroid of the respective data shard.

To address the first challenge, we propose three extra data deletion strategies, including collaborative extra deletion (CED), neighbor extra deletion (NED), and random extra deletion (RED). For the second challenge, we propose specific calculation strategies for the two scenarios: for item-level unlearning, we propose an evaluation metric that measures whether the unlearning sample can be inferred after data deletion. The intuition is that if the unlearning is highly effective, the unlearning sample should have a low probability of being inferred based on the remaining data. And for session-level unlearning, we apply the membership inference attack to validate whether the model could accurately classify the unlearned sessions.

To verify the effectiveness of the proposed method, SRU is implemented on three representative session-based recommendation models including GRU4Rec [15], SASRec [20], and BERT4Rec [31]. We conduct a series of experiments on three benchmark datasets and the result shows the effectiveness of the proposed method.

Contributions. To summarize, the main contributions lie in:

- We propose SRU to address the machine unlearning problem for session-based recommendation. Three extra data deletion strategies are proposed to improve the unlearning effectiveness, and meanwhile, we use similarity-based clustering and attention-based aggregation to keep a high recommendation performance.
- We propose an evaluation metric to verify the unlearning effectiveness of session-based recommendation. The key idea is that if the unlearning is effective, the unlearning sample should have a low probability of being inferred after data deletion.
- We apply the membership inference attack to validate the session-level unlearning effectiveness. If the session is successfully forgotten, then the MIA model should be more inclined to classify deleted conversations as non-members which means higher accuracy in identifying non-members.
- We conduct extensive experiments on three state-of-the-art session-based recommendation models and three benchmark datasets to show that SRU can achieve efficient and effective unlearning while keeping high recommendation performance.

2 RELATED WORK

In this section, we provide a literature review regarding session-based recommendation and machine unlearning.

2.1 Session-Based Recommendation

Session-based recommendation aims to capture a user's dynamic interests from her/his past interactions in the session. Early Markov chains-based models [12, 13, 27, 29] predict a user's forthcoming interests according to the last interaction in the given session. More recently, deep neural network models have been utilized to capture complex sequential signals to improve session-based recommendation. The representative session-based recommendation models can be categorized into recurrent neural network (RNN)-based models [9, 14], convolutional neural network (CNN)-based models [32], attention-based models [20, 31], and graph-based models [36]. Besides, self-supervised learning [43] and contrastive learning [24, 37, 42] have also been applied to improve session-based recommendation and plenty of models have been emerged.

In this paper, we propose a framework that enables effective and efficient unlearning for various session-based recommendation models, other than developing a new specific model. We adopt three representative models, GRU4Rec, SASRec and BERT4Rec, as backbone models for the experiments.

2.2 Machine Unlearning

The concept of machine unlearning was first proposed by [3], in response to the requirement of "the right to be forgotten". Unlearning methods can be broadly categorized into approximate unlearning methods and exact unlearning methods.

Approximate unlearning ensures that the performance of the unlearned model closely aligns with that of a retrained model. This reduces the time and computational cost of unlearning, but at the potential expense of weaker privacy assurances. The approximation can be achieved through differential privacy techniques, such as certified unlearning [39]. For instance, [33] introduced an unlearning method based on noisy stochastic gradient descent, whereas [10] achieved certified unlearning based on Newton updates. [3] proposed to use the gradient surgery which updates the model parameters using the negative gradient of the unlearning samples. [18] utilized a probabilistic model to approximate the unlearning process. [7, 25] proposed to perturb the gradients or model weights through the inverse Hessian matrix, which may incur additional computational overheads.

Exact unlearning attempts to completely remove the effect of the unlearning samples as if they have never occurred in the training process, providing a stronger privacy guarantee. However, such methods could require the model to be retrained from scratch, which is computationally expensive and time-consuming. The most representative method for efficient exact unlearning is SISA [1] since only the sub-model trained on the corresponding data shard would be retrained for an unlearning request. [8] adapted SISA for unlearning in graph neural networks. [11] modified the SISA algorithm to work for sequences of deletion requests. Another kind of method for exact unlearning involves selective influence estimators [35], which calculate the influence of the unlearning samples on the model parameters. Although such influence-based methods are effective in terms of privacy preservation, the high computational cost limits their application for real-world scenarios [39].

Recently, unlearning in the recommendation scenario tends to attract more research attention. Unlearning can not only help to protect user privacy but also improve recommendation models through eliminating the effect of noisy data and misleading information [28]. [23] and [40] proposed to use fine-tuning and the alternative least square algorithm for unlearning acceleration. [5] and [22] extended the ideas of the SISA algorithm for collaborative filtering. However, none of the existing methods is tailored for session-based recommendation. Besides, existing methods mainly focus on the unlearning efficiency, while failing to verify the effectiveness of the unlearning, i.e., to which extent the effect of the unlearning sample is removed.

3 TASK FORMULATION

In this section, we first formulate the task of session-based recommendation, upon which we define the task of item-level in a session and session-level unlearning.

3.1 Notations and Definitions

Session-based recommendation aims to predict the user's potential next action given previous interacted items in the session. We formulate the task as follows:

Definition 3.1 (Session-based recommendation)**.** Let $\mathcal{V} = \{v_1, v_2, ..., v_{|\mathcal{V}|}\}$ be the set of items, $\mathcal{D}$ denotes the training interaction sessions. $\mathcal{S}_i = [v_1^i, ..., v_t^i, ..., v_n^i] \in \mathcal{D}$ denotes the i-th specific interaction session in $\mathcal{D}$, where $v_t^i \in \mathcal{V}$ is the item interacted by the user at time step t, and n is the current length of the session. Given the historical sequence $\mathcal{S}_i$, the interaction probability over candidate item v at time step $n+1$ can be formalized as:

$$p_v^i = p(v_{n+1}^i = v|\mathcal{S}_i, \mathcal{D}) = \mathcal{M}(v|\mathcal{S}_i, \mathcal{D}), \tag{1}$$

where $\mathcal{M}$ denotes the involved recommendation model, e.g., GRU4Rec [14] and SASRec [20]. At the prediction stage, session-based recommenders select the items with the highest top-K probability p_v^i as the recommendation list for the user.

For privacy considerations or recommendation utility, an unlearning request could occur to remove the effect of certain training samples. As an illustration, a user may want to revoke some misclicks in an interaction session since the misclicks can downgrade the recommendation quality or a user could also request to hide the history of certain sensitive sessions for private concerns. In this paper, we focus on item-level and session-level unlearning in session-based recommendation, which is defined as follows:

Definition 3.2 (Item-level unlearning)**.** We denote $v_j^i \in \mathcal{S}_i$ to be the unlearning item that the user wants to revoke in the session $\mathcal{S}_i$. The goal of item-level unlearning is to obtain an unlearned model $\mathcal{M}_u$. Ideally, the unlearning sample v_j^i should have no effect on the unlearned model $\mathcal{M}_u$ as if v_j^i never occurred in the session.

Definition 3.3 (Session-level unlearning)**.** Session-level unlearning aims to revoke the effect of a whole interaction session. We denote $\mathcal{S}_i$ to be the unlearning session that the user wants the model to delete from the training dataset. The goal of session-level unlearning is to obtain an unlearned model $\mathcal{M}_u$. Ideally, the unlearning session $\mathcal{S}_i$ should have no effect on the unlearned model $\mathcal{M}_u$ as if $\mathcal{S}_i$ never occurred in $\mathcal{D}$.

4 METHODOLOGY

In this section, we describe the detail of the proposed SRU framework. As shown in Figure 2, SRU is composed of *session partition*, *attentive aggregation*, and *data deletion*. The session partition module aims to divide the training sessions into disjoint data shards and then sub-models are trained on each shard. Based on the hidden states coming from different sub-models, the attentive aggregation module fuses the hidden states for the final prediction. The data deletion module aims to improve the unlearning effectiveness. When an item-level unlearning request comes, the data deletion module first applies extra data deletion strategies to the corresponding session. Then only the sub-model and the aggregation module are retrained, achieving efficient unlearning.

4.1 Session Partition

One keystone to generating the next-item recommendation is learning signals from similar sessions. To this end, session similarity is important for recommendation accuracy. Consequently, in the session partition module, similar sessions are expected to be divided into the same data shard and thus can be trained in one sub-model. Such a division strategy can help to improve the recommendation performance since it enables more knowledge transfer within each shard.

To achieve the described division strategy, an additional session-based recommendation model $\mathcal{M}_p$ (e.g., GRU4Rec[15]) is pre-trained on $\mathcal{D}$ to obtain all training sessions' hidden states firstly. Then a k-means clustering method based on the pre-trained hidden states is used to divide training sessions. More specifically, the input of the session partition module includes the pre-trained hidden states, the number of partition shards $\mathcal{K}$, and the maximum number of sessions in each shard δ. The distance between session pairs is defined as the Euclidean distance of their hidden states. $\mathcal{K}$ sessions are randomly selected as centroids at first, then distances between sessions and centroids are calculated. Subsequently, the sessions are assigned to the shard sequentially according to the ascending order of distances. If one shard is unavailable (i.e., the number of sessions within the shard is larger than δ), the next session is assigned to the nearest available shard. After that, the new centroids are calculated as the mean of all sessions' hidden states in each corresponding shard. The above process is repeated until the centroids are no longer updated. Then we obtain the balanced-partition session as $\bigcap_{k\in[\mathcal{K}]} \mathcal{D}_k = \varnothing$ and $\bigcup_{k\in[\mathcal{K}]} \mathcal{D}_k = \mathcal{D}$. Then sub-models are trained on the data shard separately.

4.2 Attentive Aggregation

Based on the session partition, each sub-model tends to learn a clustering of similar sequential patterns. The attentive aggregation module aims to fuse the hidden states from each sub-model for the final prediction, which consists of a projection layer, an attention layer, and an output layer.

4.2.1 Projection layer. Given a session we compute its hidden representation $\mathbf{h}_k \in \mathbb{R}^d$ using each sub-model $\mathcal{M}_k$ trained on $\mathcal{D}_k$. Since sub-models are trained separately, the hidden representations could embed in different vector spaces. In order to utilize the knowledge of every sub-model, we need to project the hidden representations into a common space. Specifically, a linear transfer layer is used to conduct the projection:

$$\mathbf{h}'_k = \mathbf{W}_k \mathbf{h}_k + \mathbf{b}_k, \tag{2}$$

where $\mathbf{W}_k \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_k \in \mathbb{R}^d$ are projection parameters. Note that each sub-model $\mathcal{M}_k$ has a corresponding $\mathbf{W}_k$ and $\mathbf{b}_k$.

Besides, the data centroid of $\mathcal{D}_k$ is also projected as

$$\mathbf{c}'_k = \mathbf{W}_k \mathbf{c}_k + \mathbf{b}_k, \tag{3}$$

where $\mathbf{c}_k$ denotes the original centroid representation computed from $\mathbf{h}_k$. The data centroid representation is used in the following attention layer.

4.2.2 Attention layer. The attention layer aims to compute the importance of each sub-model for a given session. [5] also used an attention layer to fuse user and item embeddings for unlearning in collaborative filtering. However, their method cannot be applied to session-based recommendation since their attention is solely based on either user or item embedding. While in the session-based recommendation, we argue that the attention should be based on the correlations between the session and the data centroid (i.e., the attention layer should have two input sources corresponding to the session representation and the centroid representation, as shown in Figure 2).

To this end, we define the attention score for sub-model $\mathcal{M}_k$ as

$$a_k = softmax(\mathbf{g} \cdot ReLU(\mathbf{W}' \odot (\mathbf{h}'_k \odot \mathbf{c}'_k) + \mathbf{b}')), \tag{4}$$

where $\mathbf{W}' \in \mathbb{R}^{d\times f}$, $\mathbf{b}' \in \mathbb{R}^f$ and $\mathbf{g} \in \mathbb{R}^f$ are learnable attention parameters. f is the size of the attention layer. $\odot$ denotes element-wise product and $\cdot$ denotes the inner product.

Based on the attention score, the final representation of a session is formulated as

$$\mathbf{h}^f = \sum_{k=1}^{\mathcal{K}} a_k \mathbf{h}'_k. \tag{5}$$

4.2.3 Output layer. Based on the final aggregated hidden representation $\mathbf{h}^f$, a two-layer feed-forward network with ReLU activation is used to produce the output distribution over candidate items. The attentive aggregation module is trained with the cross-entropy loss over the output distribution.

4.3 Data Deletion

The data deletion module aims to improve the unlearning effectiveness. For an item-level unlearning request, conventional unlearning methods just remove the unlearning sample, while it is still possible that the removed sample can be inferred again from the remaining interactions in the session due to the existence of sequential connections and collaborative correlations. And for session-level unlearning request, the unlearned model could still infer the user's intent from other similar sessions. To address the problem, we propose three strategies, namely Collaborative Extra Deletion(CED), Neighbor Extra Deletion(NED) and Random Extra Deletion(RED).

From the view of collaborative correlations, we propose CED. In item-level unlearnling, it deletes extra items based on the similarities between the unlearning item and other items in the session.

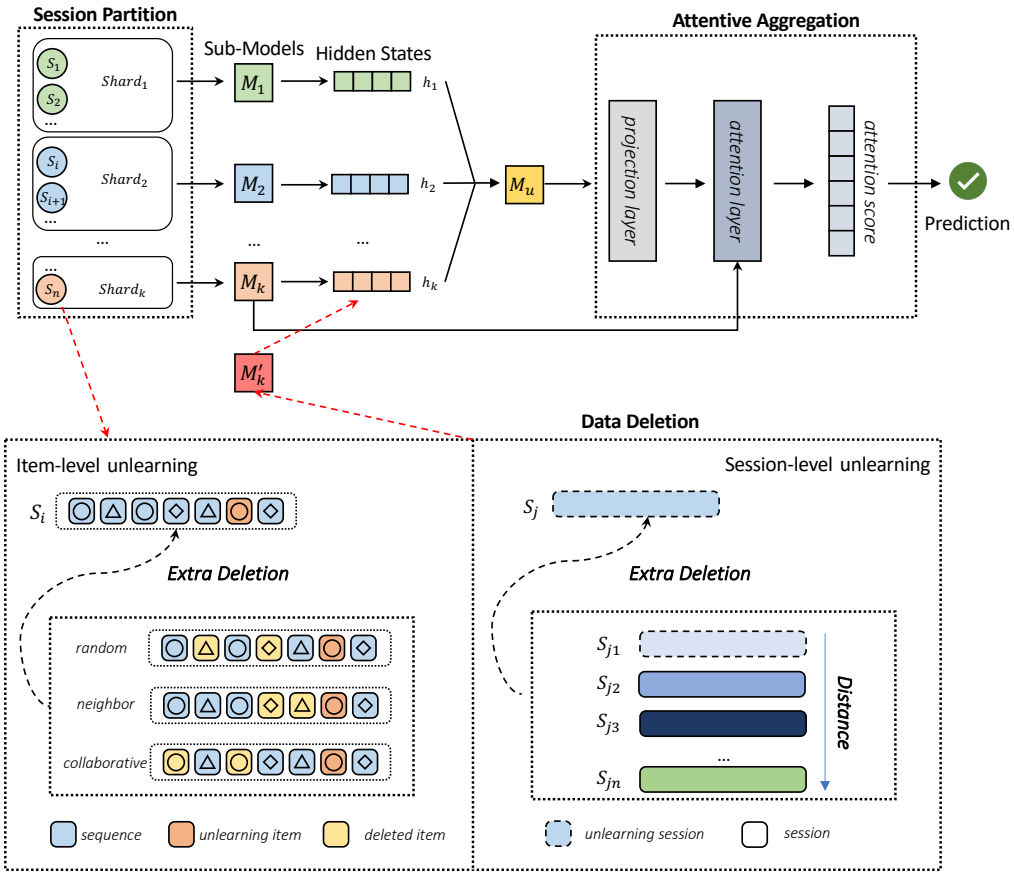

Figure 2: Overview of the proposed SRU framework. SRU is composed of session partition, attentive aggregation and data deletion modules. The data deletion module includes both session-level and item-level unlearning with extra deletion.

Given the target unlearning item v_j^i in session $\mathcal{S}_i$, the item similarity is calculated according to the Euclidean distance between item embeddings obtained from the pre-trained model $\mathcal{M}_p$. After that, the items in the session are sorted by the ascending order of the distances and the most $\mathcal{N}$ similar items are also removed from the session. Finally, the corresponding sub-model and the aggregation module are re-trained. The unlearned model can be formalized as

$$\mathcal{M}_u(v|\mathcal{S}_i'', \mathcal{D}\backslash\mathcal{S}_i \cup \mathcal{S}_i''), \text{where } \mathcal{S}_i'' = \mathcal{S}_i\backslash\text{CED}(v_j^i). \quad (6)$$

As for sequential connections, NED is proposed to remove the $\mathcal{N}$ nearest items in front of the unlearning item in chronological order. While in RED, we randomly choose $\mathcal{N}$ extra items to delete within the session.

In session-level unlearning, CED will additionally delete other conversations that are similar to the session to be forgotten, as there may be a high degree of similarity between sessions. In line with the item-level approach, given the target unlearning session $\mathcal{S}_j$ in unlearning session set $\mathcal{S}_u$, the session similarity is calculated to the Euclidean distance between session hidden states obtained by the pre-trained model $\mathcal{M}_p$. And will delete the most similar sessions. Due to the data sparsity, for example, the sparsity of Steam is 99%, we only use CED method for session-level removal.

4.4 Item-level Unlearning Effectiveness Evaluation

Item-level unlearning is a common request, for example, a user may want to hide the click of a sensitive item in a session or may dislike an item anymore. To this end, if the unlearning is effective, the unlearned item should not be inferred from the remaining items in the session or the item should not be recommended to the user again in the near future.

To this end, we define one unlearning effectiveness evaluation metric as the hit ratio (i.e., HIT@K) which measures whether the unlearned item would occur in the top-K recommendation list based on the remaining interactions in the session using the unlearned model $\mathcal{M}_u$. Such the evaluation metric can also be seen as the performance of a membership inference attack [30] which attempts to infer the unlearning items from the remaining data. If HIT@K is high, it means the unlearned item has a high probability of being re-recommended or being inferred again. On the contrary, a lower HIT@K implies better unlearning effectiveness.

Table 1: Statistics of the datasets (after preprocessing).

Dataset	#users	#items	#actions
Amazon Beauty	52,024	57,289	0.4M
Amazon Games	31,013	23,715	0.3M
Steam	334,730	13,047	3.7M

4.5 Session-level Unlearning Effectiveness Evaluation

For a session-level unlearning, such as a user may want to withdraw some out of style or unfaithful sessions in the model, we adopt the approach of directly removing these dialogues from the training set to obtain the unlearned model $\mathcal{M}_u$.

ME-MIA framework is tailor-made for sequential recommendation systems. In the detail of the membership inference attack, we follow [44] to train a target and shadow model separately, and the target model is exactly the unlearned model $\mathcal{M}_u$ we want to obtain. Because we treat the target model as a black-box, so it is necessary to build a surrogate model $\mathcal{M}_{surrogate}$ to simulate the target model which is called model extraction. Then we use the shadow model's training data to train the attack model to be a binary classifier. To better characterize the features of the given session, a feature generator is built to construct representative feature vectors from $\mathcal{M}_{shadow}$ and $\mathcal{M}_{surrogate}$. Then, the generative feature vectors are fed into the attack model $\mathcal{M}_{attack}$ to identify the session's membership. We calculate the accuracy of correctly identifying the unlearning sessions in the attack model.

From ME-MIA, we can obtain an attacker $\mathcal{M}_{attack}$ for the unlearned model $\mathcal{M}_u$. Since $\mathcal{M}_{attack}$ is built from the information of $\mathcal{M}_u$, if $\mathcal{M}_u$ has achieved a better unlearning effectiveness, then $\mathcal{M}_{attack}$ should recognize the unlearning sessions as the non-member better, which means a higher accuracy score.

5 EXPERIMENTS

In this section, we conduct experiments on three benchmark datasets to verify the effectiveness of SRU. We aim to answer the following research questions:

RQ1: How is the recommendation performance of SRU when instantiated with different session-based recommendation models?

RQ2: How is the unlearning effectiveness of SRU?

RQ3: How is the unlearning efficiency of SRU?

5.1 Experimental Settings

5.1.1 Datasets. Experiments are conducted on three publicly accessible datasets: *Amazon Beauty*, *Games* and *Steam*. The two Amazon datasets[1] are a series of product review datasets crawled from Amazon.com. In this work, we consider two item categories including "Beauty" and "Games". The Steam dataset[2] is collected from a large online video game distribution platform. For all datasets, we follow the same data prepossessing as [20]. Table 1 shows the statistics of the datasets.

5.1.2 Recommendation performance evaluation. We adopt cross-validation to evaluate the performance of the proposed methods. The ratio of training, validation, and test set is 8:1:1. We randomly sample 80% of the sessions as the training set. For validation and test sets, The evaluation is done for validation and test sets by providing the interactions in a session one by one and checking the rank of the next ground-truth item. The ranking is performed among the whole item set.

To evaluate recommendation performance, we adopt two common top-K metrics: Recall@K and NDCG@K. Recall@K measures whether the ground-truth item is in the top-K positions of the recommendation list [38]. NDCG@K is a weighted metric that assigns higher scores to top-ranked positions [19]. We use the metric HIT described in section 4.4 to evaluate item-level unlearning effectiveness and Accuracy score described in section 4.5 to evaluate session-level unlearning effectiveness.

5.1.3 Baselines. SRU is implemented with three representative session-based recommendation models: GRU4Rec [15], SASRec [20] and BERT4Rec [31].

- **GRU4Rec** [15]: This method utilizes gated recurrent units (GRU) to model user interaction sequences.
- **SASRec** [20]: This model is attention-based and uses the Transformer [34] decoder for session-based recommendation.
- **BERT4Rec** [31]: This model employs deep bidirectional self-attention to model interaction sequences.

To enable unlearning, every model is trained with:

- **Retrain:** This method retrains the whole model from scratch on the remaining dataset. It's computationally expensive.
- **SISA:** This is a fundamental exact unlearning method that randomly splits the data and averages the outputs of the sub-model.
- **SRU-N:** This is SRU with neighbor extra deletion (NED).
- **SRU-R:** This is SRU with random extra deletion (RED).
- **SRU-C:** This is SRU with collaborative extra deletion (CED).

Note that we do not compare with RecEraser [5] since it is proposed for non-sequential collaborative filtering and their data partition methods cannot be applied for session-based recommendation since the session-based recommender does not explicitly model user identifiers.

5.1.4 Hyperparameter settings. The model input is the last 10 interacted items for Beauty, and the last 20 interacted items for Games and Steam. We pad the sequences with a padding token for shorter sessions. The Adam optimizer [21] is used to train all models, with batches of size 256. The learning rate for the aggregation layer is tuned among [1e-3, 1e-2]. The default number of data shard is set as $\mathcal{K} = 8$. The extra data deletion number for unlearning is ranged from 1 to 5. The other hyperparameters are set as the recommended settings of their original papers.

5.2 Recommendation Performance (RQ1)

Table 2 shows the top-K recommendation performance of different unlearning methods when 10% random sessions need items to be unlearned in each shard.

We can see that the proposed SRU always performs better than SISA even though SRU has removed more training data. This is because when training data is unlearned, the performance of all sub-models are degraded for the training data is smaller, while SRU groups similar sessions in a shard which makes the model

[1] https://jmcauley.ucsd.edu/data/amazon/
[2] https://steam.internet.byu.edu/

Table 2: Recommendation performance comparison after unlearning 10% of data in each shard. The extra deletion number $\mathcal{N}$ ranges from 1 to 5. Best results other than Retrain are highlighted in bold. "N" is short for NDCG and "R" is short for Recall.

Beauty	GRU4Rec				SASRec				BERT4Rec			
	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20
Retrain	0.0327	0.0382	0.0550	0.0768	0.0399	0.0450	0.0632	0.0835	0.0314	0.0380	0.0558	0.0816
SISA	0.0289	0.0328	0.0460	0.0615	0.0271	0.0307	0.0428	0.0571	0.0259	0.0310	0.0464	0.0666
SRU-R	0.0304	**0.0347**	0.0489	0.0662	0.0280	**0.0323**	0.0448	**0.0617**	0.0292	0.0341	0.0509	0.0704
SRU-C	0.0286	0.0330	0.0468	0.0643	**0.0280**	0.0320	**0.0456**	0.0616	**0.0293**	**0.0348**	**0.0525**	**0.0743**
SRU-N	**0.0306**	0.0346	**0.0506**	**0.0668**	0.0274	0.0312	0.0440	0.0591	0.0291	0.0346	0.0507	0.0726
Steam	**GRU4Rec**				**SASRec**				**BERT4Rec**			
	N@10	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**
Retrain	0.0495	0.0631	0.0947	0.1489	0.0539	0.0679	0.1016	0.1574	0.0593	0.0742	0.1116	0.1711
SISA	0.0471	0.0601	0.0898	0.1412	0.0457	0.0581	0.0863	0.1357	0.0482	0.0615	0.0932	0.1460
SRU-R	**0.0490**	**0.0621**	**0.0924**	0.1444	**0.0485**	**0.0614**	**0.0914**	**0.1431**	**0.0577**	**0.0722**	**0.1077**	**0.1652**
SRU-C	0.0484	0.0616	0.0916	**0.1445**	0.0476	0.0604	0.0901	0.1411	0.0576	0.0720	0.1075	0.1648
SRU-N	0.0480	0.0612	0.0916	0.1442	0.0480	0.0608	0.0906	0.1414	0.0567	0.0710	0.1067	0.1636
Games	**GRU4Rec**				**SASRec**				**BERT4Rec**			
	N@10	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**	**N@10**	**N@20**	**R@10**	**R@20**
Retrain	0.0401	0.0495	0.0747	0.1122	0.0479	0.0580	0.0864	0.1268	0.0474	0.0596	0.0921	0.1406
SISA	0.0324	0.0377	0.0564	0.0776	0.0267	0.0318	0.0459	0.0661	0.0322	0.0402	0.0629	0.0948
SRU-R	**0.0357**	0.0424	**0.0621**	0.0887	**0.0333**	**0.0405**	**0.0596**	**0.0883**	**0.0395**	**0.0497**	**0.0752**	**0.1159**
SRU-C	0.0342	0.0410	0.0614	0.0887	0.0314	0.0378	0.0570	0.0824	0.0363	0.0462	0.0690	0.1084
SRU-N	0.0352	**0.0424**	0.0620	**0.0909**	0.0321	0.0393	0.0566	0.0851	0.0384	0.0488	0.0730	0.1146

Table 3: Session-level unlearning effectiveness comparison. Higher scores denote better results. The best results are highlighted in bold.

Beauty	GRU4Rec		SASRec		BERT4Rec	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Retrain	0.7487	0.6048	0.7661	0.7357	0.9054	0.6977
SISA	0.7412	0.5821	0.7487	0.5421	0.8743	0.5419
SRU	0.7688	0.6959	0.794	**0.7552**	0.9196	0.7049
SRU-C	**0.8040**	**0.7181**	**0.8091**	0.7275	**0.9347**	**0.7689**
Steam	**GRU4Rec**		**SASRec**		**BERT4Rec**	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Retrain	0.4081	0.5757	0.5077	0.5535	0.4472	0.5447
SISA	0.4050	0.5757	0.4992	0.5348	0.4102	0.5662
SRU	0.5085	0.5751	0.5242	0.5771	0.5507	0.5038
SRU-C	**0.5314**	**0.5999**	**0.5371**	**0.5986**	**0.5662**	**0.5766**
Games	**GRU4Rec**		**SASRec**		**BERT4Rec**	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Retrain	0.6438	**0.6797**	0.7397	0.6476	0.7808	0.6123
SISA	0.5734	0.5718	0.6783	0.5633	0.7762	0.5536
SRU	0.6433	0.6091	0.7482	0.7026	0.8182	**0.6344**
SRU-C	**0.6853**	0.6526	**0.7692**	**0.7137**	**0.8741**	0.5798

share more collaborative information to gain better recommendation performance. And it makes sense that Retrain always gets the highest scores for it can retrain the whole model on all remaining data, but sacrifices efficiency.

5.3 Unlearning Effectiveness (RQ2)

In this part, we conduct experiments to evaluate the unlearning effectiveness of different methods.

For item-level unlearning, we randomly unlearn 10% of data and set the extra deletion number $\mathcal{N}$ from 1 to 5. Table 4 shows the item-level unlearning effectiveness comparison on Beauty and Steam datasets. The results in the Games dataset show a similar conclusion. Firstly, we can see that even if the unlearning item is removed, there is a probability (e.g., more than 10% on the Steam dataset) that the item can be inferred again from the remaining interactions in the session. This observation verifies that conventional exact unlearning methods cannot achieve exact unlearning effects in the session-based recommendation scenario. Besides, we can see that the proposed SRU-R, SRU-C and SRU-N achieve better unlearning effectiveness compared with Retrain and SISA. For example, on the GRU4Rec model and trained on the Beauty dataset, the HIT@1 for SRU is 0.0577, while for Retrain is 0.0764. The observation indicates that the proposed data deletion module is essential for unlearning effectiveness. What's more, SRU-C and SRU-N achieve stable unlearning effectiveness improvement since they can help to eliminate the effect of collaborative correlations and sequential connections correspondingly, while SRU-R removes extra data randomly and has a more varied performance.

For session-level unlearning, we randomly unlearn 1% of data and Table 4 shows the accuracy of attack model's prediction for unlearning data. AUC score is the area under the ROC curve which

Table 4: Unlearning effectiveness comparison. Lower scores denote better results. The best results are highlighted in bold.

Beauty	GRU4Rec				SASRec				BERT4Rec			
	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20
Retrain	0.0764	0.1715	0.2294	0.3052	0.0619	0.1566	0.2123	0.2807	0.0700	0.1588	0.2080	0.2739
SISA	0.0685	0.1654	0.2244	0.3074	0.0681	0.1605	0.2222	0.3091	0.0763	0.1730	0.2321	0.3119
SRU-R	0.0675	0.1561	0.2122	0.2809	0.0625	0.1468	0.2042	0.2697	0.0720	0.1573	0.2131	0.2798
SRU-C	**0.0577**	**0.1335**	**0.1824**	**0.2510**	**0.0593**	**0.1429**	**0.1970**	**0.2666**	0.0661	**0.1516**	0.2058	**0.2689**
SRU-N	0.0643	0.1533	0.2028	0.2731	0.0605	0.1482	0.2039	0.2736	**0.0638**	0.1527	**0.2054**	0.2759
Steam	**GRU4Rec**				**SASRec**				**BERT4Rec**			
	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20	HIT@1	HIT@5	HIT@10	HIT@20
Retrain	0.1581	0.3992	0.5372	0.6805	0.1411	0.3636	0.4975	0.6483	0.1159	0.3292	0.4701	0.6309
SISA	0.1582	0.3979	0.5349	0.6775	0.1410	0.3646	0.4959	0.6365	0.1166	0.3282	0.4668	0.6184
SRU-R	0.1545	0.3954	0.5319	0.6739	0.1412	0.3687	0.5020	0.6417	0.0992	0.2979	0.4282	0.5749
SRU-C	0.1499	0.3882	0.5241	0.6702	0.1389	0.3686	0.5041	0.6475	0.1036	0.3088	0.4407	0.5901
SRU-N	**0.1461**	**0.3799**	**0.5136**	**0.6568**	**0.1138**	**0.3186**	**0.4422**	**0.5812**	**0.0957**	**0.2897**	**0.4205**	**0.5713**

Table 5: Comparison of unlearning efficiency (minute [m]). The best results are highlighted in bold.

Dataset		Beauty			Games			Steam		
Method		GRU4Rec	SASRec	BERT4Rec	GRU4Rec	SASRec	BERT4Rec	GRU4Rec	SASRec	BERT4Rec
Retrain		46.80m	55.60m	55.76m	31.22m	29.91m	31.14m	274.67m	368.99m	296.89m
SRU	**Sub-model**	5.80m	5.07m	7.44m	3.76m	4.75m	4.80m	33.67m	36.78m	34.07m
	Aggregation	0.72m	6.05m	5.53m	1.78m	4.40m	3.87m	25.30m	62.53m	64.30m
	Total	**6.52m**	**11.12m**	**12.97m**	**5.54m**	**9.15m**	**8.67m**	**58.97m**	**99.31m**	**98.37m**

is widely used in binary classifcation problems due to its insensitivity to the label distribution of the dataset and we use it to verify the attack model's effectiveness. Higher AUC score means the attack model has a better ability to valid data for members and non-members respectively. In this experiment setting, if the unlearning session has been removed from the unlearned model $\mathcal{M}_u$ more complete, then the attack model can recognize the unlearning session as non-member better which means a higher accuracy. We can observe that SRU-C has the highest Accuracy scores with a reasonable AUC score in all datasets and models which means that it has better unlearning effectiveness. For example, on Games and BERT4Rec, the Accuracy is 0.7808 for Retrain and 0.8741 for SRU-C, the improvement is 11.9% and the comparison between SRU-C and SISA is more obvious. This means that extra deletion method does help the model to achieve more complete oblivion.

To conclude, the proposed SRU achieves the highest unlearning effectiveness in both item-level and session-level, even better than Retrain.

5.4 Unlearning Efficiency (RQ3)

Table 5 shows the training time comparison between Retrain and SRU. We evaluate them both on NVIDIA GeForce RTX 2080 Ti and set the shard number to 8. Especially the retraining time of SRU consists of sub-model training and aggregation module training. From Table 5, we can find that SRU performs much more efficiently than Retrain. In most cases, SRU is more than three times faster than Retrain. For example, on Beauty and BERT4Rec, Retrain needs 55.76 minutes, but our SRU only needs 12.97 minutes. The efficiency improvement is more significant on the larger Steam dataset. For example, on Steam and SASRec, Retrain needs 368.99 minutes, but our SRU only needs 99.31 minutes, the improvement reaches 3.71x optimisation.

6 CONCLUSION

In this paper, we have proposed a model-agnostic unlearning framework SRU for session-based recommendation. We firstly divide unlearning requests into item-level and session-level. For an item-level unlearning request, SRU utilizes three data deletion strategies, including collaborative extra deletion (CED), neighbor extra deletion (NED), and random extra deletion (RED), to ensure the unlearned items cannot be inferred again from the remaining items in the session. And for session-level unlearning request, SRU is equipped with CED due to the high sparsity of the real dataset. Then we have retrained corresponding sub-model and the aggregation module for efficient unlearning. We have utilized a similarity-based session partition module and an attentive aggregation module to improve the recommendation performance in SRU. Besides, we have further defined an evaluation metric and adopt MIA to evaluate the unlearning effectiveness of the session-based recommendation. We have implemented SRU with three representative session-based recommendation models and conducted experiments on three benchmark datasets. Experimental results have demonstrated the superiority of our proposed methods. For future work, we plan to investigate the trade-off between unlearning effectiveness, recommendation performance, and unlearning efficiency which is also an interesting future topic.

REFERENCES

[1] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 141–159.

[2] Jonathan Brophy and Daniel Lowd. 2021. Machine unlearning for random forests. In *International Conference on Machine Learning*. PMLR, 1092–1104.

[3] Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*. IEEE, 463–480.

[4] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting Training Data from Large Language Models. *CoRR* abs/2012.07805 (2020). arXiv:2012.07805

[5] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*. 2768–2777.

[6] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models *(CCS '20)*. Association for Computing Machinery, New York, NY, USA.

[7] Kongyang Chen, Yiwen Wang, and Yao Huang. 2021. Lightweight machine unlearning in neural network. arXiv:2111.05528 [cs.LG]

[8] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph Unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. ACM. https://doi.org/10.1145/3548606.3559352

[9] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-Based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (Como, Italy) *(RecSys '17)*. Association for Computing Machinery, New York, NY, USA, 152–160. https://doi.org/10.1145/3109859.3109877

[10] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. 2020. Certified Data Removal from Machine Learning Models. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 3832–3842.

[11] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. 2021. Adaptive Machine Unlearning. arXiv:2106.04378 [cs.LG]

[12] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-Based Recommendation *(RecSys '17)*. Association for Computing Machinery.

[13] Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 191–200. https://doi.org/10.1109/ICDM.2016.0030

[14] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM. https://doi.org/10.1145/3269206.3271761

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[16] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement Learning to Rank in E-Commerce Search Engine: Formalization, Analysis, and Application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[17] Erin Illman and Paul Temple. 2019. California Consumer Privacy Act. *The Business Lawyer* 75, 1 (2019), 1637–1646.

[18] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. 2021. Approximate Data Deletion from Machine Learning Models. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 130)*, Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, 2008–2016.

[19] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[20] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Yuyuan Li, Xiaolin Zheng, Chaochao Chen, and Junlin Liu. 2022. Making Recommender Systems Forget: Learning and Unlearning for Erasable Recommendation. arXiv:2203.11491 [cs.IR]

[23] Wenyan Liu, Juncheng Wan, Xiaoling Wang, Weinan Zhang, Dell Zhang, and Hang Li. 2022. Forgetting Fast in Recommender Systems. arXiv:2208.06875 [cs.IR]

[24] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive Self-supervised Sequential Recommendation with Robust Augmentation. arXiv:2108.06479 [cs.IR]

[25] Zhuo Ma, Yang Liu, Ximeng Liu, Jian Liu, Jianfeng Ma, and Kui Ren. 2023. Learn to Forget: Machine Unlearning via Neuron Masking. *IEEE Transactions on Dependable and Secure Computing* 20, 4 (2023), 3194–3207. https://doi.org/10.1109/TDSC.2022.3194884

[26] Alessandro Mantelero. 2013. The EU Proposal for a General Data Protection Regulation and the roots of the 'right to be forgotten'. *Computer Law & Security Review* 29, 3 (2013), 229–235.

[27] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-Basket Recommendation *(WWW '10)*. Association for Computing Machinery.

[28] Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. 2023. Exploring the Landscape of Machine Unlearning: A Comprehensive Survey and Taxonomy. arXiv:2305.06360 [cs.LG]

[29] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *Journal of Machine Learning Research* 6, 43 (2005), 1265–1295.

[30] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.

[31] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[32] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. arXiv:1809.07426 [cs.IR]

[33] Enayat Ullah, Tung Mai, Anup Rao, Ryan Rossi, and Raman Arora. 2021. Machine Unlearning via Algorithmic Stability. arXiv:2102.13179 [cs.LG]

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[35] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *Proceedings of the ACM Web Conference 2023* (Austin, TX, USA) *(WWW '23)*. Association for Computing Machinery, New York, NY, USA, 651–661. https://doi.org/10.1145/3543507.3583521

[36] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 346–353. https://doi.org/10.1609/aaai.v33i01.3301346

[37] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive Learning for Sequential Recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 1259–1273. https://doi.org/10.1109/ICDE53745.2022.00099

[38] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2020. Self-supervised reinforcement learning for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 931–940.

[39] Heng Xu, Tianqing Zhu*, Lefeng Zhang, Wanlei Zhou, and Philip S. Yu. 2023. Machine Unlearning: A Survey. *ACM Comput. Surv.* (jun 2023). https://doi.org/10.1145/3603620 Just Accepted.

[40] Mimee Xu, Jiankai Sun, Xin Yang, Kevin Yao, and Chong Wang. 2023. Netflix and Forget: Efficient and Exact Machine Unlearning from Bi-linear Recommendations. arXiv:2302.06676 [cs.LG]

[41] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*.

[42] Jianming Zheng, Fei Cai, Jun Liu, Yanxiang Ling, and Honghui Chen. 2024. Multistructure Contrastive Learning for Pretraining Event Representation. *IEEE Transactions on Neural Networks and Learning Systems* 35, 1 (2024), 842–854. https://doi.org/10.1109/TNNLS.2022.3177641

[43] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 1893–1902. https://doi.org/10.1145/3340531.3411954

[44] Zhihao Zhu, Chenwang Wu, Rui Fan, Defu Lian, and Enhong Chen. 2023. Membership Inference Attacks Against Sequential Recommender Systems. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. 1208–1219.