Streaming Session-Based Recommendation: When Graph Neural Networks meet the Neighborhood

SARA LATIFI, University of Klagenfurt, Austria
DIETMAR JANNACH, University of Klagenfurt, Austria

Frequent updates and model retraining are important in various application areas of recommender systems, e.g., news recommendation. Moreover, in such domains, we may not only face the problem of dealing with a constant stream of new data, but also with anonymous users, leading to the problem of *streaming session-based recommendation* (SSR). Such problem settings have attracted increased interest in recent years, and different deep learning architectures were proposed that support fast updates of the underlying prediction models when new data arrive. In a recent paper, a method based on *Graph Neural Networks* (GNN) was proposed as being superior than previous methods for the SSR problem. The baselines in the reported experiments included different machine learning models. However, several earlier studies have shown that often conceptually simpler methods, e.g., based on nearest neighbors, can be highly effective for session-based recommendation problems. In this work, we report a similar phenomenon for the streaming configuration. We first reproduce the results of the mentioned GNN method and then show that simpler methods are able to outperform this complex state-of-the-art neural method on two datasets. Overall, our work points to continued methodological issues in the academic community, e.g., in terms of the choice of baselines and reproducibility.[1]

CCS Concepts: • **Information systems → Recommender systems**.

Additional Key Words and Phrases: Stream-based Recommendation, Session-based Recommendation, Incremental Updates, Reproducibility, Evaluation

ACM Reference Format:
Sara Latifi and Dietmar Jannach. 2022. Streaming Session-Based Recommendation: When Graph Neural Networks meet the Neighborhood. In *Sixteenth ACM Conference on Recommender Systems (RecSys '22), September 18–23, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3523227.3548485

1 INTRODUCTION

Session-based recommendation problems have attracted increased research interest in recent years. In this subclass of sequence-aware recommender systems [26] that is highly relevant in practice, the goal is to recommend items that match the short-term interests of a user in an ongoing session. A number of neural models for session-based recommendation were proposed in the past few years, see [28]. A common assumption when deploying such models in practice is that the models are periodically retrained to accommodate newly collected data. However, there are different application domains where it is desirable to update the underling models frequently. In the news domain, for example, it is highly important to consider new articles for recommendation almost immediately after their publication [21]. Similarly, considering recent intra-day trends of consumer behavior on e-commerce sites has proven to be beneficial in [13].

[1]Code and data: https://github.com/saraLatifi/SSR.

Manuscript submitted to ACM

This need for frequent model updates and the often computationally high costs of re-training complex models recently led to the design of *streaming* session-based recommendation (SSR) algorithms, e.g., [9, 14, 25]. Conceptually, these algorithms process a constant stream of new usage sessions, which they incorporate into the existing model without full retraining. SSR methods are therefore related to the class of techniques that support *online* machine learning and incremental updates [1, 3, 11].

In their recent work, Qiu et al. [25] proposed a novel method based on Graph Neural Networks (GNN). Their experiments indicate that their method is able to outperform an earlier neural stream-based method presented in [9] as well as other neural and non-neural machine learning models. However, a number of recent works in the area of session-based recommendation indicate that the most recent deep learning models are often not better than conceptually simpler methods based on, e.g., nearest neighbors or simple association rules [20]. Similarly, a number of other works exist—both in the area of recommender systems and other domains such as information retrieval or time-series forecasting [2, 6, 20, 22, 27, 31]—which indicate that it is not uncommon that established non-neural baselines are overlooked by researchers, leading to what could be seen as an "illusion of progress" [10].

In this work, we examine if this phenomenon also exists for recent SSR approaches. For this purpose, we reproduce the results reported in [25] and benchmark the GNN-based method against an effective session-based nearest neighbor method. The considered method was originally proposed in [19], extended to consider previous sessions of the current user in [15] and adapted for online learning problem in this work. Our results on the two datasets that were used for the evaluation also in [9, 25] indeed show that the conceptually simple method outperforms the GNN-based method both in terms of Hit Ratio and the MRR. Our work therefore emphasizes continuing methodological issues in this research area and calls for the consequent inclusion of well-tuned established baseline methods instead of only considering the latest neural models.

2 RELATED WORK

Work on SSR generally falls into the category of online machine learning or methods that support incremental updates. The literature in this broader area is very rich. Therefore, we here only review works that focus on the SSR problem.

In [14], Jugovac et al. implemented a replay-based evaluation protocol for updating an underlying recommendation model with new events and articles in the news domain in real-time. While their nearest-neighbor model is able to effectively capture the recent interests of a user, it was not designed to remember historical interactions of the current user. Later on, FlowRec [23] was proposed, a recommendation framework for streaming session data developed on top of scikit-multiflow. FlowRec contains different streaming models for session-based recommendations, including nearest-neighbor models as proposed in [14]. Again, however, these models are not designed to model long-term preference information. A deep learning-based approach for session-based recommendation in the news domain was proposed in [7]. This hybrid model is able to leverage side information about the news articles and supports incremental model updates.

The model by Guo et al. [9], in contrast to the works discussed so far, is able to leverage long-term preference information. To process the streaming sessions, it technically uses a reservoir technique with a weighted sampling scheme by assessing the informativeness of each session. Specifically, they proposed *(i)* a matrix factorization based attention model to capture the main intention of users from their historical interactions, *(ii)* a hybrid session-based recommender to model both the long-term and short-term preferences of a user, and *(iii)* a reservoir-based technique to tackle the large volume of the streaming data and an active sampling strategy to tackle its high velocity. We could not

include this related method in our experiments, as the authors were unfortunately unable to recover the code of their model, which was published at KDD '19.

Later, at SIGIR '20, Qiu et al. [25] proposed the Global Attributed Graph (GAG) neural network model with a Wasserstein reservoir, which aims to preserve a representative sketch of the historical data. The experiments in [25] indicate their method outperforms the earlier method by Guo et al., and we therefore use the GAG model as a representative baseline model in our experiments.

Around the same time, Xu et al. [30] developed their "GraphSAIL" framework in which they introduced three general components for incrementally training a GNN-based recommendation model: *(i)* a local structure distillation mechanism to preserve a user's long-term preference and an item's long-term characteristics, *(ii)* a global structure distillation strategy to encode the global position for each user and item node, and *(iii)* a general degree-aware self-embedding distillation component to regularize the user and item embedding learned. The authors unfortunately do not share the code of their model, which is why it is not included in our experiments.

Finally, a number of approaches for other streaming-based recommendation scenarios were proposed, for example in the context of traditional recommendation setups [29], time-aware settings [4, 33] and for sequential recommendation problems [24, 32]. A comparison of these works is beyond the scope of our work which focuses on streaming session-based recommendation.

3 EXPERIMENT DESIGN

The relevant baseline in our experiment is the recent GAG model mentioned above, and we *exactly* reproduce the experimental setting of [25], i.e., we use the identical evaluation protocol, the same metrics and data splits, and the code provided by the authors.

3.1 Compared Algorithms

We compare GAG with two non-neural session-based approaches from the literature, which were *(i)* extended to consider the past sessions of individual users as proposed in [15], and which we *(ii)* adapted to support online updates to the internal data structures and counting statistics.

GAG. Like the earlier work by Guo et al. [9], the GAG model [25] uses a reservoir technique for the streaming task. Moreover, GAG implements different mechanisms to overcome potential shortcomings of [9] such as the need for informativeness scores for every item in a session. To better model the "complicated correlations" between users and items, a graph-based approach is proposed. Essential to the model is the conversion of a user's session sequence into a session graph, where the user embeddings are associated as a global attribute to the embeddings of the interacted items, thereby enabling the model to maintain long-term user preferences. The global attribute is subsequently considered in the graph convolution process. As another technical contribution, the authors propose to use a "Wasserstein" reservoir to select the most informative training cases for updating the model.

VSKNN+. The VSKNN model [18] is a session-based nearest neighbor approach[2]. It first locates past sessions that are similar to the current one, i.e., sessions that contain interactions with the same items. Items that appear in such similar sessions are then scored as recommendation candidates by considering the similarity of the sessions.

[2]Our implementation is based on the code shared by Ludewig et al. at https://github.com/rn5l/session-rec/

More formally, the basic session-based kNN method SKNN from [18] can be summarized as follows. Given a session s, a similarity function for sessions $sim(s_1, s_2)$, and a corresponding set N_s neighbors of s, the score of an item i for a given session s can be computed as:

$$score_{kNN}(i, s) = \sum_{n \in N_s} sim(s, n) \times 1_n(i) \tag{1}$$

The specific similarity function in VSKNN puts more emphasis on overlaps in more recent interactions. The scalability of the method can be ensured with the help of specific in-memory data structures and neighbor sampling [12].

Since VSKNN does not consider past sessions of the current user, the method was recently extended in [15] with three heuristics: *(i)* extending the current session with interactions from previous sessions of the user up to a certain threshold (EXTEND), *(ii)* increasing the obtained VSKNN scores of items that the current user has previously seen by a certain percentage (BOOST), and *(iii)* applying reminding techniques (REMIND) from [16]. This turns VSKNN into a *session-aware* method [26], and the results in [15] showed that the proposed extensions help to outperform recent neural approaches to session-aware recommendations. In our present work, we consider the choice and combination of the three extension as a hyper-parameter to the method, which we name VSKNN+ in the following.

SR+. The SR (Sequential Rules) method was also proposed in [18]. It simply consists of counting item co-occurrences in sessions, where the order and the distance of the considered items is taken into account when scoring the items. Formally, a session s is considered a chronologically ordered tuple of item interaction events $s = (s_1, s_2, s_3, \ldots, s_m)$ and S_p the set of all past sessions. A user's current session is denoted as s, with $s_{|s|}$ being the last item interaction in s. A rule is created when an item q appeared after an item p in a session, even when there are other events that happened between p and q. The weight of a rule is based on the number of items between p and q; the corresponding weight function is $w_{\text{SR}}(x) = 1/(x)$, where x is the number of steps between the p and q.

Finally, the score for a recommendable item i for a given session s is as follows, where the indicator function $1_{\text{EQ}}(a, b)$ is 1 in case a and b refer to the same item and 0 otherwise [18].

$$score_{\text{SR}}(i, s) = Q \times \sum_{p \in S_p} \sum_{x=2}^{|p|} \sum_{y=1}^{x-1} 1_{\text{EQ}}(s_{|s|}, p_y) \cdot 1_{\text{EQ}}(i, p_x) \cdot w_{\text{SR}}(x - y) \tag{2}$$

where Q serves as a normalization factor:

$$Q = \frac{1}{\sum_{p \in S_p} \sum_{x=2}^{|p|} 1_{\text{EQ}}(s_{|s|}, p_x) \cdot x} \tag{3}$$

As the original SR method is a session-based approach, the SR+ method used here incorporates the BOOST and REMIND heuristics from [15] mentioned above to consider past sessions of a user.

HYBRID. We tested various ways of combining VSKNN+ and SR+ in a hybrid model. In this work, we report the results of a rather trivial combination of the outputs of the two models, where we first return the first n items—with n being a hyper-parameter in {5,10,15}—recommended by SR+ and then append items from the recommendation list of VSKNN+ without duplicates. Having the SR+ recommendations at the beginning is motivated by the fact that the recommendations by SR often led to good MRR results in past work [20].

Alternative Baselines. The authors of GAG considered a number of alternative baselines in their experiments, including trivial models based on item popularity and neural models like NARM [17]. Since these methods were all outperformed

by GAG, we did not include these baselines in our own experiments. For comparison, in [25], the performance of GAG in terms of recall was about 40-80% better than the best popularity-based method and about 5-10% better than the best neural method for each dataset.

3.2 Datasets

We conducted the experiments with the public GOWALLA[3] and LASTFM[4] datasets that were also used in [9] and [25]. The GOWALLA dataset contains check-in information from a location-based social network, and the LASTFM dataset contains information about music listening sessions. In our experiments, we used the pre-processed versions of the datasets as shared by the authors of GAG. In the GOWALLA dataset, only the 30,000 most popular locations were retrained, and events that happened on the same day were considered to be in the same session. For the LASTFM dataset, where artist recommendation is in the focus, the 10,000 most popular artists were taken into account. Interactions by a user that were observed within an 8 hour window were considered a session. For both datasets, sessions containing only one interaction or more than 20 interactions were discarded.

Table 1 shows summary statistics for the two pre-processed datasets.

Table 1. Characteristics of pre-processed datasets. #Ints.: Nb. of interactions, #U: Nb. of users, #S: Nb. of sessions, #I: Nb. of items, Avg. Length: Average session length.

Dataset	#Ints.	#U	#S	#I	Avg. Length
GOWALLA	645K	33K	198.5K	28.7K	3.2
LASTFM	2.1M	1K	298.9K	10K	6.9

3.3 Evaluation Protocol

We simulate the streaming session-based scenario as done in [9, 25], where the dataset is first divided into two parts in chronological order: *(i)* the first 60% of data is used to initially train the models; *(ii)* the second 40% of data is called candidate set and used to simulate the streaming setting. The candidate set is further divided by time into five blocks of the same size, i.e., 8% of the whole dataset.

The last 10% of the training set and the first block of the candidate set are first used as a validation set to tune the hyper-parameters. Now, the last 8% of this data (i.e., the 60%-68% fraction of the overall data), which were also just used for tuning, are provided to the trained GAG model, which is incrementally updated based on this new information. In the next step, the model is evaluated on the next 8% of the data (i.e., the 68%-76% part of the overall data). This part is then again provided to the model, which is updated (without full retraining) and evaluated on the next 8%. This process repeats until the 92%-100% fraction of the data has been used for evaluation. Overall, this gives us 4 update-and-evaluate steps. Figure 1 illustrates this protocol.

To evaluate the model's performance for a given session in the test data, we also follow the protocol used in [25] and incrementally reveal one interaction (item) after the other and let the model make a prediction regarding the next item in the session. As performance measures, we use the Hit Ratio (HR) and the Mean Reciprocal Rank (MRR) as done in previous works. Note that Recall was reported in [25], which is however equivalent to the more commonly used HR metric for session-based recommendation when there is only one positive item for each measurement.

[3]https://snap.stanford.edu/data/loc-gowalla.html
[4]http://mtg.upf.edu/static/datasets/last.fm/lastfm-dataset-1K.tar.gz

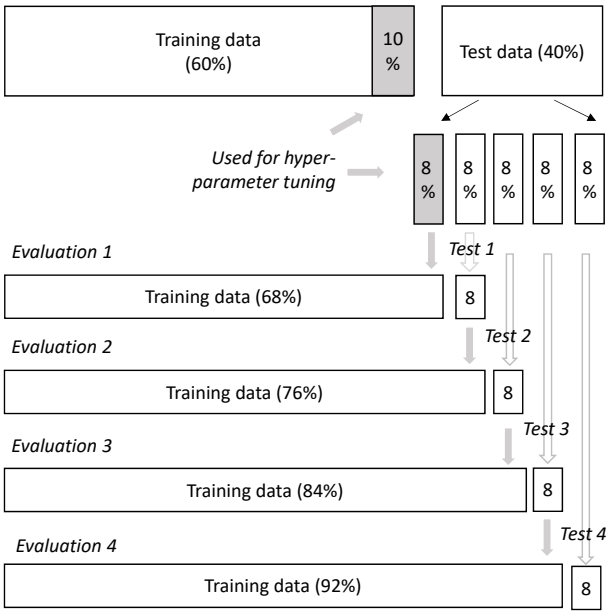

Fig. 1. Overview of the evaluation protocol. At each step, the model is updated (without full retraining) based on the new information added to the training data.

3.4 Hyper-Parameter Tuning

For the GAG model, we used the hyper-parameters according to the information from the original paper and the shared source code[5]. This is appropriate, as we used the exact same code, data, and protocol from their experiments. Interestingly, the authors only reported one set of hyper-parameters which they apparently used for the evaluation of both datasets. Using the reported hyper-parameter set, we could reproduce the results reported in [25] for the GOWALLA dataset. However, for the LASTFM dataset, we observed different (lower) values than reported in [25]. We were then in exchange with the authors of GAG, but they could not recover the hyper-parameter values that led to the their reported results. We therefore manually tuned the hyper-parameters of the GAG model further on the LASTFM dataset, e.g., by reducing the learning rate, until we observed HR values that were similar to those in the original paper. Reducing the learning rate also helped to further improve the performance on the GOWALLA dataset. Later in the paper, we report both outcomes, i.e., the results when using the original provided hyper-parameters and the ones that led to further improvements in our own experiments.

For the VSKNN+ and SR+ methods, we used a random hyper-parameter search with 100 iterations on the same validation set as for the GAG model. As a range for the hyper-parameters we relied on the ranges reported in [15]. As an optimization target, we used MRR@20 for SR+ and HR@20 for VSKNN+. The final hyper-parameters for all compared models are reported in the appendix.

4 RESULTS AND DISCUSSION

Here, we first review our main results in Section 4.1 and then discuss the findings in more detail in Section 4.2.

[5]https://github.com/RuihongQiu/GAG

4.1 Main Results

Table 2 shows the main results[6] of our experiments for both datasets. The row HYBRID shows the results for the described combination of VSKNN+ and SR+. We did not test the hybrid for the LASTFM dataset due to the relative weak performance of SR+ on this dataset. The row labeled with GAG* shows the results achieved with our own optimized set of hyper-parameters of the GAG model, with performance values slightly higher than the values achieved with the reported hyper-parameters in the original paper.

Table 2. Results for both datasets, with best performance numbers printed in bold. We also report the relative improvements of our methods over GAG* for each test set.

	GOWALLA		LASTFM	
Metrics	HR@20	MRR@20	HR@20	MRR@20
Test set 1 (68%-76%)				
HYBRID	**0.520**	**0.249**	–	–
VSKNN+	0.516	0.165	**0.315**	**0.103**
GAG*	0.475	0.224	0.295	0.095
GAG	0.454	0.212	0.282	0.090
SR+	0.414	0.231	0.145	0.063
IMPROV.	**9.47%**	**11.16%**	**6.78%**	**8.42%**
Test set 2 (76%-84%)				
HYBRID	**0.549**	**0.265**	–	–
VSKNN+	0.548	0.175	**0.308**	**0.099**
GAG*	0.503	0.238	0.290	0.093
GAG	0.477	0.224	0.279	0.090
SR+	0.440	0.247	0.145	0.065
IMPROV.	**9.15%**	**11.34%**	**6.21%**	**6.45%**
Test set 3 (84%-92%)				
HYBRID	**0.584**	**0.285**	–	–
VSKNN+	0.581	0.189	**0.302**	**0.096**
GAG*	0.534	0.252	0.283	0.087
GAG	0.510	0.236	0.272	0.084
SR+	0.478	0.269	0.142	0.060
IMPROV.	**9.36%**	**13.10%**	**6.71%**	**10.34%**
Test set 4 (92%-100%)				
HYBRID	**0.571**	**0.276**	–	–
VSKNN+	0.568	0.178	**0.313**	**0.100**
GAG*	0.539	0.256	0.296	0.093
GAG	0.509	0.233	0.278	0.087
SR+	0.484	0.265	0.149	0.064
IMPROV.	**5.94%**	**7.81%**	**5.74%**	**7.53%**

Overall, we find that GAG is consistently outperformed by either VSKNN+ or by a combination of VSKNN+ and SR+ (HYBRID) on both datasets and on both performance measures. The gains are between 5-13%. Looking closer at the results, we see some differences across the datasets. For the GOWALLA dataset, for example, SR+ works particularly well in terms of the MRR. VSKNN+, in contrast, excels in terms of the HR, and the hybrid combination proved to be helpful to obtain superior performance values in both measures.

For the LASTFM dataset, SR+ as mentioned did not lead to competitive results, which is why we also did not try the HYBRID method here. We see the reasons for the limited performance of SR+ in the particularities of the application

[6]We made a number of additional experiments not reported here, e.g., where we use VSKNN in its original form without considering past sessions of the current users. Since these other experiments led to worse results, we do not list them here.

domain and, thus, the dataset. Remember that the GOWALLA dataset contains check-in information of a social network, and we assume that consecutive check-in events may be largely influenced by geographical vicinity. Therefore, we might observe many similar check-in patterns in a smaller area, see [5] for an analysis. In the music domain and the LASTFM dataset, in contrast, observing very frequent patterns of consecutive tracks seems less likely.

Considering the results from both datasets, we find that the GNN model GAG is able to model the sequential patterns in the data well and that the incorporation of newly incoming data is effective. It however does not reach the performance levels of the simpler methods. The reasons for this are difficult to isolate, and more research seems therefore required to further improve the generally very promising neural network model.

4.2 Discussion

The reproducibility study and performance comparison presented in this paper continues a series of earlier studies mentioned above, which indicate that the progress we achieve with complex models is often smaller than we would assume. This is true also for papers that are published at highly competitive venues. In the case of streaming session-based recommendation, we again find one possible reason for this effect: researchers sometimes tend to mostly consider the latest neural methods as their main baselines but do not run simpler methods first to gauge the effectiveness of their models. Nonetheless, in the case of SSR models our result seems a bit surprising, given that several works were published in the last few years, e.g., [8, 12, 15], which highlighted the often competitive performance of k-nearest-neighbor (kNN) methods for session-based recommendation scenarios. Actually, both the authors of GAG and SSRM mention an earlier work in [14], which indicates potential advantages of kNN methods for streaming scenarios. However, they do not include such methods in their experiments because these methods were not designed to consider long-term preference information. However, as shown, these kNN methods can be extended to support online updates and to consider longer-term preference information and to thereby considerably improve their performance.

Reproducibility of published research also seems to be an open issue in this area. As our discussion of previous work shows, it is not a standard practice for authors to share their code and data. The authors of GAG are a positive exception here, and we are thankful that they also shared the pre-processed datasets with us. Given that the original code is often not available, authors may sometimes simply copy the results from an earlier paper without running the code by themselves. Such an approach is however only appropriate if we are sure that the evaluation protocol and the implementation of the metrics are identical. However, in our experience, there are often small differences in how things are implemented, which can easily make such a comparison unreliable.

Another methodological question that we see in current research in SSR is that the choice of the evaluation datasets often is not very well motivated. As we discussed earlier, a typical use case for stream-based systems in our view are news recommender systems or certain e-commerce applications. While we agree that music is often listened to in individual sessions and that there might be certain short-term community trends (e.g., songs going viral overnight) that one wants to consider immediately. For the prediction of check-ins in the context of location-based services, the need for very frequent, e.g., intra-day, model updates is not immediately clear. In our present work, we did not yet evaluate the simple methods on other and probably more relevant datasets. Such an investigation is part of our future work but goes beyond our current goal to benchmark existing neural methods in the exact same setting as they were published. Likewise, we did not analyze the performance of the methods at different list lengths or using other accuracy and beyond-accuracy measures such as diversity or novelty.

5 SUMMARY AND OUTLOOK

In several application domains of session-based recommendation it is desirable to be able to update the underlying models frequently and without the need to fully retrain them from scratch. Recent research in this area led to a number of proposals of deep learning architecture to address this problem. Our work however shows that current models can be outperformed by conceptually simpler techniques. Therefore, we conclude that future research should more often consider such simpler baselines in their experiments. Overall, however, we also believe there is huge potential for complex models in this area. In our future work, we plan to evaluate these simple methods on other datasets, and we also plan to include other recent models in our evaluation, which were published—although without public code yet—after we started this research.

APPENDIX: HYPERPARAMETER SETTINGS

The optimal hyper-parameters for our experiments can be found in Table 3. Regarding the HYBRID method, we found that a *mixed* combination of SR+ and VSKNN+ led to the best results. In this mixed approach, the final recommendation list is obtained by prepending the first five items recommended by the SR+ model to the recommendation list generated by the VSKNN+ model without duplicates.

Table 3. Optimal hyper-parameters for each method on each dataset.

Hyper-parameter	GOWALLA	LASTFM
VSKNN+		
sampling	random	random
k	500	100
sample_size	5000	5000
weighting	log	same
weighting_score	div	linear
idf_weighting	False	10
extend_session_length	12	None
boost_own_sessions	2.7	2.7
SR+		
steps	11	3
weighting	quadratic	quadratic
boost_own_sessions	3.9	3.9
GAG		
hidden_size	200	200
lr	0.003	0.003
l2	0.00001	0.00001
res_size	100	100
win_size	1	1
GAG*		
hidden_size	200	200
lr	0.001	0.001
l2	0.00001	0.00001
res_size	100	100
win_size	1	1

REFERENCES

[1] Marie Al-Ghossein, Talel Abdessalem, and Anthony BARRÉ. 2022. A Survey on Stream-Based Recommender Systems. *ACM Comput. Surv.* 54, 5 (2022).

[2] Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. 2009. Improvements That Don't Add Up: Ad-hoc Retrieval Results Since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*. 601–610.

[3] András A. Benczúr, Levente Kocsis, and Róbert Pálovics. 2019. *Online Machine Learning Algorithms over Data Streams.* Springer International Publishing, 1199–1207.

[4] Shiyu Chang, Yang Zhang, Jiliang Tang, Dawei Yin, Yi Chang, Mark A Hasegawa-Johnson, and Thomas S Huang. 2017. Streaming Recommender Systems. In *Proceedings of the 26th International Conference on World Wide Web*. 381–389.

[5] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. Friendship and Mobility: User Movement in Location-Based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. 1082–1090.

[6] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2020. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Transactions on Information Systems* 39 (2020), 1–49. Issue 2.

[7] P Moreira Gabriel De Souza, Dietmar Jannach, and Adilson Marques Da Cunha. 2019. Contextual Hybrid Session-based News Recommendation with Recurrent Neural Networks. *IEEE Access* 7 (2019), 169185–169203.

[8] Diksha Garg, Priyanka Gupta, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. Sequence and Time Aware Neighborhood for Session-based Recommendations: STAN. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19)*. 1069–1072.

[9] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. 2019. Streaming Session-based Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1569–1577.

[10] David J. Hand. 2006. Classifier Technology and the Illusion of Progress. *Statist. Sci.* 21, 1 (2006), 1–14.

[11] Steven C.H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. 2021. Online Learning: A Comprehensive Survey. *Neurocomputing* 459 (2021), 249–289.

[12] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-based Recommendation. In *In Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*. 306–310.

[13] Dietmar Jannach, Malte Ludewig, and Lukas Lerche. 2017. Session-based Item Recommendation in E-Commerce: On Short-Term Intents, Reminders, Trends, and Discounts. *User-Modeling and User-Adapted Interaction* 27, 3–5 (2017), 351–392.

[14] Michael Jugovac, Dietmar Jannach, and Mozhgan Karimi. 2018. StreamingRec: A Framework for Benchmarking Stream-based News Recommenders. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. 269–273.

[15] Sara Latifi, Noemi Mauro, and Dietmar Jannach. 2021. Session-aware Recommendation: A Surprising Quest for the State-of-the-art. *Information Sciences* 573 (2021), 291–315.

[16] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. 2016. On the Value of Reminders within E-Commerce Recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization (UMAP '16)*. 27–35.

[17] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 1419–1428.

[18] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User Modeling and User-Adapted Interaction* 28, 4-5 (2018), 331–390.

[19] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*. 462–466.

[20] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2021. Empirical Analysis of Session-based Recommendation Algorithms. *User Modeling and User-Adapted Interaction* 31, 1 (2021), 149–181.

[21] Cornelius A. Ludmann. 2017. Recommending News Articles in the CLEF News Recommendation Evaluation Lab with the Data Stream Management System Odysseus. In *Working Notes of CLEF 2017 - Conference and Labs of the Evaluation Forum (CEUR Workshop Proceedings, Vol. 1866)*. CEUR-WS.org.

[22] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PlOS ONE* 13, 3 (2018).

[23] Dimitris Paraschakis and Bengt J Nilsson. 2020. FlowRec: Prototyping Session-based Recommender Systems in Streaming Mode. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 65–77.

[24] Danni Peng, Sinno Jialin Pan, Jie Zhang, and Anxiang Zeng. 2021. Learning an Adaptive Meta Model-Generator for Incrementally Updating Recommender Systems. In *15th ACM Conference on Recommender Systems (RecSys '21)*. 411–421.

[25] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. GAG: Global Attributed Graph Neural Network for Streaming Session-based Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 669–678.

[26] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *Comput. Surveys* 51 (2018), 1–36. Issue 4.

[27] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *14th ACM Conference on Recommender Systems (RecSys '20)*. 240–248.

[28] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian. 2021. A Survey on Session-based Recommender Systems. *Comput. Surveys* 54, 7 (2021), 1–38.

[29] Weiqing Wang, Hongzhi Yin, Zi Huang, Qinyong Wang, Xingzhong Du, and Quoc Viet Hung Nguyen. 2018. Streaming Ranking Based Recommender Systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 525–534.

[30] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. 2020. Graphsail: Graph Structure Aware Incremental Learning for Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2861–2868.

[31] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. 2019. Critically Examining the Neural Hype: Weak Baselines and the Additivity of Effectiveness Gains from Neural Ranking Models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*. 1129–1132.

[32] Yang Zhang, Fuli Feng, Chenxu Wang, Xiangnan He, Meng Wang, Yan Li, and Yongdong Zhang. 2020. How to Retrain Recommender System? A Sequential Meta-learning Method. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. 1479–1488.

[33] Yan Zhao, Shoujin Wang, Yan Wang, and Hongwei Liu. 2021. Stratified and Time-aware Sampling Based Adaptive Ensemble Learning for Streaming Recommendations. *Applied Intelligence* 51, 6 (2021), 3121–3141.