Performance comparison of neural and non-neural approaches to session-based recommendation

Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation

Malte Ludewig
TU Dortmund, Germany
malte.ludewig@tu-dortmund.de

Noemi Mauro
University of Torino, Italy
noemi.mauro@unito.it

Sara Latifi
University of Klagenfurt, Austria
sara.latifi@aau.at

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

ABSTRACT

The benefits of neural approaches are undisputed in many application areas. However, today's research practice in applied machine learning—where researchers often use a variety of baselines, datasets, and evaluation procedures—can make it difficult to understand how much progress is actually achieved through novel technical approaches. In this work, we focus on the fast-developing area of session-based recommendation and aim to contribute to a better understanding of what represents the state-of-the-art.

To that purpose, we have conducted an extensive set of experiments, using a variety of datasets, in which we benchmarked four neural approaches that were published in the last three years against each other and against a set of simpler baseline techniques, e.g., based on nearest neighbors. The evaluation of the algorithms under the exact same conditions revealed that the benefits of applying today's neural approaches to session-based recommendations are still limited. In the majority of the cases, and in particular when precision and recall are used, it turned out that simple techniques in most cases outperform recent neural approaches. Our findings therefore point to certain major limitations of today's research practice. By sharing our evaluation framework publicly, we hope that some of these limitations can be overcome in the future.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Session-based Recommendation; Evaluation; Reproducibility

ACM Reference Format:
Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3347041

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3347041

1 INTRODUCTION

In recent years, we could observe an increased research interest in *session-based* recommendation problems. In such settings, the problem is not to make relevance predictions for items given the users' long-term preferences, but to make recommendations given only a few user interactions in an ongoing session [19]. While such scenarios have been addressed in the literature previously, e.g., for web usage prediction [18], they have recently received more attention, e.g., due to the availability of public datasets.

From a technical perspective, almost all session-based algorithms proposed in recent years are based on deep learning ("neural") architectures. A landmark work in this area is the GRU4REC method, which is based on Recurrent Neural Networks (RNNs) [4, 5]. Today, GRU4REC is often used as a baseline algorithm in experimental evaluations. However, recent research [7, 15] indicates that simpler methods based on nearest-neighbor techniques can outperform GRU4REC in terms of certain accuracy measures. Therefore, when new neural algorithms are published and benchmarked against GRU4REC alone, it is not clear whether or not these new methods are actually leading to progress beyond the more simple techniques.

This problem of unclear progress in applied machine learning is not entirely new. In the information retrieval (IR) field, for example, researchers already found in 2009 that the improvements reported over the years "don't add up" [1]. Recent analyses [10, 16] furthermore indicate that some neural approaches that were recently published at top conferences do not outperform long-established baseline methods, when these are well tuned. The reasons for this non-progress lie in the choice of the baselines used in the experimental evaluations or the limited efforts by the authors to fine-tune the baselines. Sometimes, another problem is the lack of reproducibility of the results. Today, publishing the code of the algorithms is more and more encouraged. However, often the code used for data pre-processing, data splitting, hyper-parameter optimization, and evaluating is not provided. Given that many of these implementation details can affect accuracy, it is often very challenging to make reliable conclusions.

With this work, our goal is to shed light on the progress in the area of session-based recommendation algorithms. We report the results of an in-depth comparison of four recent neural algorithms and a set of mostly simpler baseline algorithms. All algorithms were benchmarked under identical settings within an evaluation framework that we built upon the code from [5]. Our results indicate that the progress that is achieved with neural approaches is sometimes

very limited, and that well-tuned baselines often outperform even the latest complex models.

Generally, these observations call for improved research practices, as discussed previously in [12]. The availability of an evaluation environment for reproducible research can be one piece of this puzzle. We therefore publicly share our evaluation framework, which includes also code for data splitting, hyper-parameter optimization and a number of additional metrics.

2 BENCHMARKED ALGORITHMS

We have considered the four neural approaches shown in Table 1 in our comparison. We selected them by systematically scanning the proceedings of top-ranked conference series of the last three years. We only included works for which the source code was available and which did not use side information.

Table 1: Neural Recommendation Strategies

GRU4REC (ICLR'16, CIKM'18)	GRU4REC [5] was the first neural approach that employed RNNs for session-based recommendation. The technique was later on improved using more effective loss functions [4].
NARM (CIKM'17)	This model extends GRU4REC and improves its session modeling with the introduction of an attention mechanism. This also proved to be advantageous in the NLP field [8].
STAMP (KDD'18)	In contrast to NARM, this model does not rely on an RNN anymore. Instead, the session is modeled solely with an attention mechanism in order to improve both effectiveness and efficiency [13].
NEXTITNET (WSDM'19)	This recent model also discards RNNs to model user sessions. In contrast to STAMP, convolutional neural networks are adopted with a few domain-specific enhancements [21].

As baselines we use the five techniques that were also used in [15], as well as a recent, more complex approach based on context trees (CT) [17]. All baselines methods shown in Table 2 have the advantage that they can take new interactions immediately into account without retraining, and they only have a small set of parameters to tune. Furthermore, scalability can be ensured for the neighborhood-based techniques through adequate sampling as discussed in [7]. We initially considered additional neural approaches such as [2, 9, 11, 14], but we did not include them in our evaluation for different reasons, e.g., because the source code was not available, or the algorithm also uses side information. We also did not consider *sequential* approaches like [3, 6, 20], because they are not really designed for session-based scenarios or require user IDs in the datasets.

Table 2: Baseline Strategies

AR	Learns and applies association rules of size two. Works by simply counting pairwise item co-occurrences in the training sessions.
SR	Similar to AR, but learns *sequential* rules of size two, i.e., it counts how often one item appeared after another (possibly with elements in between) in the training sessions.
S-KNN	A session-based nearest-neighbor technique. Every item in the session is assumed to be equally important when computing similarities.
VS-KNN	Like S-KNN, but uses a similarity function that puts more emphasis on the more recent events in a session.
CT	This technique is based on context trees. It is non-parametric and showed promising results in [17].

3 DATASETS AND EVALUATION APPROACH

3.1 Datasets

We conducted experiments with seven datasets, four from the e-commerce domain and three from the music domain, see Table 3. Six of these datasets are publicly available. These datasets were also used for the comparison of algorithms in [8, 15] and [13].

Table 3: Datasets

RSC15	E-commerce dataset used in the 2015 ACM RecSys Challenge.
RETAIL	An e-commerce dataset from the company Retail Rocket.
DIGI	An e-commerce dataset shared by the company Diginetica.
ZALANDO	A non-public dataset consisting of interaction logs from the European fashion retailer Zalando.
30MU	Music listening logs obtained from Last.fm.
NOWP	Music listening logs obtained from Twitter.
AOTM	A public music dataset containing music playlists.

Some previous works on session-based recommendation use a single training-test split in their evaluation or very small subsets of the original datasets (e.g., only 1/64 of the RSC15 dataset) [4, 5, 8, 13]. In our work, we followed the approach of [15] and created, for each dataset, five subsets contiguous in time to be able to make multiple measurements in order to minimize the risk of random effects. Table 4 shows the average characteristics of these multiple subsets. Pointers to the resulting datasets and the train-test splits used in the experiments can be found online[1], together with the code of our evaluation framework. For all datasets, we removed sessions that contained only one interaction.

Table 4: Characteristics of the datasets. The values are averaged over all five splits.

Dataset	RSC15	RETAIL	DIGI	ZALANDO	30MU	NOWP	AOTM
Actions	5.4M	210k	264k	4.5M	640k	271k	307k
Sessions	1.4M	60k	55k	365k	37k	27k	22k
Items	29k	32k	32k	189k	91k	75k	91k
Days covered	31	27	31	90	90	90	90
Actions/Session	3.95	3.54	4.78	12.43	17.11	10.04	14.02
Items/Session	3.17	2.56	4.01	8.39	14.47	9.38	14.01
Actions/Day	175k	8k	8.5k	50k	7k	3.0k	3.4k
Sessions/Day	44k	2.2k	1.7k	4k	300	243	243

3.2 Experimental Procedure

Hyper-Parameter Optimization. We tuned the hyper-parameters for all methods for each dataset systematically, using a subset of the training data—covering the same amount of days as the test set—for validation. As the training process can be time-consuming and the parameter space is large, we applied a random optimization approach with 100 iterations as in [4, 8, 13] (50 iterations for NARM) to find a suitable set of parameters. All models were optimized for the Mean Reciprocal Rank (MRR@20). The ranges and the final values of the hyper-parameters for each dataset can be found online.

Protocol and Metrics. Similar to [4, 5] and other works, we used the last n days of each dataset as test data and the rest for training. For each session in the test data, we incrementally "revealed" one interaction after the other. After each revealed interaction, we

[1] https://rn5l.github.io/session-rec/index.html

computed recommendation lists and then compared the recommendations with the still hidden elements in the session.

In [5], where GRU4REC was proposed, and in subsequent works, the evaluation procedure is based on measuring to what extent an algorithm is able to predict the *immediate next* item in a session. Their corresponding measurement of the Hit Rate (HR@20) and the MRR@20 is therefore based on the existence of this next item in a given top-n recommendation list. In reality, however, usually more than one item is shown and being able to identify more than one relevant item for a given session is typically favorable over just predicting the immediate next one correctly. In this work, we therefore focus on traditional precision, recall, and mean average precision (MAP) measures, which consider all items that appear in the currently hidden part of the session as relevant. As the neural approaches are not explicitly designed to predict multiple items and for the sake of completeness, we report both types of measurements.

4 RESULTS

E-Commerce Domain. Table 5 shows the results for the domain of e-commerce.[2] On the RETAIL and the DIGI dataset, the nearest neighbor methods led to the highest accuracy results—averaged across folds—on all measures. For the ZALANDO dataset, neighborhood methods were again best, except for the MRR. The differences to the best complex model are in many cases significant.

Only for the RSC15 dataset we can observe that a neural method (NARM) is able to consistently outperform our best baseline VS-KNN on all measures. Interestingly, however, it is one of the earlier neural methods in this comparison. The results for the RSC15 dataset are generally different from the other results. The CT method, for example, was very competitive on the MRR for this dataset. STAMP, while being a very recent method, was not among the top performers except for this dataset. Given these observations, it seems that the RSC15 dataset has some unique characteristics that are different from the other e-commerce datasets.

For the larger ZALANDO and RSC15 datasets, we do not include measurements for the most recent NEXTITNET method. We found that the method does not scale well and we could not complete the hyper-parameter tuning process within weeks on our machines (also for two music datasets).

Music Domain. Table 6 shows the results for the music domain. The results are mostly aligned with the e-commerce results. On all datasets, the nearest-neighbor methods outperform all other techniques on precision, recall, MAP, and the hit rate. In terms of the MRR measure, the non-neural CT method consistently leads to the highest values. The simple SR method is again competitive in terms of the MRR, and GRU4REC as well as NARM are again among the top-performing neural approaches. The neighborhood methods in all cases are not in the leading positions in terms of the MRR and even lead to the lowest MRR performance on the AOTM dataset. The STAMP method can consistently be found at the lower ranks in this comparison.

[2]The highest value across all techniques is printed in bold; the highest value obtained by the other family of algorithms—baseline or complex model—is underlined. Stars indicate significant differences according to a Student's t-test with Bonferroni correction between the best-performing techniques from each category. $*$: $p<0.05$, $**$: $p<0.01$.

Table 5: Results for e-commerce datasets. The best values obtained for complex models and baselines are highlighted.[2]

Metrics	MAP@20	P@20	R@20	HR@20	MRR@20
			RETAIL		
S-KNN	**0.0283**	**0.0532**	**0.4707**	**0.5788**	0.3370
VS-KNN	0.0278	0.0531	0.4632	0.5745	**0.3395**
GRU4REC	0.0272	0.0502	0.4559	0.5669	0.3237
NARM	0.0239	0.0440	0.4072	0.5549	0.3196
STAMP	0.0229	0.0428	0.3922	0.4620	0.2527
AR	0.0205	0.0387	0.3533	0.4367	0.2407
SR	0.0194	0.0362	0.3359	0.4174	0.2453
NEXTITNET	0.0173	0.0320	0.3051	0.3779	0.2038
CT	0.0162	0.0308	0.2902	0.3632	0.2305
			DIGI		
S-KNN	$**$**0.0255**	$*$**0.0596**	$**$**0.3715**	$*$**0.4748**	0.1714
VS-KNN	0.0249	0.0584	0.3668	0.4729	$**$**0.1784**
GRU4REC	0.0247	0.0577	0.3617	0.4639	0.1644
NARM	0.0218	0.0528	0.3254	0.4188	0.1392
STAMP	0.0201	0.0489	0.3040	0.3917	0.1314
AR	0.0189	0.0463	0.2872	0.3720	0.1280
SR	0.0164	0.0406	0.2517	0.3277	0.1216
NEXTITNET	0.0149	0.0380	0.2416	0.2922	0.1424
CT	0.0115	0.0294	0.1860	0.2494	0.1075
			ZALANDO		
VS-KNN	**0.0158**	**0.0740**	$**$**0.1956**	$**$**0.5162**	0.2487
S-KNN	0.0157	0.0738	0.1891	0.4352	0.1724
NARM	0.0144	0.0692	0.1795	0.4598	0.2248
GRU4REC	0.0143	0.0666	0.1797	0.4925	**0.3069**
SR	0.0136	0.0638	0.1739	0.4824	0.3043
AR	0.0133	0.0631	0.1690	0.4665	0.2579
CT	0.0118	0.0564	0.1573	0.4561	0.2993
STAMP	0.0104	0.0515	0.1359	0.3687	0.2065
			RSC15		
NARM	$**$**0.0357**	$**$**0.0735**	$**$**0.5109**	$*$**0.6751**	0.3047
STAMP	0.0344	0.0713	0.4979	0.6654	0.3033
VS-KNN	0.0341	0.0707	0.4937	0.6512	0.2872
GRU4REC	0.0334	0.0682	0.4837	0.6480	0.2826
SR	0.0332	0.0684	0.4853	0.6506	0.3010
AR	0.0325	0.0673	0.4760	0.6361	0.2894
S-KNN	0.0318	0.0657	0.4658	0.5996	0.2620
CT	0.0316	0.0654	0.4710	0.6359	**0.3072**

Summary of Accuracy Measurements. Overall, across the domains we can observe that only in one single case—when using the RSC15 dataset—a rather early complex model was able to outperform relatively simple baselines. In the large majority of the cases in particular the neighborhood-based methods are better than newer neural approaches in terms of precision, recall, MAP, the hit rate and, in two cases also in terms of the MRR. When considering only the immediate next item for evaluation, and when using the MRR, the ranking of the algorithm often changes compared to the other measures. No consistent pattern was, however, found in terms of this measurement across the domains and datasets.

Some of the more recent approaches like NEXTITNET or STAMP often performed worse than GRU4REC according to our evaluation. In the original papers, they won such a comparison, although with different data subsets and evaluation procedures as in [4]. In the end,

Table 6: Results for the music domain datasets

Metrics	MAP@20	P@20	R@20	HR@20	MRR@20
NOWP					
VS-KNN	**0.0193	**0.0664	**0.1828	*0.2534	0.0810
S-KNN	0.0186	0.0655	0.1809	0.2450	0.0687
AR	0.0166	0.0564	0.1544	0.2076	0.0710
SR	0.0133	0.0466	0.1366	0.2002	0.1052
NARM	0.0118	0.0463	0.1274	0.1849	0.0894
GRU4REC	0.0116	0.0449	0.1361	0.2261	0.1076
STAMP	0.0111	0.0455	0.1245	0.1919	0.0897
CT	0.0065	0.0287	0.0893	0.1679	**0.1094**
30MU					
VS-KNN	**0.0309	**0.1090	**0.2347	**0.3830	0.1162
S-KNN	0.0290	0.1073	0.2217	0.3443	0.0898
AR	0.0254	0.0886	0.1930	0.3088	0.0960
SR	0.0240	0.0816	0.1937	0.3327	0.2410
NARM	0.0155	0.0675	0.1486	0.2956	0.1945
GRU4REC	0.0150	0.0617	0.1529	0.3273	0.2369
STAMP	0.0093	0.0411	0.0875	0.1539	0.0819
CT	0.0058	0.0308	0.0885	0.2882	*0.2502
AOTM					
S-KNN	**0.0037	**0.0139	**0.0390	**0.0417	0.0054
VS-KNN	0.0032	0.0116	0.0312	0.0352	0.0057
AR	0.0018	0.0076	0.0200	0.0233	0.0059
SR	0.0010	0.0047	0.0134	0.0186	0.0074
NARM	0.0009	0.0050	0.0146	0.0202	0.0088
CT	0.0006	0.0043	0.0126	0.0191	**0.0111
NEXTITNET	0.0004	0.0024	0.0071	0.0139	0.0065
STAMP	0.0003	0.0020	0.0063	0.0128	0.0088
GRU4REC	0.0003	0.0020	0.0063	0.0130	0.0074

it seems that progress in neural session-based recommendation is still limited, and the various reported improvements over the landmark GRU4REC method are seemingly not enough to consistently outperform much simpler techniques.

4.1 Additional Observations

Scalability. Scalability can be an issue for some of the complex models, with GRU4REC being among the faster approaches. The authors of STAMP and NARM, for example, use only 1/4 or 1/64 of the RSC15 dataset in their own experiments. Similarly, the largest dataset used for the evaluation of NEXTITNET has about 2 million sessions, which is a fraction of the original RSC15 dataset.

We measured the runtimes of training and prediction for all methods in all experiments. As an example, we report the results for RSC15 and ZALANDO in terms of the training time for one split and the average time needed to generate a recommendation list[3].

Methods like SR or VS-KNN do not learn complex models. They only need some time to count co-occurrences or prepare data structures. Also, the CT technique can be efficiently initialized. Training GRU4REC on one data split on our hardware took less than an hour. STAMP needed only slightly more time than GRU4REC, but NARM was four times slower. Finally, the most recent convolutional NEXTITNET method seems to be limited in terms of practical applicability as it needs more than one day for training on a GPU even for datasets of modest size. When datasets are used that comprise a larger set of items, e.g., the one from Zalando, the performance differences are even more pronounced. The CT method is generally fast enough when predicting for the RSC15 dataset, but it slows down rapidly when the number of items increases.

[3]Times were measured on a workstation computer with an Intel Core i7-4790k processor and a Nvidia Geforce GTX 1080 Ti graphics card (Cuda 10.1/CuDNN 7.5).

Table 7: Running times

	Training		Predicting (ms)	
Algorithm	RSC15	ZALANDO	RSC15	ZALANDO
GRU4REC (on GPU)	0.89h	1.51h	8.81	30.06
STAMP (on GPU)	1.25h	7.61h	13.79	51.84
NARM (on GPU)	4.36h	12.99h	9.72	28.69
NEXTITNET (on GPU)	26.39h	–	8.98	–
SR (on CPU)	17.35s	21.37s	3.40	8.66
VS-KNN (on CPU)	10.71s	5.48s	16.42	26.00
CT (on CPU)	5.91m	2.10h	57.66	327.83

Coverage and Popularity Bias. Previous work has indicated that some methods, in particular the simpler ones, can have a tendency to recommend more popular items [15]. At the same time, some algorithms can focus their recommendations on a small set of items that are recommended to everyone, which can be undesired in certain domains and lead to limited personalization.

To identify such potential differences, we measured the popularity bias of each algorithm by averaging the min-max normalized popularity values of the recommended items in the top-20 recommendations. Furthermore, we determined the fraction of items that ever appeared in the generated top-20 recommendations (*coverage*).

The general tendencies across datasets are as follows. In terms of the popularity bias, CT is usually very different from the other methods, and it focuses much more on popular items. For the other methods, no clear ranking was found across datasets. In many cases, however, GRU4REC is among the methods that recommend the least popular (or: most novel) items. GRU4REC also often has the highest and STAMP the lowest coverage. VS-KNN is similar to the other neural approaches in terms of coverage.

5 CONCLUSIONS

Our work indicates that even though a number of papers on session-based recommendations were published at very competitive conferences in the last years, progress seems to be still limited (or only *phantom progress*) despite the increasing computational complexity of the models. Similar to the IR domain, one main problem seems to lie in the choice of the baselines, and our work points to a potentially major limitation of today's research practice.

A general phenomenon in that context is that previous non-neural approaches—as well as simpler methods—are often disregarded in empirical evaluations, and only neural methods are used as baselines despite their possibly unclear competitiveness.

In some papers, little is also said about hyper-parameter optimization for the baselines. In addition, the code which is used in the optimization and evaluation procedures is not always shared, making reproducibility an issue. With our work, we provide a framework based on the work from [5, 15], where various algorithms can be benchmarked under the exact same conditions, using different evaluation schemes. Overall, we hope that this environment is helpful for other researchers to achieve higher levels of reproducibility and faster progress in this area.

REFERENCES

[1] 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.

[2] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News Session-Based Recommendations using Deep Neural Networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems (DLRS) '18*.

[3] Hailin Fu, Jianguo Li, Jiemin Chen, Yong Tang, and Jia Zhu. 2018. Sequence-Based Recommendation with Bidirectional LSTM Network. In *Advances in Multimedia Information Processing (PCM '18)*. 428–438.

[4] 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 (CIKM '18)*. 843–852.

[5] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings International Conference on Learning Representations (ICLR '16)*.

[6] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 505–514.

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

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

[9] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from History and Present: Next-item Recommendation via Discriminatively Exploiting User Behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1734–1743.

[10] Jimmy Lin. 2019. The Neural Hype and Comparisons Against Weak Baselines. *SIGIR Forum* 52, 2 (Jan. 2019), 40–51.

[11] Xiang Lin, Shuzi Niu, Yiqiao Wang, and Yucheng Li. 2018. K-plet Recurrent Neural Networks for Sequential Recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*. 1057–1060.

[12] Zachary C. Lipton and Jacob Steinhardt. 2018. Troubling Trends in Machine Learning Scholarship. arXiv:1807.03341 Presented at ICML '18: The Debates.

[13] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. 1831–1839.

[14] Pablo Loyola, Chen Liu, and Yu Hirate. 2017. Modeling User Session and Intent with an Attention-based Encoder-Decoder Architecture. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. 147–151.

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

[16] Dietmar Jannach Maurizio Ferrari Dacrema, Paolo Cremonesi. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys '19)*.

[17] Fei Mi and Boi Faltings. 2018. Context Tree for Adaptive Session-based Recommendation. *CoRR* (2018). arXiv:1806.03733

[18] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. 2002. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. In *Proceedings International Conference on Data Mining (ICDM '02)*. 669–672.

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

[20] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM '18)*. 565–573.

[21] 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 (WSDM '19)*. 582–590.