From the lab to production: A case study of session-based recommendations in the home-improvement domain

Pigi Kouki
pigi.kouki@relational.ai
RelationalAI

Ilias Fountalis
ilias.fountalis@relational.ai
RelationalAI

Nikolaos Vasiloglou
nik.vasiloglou@relational.ai
RelationalAI

Xiquan Cui
xiquan_cui@homedepot.com
The Home Depot

Edo Liberty
edo@pinecone.io
Pinecone

Khalifeh Al Jadda
khalifeh_al_jadda@homedepot.com
The Home Depot

ABSTRACT

E-commerce applications rely heavily on session-based recommendation algorithms to improve the shopping experience of their customers. Recent progress in session-based recommendation algorithms shows great promise. However, translating that promise to real-world outcomes is a challenging task for several reasons, but mostly due to the large number and varying characteristics of the available models. In this paper, we discuss the approach and lessons learned from the process of identifying and deploying a successful session-based recommendation algorithm for a leading e-commerce application in the home-improvement domain. To this end, we initially evaluate fourteen session-based recommendation algorithms in an offline setting using eight different popular evaluation metrics on three datasets. The results indicate that offline evaluation does not provide enough insight to make an informed decision since there is no clear winning method on all metrics. Additionally, we observe that standard offline evaluation metrics fall short for this application. Specifically, they reward an algorithm only when it predicts the exact same item that the user clicked next or eventually purchased. In a practical scenario, however, there are near-identical products which, although they are assigned different identifiers, they should be considered as equally-good recommendations. To overcome these limitations, we perform an additional round of evaluation, where human experts provide both objective and subjective feedback for the recommendations of five algorithms that performed the best in the offline evaluation. We find that the experts' opinion is oftentimes different from the offline evaluation results. Analysis of the feedback confirms that the performance of all models is significantly higher when we evaluate near-identical product recommendations as relevant. Finally, we run an A/B test with one of the models that performed the best in the human evaluation phase. The treatment model increased conversion rate by 15.6% and revenue per visit by 18.5% when compared with a leading third-party solution.

CCS CONCEPTS

• **Applied computing → Online shopping**;

KEYWORDS

session-based recommendations, evaluation using human experts, comparison of offline evaluation metrics with labels from human experts, A/B test

ACM Reference Format:
Pigi Kouki, Ilias Fountalis, Nikolaos Vasiloglou, Xiquan Cui, Edo Liberty, and Khalifeh Al Jadda. 2020. From the lab to production: A case study of session-based recommendations in the home-improvement domain. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3383313.3412235

1 INTRODUCTION

Session-based recommendations are mission-critical in e-commerce applications where customer intent across visits is weakly correlated. A representative example is the home-improvement domain where users visit the online store with a different purpose each time. As a result, their shopping history may not be very informative with respect to their current needs. For example, knowing which furniture a user purchased in the past to redesign her living room does not provide much insight into the tools she needs in the current visit to renovate her bathroom. Same-session actions, however, are very informative. For example, a customer currently shopping for a shower head and shower tiles will likely need a soap holder and grout.

There is a large body of work in the field of session-based recommender systems [20, 29]. Recently-published deep learning architectures [18, 19, 28, 30] show promise and report improved *offline* metrics compared to previous state-of-the-art systems. However, such models are hard to compare because they are evaluated on different datasets and evaluation metrics. In addition, recent studies [6, 22, 23] show that non-deep-learning baseline models, when properly tuned, can perform on par with or better than complex deep-learning architectures. In this work, we present the process of finding the most promising session-based model to put into production for a large home improvement e-commerce application, i.e., The Home Depot (THD). We first use the framework provided by Ludewig et al. [23] and evaluate fourteen session-based recommendation algorithms over three datasets from the home improvement domain, i.e., outdoors, tools, and appliances. We confirm recent findings [23] and conclude that there is no single model that universally outperforms all others across all metrics and datasets. For example, SR-GNN

RecSys '20, September 22–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09…$15.00
https://doi.org/10.1145/3383313.3412235

[30], a cutting-edge deep learning model for session-based recommendations, outperforms all other methods on the task of predicting the immediate next item. At the same time, simple variations of nearest neighbor algorithms outperform deep learning architectures when evaluating the prediction of a ranked list of items. As a result, unless we care about only one of these metrics, i.e., predicting the next item vs. a list of items, offline evaluation does not provide sufficient guidance regarding which algorithm(s) to advance to the A/B testing or production stages.

Offline evaluation is complicated not just by the range of performance metrics. Recent research [21, 27] shows the evaluation metrics can be unreliable when comparing different algorithms because they ignore problem-specific and domain-specific details. Motivated by this work, we observe that the way offline evaluation is performed is far from ideal given the particular characteristics of the home improvement domain. In particular, an algorithm is rewarded only when it recommends the exact product that was clicked by the user in the session. When a model recommends a product very similar to the one clicked by the user (but not exactly the same) the prediction is needlessly penalized.

This approach is problematic for domains where two or more products often share many similarities, differ only on minor details, yet are assigned a different product identifier. A representative domain where this is evident is the home-improvement domain. This is illustrated in Figure 1 where three algorithms A, B, and C are given a partial shopping cart <refrigerator, dishwasher, ?> and are measured on their ability to guess the third item (noted by ?) added in that cart, which we know from the actual data that it was a specific microwave. Algorithm A recommends a different microwave which shares many similarities (i.e., price, brand, dimensions, color, and style) with the actually-bought microwave. Algorithm B recommends an electric range, and Algorithm C recommends a garbage disposal part. Since all predicted products are different from the most recent product added to cart, all offline evaluation metrics that focus on measuring the prediction accuracy will consider those predictions as equally bad. However, it is clear in this example that the prediction of algorithm A is far better than the prediction of algorithm B which is more relevant than the prediction of algorithm C.

To account for cases like the one described, we performed an additional round of evaluation. Human experts of our in-house validation team were asked to provide feedback in terms of the level of relevancy of recommendations. We distinguished three levels of feedback: objective relevant, subjective relevant, and irrelevant. In the example of Figure 1 the recommendation of algorithm A would be labeled as objective relevant, the recommendation of algorithm B would be considered as subjective relevant, while the recommendation of algorithm C would be irrelevant. We selected five algorithms that performed the best in the offline experiments and evaluated their performance in about 1,000 sessions involving 35,238 recommendations, using the same datasets as in the offline evaluation (outdoors, tools, appliances).

The result of the study shows that the ranking of the models determined by various offline evaluation methods does not agree with the ranking using expert opinion. This extends recent findings [2, 24, 26, 27] showing that there is a significant discrepancy between offline metrics obtained from historical data, versus user studies, and live A/B tests. In particular, two deep learning methods (GRU4REC [13] and STAMP [19]) that performed adequately but did not rank first in the offline evaluation, were the winners in the human expert study. Interestingly, the deep learning architecture that performed best in the offline evaluation in terms of predicting the immediate next item (SR-GNN) ranked third. Baseline methods based on nearest neighbors that reported the best performance in terms of precision and recall ranked in the last positions. This suggests that, for the home improvement domain, deep learning models are more successful compared to baseline methods. Analysis of the feedback showed that the performance of GRU4REC and STAMP improved significantly when we evaluated recommendations that were very similar to the items in the ground truth as relevant. For example, in this case, we considered as relevant the very similar microwave recommended by Algorithm A in Figure 1, although its product identifier is different from the ones added to the cart. In the last part of this work, we performed an A/B test to compare the performance of STAMP (one of the winning models of the evaluation using human experts) with an existing third-party solution. The results of the A/B test showed that STAMP increased the conversion rate by 15.6% and the revenue per visit by 18.5% and the difference is considered statistically significant.

Our contributions are as follows: (1) We confirm and extend, to the home improvement domain, recent findings that baseline session-based recommendation models can actually outperform complex deep learning architectures that are considered to be state-of-the-art in popular offline metrics such as precision and recall. (2) Motivated by the limitations of offline metrics, we conduct a large study where human experts provide objective and subjective feedback for the recommendations of five algorithms in three datasets and conclude that two deep learning models are the best in terms of objective and subjective relevance. (3) We provide insights from the comparison of the results from traditional evaluation metrics and annotations provided by human subjects. The most important lesson learned is that the ranking of models in terms of their performance in the offline evaluation does not always agree with the ranking of these models when evaluating them using labels from human experts. This is mostly due to the fact that although some models are able to come up with relevant recommendations, they are needlessly penalized when using the typical offline evaluation metrics that require the recommended item to match exactly the item in the ground truth. To the best of our knowledge this is the first study that attempts to draw conclusions by having a unified view of results from both traditional offline evaluation metrics and results from human experts in the session-based recommendations setting. (4) We share the results of an A/B test conducted using one of the models that performed the best in evaluation using human experts and showcase its superiority over a third-party solution. (5) Our lessons learned towards finding the best session-based algorithm for our application scenario can be helpful to researchers in industrial and academic settings.

2 RELATED WORK

Wang et al. [29] and Ludewig and Jannach [20] offer a detailed analysis of the most recent methods and advancements in the area of session-based recommendations. This study compares different session-based recommendation methods using several standard metrics provided in the literature. It further compares the correlation of such metrics to judgments by human experts.

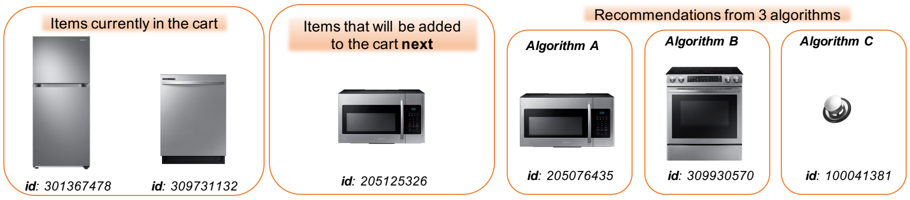

Figure 1: Given two products added in a user’s cart, three algorithms came up with three different predictions. Using traditional evaluation metrics, all predictions will be considered as irrelevant since none of the predicted product ids matches exactly the product id that the user added to her cart next. However, for a human it is obvious that there are different levels of relevance: (1) the prediction of Alg. A is very similar to the actual item added to cart, (b) the prediction of Alg. B is not similar to the item added to cart next, however, it is relevant to the items that the user has in her cart, (c) the prediction of Alg. C is irrelevant.

Comparison of session-based recommendation methods: Ludewig et al. [20, 23] provide a framework that implements a suite of state-of-the-art algorithms and baselines for session-based recommendation. The authors compare baseline and state-of-the-art algorithms in session-based recommendations using offline evaluation metrics. They conclude that the improvement gained by deep learning architectures is often limited. The authors further performed a user study in the music domain, where they evaluated five session-based algorithms [21] in terms of their effect on the quality perception of the users. The finding is that, for the task of next track recommendation, simple mechanisms are competitive with complex deep learning architectures. Finally, the authors in [22] conclude that there is a need for more studies to understand the characteristics of successful recommender systems in different domains. Working in a similar direction, we first apply the framework to three datasets in the home improvement domain and corroborate the authors’ findings. Motivated by the unique characteristics of the home improvement domain and the inconclusive offline evaluation results, we further evaluate the best performing algorithms using human experts. To the best of our knowledge, our work is the first that contrasts offline results with human perceptions in the home improvement domain. Analysis of the study showed, for the first time, that in the home improvement domain there is indeed merit from using deep learning models, as all models in this category outperformed the baselines. In addition, our work introduces the idea that recommendations that are similar to the actual user actions should be considered as equally good. Finally, we performed an A/B test with real users showing the improved performance of a particular deep learning model in the home improvement domain.

Limitations of using offline evaluation metrics in recommender systems: Wagstaff [27] observes that evaluation metrics (such as classification accuracy and RMSE) cannot be used reliably to compare different algorithms since they ignore or remove problem-specific details. In the home improvement domain, it may be too strict to require an algorithm to predict the *exact* same product that a user clicked or bought. An algorithm that can predict *similar* items is probably as good. Cremonesi et al. [5] performed a user study where seven different movie recommender systems were evaluated by real users. The study showed that the objective quality of an algorithm is not a reliable predictor of the quality of the recommendations as they are perceived by the users. Garcin et al. [7] evaluated the performance of three algorithms in the news recommendation domain, first in an offline setting and then by running an A/B test. The ranking of the models in the online experiment was the reverse of that in the offline evaluation, which puts in question the reliability of the offline evaluation studies. Beel and Langer [2] compare results from three different evaluation methods: offline, online, and user studies. The paper concludes that in the domain of research paper recommendation, results from offline evaluation contradict results from online evaluations and user studies, while results from online evaluations correlate well with results from user studies. Gomez-Uribe and Hunt [9] admit that offline evaluation is convenient since a large number of algorithms can be tested relatively quickly, which prunes the candidate variants for the A/B tests. However, the authors underline the fact that offline evaluation assumes users would have behaved the same given the new algorithm. This is a challenging assumption because the whole purpose of a new algorithm is to change user behavior. As a result, offline evaluation is used only as an intermediate step to choose which algorithms to move to the A/B test. Rossetti et al. [26] conduct a study comparing offline and online performance of four different movie recommendation algorithms. The study showed that the online and offline ranking of the algorithms is different. Kamehkhosh and Jannach [15] study whether offline evaluation metrics correlate with users’ quality perceptions in the music recommendation domain. Their work showed that methods that focus on both coherence of songs recommended and on improving accuracy yield the best performance in both offline evaluation and user studies. The studies described above outline the discrepancy between offline experiments, user studies, and live experiments. This discrepancy motivated us to compare three evaluation modes for session-based recommendations in the home improvement domain: evaluation using traditional offline metrics, evaluation using labels from human experts, and evaluation using an A/B test. To our knowledge, this is the first study of its kind.

3 BENCHMARK ALGORITHMS AND DATASETS

Table 1 presents the fourteen models (eight baseline and six deep learning) we evaluated for session-based recommendations. Detailed description of the models can be found in the original works and in [20, 22]. We used three real-world datasets coming from THD that represent the diversity of products: outdoors, tools, and appliances. We sampled sessions for a specific representative time period and focused only on add-to-cart actions with greater than two and less than seven products added to the cart. For each category, we discarded sessions containing a product from a different category. The detailed dataset statistics (including train, validation, and test splits) are reported in Table 2.

Model	Description
Baselines	
AR [1]	A simplified version of Association Rules where the algorithm recommends the item that most-frequently co-occurs with the last item in the session.
SR [16]	Sequential Rules is a variation of AR that weights the co-occurrence based on the order of the items in the session.
SKNN [3, 11, 14, 17]	Session-based K-Nearest Neighbor considers all the items of a session: given a session, the method computes similarities between sessions and recommends items that appear in the most similar sessions.
V-SKNN [20]	Vector multiplication SKNN extends the SKNN model by assigning higher weight to the most recent items of a session when computing similarities between sessions.
CT [25]	Context Trees is a non-parametric variable-order Markov model that is able to learn static sequential patterns.
STAN [8]	Sequence and Time Aware Neighborhood extends V-SKNN by using three different decay factors in the process of computing session-level similarity and item relevance.
V-STAN [22]	Vector Sequence and Time Aware Neighborhood combines STAN and V-SKNN by adding a sequence-aware item scoring process and an inverse-document-frequency weighting scheme to promote less popular items.
SMF [20]	Session-based Matrix Factorization model that combines factorized Markov chains with classic matrix factorization.
Deep Learning	
GRU4REC [12, 13]	GRU4REC, uses gated recurrent units [4] to capture dependencies among items in a session.
NARM [18]	Neural Attentive Recommendation Machine extends GRU4REC by introducing a hybrid encoder with an attention mechanism into the recurrent network to model the user's sequential behavior.
STAMP [19]	Short-Term Attention/Memory Priority model captures both users' general interests from the long-term memory of a session context and users' current interests from the short-term memory of the last clicks.
NEXTITNET [31]	NEXTITNET is a generative model that directly estimates the distribution of the output item sequence based on the input item sequence.
CSRM [28]	Collaborative Session-based Recommendation Machine is a hybrid deep learning model that applies collaborative neighborhood information to session-based recommendations.
SR-GNN [30]	Session-based Recommendation with Graph Neural Networks models sessions as directed graphs which allows to capture complex transition of items. An attention network is also deployed to fuse the global preference and the local interest of the current session.

Table 1: Benchmark models evaluated for session-based recommendation in the home improvement domain.

Dataset	Split	Actions	Sessions	Items
Outdoors	Training	2,271,458	722,260	43,798
	Validation	114,247	35,383	14,398
	Test	377,230	126,442	21,193
Tools	Training	1,346,326	445,326	25,066
	Validation	42,066	14,215	6,399
	Test	187,333	64,341	13,188
Appliances	Training	456,686	161,649	10,079
	Validation	14,294	5,030	2,924
	Test	110,164	38,595	5,776

Table 2: Dataset statistics. Each action represents the fact that a user added to her cart an item (i.e., actions are add-to-cart actions) and each session contains an ordered set of add-to-cart actions.

4 EVALUATION USING OFFLINE TRADITIONAL METRICS

In the first part of this work we use the benchmark algorithms described in Table 1 and apply them to the datasets reported in Table 2. We evaluate their performance using eight popular metrics and conclude that offline evaluation does not provide enough evidence in the process of selecting the best model to advance to an A/B test.

4.1 Evaluation Setup and Metrics using Traditional Metrics

Data splits: For each category we split the data (Table 2) into three sets in subsequent time periods: (1) training set from the first time period, (2) validation set from the second time period immediately after the first time period, and (3) test set from the third time period immediately after the second time period. Since none of the models can handle cold-start recommendations, we pruned all sessions that contained new products in the validation and test set.

Hyperparameter tuning: To find the best parameters for each model (baselines and deep neural networks) we performed hyperparameter tuning using the validation set. We followed the same process as described in [22, 23], i.e., we applied a random search approach with 100 iterations (50 iterations for NARM and SR-GNN that were significantly slower compared to the other models). We used the same values for the search space for all hyperparameters as in previous work [22, 23]. Finally, all models were optimized for mean reciprocal rank.

Evaluation Metrics: All results are reported on the test set. Metrics fall into three categories based on (1) predicting the immediate next item that was added to cart (2) predicting the N next items added to cart (3) coverage and popularity-based metrics. For (1) these metrics are: hit rate (HR), mean reciprocal rank (MRR), and normalized discounted cumulative gain (NDCG) [10]. For (2) the metrics are: precision (Prec), recall (Rec), and mean average precision (MAP) [10]. For (3) the metrics are the transactional coverage (Cov) and popularity (Pop). Coverage is defined as the fraction of number of distinct items that a model can predict in the top N positions over the number of distinct items that appear in the training set. Popularity is defined as the average popularity of all recommended items in the top N positions, where popularity for each item is computed from the training set. Although deep neural network models are optimized for the first task (i.e., predict the immediate next item), in a real-world application a session-based recommendation algorithm is expected to recommend a list of items to add next to the user's cart. For this reason, metrics that measure the accuracy of a list of recommendations (i.e., Prec, Rec, MAP) are also of great importance.

Additionally, in our application scenario, the "rich get richer" effect is evident, since a small percentage of products are very popular and have a large number of purchases, where the vast majority of products belong to the long tail with a very small number of purchases. As a result, increasing customer awareness of the full catalog of a retailer is also a goal, in addition to optimizing for conversions. In other words, it is highly desirable for an algorithm to report high coverage and low popularity while maintaining conversion. To compute the evaluation metrics for each model we follow the same process as in previous work [22, 23] and for each session we reveal each interaction sequentially. For each revealed interaction, we generate recommendation lists and compare them to the ground truth data in order to compute the values for the evaluation metrics.

4.2 Results from Offline Evaluation

The results of the offline evaluation using the traditional accuracy and coverage metrics are shown in Table 3. We also report training times using CPU for baseline methods and GPU for deep learning methods. All deep learning models are trained using one single GPU. We choose the number of top-N recommendations to evaluate as $N = 5$ which is smaller compared to what the literature reports ($N = 20$) for two reasons: first, we wanted to be able to compare the results with the evaluation using human experts and second, in our e-commerce scenario, users can directly look at the top five recommendations (which was also the case for the A/B test conducted). For each category, we report with bold the best performing value and we underline the second to best value from the other family of algorithms. For each accuracy metric, we conduct a paired t-test between the best performing deep learning model and baseline and indicate with an asterisk the cases that the difference is considered statistically significant (p<.001). Below we present the major insights, grouped by the three different metric categories discussed above. Next, we discuss the overall performance of baseline and deep learning approaches and also compare our findings with very recent findings of the community.

Overall Comparison in terms of predicting the immediate next item: For all three categories, SR-GNN consistently outperforms all the other models in terms of HR, MRR, and NDCG. The second-best model(s) in terms of HR and NDCG is either STAN or V-STAN for all categories (STAN is the second-best for tools while V-STAN is the second-best for outdoors and appliances). In terms of MRR, SR-GNN is followed by CT for outdoors and appliances, while SR-GNN is followed by V-STAN for tools.

Overall Comparison in terms of predicting a list of next items: STAN and V-STAN outperform all the baseline and deep learning models in terms of precision, recall, and MAP for most of the datasets. The only exception is the recall for tools and appliances where STAN/V-STAN come second after SR-GNN. In terms of precision and MAP the performance of SR-GNN is the best among the deep neural network models, however, it is worse compared to V-STAN (outdoors, appliances) and STAN (tools).

Overall Comparison in terms of coverage and popularity: V-SKNN and GRU4REC are the models that achieved the best performance in terms of coverage. For outdoors and tools V-SKNN is the best, followed by GRU4REC, while the reverse is true for appliances. For popularity, GRU4REC consistently outperforms all other models followed by SMF. In terms of baselines, V-SKNN is the best (but worse compared to GRU4REC and SMF).

Comparison among baselines: Across the baseline models, STAN and V-STAN outperform all other models in terms of all accuracy metrics (HR, MRR, NDCG, precision, recall) with the exception of MRR for outdoors and appliances. Consequently, our findings agree with [22], i.e., V-STAN is a very strong approach that should be included as a baseline in future evaluations of session-based recommendations. V-SKNN and SKNN that perform similarly in terms of accuracy metrics, follow the performance of STAN and V-STAN in almost all accuracy metrics (HR, NDCG, precision, recall, MAP). At the same time, V-SKNN performs the best among the baselines in terms of coverage and popularity.

Comparison among deep learning approaches: Among all deep learning models, SR-GNN is the clear winner in terms of all accuracy metrics. This is a new observation which is different from the main finding of [22] where it was not clear how to rank the deep learning models, because of the variations of the performance of the algorithms in different datasets. We believe that this may be due to the particular characteristics of our domain. Also, this new finding suggests that every e-commerce application needs to perform this step of offline evaluation with traditional metrics using their data in order to find which are the best performing methods. Other than SR-GNN, we observe that there is not a clear ranking for the other models, since we do not observe a consistent behavior among accuracy metrics and across datasets. It is notable though that most of the times, deep learning models (except for SR-GNN) perform worse compared STAN and/or V-STAN. In terms of coverage and popularity, GRU4REC ranks first, which agrees with [22].

Overall comparison: There is not a clear winning method. Although it is expected that there is a trade-off between coverage and accuracy metrics, in this case there is also no winning model across *all* accuracy metrics. The big question for each e-commerce application seems to be whether it focuses on optimizing for predicting the immediate next item or a list of next items. In the first case, it looks like SR-GNN is the best choice based on this offline evaluation, while for the second case, baseline models such as STAN and V-STAN look like a better option. In cases where an e-commerce application focuses a lot on increasing the coverage, then good options are either V-SKNN or GRU4REC. Training times and scaling is also a concern in real e-commerce applications since the models need to train regularly using a large number of sessions. There are deep learning models that can train very fast on a GPU (such as GRU4REC and STAMP), making the process of putting those into production relatively easy. However, other models (i.e., SR-GNN, NARM) need a relatively long amount of time to train which makes their adoption by real e-commerce applications more difficult.

5 EVALUATION USING HUMAN EXPERTS

We now present evaluation of the models using human experts. This is a necessary step for two reasons. First, offline measurements did not produce a clear winner, and, second, traditional metrics (HR, MRR, NDCG, Prec, Rec, MAP) cannot capture the subtleties of the home improvement domain, discussed in Section1.

Dataset	Metrics	HR@5	MRR@5	NDCG@5	Prec@5	Rec@5	MAP@5	Cov@5	Pop@5	Training Time
Outdoors	**AR**	0.451	0.278	0.377	0.140	0.386	0.079	0.547	0.118	6.8s
	SR	0.457	0.290	0.386	0.137	0.385	0.078	0.525	0.110	4.1s
	SKNN	0.504	0.281	0.420	0.159	0.434	0.091	0.560	0.118	5.6s
	V-SKNN	0.506	0.284	0.421	0.160	0.436	0.092	**0.626**	0.105	11.3s
	CT	0.469	0.301	0.398	0.140	0.392	0.080	0.507	0.133	281s
	STAN	0.509	0.288	0.428	0.160	0.436	0.092	0.569	0.113	5s
	V-STAN	0.516	0.290	0.434	***0.162**	***0.441**	***0.094**	0.588	0.100	5.8s
	SMF	0.479	0.300	0.403	0.144	0.402	0.082	0.364	0.087	1.66h
	GRU4REC	0.470	0.293	0.396	0.140	0.393	0.080	0.590	**0.064**	0.26h
	STAMP	0.467	0.291	0.390	0.146	0.405	0.083	0.506	0.116	0.39h
	NEXTITNET	0.484	0.305	0.407	0.144	0.402	0.082	0.364	0.117	1.8h
	NARM	0.483	0.297	0.403	0.147	0.407	0.083	0.535	0.111	9.4h
	CSRM	0.484	0.296	0.402	0.147	0.407	0.083	0.510	0.109	1.12h
	SR-GNN	***0.526**	***0.337**	***0.448**	0.157	0.439	0.091	0.423	0.123	30.5h
Tools	**AR**	0.379	0.234	0.318	0.115	0.327	0.066	0.581	0.067	3.2s
	SR	0.380	0.240	0.321	0.115	0.328	0.067	0.567	0.064	3.4s
	SKNN	0.428	0.235	0.354	0.134	0.374	0.078	0.599	0.071	3s
	V-SKNN	0.417	0.236	0.347	0.128	0.360	0.074	**0.700**	0.055	3.3s
	CT	0.374	0.236	0.316	0.113	0.324	0.066	0.526	0.090	245s
	STAN	0.431	0.238	0.360	***0.135**	0.376	***0.080**	0.604	0.067	3.2s
	V-STAN	0.428	0.240	0.358	0.134	0.373	0.079	0.609	0.066	3s
	SMF	0.396	0.243	0.330	0.120	0.343	0.069	0.395	0.049	0.60h
	GRU4REC	0.379	0.233	0.317	0.115	0.327	0.066	0.599	**0.028**	0.16h
	STAMP	0.387	0.240	0.323	0.119	0.340	0.069	0.592	0.061	0.2h
	NEXTITNET	0.387	0.238	0.322	0.118	0.337	0.068	0.384	0.069	0.85
	NARM	0.399	0.240	0.330	0.123	0.348	0.071	0.596	0.064	2.9h
	CSRM	0.401	0.241	0.331	0.120	0.341	0.069	0.553	0.065	0.25h
	SR-GNN	***0.440**	***0.275**	***0.372**	0.133	**0.378**	0.078	0.506	0.068	12.5h
Appliances	**AR**	0.523	0.344	0.450	0.160	0.459	0.097	0.644	0.113	1s
	SR	0.536	0.359	0.436	0.160	0.467	0.098	0.621	0.107	1s
	SKNN	0.579	0.312	0.485	0.180	0.511	0.110	0.674	0.111	0.95s
	V-SKNN	0.578	0.312	0.487	0.182	0.512	0.111	0.743	0.103	2.1s
	CT	0.537	0.375	0.473	0.160	0.465	0.098	0.580	0.140	45.8s
	STAN	0.580	0.316	0.491	0.183	0.517	***0.112**	0.667	0.108	1s
	V-STAN	0.583	0.318	0.493	***0.184**	0.517	***0.112**	0.667	0.107	1s
	SMF	0.560	0.372	0.483	0.171	0.490	0.105	0.480	0.096	0.10h
	GRU4REC	0.514	0.338	0.441	0.159	0.455	0.096	**0.769**	**0.066**	0.05h
	STAMP	0.524	0.355	0.453	0.161	0.469	0.098	0.690	0.106	0.07h
	NEXTITNET	0.562	0.379	0.487	0.167	0.483	0.102	0.670	0.109	0.19h
	NARM	0.552	0.357	0.469	0.169	0.484	0.102	0.660	0.108	0.48h
	CSRM	0.555	0.359	0.474	0.170	0.489	0.104	0.639	0.107	0.08h
	SR-GNN	***0.602**	***0.396**	***0.518**	0.181	***0.522**	0.111	0.594	0.106	1.9h

Table 3: Results on offline evaluation metrics on three datasets: outdoors, tools, and appliances. The highest value across all algorithms is shown in bold. The highest value obtained by the other family of algorithms (baseline or deep neural network) is underlined. The training time reported is when using a CPU for baseline models and a GPU for deep learning models. Stars indicate significant differences according to a paired t-test between the best performing baseline and deep learning model for each metric (p<.001).

5.1 Algorithm Selection

Getting labels from humans is an expensive task, especially when they are experts in the field. Since evaluating all fourteen algorithms was cost-prohibitive, we chose five algorithms based on their offline performance, namely V-SKNN, V-STAN, GRU4REC, STAMP, and SR-GNN. V-SKNN showed the best or second-best coverage in all datasets without falling too far behind in terms of accuracy metrics. V-STAN reported best or second-best performance in terms of precision, recall, and MAP in all datasets. GRU4REC reported the best or second-best coverage in two datasets and satisfactory performance in the accuracy metrics. STAMP performed slightly better compared to the GRU4REC model in terms of accuracy metrics with a small decrease in coverage. SR-GNN reported the best performance in terms of HR, MRR, and NDCG in all datasets and best or second-best performance for the other accuracy metrics. Note that GRU4REC, STAMP, and SR-GNN are deep learning models while V-SKNN and V-STAN are not.

5.2 Labeling Task

The next step was to generate the test set for the expert validators. To this end, we used a subset of the predictions produced by the models using the test set described in Section 4. To unify the task, we selected uniformly at random shopping sessions of 5 items $S = \{i_1, i_2, i_3, i_4, i_5\}$ where i_t is an item from the catalog and t is its insertion order to the cart. We use the items in $\mathrm{S}_1 = \{i_1, i_2\}$ as the context for each model to generate a set of 5 item recommendations $\mathrm{R} = \{r_1, ..., r_5\}$. We use the set $\mathrm{S}_2 = \{i_3, i_4, i_5\}$ as the ground truth for the recommended items. We asked the validators to compare *each* recommended item $r_i \in \mathrm{R}$ with the items in S_1 and S_2 and provide a label.

Figure 2 shows an example of a session with the recommendations generated by the model. For each session, we showed the validators a picture that consisted of the three sets S_1, S_2, and R along with descriptions. S_1 is presented as "items currently in the cart", S_2 as "items that will be added to the cart next", and R as "recommendations". To give validators enough context for each item (product), we provided its picture, identifier, title, and price. There was also the option to click on the picture of each product and get access to its full information as it appears on the web page of THD. For each triplet $< \mathrm{S}_1, \mathrm{S}_2, r_i >$ where $r_i \in \mathrm{R}$, the validators had to provide one of three labels:

Objective relevant: a recommendation was assigned this label if the item was the same or very similar to one of the items in S_2. We consider two products to be very similar if they are interchangeable which is indicated when they serve the same functionality, belong to the exact same category (e.g., outdoor lounge chair), share the same style (e.g., same color), and share the same or similar values for basic attributes such as price and material. This was indicated by a 'thumbs up' icon in the labeling interface.

Subjective relevant: recommendations were assigned this label when they were not objective relevant but they made good sense from a customer perspective when taking into account the items in S_1. This was indicated by a 'check'.

Irrelevant: a recommendation was assigned this label if the recommendation was neither objective nor subjective relevant. This was indicated by a 'thumbs down' icon.

For example, in Figure 2, the first two recommendations received a 'thumbs up' which corresponds to objective relevant. The outdoor patio chaise lounge and rocking chair were *not* exact matches to those in S_2, but they were objectively interchangeable with i_4 and i_3 respectively. The third and fourth recommendations received a subjective relevant label ('check'). An outdoor patio ottoman did not appear in S_2, but it was relevant given the fact that the customer already added an outdoor patio swivel lounge chair to the cart. An outdoor accent table could be considered interchangeable with the coffee table in S_2, but they belong to different categories (outdoor coffee table vs. outdoor side table) and therefore they were assigned the subjective relevant label. Given that the cart contained a loveseat and a lounge chair, a picnic table was not deemed reasonable or relevant by the experts ('thumbs down').

In this experiment, recommendations were not penalized if their ordering did not match the order that the items were added to the cart. Finally, the labels provided for the recommendations within the same session were independent from each other. That is, the decision about the label that would be assigned to $< \mathrm{S}_1, \mathrm{S}_2, r_1 >$ did not affect the decision about the label what would be assigned to $< \mathrm{S}_1, \mathrm{S}_2, r_2 >$. For example, if in Figure 2 there was a sixth recommendation of an outdoor patio rocking chair which was similar to the chair in S_2 then this would be considered as objective relevant and there would be no penalty for providing two recommendations of outdoor patio rocking chairs.

5.3 Results from Evaluation using Human Experts

We evaluated sessions from three datasets; 327 sessions in outdoors, 339 sessions in tools, and 318 sessions in appliances. All models were evaluated using the exact same sessions. This resulted in 24,600 annotations: 8,175 for outdoors, 8,475 for tools, and 7,950 for appliances. Since the validators were part of the professional validation team, there was no need to identify spam users (as opposed to experiments within Amazon Mechanical Turk). We ensured that our validators had clear instructions. To this end, we provided examples before validation, we conducted Q&A sessions, and we also inspected the labels provided at the beginning of the evaluation process for each category, pointing out possible shortcomings in the labeling process that were subsequently fixed.

For some sessions, more than one models came up with the exact same recommendations. This resulted in having multiple labels for the same triplet $< \mathrm{S}_1, \mathrm{S}_2, r_i >$. Since there were multiple validators involved in the process of evaluation and the task required to use their critical thinking, there was a possibility that these labels did not agree. To resolve the disagreements, we applied a simple majority vote algorithm. We first computed the total number of labels available for each triplet $< \mathrm{S}_1, \mathrm{S}_2, r_i >$ and then computed the number of same labels for each triplet (votes). Since we had five algorithms that provided recommendations, the maximum number of labels for a particular $< \mathrm{S}_1, \mathrm{S}_2, r_i >$ was five. This happened when all five algorithms came up with the exact same recommendation. We then worked as follows: if the number of total labels was four or five and the number of same labels was at least three, then we assigned this majority label to the triplet. For example, if we had four labels in total and three were objective relevant then we would assign the label objective relevant to the triplet $< \mathrm{S}_1, \mathrm{S}_2, r_i >$. Similarly, if there were three labels in total and at least two labels were the same, then we assigned this majority label to the triplet. To account for cases where we had only one or two labels for a triplet $< \mathrm{S}_1, \mathrm{S}_2, r_i >$, we ran a second round of validation to ensure that all triplets were labeled at least three times in order to have sufficient evidence for the majority vote algorithm. The second round of labeling increased the number of recommendations that were labeled to 11,522 for outdoors, 12,507 for tools, and 11,209 for appliances (35,238 in total). After the second round of evaluation, we applied again the same majority vote algorithm, and discarded a triplet if there was no agreement among the validators. This approach resulted in 98.4% of the sessions having a majority label. Table 4 presents the overall results of the study. For each dataset and model, we present the following metrics:

- **%triplets used**: the number of triplets, $< \mathrm{S}_1, \mathrm{S}_2, r_i >$, with a majority agreement among validators over the total number of triplets.
- **#triplets**: the number of triplets where there was agreement among validators.
- **Exact relevant**: The ratio of exact relevant recommendations over the total number of recommendations. For a triplet $< \mathrm{S}_1, \mathrm{S}_2, r_i >$ a recommendation r_i is considered as exact relevant if r_i matches **exactly** (i.e., same product identifier) one of the items in S_2. This metric is more strict than objective relevant as, in the latter, the

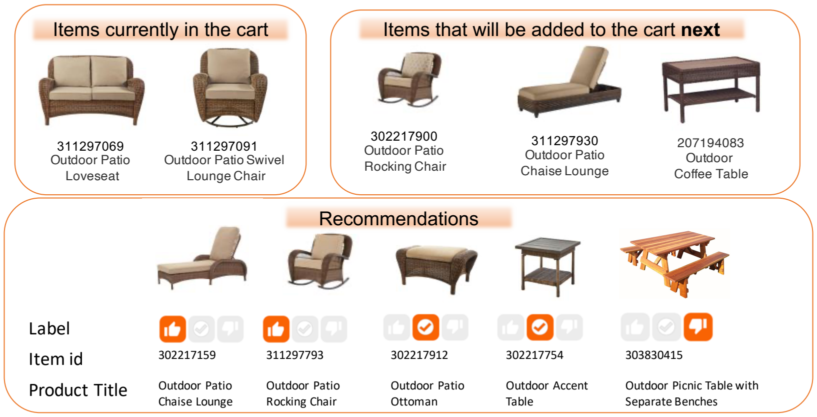

Figure 2: Validation example: for one session, two items are currently in the cart (S_1) and three items will be added next (S_2). The validators provided labels for five recommendations (R). The first and second recommendations are assigned the 'thumbs up' label (objective relevant) since they are very similar to items in S_2. The third and fourth recommendations are assigned the 'check' label (subjective relevant) since, although these recommended items are neither in S_2 nor similar to any item in S_2, they are still considered relevant given the items in S_1. Finally, the last recommendation is assigned the 'thumbs down' label (irrelevant) since it is not relevant to items in S_1 or S_2.

recommended item does not necessarily have to match the product identifier of the item in S_2.

- **Objective relevant**: the ratio of the number of objective relevant labels over the number of triplets used.
- **Subjective relevant**: the ratio of the number of subjective relevant labels over the number of triplets used.
- **Objective not exact relevant**: the ratio of the number of objective relevant labels for the case that the recommended product was very similar but not identical to the product in S_2 over the number of triplets used. This metric captures how often a model recommends items that are very similar but not identical to the item added to cart. The values for this metric are crucial for the study, since they will provide an insight into how often a model is penalized using offline evaluation metrics despite making predictions very close to the actual items that were added to cart.
- **Overall relevant**: the summation of the values for objective relevant and subjective relevant that shows the percent of relevant recommendations made by each model.

The exact relevant metric is computed offline since it does not require input from validators. To compute the rest of the metrics, we used only the triplets where validators agreed. For each metric, we conduct a paired t-test between the two best performing models and indicate with asterisks the cases that the difference is considered statistically significant (***p<.001, **p<.01, *p<.05). We now present the main insights from our study and also compare the results of this study (Table 4) with the results of the offline evaluation (Table 3).

Exact Relevant: V-SKNN reports very good performance in terms of this metric, since this model ranked first for outdoors and tools and second-best for appliances. Note though that this improvement is considered statistically significant only for the case of tools. GRU4REC ranks second-best for outdoors and tools. For this metric, it is difficult to distinguish which models are best because the performance improvements are marginal for two out of three datasets. On a separate note, exact relevant can be considered very similar to the precision metric reported in the evaluation using traditional metrics in Table 3, however the ranking of the models is different in this case. We believe that this happened because we worked on a subset of the dataset used in Section 4 focusing on actions with five items.

Objective relevant: GRU4REC is the model that ranked first for this metric in all datasets and the difference is always considered statistically significant. Taken together with the performance of GRU4REC on the exact relevant metric (not ranked 1st) and the objective not exact relevant metric (by far the winning model), we conclude that GRU4REC is getting this boost in performance for the objective relevant metric because it recommends products that, although not identical to items added to cart, are still very similar. It is not straightforward to determine which model ranks second for this metric, since there is variation depending on the dataset. Regardless, the important observation is that the ranking of the models by the exact relevant metric is not in agreement with the ranking of the models for the objective relevant metric. This is an important finding, showing that the offline evaluation metrics are not enough since they are too strict and penalize the models even when they make predictions very similar to the ground truth. This also suggests that researchers should not rely solely on the results of offline evaluation to select the most appropriate model for their system.

Objective not exact relevant: For all models, the percentage of recommendations that are very similar but not identical to the items added to cart is very high. We observe that the lowest increase in absolute value is 6.3% while the highest increase is 22.7% which is a huge boost for any model. The results around this metric indicate that our motivation regarding the limitations of the offline evaluation metrics has merit: these metrics indeed penalize the models since they require exact match of the product identifiers. In terms of comparison among models, GRU4REC ranks first for this metric, always followed by STAMP. The difference is always statistically significant. This suggests that both of these models are able to not only recommend the exact products that the users added to cart, but they can also recommend items very similar to these products. On the other hand, baseline models such as V-SKNN and V-STAN are

Dataset	Model	% triplets used	#triplets	Exact relevant	Objective relevant	Subjective relevant	Objective not exact relevant	Overall relevant
Outdoors	V-SKNN	98.2%	1,611	**0.220**	0.285	0.234	0.064	0.519
	V-STAN	98.7%	1,622	0.209	0.277	0.191	0.065	0.468
	GRU4REC	98.0%	1,593	0.218	***0.402	0.275	***0.183	0.677
	STAMP	98.9%	1,607	0.210	0.357	***0.329	0.145	**0.686**
	SR-GNN	98.9%	1,607	0.215	0.344	0.255	0.129	0.599
Tools	V-SKNN	96.3%	1,599	**0.245	0.396	0.211	0.155	0.607
	V-STAN	98.1%	1,629	0.224	0.282	0.195	0.063	0.477
	GRU4REC	97.1%	1,610	0.228	*0.420	0.201	**0.197	0.621
	STAMP	97.4%	1,612	0.226	0.390	**0.258	0.169	*0.648
	SR-GNN	96.6%	1,603	0.215	0.313	0.213	0.102	0.526
Appliances	V-SKNN	99.1%	1,571	0.284	0.388	0.121	0.113	0.509
	V-STAN	98.9%	1,552	**0.286**	0.396	0.108	0.117	0.504
	GRU4REC	99.2%	1,574	0.268	***0.488	0.169	***0.227	***0.657
	STAMP	98.9%	1,563	0.218	0.386	**0.180**	0.173	0.566
	SR-GNN	98.6%	1,588	0.280	0.418	0.146	0.141	0.564

Table 4: Evaluation results using human experts on three datasets: outdoors, tools, and appliances. Exact relevant is computed by comparing items in R with items in S_2. All the other metrics are computed using the feedback from validators. The highest value across all algorithms is shown in bold. The second-best value is underlined. Significance levels: *** p < .001, ** p < .01, * p < .05.

not able to perform well in this metric, suggesting that they cannot learn deep interactions between the items and the sessions, instead focusing on predicting the exact same item that a user added to cart.

Subjective relevant: STAMP outperforms the other models in all datasets (difference is considered statistically significant for outdoors and tools). GRU4REC ranks second for outdoors and appliances, while SR-GNN ranks second for tools. This suggests that STAMP can capture the user intent and recommend relevant items to the ones in the cart.

Overall relevant: STAMP and GRU4REC are the two best-performing models, but, at the same time it is not straightforward to identify the clear winner: STAMP scored first for outdoors and tools and second for appliances, while the reverse is true for GRU4REC.

6 A/B TEST

A/B testing is the most authoritative means to determine the exact effectiveness of an algorithm for a business application [9] as it can compute net business-impact. However, a bad A/B test design choice could also impact customer experiences in real life and cause tangible business damages. To mitigate this potential risk, we performed the above two rounds of evaluation that allowed us to decide which is the best candidate model for our application scenario. In what follows, we explain our choice to use STAMP to test against an existing third-party solution and the challenges that we had to address to efficiently use such a model. Finally we present the results of the A/B test.

6.1 Session-based Recommendations in Production with STAMP

Based on the results of both offline evaluation and the evaluation from human experts, we selected STAMP to participate in an A/B test for the following reasons: (1) STAMP performed well in all metrics reported in the offline evaluation (Table 3) with a very low training time compared to other deep-learning models such as SR-GNN. (2) STAMP reported the best overall relevant accuracy in the evaluation using human experts in two out of three datasets (Table 4). Additionally, STAMP ranked first in terms of subjective-relevant score in all datasets.

We measured STAMP against the control which is a leading third-party software for retail recommendation. Since it is provided as a black box solution we cannot describe its inner workings. Putting a complex deep-learning model such as STAMP [19] into production is a challenging task: STAMP requires computing the tri-linear product of the session embedding, the last item embedding, and the embedding of each product in the catalog for each event that involves any change of items in the cart. Since the speed of serving recommendations for millions of items and for millions of customers in real-time is critical for the success of a model, we leveraged Pinecone[1], a commercially available real-time ranking platform that can efficiently handle both the deep learning model transformations and vector search. Pinecone allows the lookup of product embeddings and computation of the session embedding on the fly, as well as a speedy tri-linear product computation against millions of products in the catalog. STAMP was decomposed into a query transformer and an item transformer both of which produced vectors whose dot product was the recommendation score. Shopping items were indexed by the platform by applying the item transformer offline and storing the resulting vectors in replicated nearest-neighbor retrieval servers. Query transformers were applied in real time to shopping carts as well as other session information. The service then used the query vector to retrieve highest scoring items. 99% of the customers experienced less than the 60ms latency when running the STAMP model inference (i.e., $p99 < 60$ms).

6.2 Experiment Design and Results

The experiment was conducted on the shopping cart page. After a customer added one or more items to the cart, and before checkout, the model recommended five products that ranked in the first positions. The A/B test followed the common randomized experiment protocol. The subjects of the experiment were customers who visited the shopping cart page of the site via a browser on a personal computer in the United States. Customers were split between the control and the treatment uniformly at random (50:50 split on the full traffic). Each customer consistently received either the treatment or the control. The metrics used in the A/B test were conversion rate and revenue per visit. Conversion rate is defined as the probability that a customer who is shown a recommendation, clicks on one of

[1] www.pinecone.io

the items, and *eventually* purchases it. Revenue per visit is akin to conversion rate but weighed according to product prices. It is defined as the total cart prices of all carts which contained recommended (and clicked) items divided by the number of website visits. The division only serves as a normalization term for the population size. The experiment showed the overall conversion rate of the treatment (STAMP) was 15.6% higher that that of the control (third-party). Finally, when measuring the revenue per visit we observed a 18.5% increase from the control to the treatment. Both differences are considered statistically significant with p-value < .001.

7 CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

In this work, we studied the performance of a set of models for session-based recommendations in the home improvement domain with the goal of finding the best model to put into production in a real e-commerce application, i.e., THD. To this end, we conducted three different evaluation procedures. First we performed an offline evaluation where we compared the performance of fourteen baseline and deep learning models using traditional metrics. We confirmed recent findings that baseline models are very strong in terms of precision and recall metrics. Next, we selected five algorithms that performed the best and ran a second round of evaluation using human experts. We found that SR-GNN that scored the best in metrics such as HR, MRR, NDCG scored third in the evaluation using human experts while V-STAN that ranked first in terms of precision, recall, and MAP, ranked last in this round. Out of the five models, the ones that ranked in the first three positions were all deep learning models (STAMP, GRU4REC, SR-GNN), suggesting that this family of models is more suitable for our application scenario. In the third round of evaluation, we ran an A/B test with real users where we compared STAMP with a third-party solution showing that STAMP statistically significantly improves conversion rate and revenue per visit.

In the future, we plan to build upon our work and further improve our findings as follows: (1) Although GRU4REC and STAMP performed best, for business reasons, we needed to avoid the risk of putting two new methods in front of real users. As a result, we compared one of the best two models (STAMP) with the third-party solution that, again for business reasons, we could not disclose the details of its inner workings. However, we stress that the incumbent solution is the result of significant engineering and science efforts by a large commercial entity. In our future work, and, based on our finding that STAMP is better compared to the third-party solution, we plan to conduct another round of A/B test between STAMP and GRU4REC. (2) The session-based recommendation models in this paper do not leverage the item content and, as a result, we cannot recommend items that do not appear in the training data. To overcome this limitation, we plan to leverage the attributes of the items (e.g., product title, description). (3) During the two evaluations, we found that different models reported better in different metrics. In the future, we plan to experiment with ensembling different models to get a combined benefit and overcome the limitations of single models. (4) Finally, we also plan to integrate the validators' feedback in the training data and establish a continuous human-in-the loop AI process.

ACKNOWLEDGMENTS

We would like to thank Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach for making the code of their framework available to the research community [23]. We also thank Mingming Guo, Cassi Hunt, Steve Bertolani, and Amir Sadoughi for providing insightful comments.

REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. 1993. Mining Association Rules Between Sets of Items in Large Databases. In *SIGMOD '93*.

[2] J. Beel and S. Langer. 2015. A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems. In *Research and Advanced Technology for Digital Libraries*.

[3] G. Bonnin and D. Jannach. 2014. Automated generation of music playlists: Survey and experiments. *CSUR* 47, 2 (2014).

[4] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *SSST '14*.

[5] P. Cremonesi, F. Garzotto, and R. Turrin. 2012. Investigating the Persuasion Potential of Recommender Systems from a Quality Perspective: An Empirical Study. *TiiS* 2, 2 (2012).

[6] M. Dacrema, P. Cremonesi, and D. Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. In *RecSys '19*.

[7] F. Garcin, B. Faltings, O. Donatsch, A. Alazzawi, C. Bruttin, and A. Huber. 2014. Offline and online evaluation of news recommender systems at swissinfo.ch. In *RecSys '14*.

[8] D. Garg, P. Gupta, P. Malhotra, L. Vig, and G. Shroff. 2019. Sequence and time aware neighborhood for session-based recommendations: Stan. In *SIGIR '19*.

[9] C. Gomez-Uribe and N. Hunt. 2016. The Netflix Recommender System: Algorithms, Business Value, and Innovation. *TMIS* 6, 4 (2016).

[10] A. Gunawardana and G. Shani. 2015. *Evaluating Recommender Systems*. Recommender Systems Handbook.

[11] N. Hariri, B. Mobasher, and R. Burke. 2012. Context-aware music recommendation based on latenttopic sequential patterns. In *RecSys '12*.

[12] B. Hidasi and A. Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *CIKM '18*.

[13] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. 2016. Session-based recommendations with recurrent neural networks. In *ICLR '16*.

[14] D. Jannach and M. Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *RecSys '17*.

[15] I. Kamehkhosh and D. Jannach. 2017. User Perception of Next-Track Music Recommendations. In *UMAP '17*.

[16] I. Kamehkhosh, D. Jannach, and M. Ludewig. 2017. A Comparison of Frequent Pattern Techniques and a Deep Learning Method for Session-Based Recommendation. In *TempRec '17*.

[17] L. Lerche, D. Jannach, and M. Ludewig. 2016. On the value of reminders within e-commerce recommendations. In *UMAP '16*.

[18] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma. 2017. Neural Attentive Session-Based Recommendation. In *CIKM '17*.

[19] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *KDD '18*.

[20] M. Ludewig and D. Jannach. 2018. Evaluation of session-based recommendation algorithms. *UMUAI* 28, 4 (2018).

[21] M. Ludewig and D. Jannach. 2019. User-centric evaluation of session-based recommendations for an automated radio station. In *RecSys '19*.

[22] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach. 2019. Empirical Analysis of Session-Based Recommendation Algorithms. *CoRR* abs/1910.12781 (2019).

[23] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach. 2019. Performance Comparison of Neural and Non-Neural Approaches to Session-Based Recommendation. In *RecSys '19*.

[24] S. McNee, J. Riedl, and J. Konstan. 2006. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06*.

[25] F. Mi and B. Faltings. 2018. Context tree for adaptive session-based recommendation. *arXiv preprint arXiv:1806.03733* (2018).

[26] M. Rossetti, F. Stella, and M. Zanker. 2016. Contrasting Offline and Online Results when Evaluating Recommendation Algorithms. In *RecSys '16*.

[27] K. Wagstaff. 2012. Machine Learning that Matters. In *ICML' 12*.

[28] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. de Rijke. 2019. A Collaborative Session-Based Recommendation Approach with Parallel Memory Modules. In *SIGIR'19*.

[29] S. Wang, L. Cao, and Y. Wang. 2019. A Survey on Session-based Recommender Systems. *CoRR* abs/1902.04864 (2019).

[30] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *AAAI '19*.

[31] F. Yuan, A. Karatzoglou, I. Arapakis, J. Jose, and X. He. 2019. A simple convolutional generative network for next item recommendation. In *WSDM '19*.