From Sequences to Profiles: Generating Universal Behavioral Profiles exploiting Recurrent Neural Networks

From Sequences to Profiles: Generating Universal Behavioral Profiles exploiting Recurrent Neural Networks

Simone Colecchia
Politecnico di Milano
Milano, Italy
simone.colecchia@mail.polimi.it

Mauro Orazio Drago
Politecnico di Milano
Milano, Italy
mauroorazio.drago@mail.polimi.it

Jihad Founoun
Politecnico di Milano
Milano, Italy
jihad.founoun@mail.polimi.it

Paolo Gennaro
Politecnico di Milano
Milano, Italy
paolo1.gennaro@mail.polimi.it

Ernesto Natuzzi
Politecnico di Milano
Milano, Italy
ernesto.natuzzi@mail.polimi.it

Luca Pagano
Politecnico di Milano
Milano, Italy
luca1.pagano@mail.polimi.it

Sajjad Shaffaf
Politecnico di Milano
Milano, Italy
sajjad.shaffaf@mail.polimi.it

Giuseppe Vitello
Politecnico di Milano
Milano, Italy
giuseppe1.vitello@mail.polimi.it

Andrea Pisani
Politecnico di Milano
Milano, Italy
Politecnico di Torino
Torino, Italy
andrea.pisani@polito.it

Maurizio Ferrari Dacrema
Politecnico di Milano
Milano, Italy
maurizio.ferrari@polimi.it

Abstract

This paper presents the solution developed by the EmbedNBreakfast team for the ACM RecSys Challenge 2025, for the construction of *Universal Behavioral Profiles*: general-purpose user representations derived from historical interactions. We propose a representation-learning framework that combines Recurrent Neural Networks, attention mechanisms, and collaborative filtering to jointly optimize embeddings across several predictive objectives. Our method achieved 2nd place on the Academic Leaderboard and 5th Overall, demonstrating the effectiveness of unified, representation-based modeling for diverse behavior prediction tasks.

CCS Concepts

• **Information systems** → **Recommender systems**; **Personalization**; *Learning to rank*; • **Computing methodologies** → *Neural networks*; *Supervised learning by classification.*

Keywords

ACM RecSys Challenge 2025, Universal Behavioral Profiles, Recommender Systems, Neural Networks

RecSysChallenge '25, Prague, Czech Republic

ACM ISBN 979-8-4007-2099-4/25/09
https://doi.org/10.1145/3758126.3758133

ACM Reference Format:
Simone Colecchia, Mauro Orazio Drago, Jihad Founoun, Paolo Gennaro, Ernesto Natuzzi, Luca Pagano, Sajjad Shaffaf, Giuseppe Vitello, Andrea Pisani, and Maurizio Ferrari Dacrema. 2025. From Sequences to Profiles: Generating Universal Behavioral Profiles exploiting Recurrent Neural Networks. In *RecSys Challenge 2025 (RecSysChallenge '25), September 22, 2025, Prague, Czech Republic.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3758126.3758133

1 Introduction

Forecasting user behavior for online platforms is a challenging goal, especially considering the need to combine several tasks such as recommendation, churn prediction, and lifetime value estimation. The ACM RecSys Challenge 2025[1], organized by Synerise[2], encourages a unified alternative: learning *Universal Behavioral Profiles* from rich interaction logs, evaluated on multiple downstream tasks using a fixed prediction head.

The challenge provides event-level logs and requires participants to submit embeddings representing users of a specified subset. These are automatically evaluated on a mix of open and hidden tasks via a common downstream architecture. Our team's approach[3] combines multi-task learning, sequential modeling, and collaborative signals to produce robust, versatile user profiles.

2 Problem Formulation

The goal is to learn embeddings from user event histories that generalize across diverse downstream tasks. Participants submit

[1]https://www.recsyschallenge.com/2025/
[2]https://www.synerise.com/
[3]Code: https://github.com/recsyspolimi/recsys-challenge-2025-synerise

embeddings of a specified subset of users for evaluation through an automatic system based on a deep neural predictor. The architecture and implementation of this model are made available to participants, allowing them to reproduce the evaluation process offline.

Evaluation covers both *open* tasks, whose definitions are known to the participants, and *hidden* tasks, which are not known and are used to assess the generalizability of the embeddings.

2.1 Data Description

The dataset provided by Synerise spans a six-month period and includes more than 150 million interactions. Notably, the competition was structured into two distinct *phases*, characterized by different datasets. The fundamental structure, event types, and overall interaction distribution of the datasets remained largely consistent across both phases; however, the evaluation and results presented in this paper pertain exclusively to the second and final phase of the Challenge.

The dataset consists of five event types:

- `product_buy`, `add_to_cart`, `remove_from_cart`: Log interactions tied to a specific numeric item ID named Stock Keeping Unit (SKU), representing product-level engagement.
- `page_visit`: Records anonymized URL IDs for visited pages.
- `search_query`: Anonymized interaction logs containing only the users' search queries. Each query is represented by a 16-dimensional quantized embedding generated from a Large Language Model (LLM).

The dataset includes additional metadata. Each SKU is associated with a numeric `category` ID, a `price` bucket (based on 100 quantiles), and a quantized 16-dimensional embedding of the product description obtained in the same way as search query embeddings.

2.2 Open Tasks

The open tasks are the following:

Churn Prediction: Binary classification task. Target $y_u^{(\text{churn})} = 1$ if user u has no `product_buy` events in the future 14-day prediction window, 0 otherwise.

Category Propensity: Multi-label prediction of purchases in the 100 most often purchased product categories. Let $C = \{c_1, \ldots, c_{100}\}$ denote the set of these categories. Target $\mathbf{y}_u^{(\text{cat})} \in \{0,1\}^{100}$, with $y_{u,c}^{(\text{cat})} = 1$ if user u buys from category c.

SKU Propensity: Multi-label prediction for the 100 most often purchased products, defined by their SKU. Let $\mathcal{S} = \{s_1, \ldots, s_{100}\}$ denote the set of these SKUs. Target $\mathbf{y}_u^{(\text{SKU})} \in \{0,1\}^{100}$, with $y_{u,s}^{(\text{SKU})} = 1$ if SKU s is purchased by user u.

Each task is evaluated using the AUROC metric. For category and SKU Propensity, scores consider novelty and diversity as follows:

$$\text{Score} = 0.8 \cdot \text{AUROC} + 0.1 \cdot \text{Novelty} + 0.1 \cdot \text{Diversity}. \quad (1)$$

Leaderboard rankings are computed using Borda count over all tasks, thereby encouraging the development of models with strong generalization capabilities across heterogeneous objectives.

3 Feature Engineering

To enable effective sequential modeling of user behavior, we constructed a unified interaction dataset by aggregating events of all types. Each user's behavioral history is represented as a time-ordered sequence of interaction vectors $x_1, x_2, \ldots, x_n$, where each $x_i \in \mathbb{R}^d$ encodes the event features of a single event. The dimensionality d reflects the total number of engineered features, including:

Counting features: Event-level counts.

Conversion rates and entropies: Statistics computed from user and item histories, such as user-wise conversion rates and entropy of item/category interactions.

Clustering-derived features: Semantic structure was introduced by clustering product name and search query embeddings using KMeans; resulting cluster IDs and centroids were included as features.

To reduce redundancy, search and visit events were grouped into sessions [9], using a 40-minute inactivity threshold. Aggregated features characterizing each session were appended to the corresponding interaction vectors. They consist of:

Duration: The amount of time the session lasted.

Nearby Event Count: The number of product-related events occurring during the session and up to 5 minutes after.

Top Product Flags: Binary flags indicating if any nearby event involved a top-selling product (SKU) or a top-selling category.

4 Model Architecture

Hybrid solutions have shown strong results in past Challenges [1, 2, 4, 8]. Thus, our solution involves a hybrid of multiple variants of three base models. This section outlines the base models that we explored for learning user embeddings, including sequence-based neural models and factorization approaches.

4.1 GRU-based Core Architecture Design

All base models utilize a GRU [7] encoder as the fundamental component for processing variable-length user interaction sequences. The encoder processes the event features through a GRU layer, followed by projection layers that map the final hidden state to a fixed-dimensional latent representation. This design choice ensures a consistent representation size across users, regardless of the sequence length.

4.1.1 ***Base Model 1: Sequence Reconstruction Autoencoder (SRA).*** This model adopts a symmetric encoder-decoder architecture [6] [11] trained via sequence reconstruction. The encoder consists of a GRU layer followed by a linear projection into the latent space, producing compact user representations. The decoder mirrors the encoder and attempts to reconstruct the original input sequence from the latent vector.

This model leverages the full behavioral history as both input and reconstruction target, thus maximizing the temporal context used to learn robust embeddings. The reconstruction loss is defined as the average mean squared error between original and reconstructed sequences:

$$\mathcal{L}_{\text{SRA}} = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 . \quad (2)$$

4.1.2 ***Base Model 2: Future Interaction Prediction (FIP).*** This architecture modifies the standard autoencoder paradigm by introducing a temporal shift, such that the model is trained on the first T

months of interaction data to predict interactions in the $(T+1)$-th month, in order to more closely resemble the downstream task.

We define the training loss by focusing exclusively on future interactions, beyond the training cutoff T. Specifically, let $\mathcal{F} = \{i \in \mathbb{N} \mid timestamp(x_i) > T\}$ be the index set of future interactions, and $\hat{x}_i$ the model's prediction for event x_i. The objective becomes:

$$\mathcal{L}_{\text{FIP}} = \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} \|x_i - \hat{x}_i\|^2 \tag{3}$$

This loss encourages the encoder to produce latent embeddings that summarize past behaviors in a way that is informative for the prediction of future events.

4.1.3 *Base Model 3: Multi-Task Learning with Explicit Objectives (MTL).*

The last model attempts to integrate the previous encoder-decoder architecture with the downstream tasks. This is achieved by replacing the decoder with three task-specific prediction heads, so that the learned embedding could be better optimized for all tasks. The architecture first processes input sequences using a refined GRU encoder which includes skip connections. The resulting outputs are then fed into a transformer encoder layer, which applies self-attention [12]. The attention mechanism enables the model to dynamically weight different temporal positions in the user's interaction history, capturing long-range dependencies that might be overlooked by the GRU alone.

The resulting latent user representation is then passed to the three task-specific heads [5], each comprising a shallow, fully-connected network, predicting the open tasks described in Section 2. A central design choice in this model is the intentional use of relatively low-capacity prediction heads, especially compared to the significantly more complex architectures employed in the neural predictor of the leaderboard's automated evaluation pipeline. This constraint prevents the heads from overfitting the supervised objectives and encourages the encoder to learn general-purpose representations.

The model is trained using a composite loss that combines the binary cross-entropy (BCE) objectives of the three prediction tasks in a weighted fashion:

$$\mathcal{L}_{\text{MTL}} = \lambda_1 \cdot \text{BCE}(\hat{y}^{(\text{churn})}, y^{(\text{churn})}) + \lambda_2 \cdot \frac{1}{|C|} \sum_{c=1}^{|C|} \text{BCE}(\hat{\mathbf{y}}_c^{(\text{cat})}, \mathbf{y}_c^{(\text{cat})}) + \lambda_3 \cdot \frac{1}{|\mathcal{S}|} \sum_{s=1}^{|\mathcal{S}|} \text{BCE}(\hat{\mathbf{y}}_s^{(\text{SKU})}, \mathbf{y}_s^{(\text{SKU})}) \tag{4}$$

where C and $\mathcal{S}$ denote the sets defined in Section 2.2.

As in the future prediction model, the MTL architecture is trained on user interaction data from the first T months observation window, while the $(T+1)$-th month is used to derive ground-truth labels for all three supervised tasks. Accordingly, $\hat{y}^{(\text{churn})}$ denotes the predicted churn probability, $\hat{\mathbf{y}}^{(\text{cat})}$ the predicted category propensities, and $\hat{\mathbf{y}}^{(\text{SKU})}$ the predicted SKU propensities. The targets $y^{(\text{churn})}$, $\mathbf{y}^{(\text{cat})}$, $\mathbf{y}^{(\text{SKU})}$ are defined as described in Section 2.

The weighting coefficients $\lambda_1, \lambda_2, \lambda_3$ control the relative influence of each task-specific objective on the shared representation learning, allowing us to steer the learned embeddings toward general-purpose or task-biased representations, depending on the desired evaluation outcome.

4.2 Matrix Factorization with BPR for Category and SKU Propensity

In addition to the GRU-based models defined above, we employed Matrix Factorization with Bayesian Personalized Ranking (BPR) [10] to estimate user preferences over product categories and SKUs. For both category-level and SKU-level predictions, we constructed a user-item interaction matrix by aggregating separate implicit interaction matrices (one for each type of product-related event, i.e. `product_buy`, `add_to_cart`, `remove_from_cart`) weighted by specifically tuned coefficients.

Each matrix ($\mathbf{X}^{(\text{cat})}$ and $\mathbf{X}^{(\text{SKU})}$) was factorized using BPR-based optimization to learn user representations $\mathbf{w}_u$ and item latent factors $\mathbf{h}_i$. Finally, predictions were obtained by projecting the user representations onto the latent factors of the items in sets C and $\mathcal{S}$:

$$\hat{\mathbf{y}}_u^{(\text{cat})} = \left[\mathbf{w}_u^\top \mathbf{h}_i^{(\text{cat})}\right]_{i \in C}, \qquad \hat{\mathbf{y}}_u^{(\text{SKU})} = \left[\mathbf{w}_u^\top \mathbf{h}_i^{(\text{SKU})}\right]_{i \in \mathcal{S}} \tag{5}$$

5 Evaluation Strategy

To ensure consistency between local validation and the official submission evaluation pipeline, we relied on the deep neural predictor provided by the organizers (see Section 2) for hyperparameter tuning and model selection. However, this approach required careful handling to prevent data leakage, where a model could learn from the target validation data during the representation learning phase, leading to inflated and unreliable results. To address this, we implemented a temporal shift strategy. For our predictive models (MTL and FIP), we trained them on a four-month window to predict the fifth month. We then generated embeddings using the data from the fifth month and passed these to the validation pipeline. This method ensures that the representations are generated without any exposure to the downstream validation targets. Such strategy was unnecessary for our SRA model; its reconstruction objective does not risk target leakage, so it was trained on the complete five-month period before its embeddings were evaluated using the same pipeline. A similar temporal split was applied to the MF model with BPR (see Section 4.2), which was trained on the first five months of data and validated on the sixth. This model's training set included a weighted aggregation of all 5 product-related interaction types. The AUROC metric was applied for validation to guide hyperparameter optimization. Notably, the validation set only included `product_buy` events, and focused only on the 100 most purchased products, in order to closely align the model's fine tuning to the Challenge's Propensity tasks.

6 Hybrid Models

The final hybrid model [3] was constructed through a multi-stage process of combination and concatenation of individual models. The final ensemble combines several models, each trained with a distinct objective to capture different facets of user behavior. The components were integrated as illustrated in the diagram (Fig. 1):

Table 1: Evaluation scores of our best models on the official submission system (leaderboard results).

Model	Churn	Cat	SKU	H1	H2	H3
SRA First Phase	0.7242	0.7937	0.7884	0.7420	0.7505	0.7961
SRA	0.7251	0.7929	0.7910	0.7426	0.7347	0.7968
MTL First Phase	0.7287	0.7889	0.7840	0.7441	0.7939	0.7939
MTL	0.7339	0.7965	0.7921	0.7530	0.8093	0.8000
SRA+MTL	0.7337	0.8017	0.8002	0.7545	0.8038	0.8038
SRA+MTL Reg.	0.7348	0.8033	0.8009	0.7558	0.8053	0.8047
MTL Propensity	0.7284	0.8003	0.8005	0.7443	0.8055	0.7988
FIP	0.7307	0.7969	0.7986	0.7486	0.8016	0.8000
GRU models	0.7350	0.8068	0.8070	0.7561	0.8119	0.8061
GRU + BPR-Cat	0.7360	0.8153	0.8122	0.7567	**0.8136**	0.8059
GRU + BPR-Cat+SKU	0.7364	0.8161	0.8130	0.7570	0.8005	0.8063
Final Hybrid (w/ MTL Churn)	**0.7368**	**0.8169**	**0.8137**	**0.7575**	0.7998	**0.8066**

SRA: the Sequence Reconstruction Autoencoder trained on all users and the whole 6 months.

SRA First Phase: as above but trained on the first phase dataset.

SRA Regularized: linear combination of SRA model and its *First Phase* version, using weights 0.7 and 0.3 respectively (see fig. 1).

MTL: Multi-Task Learning model trained on the first 5 months, using the 6th month as target on both Churn and Propensity tasks. The chosen task weighting coefficient are $\lambda_1 = 0.3, \lambda_2 = 0.35, \lambda_3 = 0.35$.

MTL First Phase: as above but trained on the first phase dataset.

MTL Regularized: linear combination of MTL model and its *First Phase* version, using weights 0.7 and 0.3 respectively (see fig. 1).

MTL Propensity: MTL model trained only on Propensity tasks ($\lambda_1 = 0, \lambda_2 = 0.5, \lambda_3 = 0.5$).

FIP: Future Interaction Prediction model, trained on all users on the first 5 months, the 6th month as target.

BPR SKU/Category: Matrix Factorization with BPR loss as described in sec. 4.2.

MTL Churn: MTL model, with a 32-dimensional latent space, trained only on the Churn task ($\lambda_1 = 1, \lambda_2 = 0, \lambda_3 = 0$).

7 Results

Table 1 summarizes the results obtained by our main models and their combinations in the official leaderboard. Several key insights emerge from this analysis.

A notable finding concerns the role of models trained on data from the previous competition phase (*"First Phase"*). As shown, both the *SRA First Phase* and *MTL First Phase* models underperform compared to their counterparts trained on the final dataset (*SRA* and *MTL*). However, integrating them proved beneficial: comparing the results of the base model concatenation (*SRA+MTL*) with its regularized variant (*SRA+MTL Reg.*), we observe consistent gains across all tasks, both public and hidden. This suggests that, despite their weaker standalone results, the *First Phase* models captured complementary patterns that enriched the final embeddings. While this *First Phase* models may not have been decisive for the final ranking, they proved useful as an effective regularization strategy, enhancing the generalization of our representations.

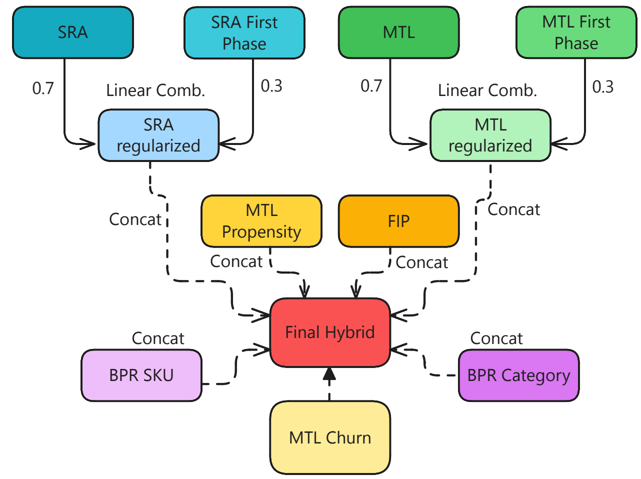

Figure 1: Structure of the final hybrid.

We further enriched our solution by concatenating representations from the *FIP* model, which is trained to predict future interactions, and the *MTL Propensity* model, optimized solely for the Propensity tasks. This choice was motivated by the observation that removing the Churn objective from the loss improved training stability. The combined results of these models are reported under the *GRU models* row in Table 1.

Furthermore, a significant boost in effectiveness on the Propensity tasks was achieved by appending scores from the Matrix Factorization model described in Section 4.2. As shown, including BPR scores for categories and SKUs led to substantial gains on the corresponding tasks.

Finally, the *MTL Churn* model contributed to an improved effectiveness on Churn Prediction.

8 Conclusion

In this paper, we presented the solution developed by the EmbedNBreakfast team for the ACM RecSys Challenge 2025, centered on the construction of Universal Behavioral Profiles through sequential modeling and collaborative filtering. Our approach integrated GRU-based architectures with attention mechanisms to capture the temporal dynamics of user behavior, and complemented these with a Matrix Factorization model trained via BPR to enhance collaborative signals. We demonstrated that combining multiple learning paradigms enables the construction of user embeddings that generalize effectively across diverse behavioral prediction tasks. Our ensemble strategy, which included both regularized and complementary models, proved effective in improving stability and robustness, leading to a strong effectiveness in both public and hidden tasks. These results underscore the potential of hybrid architectures in building scalable, task-agnostic user profiles.

Acknowledgments

We are grateful to our supervisors, Maurizio Ferrari Dacrema and Andrea Pisani, for their constant guidance and support. We also thank Michael Benigni and Riccardo Nembrini for their helpful suggestions, generously offered despite no obligation to assist.

References

[1] Andrea Alari, Lorenzo Campana, Federico Giuseppe Ciliberto, Saverio Maggese, Carlo Sgaravatti, Francesco Zanella, Andrea Pisani, and Maurizio Ferrari Dacrema. 2024. Exploiting Contextual Normalizations and Article Endorsement for News Recommendation. In *Proceedings of the Recommender Systems Challenge 2024, RecSysChallenge 2024, Bari, Italy, October 14-18, 2024*. ACM, 17–21. https://doi.org/10.1145/3687151.3687154

[2] Paolo Basso, Arturo Benedetti, Nicola Cecere, Alessandro Maranelli, Salvatore Marragony, Samuele Peri, Andrea Riboni, Alessandro Verosimile, Davide Zanutto, and Maurizio Ferrari Dacrema. 2023. Pessimistic Rescaling and Distribution Shift of Boosting Models for Impression-Aware Online Advertising Recommendation. In *ACM RecSys Challenge 2023, Singapore, 19 September 2023*. ACM, 33–38. https://doi.org/10.1145/3626221.3627288

[3] Robin D. Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Model. User Adapt. Interact.* 12, 4 (2002), 331–370. https://doi.org/10.1023/A:1021240730564

[4] Luca Carminati, Giacomo Lodigiani, Pietro Maldini, Samuele Meta, Stiven Metaj, Arcangelo Pisa, Alessandro Sanvito, Mattia Surricchio, Fernando Benjamín Pérez Maurera, Cesare Bernardis, and Maurizio Ferrari Dacrema. 2021. Lightweight and Scalable Model for Tweet Engagements Predictions in a Resource-constrained Environment. In *RecSysChallenge '21: Proceedings of the Recommender Systems Challenge 2021 (RecSysChallenge 2021), October 1, 2021, Amsterdam, Netherlands*. ACM, 8 pages. https://doi.org/10.1145/3487572.3487597

[5] Rich Caruana. 1998. Multitask Learning. In *Learning to Learn*, Sebastian Thrun and Lorien Y. Pratt (Eds.). Springer, 95–133. https://doi.org/10.1007/978-1-4615-5529-2_5

[6] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1724–1734. https://doi.org/10.3115/V1/D14-1179

[7] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR* abs/1412.3555 (2014). arXiv:1412.3555 http://arxiv.org/abs/1412.3555

[8] Nicola Della Volpe, Lorenzo Mainetti, Alessio Martignetti, Andrea Menta, Riccardo Pala, Giacomo Polvanesi, Francesco Sammarco, Fernando Benjamin Perez Maurera, Cesare Bernardis, and Maurizio Ferrari Dacrema. 2022. Lightweight Model for Session-Based Recommender Systems with Seasonality Information in the Fashion Domain. In *Proceedings of the Recommender Systems Challenge 2022* (Seattle, WA, USA) *(RecSysChallenge '22)*. Association for Computing Machinery, New York, NY, USA, 18–23. https://doi.org/10.1145/3556702.3556829

[9] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4 (2018), 66:1–66:36. https://doi.org/10.1145/3190616

[10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461. https://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf

[11] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi (Eds.). ACM, 111–112. https://doi.org/10.1145/2740908.2742726

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html