Enhancing News Recommendation with Real-Time Feedback and Generative Sequence Modeling

Qi Zhang
VMALL, Huawei Device Co., Ltd.
Wuhan, China
zhangqi193@huawei.com

Jieming Zhu
Huawei Noah's Ark Lab
Shenzhen, China
jiemingzhu@ieee.org

Jiansheng Sun
VMALL, Huawei Device Co., Ltd.
Wuhan, China
sunjiansheng1@huawei.com

Guohao Cai
Huawei Noah's Ark Lab
Shenzhen, China
caiguohao1@huawei.com

Ruining Yu
Beijing Jiaotong University
Beijing, China
ufpufwp@outlook.com

Bangzheng He
VMALL, Huawei Device Co., Ltd.
Shenzhen, China
hebangzheng@huawei.com

Liangbi Li
VMALL, Huawei Device Co., Ltd.
Shenzhen, China
liliangbi@huawei.com

ABSTRACT

Personalized news recommendation is a crucial technology for helping users discover news articles tailored to their interests. Key challenges in this field include modeling user preferences based on implicit behaviors, accounting for the influence of the news agenda on user interests, and managing the rapid decay of news items. The ACM RecSys Challenge 2024, organized by Ekstra Bladet, provide a large-scale news dataset for benchmarking news recommendation research. In this paper, we present our solution to the challenge. We propose real-time feedback learning mechanisms to capture users' immediate interests and explore generative sequence modeling techniques to learn impression-level user behaviors. Furthermore, we develop an ensemble method to combine tree models and deep models to improve recommendation accuracy. Based on this solution, our team ("hrec") achieved an impressive AUC score of 0.8667 on the final test set, securing the fifth place in the competition.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

RecSys Challenge, News Recommendation, Real-Time Feedback, Generative Sequence Modeling

ACM Reference Format:
Qi Zhang, Jieming Zhu, Jiansheng Sun, Guohao Cai, Ruining Yu, Bangzheng He, and Liangbi Li. 2024. Enhancing News Recommendation with Real-Time Feedback and Generative Sequence Modeling. In *ACM RecSys Challenge 2024 (RecSys Challenge '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3687151.3687158

RecSys Challenge '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-1127-5/24/10
https://doi.org/10.1145/3687151.3687158

Table 1: Top-5 teams on the leaderboard

Rank	Team Name	Score
1	:D	0.8924
2	BlackPearl	0.8815
3	Tom3TK	0.8707
4	axi2017	0.8692
5	hrec	0.8667

1 INTRODUCTION

The RecSys 2024 Challenge[1], orchestrated by experts from diverse institutions based on data from Ekstra Bladet, centers on online news recommendation, tackling both technical and normative obstacles. Participants are offered the Ekstra Bladet News Recommendation Dataset (EB-NeRD), which supports the creation of models to anticipate user clicks on news articles, leveraging user browsing history, session particulars, and personal metadata to rank candidate articles based on user preferences.

Our solution utilizes XGBoost [1] as the main model, which has scalability and efficiency in handling large datasets and can complete training in approximately one day. Due to the large amount of data, choosing XGBoost ensures that our system can quickly process large amounts of data, enabling timely updates and adaptation to user behavior patterns. When conducting comprehensive growth analysis and extracting precise features, XGBoost's ability to efficiently manage large datasets enables us to fully integrate test set data. This integration can capture subtle user interaction patterns, such as tracking clicks since March, which is crucial for enhancing user engagement and ensuring content relevance.

Besides, we employ deep learning-based models to enhance understanding of feature interactions and sequential behaviors. Generative sequence modeling is adopted to refine the ranking of news recommendations based on user interactions and contextual information. Specifically, our method includes capturing real-time

[1] https://www.recsyschallenge.com/2024/index.html

features, integrating prediction scores from models such as Deep and Cross Network (DCN) [4] and Deep Interest Network (DIN) [7]. Analyzing real-time user actions before clicking and effectively predicting intentions can help improve recommendation relevance. Inspired by recent developments in Large Language Models, we introduce the generative sequence modeling technique to create a more flexible and context-aware Click-Through Rate (CTR) prediction model. This approach has the potential to improve the accuracy and responsiveness of recommendation systems by more effectively capturing the dynamics of user interests in real-time. The multilingual support provided by UNBERT [6] allows us to directly analyze textual data from articles' title and body, expanding its applicability and influence. This hybrid method enriches our feature set and produces excellent performance metrics, ensuring that our recommendations are timely and relevant. As shown in Table 1, our team "hrec" achieved the 5th place on the final leaderboard[2].

Our contributions can be summarized as follows:

- Advanced Feature Engineering. We adopt tree-based models like XGBoost for efficiently handling large datasets and deep-learning models such as DCN, DIN and UNBERT for capturing higher-order feature interactions, sequential behaviors and textual context. The combination of these models allows for a more nuanced understanding of user preferences and interactions.
- Integration of Real-Time Feedback. We introduce real-time feedback mechanisms to enhance the accuracy of news recommendations, which can adapt to users' immediate interests and preferences, thereby improving the relevance and timeliness of recommended content.
- Generative Sequence Modeling. Inspired by large language models, we process sequence inputs in an auto-regressive manner to capture impression-level and session-level user behaviors. By framing the CTR prediction task to include both positive and negative feedback from user behaviors, our model can better adapt to changes in users' interests.

2 METHODS

In this section, we first introduce the methods we explored and the overall workflow for model ensemble. Next, we discuss two important techniques that significantly enhance our recommendation performance. Figure 1 provides an overview of our solution.

2.1 Overall Workflow

2.1.1 Tree Models. Tree-based Models, such as LightGBM [2] and XGBoost [1], are known for their predictive power and relatively fast training times even with large datasets. They can also handle both categorical and numerical features well, making them versatile for different types of tasks. Therefore we firstly employ XGBoost here as our prediction model for the quick online-offline validation, as well as the feature engineering and selection.

The feature importance derived from XGBoost provides insights into the factors that significantly influence click-through rates in news recommendations. Renowned for its speed and computational efficiency, particularly when dealing with extensive data sets, our XGBoost model generates a comprehensive set of 1,102 features. To streamline the training process and manage complexity, we meticulously select 300 features based on their feature importance scores. Furthermore, XGBoost's versatility in supporting multiple objective functions enables us to compare the performance of binary and ranking objectives. Our analysis reveals that a model utilizing pairwise loss consistently outperforms one relying on binary logistic loss, highlighting the crucial role of rankings or item order in prediction, rather than focusing solely on absolute values.

[2]Our best result achieved an AUC score of **0.8714**, but unfortunately, we failed the submission.

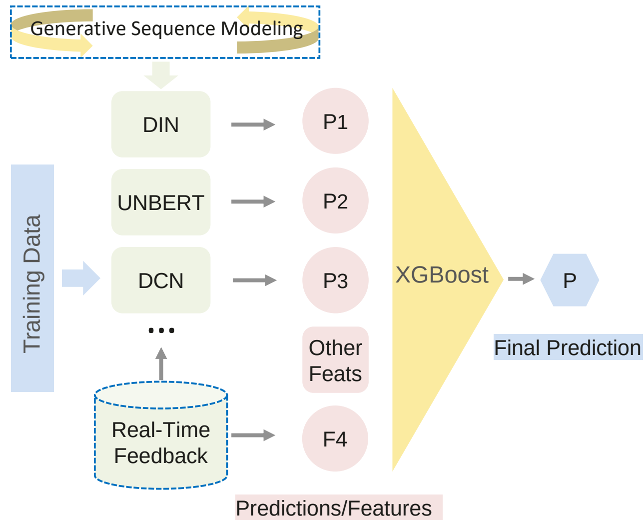

Figure 1: The overall workflow of our solution.

2.1.2 Deep Models. In order to leverage the rich sequence features and embedding features in the competition data which is hard to be introduced to tree-base model directly, we further explore a set of deep recommendation models for CTR prediction.

We firstly apply DCN to capture the higher-order feature interactions. The manually crafted features are not sufficient or effective. A more efficient feature extraction method is required, DCN uses the cross network structure to implement automatic, controllable, and efficient construction of the cross features, and explicitly presents a construction process of the cross features. In addition, the users' click histories collected over 21 days period are provided, hence we introduce DIN to involve this kind of information that reflect user interests recently. DIN can improve the correlation between the candidate news and historical behaviors by capturing strongly correlated the historical behaviors, and creates a splicing vector that varies with the candidate news through the attention mechanism. The longer the historical sequence, the better performance in our DIN experiment. Furthermore, the considerable textual data plays an important role in news recommendation, in order to obtain a good news representation as well as assimilate this textual representation into user's historical behavior, UNBERT [6] is introduced to enhance the semantic representation and captures multi-grained user-news matching signals within the behavior.

These deep models, with an end-to-end training workflow, are easy to handle the categorical features, sequence features, and even embedding features extracted from pretrained models, which, however, are difficult to train due to this large-scale data, especially for long sequences. Besides, most of our handcrafted features are numerical, which are not easily accommodated by these deep models, and their performance is worse than XGBoost's when using a large set of numeric features. Therefore, we also study how to integrate deep models and tree models via ensemble methods.

2.1.3 Ensemble. Ensemble methods are techniques that combine multiple models to improve the overall predictive performance, to leverage the strengths of individual models and offset their weaknesses by averaging their predictions or making decisions based on the consensus of the group. We first apply voting ensemble through weighted/robust average of probabilities, but it doesn't work in our approach, which is due to large gap of performance between deep models and tree models in our opinion. In order to involve these weak labels from deep models, we directly use multiple deep models' predictions as input features in our XGBoost training, and then learns to make a final decision. This allows for better transfer of knowledge between models in learning process, which improve the overall performance actually.

2.2 Enhancements

2.2.1 Real-Time Feedback. Real-time feedback plays a crucial role in modern online news recommendation systems, significantly impacting the Click-Through Rate (CTR) prediction task. In this challenge, the real-time feedback is manifested as real-time features for ranking model. Real-time features enable recommendation systems to capture users' instant interests and preferences more accurately. In practice, three categories of real-time features, as shown in Table 2, are implemented in our experiments.

Table 2: Real-time Feautures.

Category	Feature Name	Example
User Interaction	Recent Click History	ReadPageviewsByUserLastHour
	Dwell Time	ReadTimeMeanByUserLastMin
	Scroll Depth	ScrollMeanByUserLastDay
Content Popularity	Click-Through Rate	ReadPageviewsByArticleLastHour
	View Count	CategoryImpCountLastMin
Temporal	Time of Day	Hour
	Day of Week	Weekday
	Content Freshness	ArticleLastUpdateTimeGap

By capturing the most recent click history and interaction patterns, the system can adapt its recommendations swiftly, leading to higher CTR. We construct recent click history features that track the articles a user has clicked on in the last few minutes or hours, including the statistical metrics of category, dwell time, and scrolling depth. For instance, if a user has been reading articles about a breaking news event, the system can quickly prioritize related content, increasing the likelihood of clicks.

By incorporating temporal features, recommendation systems can align their suggestions with users' daily habits. Real-time temporal features allow the system to capture such patterns, potentially boosting CTR. For example, a user might prefer light entertainment news in the evening but business news in the morning. We construct many temporal features such as minute, hour, day, weekday, and temporal interval between adjacent items from the impression time fields of behavior and history data.

Table 3: Results of different models.

Type	Model	AUC
Deep	UNBERT	0.7207
	DIN	0.7909
	DCN	0.7935
Tree	XGBoost	0.8294
Ensemble	XGBoost + Deep (average)	0.8296
	XGBoost + Deep (learning)	0.8354

Real-time popularity features such as the number of views an article receives in short time intervals, help recommendation systems identify and promote trending content quickly. News articles experiencing sudden spikes in views or social media engagements can be prioritized, capitalizing on their momentum to increase CTR. This is particularly valuable in the fast-paced news environment where trending stories can rapidly gain and lose relevance.

2.2.2 Generative Sequence Modeling. Real-time features, however, only offer coarse-grained statistics for a specified period and do not capture impression-level or session-level user behaviors. To address this, we further explore generative sequence modeling techniques.

Existing CTR prediction models can be broadly categorized into two types. The first type (e.g., DCN [4] and FINAL [8]) frames the CTR prediction task as Equation (1), focusing mainly on interactions among features x (including user features, item features, and context features).

$$\hat{y} = f(x) = f(user, item, context) \tag{1}$$

The second type of models (e.g., DIN [7]) emphasizes sequential user behavior modeling, as shown in Equation (2).

$$\hat{y} = f([s_1, s_2, \cdots, s_l], x_{others}) \tag{2}$$

Here, $[s_1, s_2, \cdots, s_l]$ denotes the user behavior sequence of length l, and x_{others} represents other features. However, such sequential models only capture positive feedback from user behaviors (e.g., recently clicked sequences) and cannot adapt swiftly to changes in user interests indicated by negative signals.

In this work, inspired by generative sequence modeling techniques in large language models, we propose the following formulation for sequential CTR prediction:

$$\hat{y} = f\big((x_1, y_1), \cdots, (x_t, y_t), x_{t+1}\big) \tag{3}$$

Unlike the formulation in DIN, we construct each input with t previous samples $[(x_1, y_1), \cdots, (x_t, y_t)]$ within the same session, including both positive and negative samples as few-shot demonstrations. Similar to recent work in Generative Recommender [5], we process this sequence input in an autoregressive manner, predicting the label y_{t+1} based not only on x_{t+1} but also on the in-context demonstrations. This approach enables the model to infer immediate user interests from recent samples and labels, making it more adaptive to real-time interest evolution.

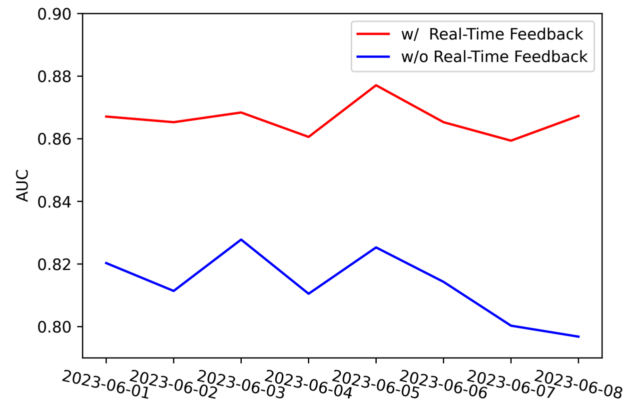

Figure 2: Test AUC w/ vs w/o real-time feedback.

However, training such a generative recommendation model from scratch is very resource-intensive and time-consuming, given the large volume of the EBNeRD dataset (about 490 million samples). Therefore, we decouple the training into two phases. First, we reuse the DIN model trained according to Equation (2) and extract the hidden representation for each sample. This results in the following sequence input for each session $[h_1, y_1, \cdots, h_t, y_t, h_{t+1}, ?]$. h_i represents a fixed 512-dimensional vector, and y_i denotes click labels (i.e., 0/1) encoded using two learnable embedding vectors. The input can thus be viewed as a token sequence, with the task being to predict the next label token (denoted by "?"). For simplicity, we use a causal transformer to model the given sequence and predict each label token autoregressively.

Meanwhile, as observed in recent work [3], leveraging listwise or pairwise loss can mitigate the issue of gradient vanishing for negative samples. Consequently, we employ the pairwise ranking loss in XGBoost and the Combined-Pair [3] loss in deep models. Finally, the generated click probabilities are fed into the ensemble for final ranking.

3 EXPERIMENTS

3.1 Training Details

All experiments are conducted on Ubuntu servers with Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz, 1000GB RAM, and NVIDIA Tesla V100 GPUs. We implement our deep models based on the FuxiCTR [9] framework[3], which are trained on the training set and generate the prediction scores on validation and test set. The models are trained for 2 epochs using AdamW optimizer with a batch size of 7168 and learning rate of 5e-4. We simply train the XGBoost on the validation set for 800 estimators with learning rate of 0.1 and the predict final scores on test set. Our source code is available at https://github.com/doubleQ2018/recsys-challenge-2024.

3.2 Results

The final leaderboard test set results are shown in Table 1, where teams were assessed using Area Under the Curve (AUC) metric.

[3]https://github.com/reczoo/RecSys2024_CTR_Challenge

Table 4: Ablation study of each useful component of our approach on small dataset.

Method	AUC	MRR	nDCG@5	nDCG@10
BASE	0.8294	0.6267	0.6924	0.7117
BASE w/o rtf	0.8091	0.5950	0.6609	0.6862
BASE w/o gsm	0.8124	0.6144	0.6802	0.7023

This competition saw an extensive participation of more than 100 teams, and our team, hrec, achieved the 5th place in the competition.

To analyze the effectiveness of our methodology, Table 3 presents a comparative evaluation. It is evident that XGBoost outperforms the deep learning models, indicating that the numerical real-time features incorporated in our strategy are particularly advantageous when combined with tree-based approaches. The ensemble method, xgboost + deep (learning), demonstrates significant enhancement, surpassing the standalone XGBoost model, while the simple averaging strategy, xgboost + deep (average), fails to yield additional improvements beyond XGBoost alone. This observation underscores the synergy between XGBoost and deep learning in our proposed solution.

3.2.1 *Ablation Study.*

Our final submission is conducted by an ensemble learning with real-time feedback and generative sequence modeling. To individually verify the effectiveness of each component in our approach, we test our model on the small dataset by disabling one component to evaluate the its contribution, yielding BASE (the complete approach), BASE w/o rtf (base without real-time features) and BASE w/o gsm (base without generative sequence modeling), as shown in Table 4. The results unambiguously demonstrate the profound boost in performance brought about by real-time feedback. Moreover, generative sequence modeling stands out in enhancing the ranking capabilities, significantly outperform the conventional point-wise binary target approach. This highlights the integral role of these techniques in our overall approach.

Figure 2 provides a compelling visual demonstration, where the decline in performance is clearly observed without real-time feedback as the day progresses. Conversely, the model that incorporates real-time feedback exhibits consistent stability. This striking contrast underscores the substantial advantage of real-time feedback in our strategy – not only does it enhance the overall efficacy, but also ensures the model's resilience, preventing the typical degradation over time.

4 CONCLUSION

In this paper, we present our solution for the RecSys Challenge 2024, specifically targeting online news recommendations. Our innovative final model relies on a powerful ensemble technique, seamlessly fused with real-time feedback mechanisms and generative sequence modeling tailored to news recommendation. We initially train multiple state-of-the-art methods to generate pseudo-supervised signals. These synthetic labels are then skillfully integrated with real-time features into a highly effective tree-based gradient boosting algorithm, driving the model toward superior final predictions. Our experimental results demonstrate the effectiveness of our approach, securing a top-5 position in the competition.

REFERENCES

[1] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 785–794.

[2] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, Vol. 30.

[3] Zhutian Lin, Junwei Pan, Shangyu Zhang, Ximei Wang, Xi Xiao, Shudong Huang, Lei Xiao, and Jie Jiang. 2024. Understanding the Ranking Loss for Recommendation with Sparse User Feedback. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

[4] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17*. 12:1–12:7.

[5] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, Yinghai Lu, and Yu Shi. 2024. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. In *International Conference on Machine Learning (ICML)*.

[6] Qi Zhang, Jingjie Li, Qinglin Jia, Chuyuan Wang, Jieming Zhu, Zhaowei Wang, and Xiuqiang He. 2021. UNBERT: User-News Matching BERT for News Recommendation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. 3356–3362.

[7] Guorui Zhou, Xiaoqiang Zhu, Chengru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep Interest Network for Click-Through Rate Prediction. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1059–1068.

[8] Jieming Zhu, Qinglin Jia, Guohao Cai, Quanyu Dai, Jingjie Li, Zhenhua Dong, Ruiming Tang, and Rui Zhang. 2023. FINAL: Factorized Interaction Layer for CTR Prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2006–2010.

[9] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open Benchmarking for Click-Through Rate Prediction. In *The 30th ACM International Conference on Information and Knowledge Management (CIKM)*. 2759–2769.