International Journal of Applied Science; Vol. 8, No. 1; 2025
ISSN 2576-7240 E-ISSN 2576-7259
https://doi.org/10.30560/ijas.v8n1p1

Exploration & Exploitation: A Unified Framework for Data Purification & Augmentation in Recommendation Systems

Cheng Junwei[1]

[1] Beijing University of Posts and Telecommunication, Beijing, China

Correspondence: Cheng Junwei, Beijing University of Posts and Telecommunication, Beijing, China. E-mail: junwei_cheng@bupt.edu.com

Received: December 7, 2024; Accepted: January 10, 2024; Published: January 14, 2025

Abstract

In the context of recommendation scenarios, the utilization of data purification and data augmentation methodologies has demonstrated their efficiency in enhancing the quality of representations. Nevertheless, within an open environment, the constrained nature of interaction data and the diverse range of interaction intentions present formidable challenges, giving rise to insufficient generalization ability and generalization bias in these methodologies. To address this issue, in this paper we introduce Exploitation & Exploration: A Unified Framework for Data Purification & Augmentation in Recommendation Systems, which not only ensures the precise purification of current data but also delves into potential noisy data for further exploration. To be specific, based on the traditional collaborative filtering method calculating user-item correlation, we first implement an efficient multi-head SENet block to remove potential noise from the interaction data. After this, we deploy a diffusion module to remove the added adversarial noise based on its ability to denoise all kinds of noise. And finally we use mutual-learning method to coordinate two parts' learning. We conducted experiments on three publicly available datasets, evaluating our model against current state-of-the-art algorithms in recommendation robustness tasks. The experimental results validate the effectiveness of our model.

Keywords: robust recommendation, data purification, data augmentation

1. Introduction

Recommendation systems predict users' future needs by analyzing user-item interactions. However, some individuals manipulate outcomes for profit, creating challenges in maintaining system robustness. To address this, it's crucial to reduce disruptive data and build accurate representations.

Data purification removes irrelevant or redundant information, helping the model focus on meaningful data. Adversarial learning generates challenging examples to train the model, improving its ability to handle complex data and enhancing robustness. While effective, traditional methods have limitations: denoising can only remove limited noise, and data augmentation may introduce bias, harming the model's generalization ability.To overcome these challenges, combining data purification and augmentation is essential. However, this integration faces two key issues: lack of labeled data and the need for dynamic combination. To solve this, we redefine recommendation system robustness as a representation learning task and introduce a new framework: Exploration & Exploitation: A Unified Framework for Data Purification & Augmentation in Recommendation Systems. This approach improves representations and enhances system robustness. Our Contributions are as follows:

1) Novel Framework: We redefine the problem and propose the first unified framework combining data purification and augmentation for recommendation systems.

2) Self-Supervised Learning Module: It iteratively removes noise from interaction data, producing purified embeddings. Then, a diffusion-based adversarial augmentation process further enhances the data by eliminating remaining noise. Finally, mutual learning dynamically integrates these processes.

3) Experimental Validation: Tests on three public datasets show our approach outperforms state-of-the-art methods in improving recommendation system robustness.

2. Related Works

2.1 Data Purification

The mainstream purification methods can be categorized into fine-grained methods and coarse-grained methods. In fine-grained methods, Liu et al.[1]proposed a two-stage automatic feature selection algorithm that embeds a regularized optimizer into model parameters to automatically identify and remove redundant interaction features. For item-level interactions, Gao et al.[2] employed meta-learning to use early-stage training information to guide subsequent learning, while Zhi et al.[3] used explicit feedback to guide the learning and denoising of implicit feedback. For noise within user-item interactions, Qin et al.[4] proposed reinforcement learning and contrastive learning methods to filter out item baskets unrelated to or negatively impacting the recommendation target, thereby improving recommendation performance. Noise in inter-item dependencies arises because not all interactions between user sequences or candidate items are strongly correlated. Zhang et al.[5], Cao et al.[6], and Chen et al.[7] modeled long-sequence dependencies using general selection modules (GSU) and fine-grained modeling modules (ESU) as interaction sequences became increasingly complex.

In coarse-grained denoising methods, one approach is to treat noise as a set of items and remove it by uncovering specific relational structures[8] or retrieving meta-paths from knowledge graphs[9]. For decisions involving multimodal information and the noise within it, Chen et al. [10] analyzed the impact of different parts within each modality on recommendation results using attention mechanisms, among other techniques. Li et al.[11] formalized the selection of beneficial modalities as a multi-agent collaborative Markov decision process using multi-agent reinforcement learning.

2.2 Data Augmentation

With the widespread application of recommendation systems, some external users have started attempting to manipulate recommendation outcomes by constructing malicious adversarial samples[12] to achieve profit-driven goals. For instance, Chen et al.[13] introduced discrete perturbations into user-item interactions to enhance the model's ability to resist interaction-level attacks. A more popular approach involves parameter perturbation-based enhancement, where models are trained by adding more perturbation factors at the parameter level to handle increasingly complex and diverse interaction data[14]. Similarly, adversarial perturbation methods based on user information[15]generate influential fake users to help the recommendation system learn the distribution of users that are more likely to cause interference, thereby improving the model's robustness. While existing methods demonstrate some effectiveness, they are often tailored to specific attack scenarios. Additionally, existing studies[16] indicate that these methods tend to overfit robustness features specific to adversarial samples, while neglecting certain non-robust features, which leads to a decline in predictive performance.

3. Method

3.1 Notations and Task Formulation

$\mathrm{u} \in \mathcal{U}$ and $\mathrm{v} \in \mathcal{V}$ denote users and items' information respectively. We use $\mathcal{D}$ to represent all training instances, and $(\mathrm{u},\mathrm{v}) \in \mathcal{D}$ is one training user-item instance. The matrix factorization model aims to minimize the empirical risk according to the following loss function:

$$\mathcal{L}(\mathcal{D};\Theta) = \sum_{(\mathrm{u},\mathrm{v})\in\mathcal{D}} \|uv^T - r\|^2 + \lambda \parallel \mathrm{U} \parallel^2 + \lambda \parallel \mathrm{V} \parallel^2 \tag{1}$$

Where $\mathrm{u} \in \mathrm{R}^{\mathrm{d}}, \mathrm{U} = [\mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_{\mathrm{m}}], \mathrm{v} \in \mathrm{R}^{\mathrm{d}}, \mathrm{V} = [\mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_{\mathrm{n}}]$. u and v represent embedding of u and v respectively, m and n are quantity of users and items in $\mathcal{D}$, and r is the score ranging from 1 to 5. Based on the learned model parameter $\Theta = \{\mathrm{U},\mathrm{V}\}$, we use x to denote the interaction emb of u and v. The main aim of this study is to train a data purification & augment module and use mutual learning part to combine them together. The structure of our model is as follows:

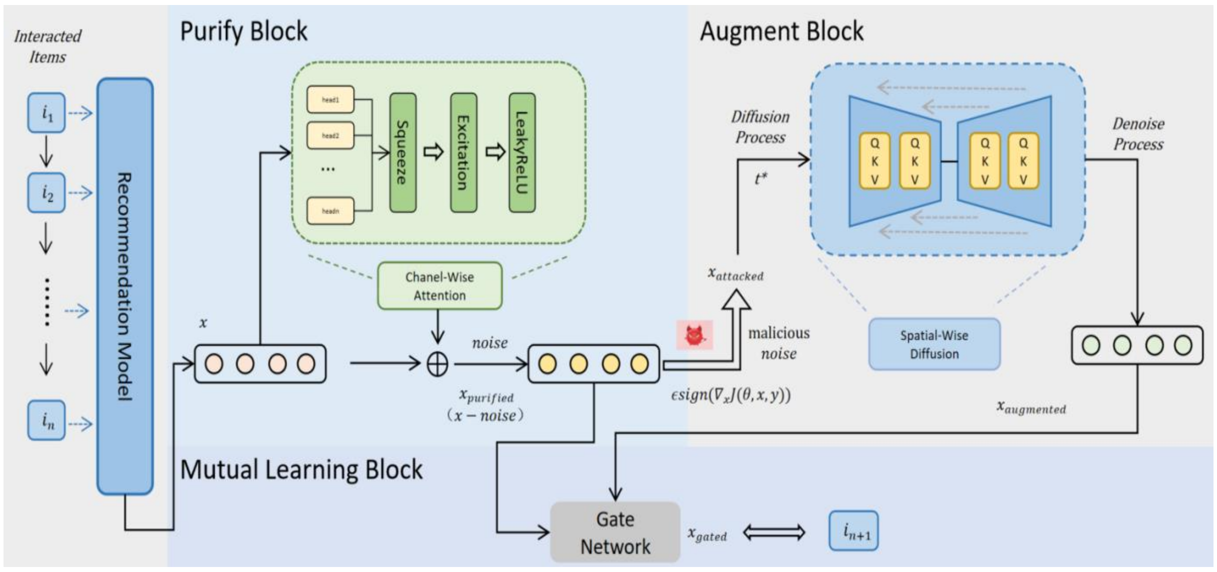

Figure 1. Structure of Data Purification & Augmentation Model in Recommendation System

3.2 Data Purification

For the goal of the data purification module, we model it as follows:

$$argmin\ L(f_{downstream}(x - f_{purify}(x)), y) \tag{2}$$

The goal of the above function is to remove the noise contained in *x* by using the data purification module. In this part, we use the SENet model and make relevant modifications to further improve its efficiency for feature extraction. Aiming at the problem that traditional SeNet networks have weak feature extraction capability and models such as DCN[17] have high feature cross-complexity, we proposes a multi-head SeNet structure that focuses on representation learning:

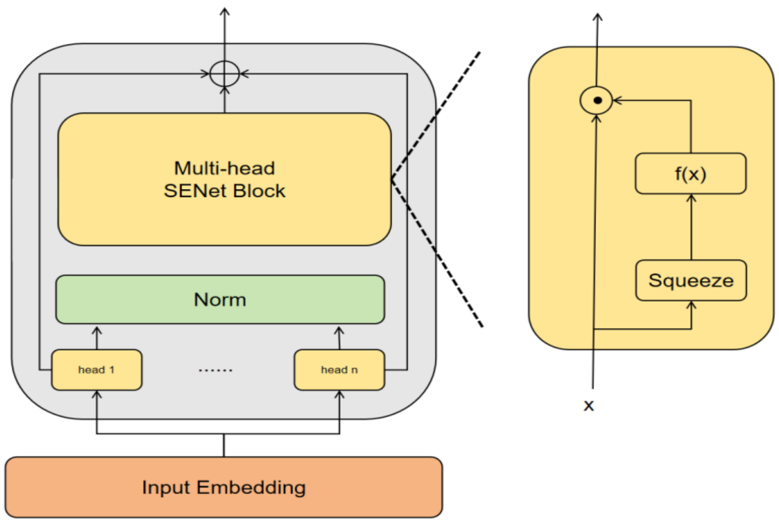

Figure 2. Structure of multi-head SENet Block

We assume that $X = [x_{h1}, x_{h2}, \ldots, x_{hn}]$, where hi is the *ith* head of attention. Then in the squeeze stage, we use this formula to calculate:

$$z_{hi} = F_s q(v_i) = 1/k \sum_{(t=1)}^{k} v_i^t \tag{3}$$

Where v_i is the *ith* dimensional feature of *x*. Due to the squeeze operation on *n* heads at the same time, the attention of different features can be calculated from *n* dimensions to better retain and fuse information.Then in the activation phase, multi-layer perceptrons is introduced to act on the output of the squeeze calculation:

$$S = F_{ex}(Z, W) = \delta\big(W_2\delta(W_1 Z)\big) \tag{4}$$

Then, using the idea of residual learning, the calculated invalid information is removed and the purified expression is sent to the downstream data enhancement module:

$$x_{purified} = x - S * x \tag{5}$$

3.3 Data Augmentation

In this part, we mainly generates adversarial noise through various schemes to achieve the training of the generative diffusion model. In this part, we mainly use gradient-based adversarial noise η. It is then added to the input as an adversarial sample

$$x_{attacked} = x_{purifed} + \eta \tag{6}$$

After obtaining the adversarial samples, we use the diffusion process to remove the noise in the obtained data. The training process of diffusion model is divided into two processes: denoising and denoising. Among them, the process from x_0 to x_T is the forward diffusion process, which is a series of noise addition operations to the original image until it becomes pure noise. And the corresponding denoising process is a reverse process, from noise to normal image. This process can be used to calculate the posterior probability through the Bayesian probability formula, then we can turn posterior probability into prior probability and convert it into the form of normal distribution:

$$P(x_{t-1} \mid x_t) = N\left(\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right), \frac{(1-\alpha_t)(1-\alpha_{\bar{t}-1})}{1-\overline{\alpha_t}}\right) \tag{7}$$

Because the coefficients about α are all constants, we only need to predict ϵ. Due to the problems that in adversarial training overfitting adversarial samples brings generalization bias and poor performance on normal data sets, we deploy diffusion to avoid this problem It does not depend on the attack form and the specific assumptions of the classification model. For the time step *t* of the noise process, we formalize it in the following format:

$$\frac{\partial D_{KL}(p_t \| q_t)}{\partial t} \le 0 \tag{8}$$

The diffusion timestep is the point when the distribution of clean data is closest to the distribution of attacked data, at which the derivative of their KL divergence with respect to *t* is less than 0. Given an adversarial sample x_a, the goal is to recover the original sample x to maximize the posterior distribution. The defense against adversarial samples is solved by optimizing the problem of finding the original sample that maximizes the posterior distribution by maximizing the log-likelihood of the posterior distribution. A variational posterior distribution is introduced to approximate the true posterior distribution in the original optimization target. Its variational upper bound is:

$$-logp(x_a) \le \mathbb{E}_{q(x)}[-logp(x_a \mid x)] + KL(q(x) \parallel p_\theta(x)) \tag{9}$$

So we simplifies the loss function into this form:

$$\mathcal{L}_{MSE}(x, x_a, \theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\epsilon\sim\mathcal{N}(0,I)}\left[\|D_\theta(x + \sigma_t\epsilon; t) - x\|_2^2 + \lambda\|x - x_a\|_2^2\right] \tag{10}$$

$D_\theta(x + \sigma_t\epsilon; t)$ is the output after denoising process with added Gaussian Noise at timestep *t*. The train process is as follows:

Algorithm 1: Data Augmentation Based on Diffusion Process

```
Input： Input from data purification module $x_{purified}$, Input with adversarial noise $x_{attacked}$, Diffusion denoising part $D_\theta$, Gaussian Noise added timestep $t$， Epoch $n$, Learning rate $\eta$
Output：Augmented representation $x_{augmented}$
$x_0 = x_{attacked}$
for i ϵ 0,…, M − 1 do
  Sample $\epsilon_1, \epsilon_2$ from $\mathcal{N}(0; I)$
    $x_{i,t} = x_i + \sigma_t\epsilon_1$
    $x_{a,t} = x_{attacked} + \sigma_t\epsilon_2$
Calculate grad of $x_i$ according to $\mathcal{L}_{MSE}$:
    $grad = \nabla_{x_i}\left[\left\|D_\theta(x_{i,t};t) - x_i\right\|_2^2 + \left\|D_\theta(x_{i,t};t) - D_\theta(x_{a,t};t)\right\|_2^2\right]$;
    $x_{i+1} = x_i - \eta \cdot grad$;
End
Return output $x_{augmented}$
```

Then me deliver the output $x_{augmented}$ into the mutual learning module below.

3.4 Mutual Learning

In this part, the above two module are coordinated through a unified loss function：

$$L_{\text{total}} = L_{fused} + T^2\left(L_{KL(x_{\text{purified}} \| x_{fused})} + L_{KL(x_{\text{augmented}} \| x_{\text{fused}})}\right) \tag{11}$$

L_{total} represents total loss function and uses MSE loss function. x_{fused} is fused by concatenate $x_{purified}$ and $x_{augmented}$ and send them through multi-layer perceptrons. $L_{KL(x_{purified} \| x_{fused})}$ and $L_{KL(x_{augmented} \| x_{fused})}$represent the difference between fused result and initial distribution $x_{purified}$ or $x_{augmented}$ to ensure the consistency of the results before and after fusion. And *T* is an adjustable hyperparameter to control the weight of the loss function.

4. Experiments

In this section, we experiment our proposed approach and a variety of robustness enhancement methods under different adversarial attack scenarios. The experimental results verify the effectiveness of our proposed method

4.1 Dataset and Experimental Settings

In our experiment, we used the dataset of MovieLens 1M, MovieLens 100K and YELP. According to the network layers and parameter requirements of the model, this task runs on the online server with Intel Xeon Gold CPU@3.30Hz*64 512G and two NVIDIA V100 graphics cards. The operating system of this task is based on Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-62-generic x86_64), Python 3.8, PyTorch 1.11.

4.2 Baselines and Metrics

4.2.1 Attack Methods：

Random Attack: Assigns the highest score to the target product, while other product scores are randomly assigned based on a Gaussian distribution derived from overall ratings.

Average Attack: Gives the highest score to the target product and generates scores for other products using a Gaussian distribution centered around the average score of selected products.

AUSH Attack[18]: Uses GAN-based models to generate synthetic user data from real user data, augmenting the dataset with stealthy synthetic users.

TNA Attack: Focuses on highly interactive users. It optimizes score differences between the target product and other products within this user group.

DADA Attack[19]: Targets vulnerable users who are more susceptible to manipulation, aiming to maximize impact while minimizing detection, creating a more efficient and far-reaching attack.

4.2.2 Defense Baseline:

PCMF[20]: Enhances robustness by addressing the error matrix from discrepancies between predicted and actual ratings in matrix factorization.

APT: Simulates poisoning attacks by injecting synthetic users to minimize empirical risk, improving adversarial resistance.

GDA: Uses adversarial training with generative data augmentation to optimize the model by maximizing loss, boosting performance under attacks.

RGCF[21]: Employs graph neural networks to filter out unreliable interactions, improving recommendation quality and system reliability.

4.2.3 Metrics

Here we use HitRatio(HR@K) as the evaluation metric. It measures the proportion of recommended items that a user actually interacts with or selects from the list provided. 4.3 Model Parameters

In our experiments, we set the value of k to 50 for the ML-100K and ML-1M datasets., and 1000 for the Yelp dataset due to its sparsity. We randomly selected 10 products as attack samples and then recorded the model's performance under various attack conditions.

4.4 Overall Performance Comparison

Table 1. The performance of target items (robustness)

Dataset	Attack	Target item = 10							
		Original	Attacked	AT	PCMF	RGCF	GDA	APT	Ours
ML-100K (HR@50)	Random	4.34	19.83	16.35	9.83	11.63	11.70	6.77	6.19
	Average	4.34	31.33	24.38	9.04	15.04	19.89	9.17	9.32
	AUSH	4.34	41.37	29.22	20.11	30.02	27.63	18.99	16.02
	TNA	4.34	46.68	27.63	21.40	26.31	23.35	19.34	17.65
	DADA	4.34	52.39	33.98	25.08	29.83	31.07	17.22	15.67
ML-1M (HR@50)	Random	0.01	12.35	8.33	8.83	9.64	8.84	8.13	7.49
	Average	0.01	31.44	14.73	10.61	18.31	16.64	9.17	9.86
	AUSH	0.01	43.35	23.32	18.17	19.35	21.44	9.32	8.65
	TNA	0.01	47.67	27.66	16.80	23.34	21.47	13.44	14.72
	DADA	0.01	56.94	32.33	19.67	26.08	23.36	16.74	14.97
Yelp (HR@1000)	Random	1.31	4.33	4.21	2.87	3.15	3.66	2.94	2.31
	Average	1.31	5.72	5.11	3.14	4.77	5.19	3.07	3.48
	AUSH	1.31	13.61	12.91	9.35	9.81	8.17	6.35	5.48
	TNA	1.31	7.34	7.41	7.37	6.35	6.85	4.41	5.64
	DADA	1.31	14.98	11.36	7.24	10.15	11.33	6.14	6.27

Compared with the initial model, the HitRatio of the attacked goods specified at the time of attack will increase significantly in the attacked model, while the recommended model using the robustness enhancement method will effectively suppress the increase in the click-through rate of this part of maliciously attacked goods. Our method presents better robustness than most of the current comparison baseline defense performance.

4.5 Ablation Study

In order to further verify the effectiveness of the proposed algorithm, we also conducted relevant ablation experiments to verify the results:

Table 2. The ablation experoment of each part

Dataset	Attack	Original	Attacked	AT
ML-100K (HR@50)	Random	4.34	19.83	16.35
	Average	4.34	31.33	24.38
	AUSH	4.34	41.37	29.22
	TNA	4.34	46.68	27.63
	DADA	4.34	52.39	33.98
ML-1M (HR@50)	Random	0.01	12.35	8.33
	Average	0.01	31.44	14.73
	AUSH	0.01	43.35	23.32
	TNA	0.01	47.67	27.66
	DADA	0.01	56.94	32.33
Yelp (HR@1000)	Random	1.31	4.33	4.21
	Average	1.31	5.72	5.11
	AUSH	1.31	13.61	12.91
	TNA	1.31	7.34	7.41
	DADA	1.31	14.98	11.36

Ablation experiments are conducted on the contribution of the data purification module to further analyze the difference between directly enhancing the data and using the data purification method to remove the natural noise in the data, and then using the data enhancement method to remove the counter noise in the data. The results of ablation experiments show that the additional data purification module can effectively cooperate with the data enhancement module to provide at least 10%+ additional data enhancement capability, thus enhancing the robustness of the model.

5. Conclusion

In this paper, aiming at the problems of lack of generalization and generalization bias in traditional data purification and data augmentation work, we propose a innovative framework for mutual learning of data purification and data augmentation tasks. Realizing collaborative enhancement through self-supervised learning strategy and diffusion model-based adversarial noise removal. A model-agnostic robust enhancement method for recommendation is proposed to solve the problem that most current methods need to design a special model structure according to the scene.

We conducted experiments on three real-world data sets to verify the effects of our model. The results show that the effect of our model greatly exceeds the current baseline model, and the ablation test also validates the mutual enhancement effect of the two tasks of data purification and data enhancement.

Reference

[1] Liu, B., Zhu, C., Li, G., Zhang, W., Lai, J., Tang, R., He, X., Li, Z., & Yu, Y. (2020). AutoFIS: Automatic feature interaction selection in factorization models for click-through rate prediction. *KDD, 2636-2645*.

[2] Gao, Y., Du, Y., Hu, Y., Chen, L., Zhu, X., Fang, Z., & Zheng, B. (2022). Self-guided learning to denoise for robust recommendation. *SIGIR, 1412-1422*. https://doi.org/10.1145/3477495.3532056

[3] Bian, Z., Zhou, S., Fu, H., Yang, Q., Sun, Z., Tang, J., Liu, G., Liu, K., & Li, X. (2021). Denoising user-aware memory network for recommendation. *RecSys, 400-410*. https://doi.org/10.1145/3460231.3471690

[4] Qin, Y., Wang, P., & Li, C. (2021). The world is binary: Contrastive learning for denoising next basket recommendation. *SIGIR, 859-868*. https://doi.org/10.1145/3404835.3462906

[5] Zhang, Y., Chen, E., Jin, B., Wang, H., Hou, M., Huang, W., & Yu, R. (2022). Clustering-based behavior sampling with long sequential data for CTR prediction. *SIGIR, 2195–2200*. https://doi.org/10.1145/3404835.3463002

[6] Cao, Y., Zhou, X., Feng, J., Huang, P., Xiao, Y., Chen, D., & Chen, S. (2022). Sampling is all you need on modeling long-term user behaviors for CTR prediction. *CIKM, 2974–2983*. https://doi.org/10.1145/3340531.3412047

[7] Chen, Q., Pei, C., Lv, S., Li, C., Ge, J., & Ou, W. (2021). End-to-end user behavior retrieval in click-through rate prediction model. *CoRR, abs/2108.04468*. https://arxiv.org/abs/2108.04468

[8] Zhang, Y., Ai, Q., Chen, X., & Wang, P. (2018). Learning over knowledge-base embeddings for recommendation. *CoRR, abs/1803.06540*. https://arxiv.org/abs/1803.06540

[9] Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., & Li, Y. (2019). Metapath-guided heterogeneous graph neural network for intent recommendation. *KDD, 2478-2486*. https://doi.org/10.1145/3293663.3293685

[10] Chen, X., Zhang, Y., Xu, H., Cao, Y., Qin, Z., & Zha, H. (2018). Visually explainable recommendation. *CoRR, abs/1801.10288*. https://arxiv.org/abs/1801.10288

[11] Li, K., Wang, P., & Li, C. (2022). Multi-agent RL-based information selection model for sequential recommendation. *SIGIR, 1622-1631*. https://doi.org/10.1145/3404835.3462904

[12] Christakopoulou, K., & Banerjee, A. (2019). Adversarial attacks on an oblivious recommender. *RecSys, 322-330*. https://doi.org/10.1145/3293663.3293677

[13] Chen, H., Zhou, K., Lai, K. H., Hu, X., Wang, F., Yang, H. (2022). Adversarial graph perturbations for recommendations at scale. *SIGIR, 1854-1858*. https://doi.org/10.1145/3397271.3401300

[14] Chen, H., & Li, J. (2019). Adversarial tensor factorization for context-aware recommendation. *RecSys, 363-367*. https://doi.org/10.1145/3293663.3293665

[15] Wu, C., Lian, D., Ge, Y., Zhu, Z., Chen, E., & Yuan, S. (2021). Fight fire with fire: Towards robust recommender systems via adversarial poisoning training. *SIGIR, 1074-1083*. https://doi.org/10.1145/3404835.3462902

[16] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., & Madry, A. (2019). Adversarial examples are not bugs, they are features. *NeurIPS, 125-136*. https://doi.org/10.5555/3328779.3328937

[17] Wang, R., Fu, B., Fu, G., & Wang, M. (2017). Deep & cross network for ad click predictions. *ADKDD@KDD, 12*(1), 12–7. https://doi.org/10.1145/3107323.3107325

[18] Lin, C., Chen, S., Li, H., Xiao, Y., Li, L., & Yang, Q. (2020). Attacking recommender systems with augmented user profiles. *CIKM, 855–864*. https://doi.org/10.1145/3340531.3411913

[19] Li, S., Kawale, J., & Fu, Y. (2015). Deep collaborative filtering via marginalized denoising auto-encoder. *CIKM, 811–820*. https://doi.org/10.1145/2806416.2806507

[20] Li, H., Di, S., & Chen, L. (2022). Revisiting injective attacks on recommender systems. *NeurIPS*.

[21] Huang, H., Mu, J., Gong, N. Z., Li, Q., Liu, B., & Xu, M. (2021). Data poisoning attacks to deep learning based recommender systems. *NDSS*.