Model-Agnostic Poisoning Attacks on Recommender Systems via PPO

Vincenzo Agate, Giuseppe Lo Re, Marco Morana and Antonio Virga
*Department of Engineering, University of Palermo, Palermo, Italy.
Email: vincenzo.agate@unipa.it, giuseppe.lore@unipa.it, marco.morana@unipa.it, antonio.virga01@unipa.it

***Abstract*—Recommender systems have become pivotal in modern digital platforms, guiding user choices and driving engagement. However, their widespread adoption has also made them a prime target for adversarial attacks, especially data poisoning attacks that subtly manipulate recommendations. Existing approaches often generate unrealistic fake profiles, making them vulnerable to detection by anomaly-based defenses. In this paper, we propose a novel, model-agnostic poisoning framework that combines contrastive learning and reinforcement learning with Proximal Policy Optimization (PPO) to craft highly realistic fake profiles derived from cross-domain user data. By interacting exclusively with a surrogate recommender trained on a compatible domain, our framework identifies and fine-tunes influential user profiles to maximize the impact on a black-box target system. Our experimental evaluation on real-world datasets shows that our approach successfully promotes target items across diverse recommendation models with minimal injection effort, outperforming baseline strategies.**

***Index Terms*—Data Poisoning, Recommender Systems, Reinforcement Learning.**

I. Introduction

In the digital era, Recommender Systems (RS) have become essential tools for mitigating the challenge of information overload on online platforms. By assisting users in navigating extensive choices and providing personalized recommendations, these systems enhance both user experience and satisfaction [1]. Recommender Systems are widely applied across various domains, including e-commerce, streaming services, and social media, where they not only improve user engagement but also contribute to increased revenue for platforms. However, their pivotal role also exposes them to targeted threats, such as adversarial attacks, which can manipulate recommendation outputs to unfairly promote or suppress specific products or services [2].

One of the most prevalent attacks on Recommender Systems is data poisoning, where adversaries inject malicious data, such as fake user profiles, into the system to manipulate its recommendation models. These attacks represent a significant threat, as they enable adversaries to influence recommendations for commercial, political, or strategic purposes [3].

Many existing approaches focus on generating fake profiles from scratch. Although effective to some extent, such methods suffer from critical drawbacks: the synthetic profiles they create are often unrealistic and can be readily identified by anomaly-based defense mechanisms.

A more insidious threat to recommender system robustness involves exploiting compatible domains or those with partial item overlaps, as seen between platforms with similar catalogs (e.g., eBay and Amazon, Netflix and IMDb, Spotify and Apple Music). This approach leverages authentic user profiles by copying and adapting them to the context of the target domain. By making minor adjustments to existing user behaviors, it becomes possible to create pseudo-realistic profiles that are more convincing and effective than synthetically generated ones. This strategy is particularly advantageous in black-box scenarios, where direct access to the structure or parameters of the target system is unavailable. A significant challenge in attacks of this nature lies in identifying the most influential profiles, those that maximize the impact on the recommendation system [4]. Given that source domains often contain millions of user profiles, traditional methods address this issue by reducing dataset dimensionality through clustering-based selection techniques or by employing influence functions to quickly estimate the profiles' impact. While these strategies help manage computational complexity, they come with inherent limitations: classical clustering techniques may discard potentially valuable profiles, and heuristic-based approaches often result in suboptimal solutions.

To overcome these challenges, our attack framework leverages contrastive learning (*CL*) to reduce the dimensionality of the profile search space while retaining a diverse and influential pool of profiles. Contrastive learning optimizes the representation of user-item interactions in a latent space, maximizing the similarity between instances within the same cluster and minimizing similarity across different clusters. This process enables more effective clustering and improves the selection of profiles for a successful attack.

To address the challenge of identifying the most influential profiles, we propose an innovative approach based on reinforcement learning (RL), leveraging the Proximal Policy Optimization (PPO) algorithm [5]. PPO is renowned for its stability and its capacity to balance exploration and exploitation, both critical in high-dimensional scenarios such as this.

Our framework includes a PPO agent that evaluates all available profiles in the compatible domain obtained through contrastive learning (*CL*) and determines minor changes to the profiles to maximize their impact on a target system; we also leverage a surrogate model trained on the compatible domain, to overcome the common limitation of restricted access to the target system. This surrogate model allows the PPO agent to learn an effective policy in creating profiles to inject into the target system, providing transferable feedback that will be useful in launching an attack.

The main contributions of the work can be summarized as follows:

- a novel attack framework based on reinforcement learning and Proximal Policy Optimization (PPO) for black-box recommender systems is proposed. The framework learns

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

effective attack strategies by interacting with a surrogate model and leveraging user profiles from a compatible domain;
- in order to identify the most influential profiles, we reduce the dimensionality of the search space through a combination of contrastive learning and clustering, preserving diverse and impactful representations;
- selected profiles from a compatible domain are slightly modified to retain realistic user behavior patterns, enhancing the stealthiness of the injection.
- the effectiveness of our approach is validated by comprehensive experiments on real-world datasets across multiple state-of-the-art recommender models.

The rest of the paper is organized as follows: Section II reviews related work. Problem formulation is presented in Section III. Section IV describes the attack framework and architecture of the proposed solution. Section V details the experiments and results obtained from this study. Finally, Section VI outlines the conclusions of this work.

II. Related Work

In recent years, security in recommender systems has received increasing attention, particularly in the context of adversarial attacks. These attacks can compromise recommendation quality and system reliability by exploiting system vulnerabilities in analyzing user-item interactions.

In a poisoning attack, the attacker attempts to contaminate the training data and then tamper with the model so as to compromise its integrity and lead the model to generate undesired or malicious predictions [6]. This allows adversaries to attack even the most advanced training algorithms and complex models [7].

The transferability of poisoning attacks is evident when an attack designed to compromise a surrogate model remains effective against the target model, without requiring direct access to it. However, executing transferable poisoning attacks presents significant challenges, particularly for availability attacks, as these require compromising the global behavior of the target model. In contrast, transferable integrity attacks are relatively simpler to implement, as they focus on more specific objectives, such as influencing predictions on a limited set of data points. These attacks typically require a lower level of perturbation to achieve their goal [8], [9].

Although adversarial training has long been applied in domains such as intrusion-detection [10], where the goal is to push classifiers toward outright misclassification, adversarial activity in recommender systems pursues a subtler objective: covertly steering user choices by manipulating the recommendations themselves, while the system otherwise appears to function normally. The data injected into the training set typically relates to fake users *(shilling attack)* or fake ratings, in an attempt to change the resulting recommendations into what the attacker wishes to accomplish, to promote or demote one or more items. Most of the attack methods in the literature are tested and presented to attack white-box or gray-box systems [11], [12], where an attacker is able to access the entire information set (i.e. models and parameters) with which the target recommender is defined.

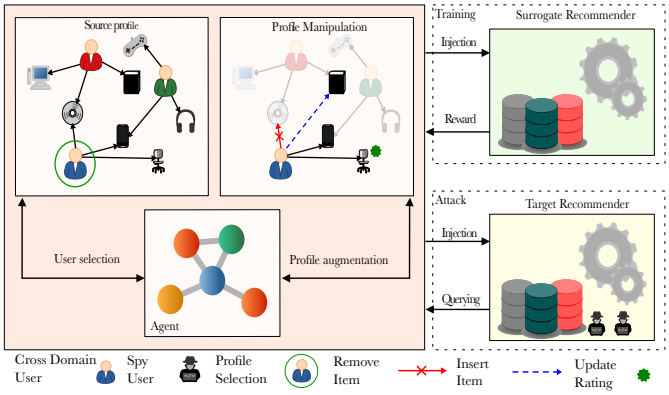

Fig. 1: Overview of the proposed poisoning attack framework.

An interesting approach in poisoning attacks is the use of the influence function to identify and exploit the most influential users in the recommendation system. These users, whose behavior has a significant impact on the prediction models, are selected to maximize the effectiveness of the attack with a reduced number of manipulations, as demonstrated in several recent works that have adopted this strategy to optimize the injection of poisoned data [13]. LOKI [14] is a recent RL-based method that generates fake profiles for matrix factorization models in black-box settings. However, it assumes full access to the training data, produces synthetic profiles easily detected by simple defenses, and is not applicable to deep models such as RNNs or transformers due to the prohibitive cost of Hessian computation [15]. An emerging trend in poisoning attacks is the application of reinforcement learning (RL) techniques to optimize manipulation strategies. These approaches leverage RL's ability to model sequential and adaptive scenarios, allowing attackers to learn optimal policies for poisoning data dynamically and maximize their impact on the recommendation system. Several recent studies have shown that the use of RL can significantly improve the effectiveness of attacks by adapting to the target model's responses during the attack [3]. CopyAttack [16] leverages reinforcement learning to select user profiles from a compatible source domain for black-box poisoning. While effective under certain conditions, it does not enforce profile diversity or realism, leading to homogeneous and easily detectable fake users. Moreover, it lacks mechanisms to adaptively fine-tune profiles for different target models, which limits its transferability across heterogeneous recommenders. In contrast, our framework addresses these limitations by: (i) applying contrastive learning and clustering to ensure a diverse pool of influential source profiles; (ii) using PPO-based fine-tuning on real profiles to preserve credibility; (iii) optimizing attack policies via surrogate interactions to enhance transferability across multiple target recommender architectures.

Therefore, our contribution lies in delivering a diverse, realistic, and model-agnostic poisoning strategy that is better suited for stealthy and robust attacks in real-world mobile recommendation scenarios.

III. Problem Formulation

We consider a target recommender system $\mathcal{T}$ treated as a black-box model: its internal algorithms and parameters are unknown, but its output recommendations are observable. Formally, $\mathcal{T}$ consists of a user set $\mathcal{U}_\mathcal{T}$, an item set $\mathcal{I}_\mathcal{T}$, and a sparse interaction matrix $\mathcal{R}_\mathcal{T}$, where each element $r(u,i)$ encodes an interaction between user $u \in \mathcal{U}_\mathcal{T}$ and item $i \in \mathcal{I}_\mathcal{T}$.

The goal of $\mathcal{T}$ is to produce an ordered list of top-k items for each user based on observed interactions. In this setting, an attacker aims to manipulate $\mathcal{T}$ by injecting fake user profiles to promote or demote specific items, or to degrade overall recommendation quality.

For promotion, the objective is to increase the likelihood that target items $\mathcal{I}^* \subseteq \mathcal{I}_\mathcal{T}$ appear in the top-k recommendations for a target user subset $\mathcal{U}_\mathcal{T}^{\text{target}} \subseteq \mathcal{U}_\mathcal{T}$. Demotion instead aims to reduce the ranking of certain items $\mathcal{I}' \in \mathcal{I}_\mathcal{T}$ by amplifying the competition from non-target items. In practice, both strategies exploit the same ranking mechanisms by altering relative item popularity.

Formally, the attacker's goal is to maximize:

$$\max \sum_{u \in \mathcal{U}_\mathcal{T}^{\text{target}}} \sum_{i^* \in \mathcal{I}^*} \mathbb{P}(i^* \in \mathcal{L}(u)), \tag{1}$$

where $\mathbb{P}(i^* \in \mathcal{L}(u))$ is the probability that item i^* appears in the top-k list $\mathcal{L}(u)$ for user u. Conversely, a demotion objective seeks to minimize this probability for items $\mathcal{I}'$.

The attacker uses a *data poisoning* strategy, injecting a set of fake profiles $\mathcal{U}_\mathcal{F} = \{u_f^1, u_f^2, \ldots, u_f^{|\mathcal{U}_\mathcal{F}|}\}$ crafted to bias the ranking. These profiles must be realistic enough to evade detection and diverse enough to influence user-item correlations effectively.

Finally, the attacker adapts its strategy by observing the recommendations generated for sentinel users, refining the fake profiles iteratively to maximize the attack impact without direct access to the target model's internals.

The task of training an agent to learn the best policy for identifying and modifying the most influential profiles can be modeled as a sequential decision-making process, formalized through a *Markov Decision Process (MDP)* [17], [18]. An MDP is formally represented by a tuple composed of four elements $(\mathcal{S}, \mathcal{A}, \mathcal{P}(s_t, a_t), \mathcal{R})$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ the set of actions, $\mathcal{P}(s_t, a_t)$ the transition probability, and $\mathcal{R}$ the reward function. In our scenario, each state $s_t \in \mathcal{S}$ at time t in the **State Space** ($\mathcal{S}$) describes the current configuration of the attack and includes:

$$s_t = \{\mathcal{U}_\mathcal{F}{}^{(t)}, \mathcal{I}^*, \mathcal{U}_\mathcal{T}^{\text{target}}\}, \tag{2}$$

where $\mathcal{U}_\mathcal{F}{}^{(t)}$ represents the set of fake profiles injected up to time t, $\mathcal{I}^*$ is the set of target items to be promoted, and $\mathcal{U}_\mathcal{T}^{\text{target}}$ is the subset of users to be influenced. This structure allows the agent to strategically select and manipulate user profiles and target items based on their impact on the recommendation system.

The **Action Space** ($\mathcal{A}$) defines the set of actions available to the agent at each state. In our scenario, an action a_t consists of selecting a real user profile u_b from the source domain $\mathcal{U}_\mathcal{B}$, characterized by its interaction matrix $\mathcal{R}_\mathcal{B}$ and associated items $\mathcal{I}_\mathcal{B}$, and applying a controlled manipulation to it. The selected profile is adjusted to maintain realistic behavior while maximizing its impact on the target system. Specifically, the agent can add new items to increase the relevance of specific targets, remove interactions that introduce noise or reduce effectiveness, or modify existing ratings to simulate stronger or weaker preferences. This formulation allows the agent to explore a rich space of subtle, plausible modifications while staying within constraints that preserve stealthiness. The state space reflects the current configuration of the attack and its complexity depends on three main factors: the number of fake profiles already injected, the target items, and the target users that the attacker aims to influence. Similarly, the action space is defined by selecting a candidate user profile from the source domain and applying a possible manipulation, such as adding, removing, or modifying an interaction. Although the theoretical size of this space can be large, the contrastive clustering phase drastically reduces the number of candidate profiles, ensuring that both state and action spaces remain compact and manageable in practice.

The **Transition Probability** ($\mathcal{P}(s_t, a_t)$) defines the likelihood of moving from one state to another after performing an action. In this context, the transition is driven by the effect of the selected action a_t on the current state s_t, that is, how the manipulation of the user profile influences the representation of the target item and the overall recommendation outcomes.

The **Reward Function** ($\mathcal{R}$) quantifies the effectiveness of each action in promoting the target item i^*. Specifically, the reward r_t is defined as the percentage variation, at time t, of the Hit Ratio at K (HR@K), computed as:

$$HR@K = \frac{1}{|\mathcal{U}_\mathcal{T}^{\text{target}}|} \sum_{u \in \mathcal{U}_\mathcal{T}^{\text{target}}} \mathbb{I}[i^* \in \text{Rec}(u)],$$

where $\mathbb{I}[\cdot]$ is an indicator function that equals 1 if i^* appears in the recommendation list for user u, and 0 otherwise. In other words, the reward function reflects the incremental impact of the action on the system's ability to display the target item, directly measuring the attack's success. The ranking criterion of the recommender system is user-based: each user receives an ordered top-k list of items according to the predicted relevance scores, and the HR@K is computed as the fraction of target users for whom the promoted item appears in the recommendation list.

Figure 1 summarizes the overall workflow, highlighting the key modules: cross-domain profile selection, profile manipulation, and PPO-based policy optimization.

IV. Proposed Framework

In this section, we describe our model-agnostic poisoning attack framework for black-box recommender systems. The framework does not rely on access to the internal parameters or gradients of the target model; instead, it learns an attack policy exclusively through interactions with a surrogate recommender. This design enables the strategy to generalize across different recommendation architectures, thereby ensuring its model-agnostic nature. The core idea is to use reinforcement learning, specifically Proximal Policy Optimization (PPO), to

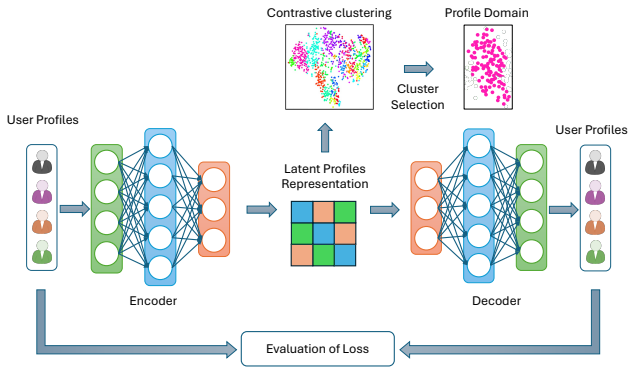

Fig. 2: Selection of the most influential subset of user profile leveraging Contrastive Learning.

train an agent that selects and refines realistic fake user profiles to maximize the promotion [19] or demotion of target items. PPO is chosen for its stability and scalability in large-scale scenarios. A distinctive feature of our framework is the integration of cross-domain clustering for profile selection. Instead of relying solely on users who have directly interacted with the target item, the method leverages contrastive learning and clustering to identify similar user profiles across domains [20], [21], [22]. This intelligent selection enhances the diversity and impact of injected profiles while keeping the attack budget minimal.

The proposed framework consists of three phases: (i) selection of influential cross-domain profiles, (ii) optimization of the attack policy via PPO, and (iii) interaction with the target system to iteratively refine the poisoning strategy through black-box feedback.

A. Selection of Influential Profile Domain

The goal of this phase is to narrow the search space for the reinforcement learning agent by selecting a diverse and realistic subset of user profiles from a cross-domain dataset. Figure 2 illustrates this process. Starting from a compatible source domain with overlapping items, we apply a contrastive clustering strategy to identify users whose behavior patterns are relevant for manipulating the target recommender. Compatible domains are identified by partial overlap in item catalogs and behavioral similarity across users. For example, MovieLens and Netflix share a substantial set of movies, enabling realistic cross-domain transfer.

We build on contrastive learning [23] to learn robust user representations that capture behavioral similarity. Given a dataset $\mathcal{D} = \{(u, i, r)\}$, we construct *positive pairs* (users who interact with the same target item) and *negative pairs* (users with disjoint interactions). Data augmentation is applied at the interaction level via masking and rating perturbation: $r \to r + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2)$.

The contrastive loss [24] encourages an encoder to pull positive pairs closer in latent space while pushing apart negative pairs:

$$\mathcal{L}_{CL} = -\frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \log \frac{\exp(\text{sim}(\mathbf{z}_{\tilde{u}_1}, \mathbf{z}_{\tilde{u}_2})/\tau)}{\sum_{v \in \mathcal{U}'} \exp(\text{sim}(\mathbf{z}_{\tilde{u}_1}, \mathbf{z}_v)/\tau)},$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is a temperature parameter.

Once trained, the encoder maps users into a latent space where we apply DBSCAN clustering to extract homogeneous groups. We then identify the cluster C_t containing users who have interacted with target items and expand it with similar users based on an affinity metric:

$$\text{Affinity}(u, U_t) = \frac{1}{|U_t|} \sum_{u_t \in U_t} \exp(-\|\mathbf{z}_u - \mathbf{z}_{u_t}\|_2).$$

Users with affinity above a threshold α are included, yielding the final subset $\mathcal{U}^*_{\mathcal{T}}$.

This selection reduces the RL agent's search space and ensures that the injected profiles remain realistic and varied, increasing the stealth and impact of the poisoning attack.

B. Optimization of the Attack Policy with PPO Algorithm

To effectively train the agent to maximize the impact of its attack policy on the recommendation system, we employ Proximal Policy Optimization (PPO) [5]. PPO is a state-of-the-art policy-gradient algorithm that optimizes a parameterized stochastic policy $\pi_\theta(a_t \mid s_t)$, with parameters θ, by directly maximizing the expected cumulative reward while ensuring stable updates and avoiding excessively large policy changes.

The objective of the agent in the context of attacking a recommender system is to learn a policy that maximizes the discounted cumulative reward over time, defined as:

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \tag{3}$$

where $\tau = (s_0, a_0, r_0, s_1, \ldots)$ is a trajectory sampled from the policy, r_t is the reward at time t, and $\gamma \in [0, 1)$ is the discount factor.

PPO improves the policy π_θ while constraining its divergence from the previous policy $\pi_{\theta_{\text{old}}}$. This is achieved using a clipped surrogate objective:

$$L^{\text{PPO}}(\theta) = \mathbb{E}_{s_t, a_t \sim \pi_{\theta_{\text{old}}}} \Big[\min \big(r_t(\theta) A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) A_t \big) \Big], \tag{4}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, A_t is the advantage function, and $\epsilon > 0$ controls the clipping range to ensure stable updates. The advantage function A_t is computed using temporal difference (TD) methods, as described in [5].

In our attack context, the reward r_t at each step is defined based on the effectiveness of the manipulation in promoting the target item i^*, as described in the previous subsection. The PPO algorithm alternates between collecting trajectories using the current policy and optimizing the clipped surrogate objective $L^{\text{PPO}}(\theta)$ with respect to θ via gradient ascent:

$$\theta \leftarrow \theta + \alpha \nabla_\theta L^{\text{PPO}}(\theta), \tag{5}$$

where α is the learning rate. Unlike TRPO, PPO uses a simpler and more scalable objective, which makes it well suited for our context, where the reward signal is sparse and the policy must be refined through repeated interactions with a surrogate

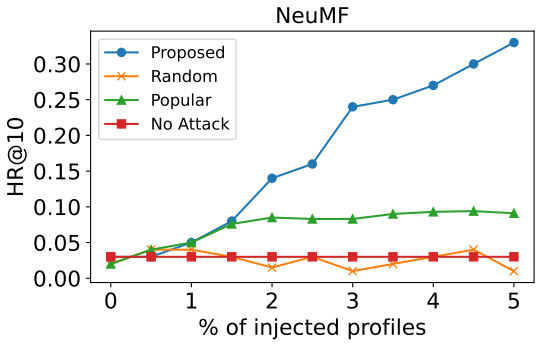

Fig. 3: Impact of the attack budget on the HR@10 for NeuMF.

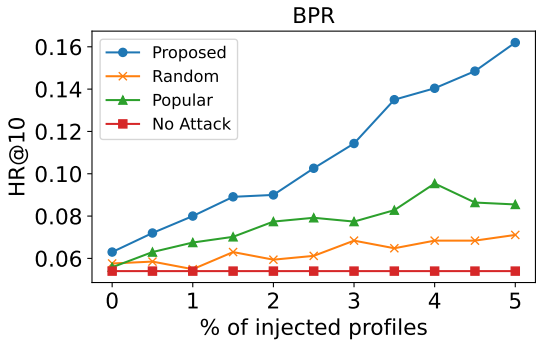

Fig. 4: Impact of the attack budget on the HR@10 for BPR.

recommender. We adopt a surrogate recommender trained on a compatible source domain to approximate the dynamics of the target. While the surrogate does not replicate the exact architecture, it captures similar user–item relationships, ensuring transferable feedback. PPO further generalizes the learned attack policy across unseen architectures. In our experiments, we adopt a simple matrix factorization model (Singular Value Decomposition - SVD) as the surrogate recommender.

By leveraging PPO, the agent iteratively refines its policy to maximize the HR@K metric for the target item i^* while ensuring stable and efficient updates. This enables our framework to adaptively select and modify user profiles, enhancing the effectiveness of poisoning attacks in realistic black-box scenarios.

C. Interaction with the Target System

The final phase of the attack framework involves the interaction of the agent with the target recommendation system. The agent selects the most influential profiles from the augmented user set $\mathcal{U}_{\mathcal{T}}^{*}$ and manipulates them to maximize the impact on the target system. The agent then observes the changes in the recommendation results for the target users $\mathcal{U}_{\mathcal{T}}^{\text{target}}$ and evaluates the effectiveness of the attack. This feedback is used to update the policy and improve the attack strategy iteratively. The agent interacts with the target system iteratively to maximize the impact of the attack, while respecting a strict budget constraint on the number of injected profiles to preserve stealthiness.

V. Experimental Evaluation

This section presents our experimental settings, baselines, and evaluation results. We validate the proposed attack on real-world datasets and compare its performance against representative baselines using practical attack budgets.

We use two real-world cross-domain datasets: *MovieLens 32M* [25] and *Netflix Prize* [26]. To ensure temporal consistency and preserve user behavior patterns, only interactions before *December 31, 2005* are retained. For computational efficiency, the *Netflix Prize* dataset is further sampled to approximately one million interactions while maintaining its diversity and representativeness.

For the target recommender, only items overlapping between the two datasets are considered, enabling effective cross-domain transfer. As a surrogate, we adopt a simple matrix factorization model (SVD) trained on the source domain to simulate a realistic black-box scenario and learn transferable strategies. For training, we implement 15 parallel environments, each simulating an independent recommender system under attack. The agent runs 100,000 poisoning steps per environment (1.5 million total), with each environment reset every 4000 steps to enforce a budget constraint per episode. This constraint limits the number of injected profiles, ensuring that the attack remains stealthy; in our setup, the injected profiles never exceed 5% of the user base per episode. This setup accelerates convergence and promotes robust policy learning that generalizes across recommendation models.

Since most poisoning attacks assume white-box access, which is not realistic here, we define three baselines for comparison in a black-box setting: **No Attack** (natural system behavior), **Random Attack** (randomly sampled user profiles injected without any strategy), and **Popular**, which promotes target items by linking them to the most popular items to exploit their visibility. We test the attack framework against our proposed method using two state-of-the-art recommenders: NeuMF, which combines matrix factorization with neural networks to capture linear and non-linear patterns, and BPR, a Bayesian personalized ranking model optimized for implicit feedback. Effectiveness is evaluated using the *Hit Ratio at K (HR@K)*, which measures the proportion of users whose top-K recommendations include the target item.

Figures 3 and 4 show how the Hit Ratio of the target items varies as the percentage of injected profiles increases from 0% to 5%. Our method consistently outperforms the Random and Popular baselines across all scenarios. For example, with NeuMF, our approach achieves an HR@K of about 0.25 with only 3% of injected profiles, compared to 0.02 (Random) and 0.095 (Popular). This represents more than a 160% increase over the strongest baseline, with a minimal injection budget.

This improvement is driven by two key components: the contrastive clustering phase, which identifies the most influential and diverse profiles, and the PPO-based policy refinement, which adaptively adjusts them for maximum effect, outperforming static or heuristic strategies. Performance remains robust even under tight budget constraints, confirming the practicality of the method in realistic black-box scenarios where attackers have limited control. However, we observe diminishing returns as the injection budget grows, indicating that profile diversity and quality matter more than simply increas-

ing volume. The consistent gains on NeuMF and BPR, two architecturally distinct models, confirm the model-agnostic nature of our attack and the transferability of policies learned solely through surrogate interaction. These results confirm the effectiveness, efficiency, and stealthiness of the proposed approach, highlighting the potential risks posed by transferable poisoning attacks in real-world recommender systems.

VI. Conclusions

In this paper, we presented a novel, model-agnostic poisoning attack framework for black-box recommender settings. Our approach combines contrastive learning with reinforcement learning based on Proximal Policy Optimization (PPO) to identify and adapt realistic cross-domain user profiles that maximize the impact on the target system while preserving stealthiness. By leveraging a surrogate recommender trained on a compatible domain, the framework iteratively refines the injection strategy without requiring access to the internal parameters of the target model.

Through comprehensive experiments on real-world datasets and multiple recommendation algorithms, we demonstrated that the proposed method consistently outperforms existing baselines in terms of target item promotion, even with a minimal injection budget. These results highlight the transferability and practical threat of realistic poisoning attacks that exploit cross-domain knowledge. Beyond the experimental validation, our approach has potential implications in several real-world application scenarios. For instance, in e-commerce platforms, an adversary could artificially promote specific products by injecting realistic cross-domain profiles; in streaming services, targeted content could be covertly boosted to manipulate user engagement; and in app stores or review-based platforms, attackers could influence rankings by altering recommendation outcomes. These examples highlight both the applicability of our framework and the urgency of designing robust defenses to mitigate such threats.

As natural directions for future work, we plan to further explore scalability aspects of PPO in very large-scale recommenders, extend the surrogate training beyond matrix factorization models to more complex deep learning-based architectures such as transformers, and complement our study with a systematic evaluation of detectability [27] against anomaly-based defenses.

References

[1] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *Journal of Big Data*, vol. 9, no. 1, p. 59, 2022.

[2] Y. Deldjoo, T. D. Noia, and F. A. Merra, "A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 2, pp. 1–38, 2021.

[3] T. T. Nguyen, N. Quoc Viet hung, T. T. Nguyen, T. T. Huynh, T. T. Nguyen, M. Weidlich, and H. Yin, "Manipulating recommender systems: A survey of poisoning attacks and countermeasures," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–39, 2024.

[4] Z. Hammoudeh and D. Lowd, "Training data influence analysis and estimation: A survey," *Machine Learning*, vol. 113, no. 5, pp. 2351–2403, 2024.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[6] Z. Tian, L. Cui, J. Liang, and S. Yu, "A comprehensive survey on poisoning attacks and countermeasures in machine learning," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–35, 2022.

[7] V. Agate, P. Ferraro, and S. Gaglio, "A cognitive architecture for ambient intelligence systems," in *6th International Workshop on Artificial Intelligence and Cognition, AIC 2018. CEUR Workshop Proceedings*, vol. 2418, 2019, p. 52 – 58.

[8] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks," in *28th USENIX security symposium*, 2019, pp. 321–338.

[9] O. Suciu, R. Marginean, Y. Kaya, H. Daume III, and T. Dumitras, "When does machine learning FAIL? generalized transferability for evasion and poisoning attacks," in *27th USENIX Security Symposium*, 2018, pp. 1299–1316.

[10] V. Agate, S. Drago, P. Ferraro, and G. Lo Re, "Anomaly detection for reoccurring concept drift in smart environments," in *18th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2022, pp. 113–120.

[11] M. Fang, N. Z. Gong, and J. Liu, "Influence function based data poisoning attacks to top-n recommender systems," in *Proceedings of The Web Conference 2020*, 2020, pp. 3019–3025.

[12] C. Lin, S. Chen, H. Li, Y. Xiao, L. Li, and Q. Yang, "Attacking recommender systems with augmented user profiles," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 855–864.

[13] C. Wu, D. Lian, Y. Ge, Z. Zhu, and E. Chen, "Influence-driven data poisoning for robust recommender systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11 915–11 931, 2023.

[14] H. Zhang, Y. Li, B. Ding, and J. Gao, "Loki: a practical data poisoning attack framework against next item recommendations," *IEEE Trans. on Knowledge and Data Engineering*, vol. 35, pp. 5047–5059, 2022.

[15] Z. Yue, Z. He, H. Zeng, and J. McAuley, "Black-box attacks on sequential recommenders via data-free model extraction," in *Proceedings of the 15th ACM Conference on Recommender Systems*. ACM, 2021, p. 44–54.

[16] W. Fan, X. Zhao, Q. Li, T. Derr, Y. Ma, H. Liu, J. Wang, and J. Tang, "Adversarial attacks for black-box recommender systems via copying transferable cross-domain user profiles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 12 415–12 429, 2023.

[17] G. Gupta and R. Katarya, "A study of recommender systems using markov decision process," in *2018 Second International Conference on Intelligent Computing and Control Systems*, 2018, pp. 1279–1283.

[18] V. Padhye, K. Lakshmanan, and A. Chaturvedi, "Proximal policy optimization based hybrid recommender systems for large scale recommendations," *Multimedia Tools and Applications*, vol. 82, no. 13, pp. 20 079–20 100, 2023.

[19] V. Agate, A. De Paola, G. Lo Re, and M. Morana, "A simulation software for the evaluation of vulnerabilities in reputation management systems," *ACM Transactions on Computer Systems (TOCS)*, vol. 37, no. 1-4, pp. 1–30, 2021.

[20] T. Zang, Y. Zhu, H. Liu, R. Zhang, and J. Yu, "A survey on cross-domain recommendation: taxonomies, methods, and future directions," *ACM Transactions on Information Systems*, vol. 41, pp. 1–39, 2022.

[21] Y. Zheng, "Methodologies for cross-domain data fusion: An overview," *IEEE transactions on big data*, vol. 1, no. 1, pp. 16–34, 2015.

[22] Y. Li, S. Bai, Y. Zhou, C. Xie, Z. Zhang, and A. Yuille, "Learning transferable adversarial examples via ghost networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 11 458–11 465.

[23] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 8547–8555.

[24] H. Tang, G. Zhao, Y. He, Y. Wu, and X. Qian, "Ranking-based contrastive loss for recommendation systems," *Knowledge-Based Systems*, vol. 261, p. 110180, 2023.

[25] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[26] J. Bennett and S. Lanning, "The netflix prize," in *Proceedings of the KDD Cup Workshop 2007*. New York: ACM, Aug. 2007, pp. 3–6.

[27] V. Agate, F. M. D'Anna, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana, "A behavior-based intrusion detection system using ensemble learning techniques." in *CEUR Workshop Proceedings, 6th Italian Conference on Cybersecurity, ITASEC 2022*, vol. 3260, 2022, pp. 207–218.