# MSAP: Multi-Step Adversarial Perturbations on Recommender Systems Embeddings

**Vito Walter Anelli[†], Alejandro Bellogín[‡], Yashar Deldjoo[†], Tommaso Di Noia[†], Felice Antonio Merra[†*]**
[†]Politecnico di Bari, Italy, firstname.lastname@poliba.it,
[‡]Universidad Autónoma de Madrid, Spain, firstname.lastname@uam.es

## Abstract

Recommender systems (RSs) have attained exceptional performance in learning users' preferences and finding the most suitable products. Recent advances in adversarial machine learning (AML) in computer vision have raised interests in recommenders' security. It has been demonstrated that widely adopted model-based recommenders, e.g., BPR-MF, are not robust to adversarial perturbations added on the learned parameters, e.g., users' embeddings, which can cause drastic reduction of recommendation accuracy. However, the state-of-the-art adversarial method, named fast gradient sign method (FGSM), builds the perturbation with a single-step procedure.

In this work, we extend the FGSM method proposing multi-step adversarial perturbation (`MSAP`) procedures to study the recommenders' robustness under powerful methods. Letting fixed the perturbation magnitude, we illustrate that `MSAP` is much more harmful than FGSM in corrupting the recommendation performance of BPR-MF. Then, we assess the `MSAP` efficacy on a robustified version of BPR-MF, i.e., AMF. Finally, we analyze the variations of fairness measurements on each perturbed recommender. Code and data are available at `https://github.com/sisinflab/MSAP`.

## Introduction

Recommender systems (RSs) are machine-learning (ML) models involved in decision-making tasks to show to the customers personalized lists of relevant products learned from their historical interactions. However, *adversarial machine learning* (AML) (Huang et al. 2011) has revealed security breaches of ML models in several tasks with a particular focus on computer vision (CV) domain (Akhtar and Mian 2018). In (Szegedy et al. 2014), the authors formalize the first adversarial attacks against DNNs for *image classification* finding that a human-imperceptible pixel changing is sufficient to confuse the network to classify a pandas image into the wrong gibbon class. This perturbation, named *adversarial perturbation*, consists in adding a norm-constrained amount of noise to enforce the model to make a wrong prediction. Starting from this work, several attacks (Kurakin, Goodfellow, and Bengio 2017; Madry et al. 2018; Carlini and Wagner 2017), as well as defenses (Goodfellow, Shlens, and Szegedy 2015), have been studied in CV domain with the goal to make reliable ML models.

(He et al. 2018) proposed the pioneering work of AML for RSs. The authors clarified that attacks and defenses should be treated differently in the CV and RS tasks since image data are continuous-valued matrices, while recommender data are discrete interactions (0/1 feedback). For this reason, they tested adversarial perturbations on the model parameters, e.g., the embedding matrices of matrix-factorization (MF) models. They discovered that the fast gradient sign method (FGSM) (Goodfellow, Shlens, and Szegedy 2015), a *single-step adversarial perturbation* procedure, leads to five times larger deterioration of recommendation accuracy than the one caused by random variation. This finding shows the weaknesses of model-based recommenders in learning embeddings that will cause drastic performance degradation when subjected to small changes. For instance, this change can be caused when new users, or items, are added to the system. Furthermore, they successfully applied an *adversarial training* procedure (Goodfellow, Shlens, and Szegedy 2015) on BPR-MF, named AMF, demonstrating more robust RS performance under FGSM perturbations. These techniques have been tested on multimedia recommendation systems (Tang et al. 2020), deep RSs (Yuan, Yao, and Benatallah 2019a; 2019b), and tensor factorization approaches (Chen and Li 2019).

In this work, inspired by the evidence in the CV domain that iterative attacks are more effective than single-step ones (Kurakin, Goodfellow, and Bengio 2017), we present two *multi-step adversarial perturbation* (`MSAP`) techniques, namely basic iterative method (BIM) and projected gradient descent (PGD), applied on the embeddings of two state-of-the-art MF models (He et al. 2018; Rendle et al. 2009) to answer the following research questions:

- Does MSAP outperform single-step attacks in degrading the quality of the system with respect to accuracy and beyond-accuracy evaluation measures?
- Is the *adversarial regularization* of RSs still useful against the presented multi-step generated noise?
- Are adversarial perturbations, and in particular the `MSAP`, able to impact in a significant direction on the observed fairness of recommender models?

To this end, we evaluate MSAP against two model-based recommenders, i.e., BPR-MF and its adversarial robustified version AMF, on two datasets, i.e., `ML-1M` and `LastFM`.

*The authors are listed in alphabetical order. Corresponding author: Felice Antonio Merra (`felice.merra@poliba.it`).

## Related Work

### Collaborative Recommendation

Recommendation models proposed in the last thirty years are categorized into collaborative filtering (CF), content-based, and hybrid RS (Ricci, Rokach, and Shapira 2015). The first category learns users' preferences from historical user-item interactions. The second category suggests unseen products based on the content-based similarity of user consumed items and other unseen items. The last class combines both techniques to augment user-item interactions with side information. Model-based CF models such as BPR-MF (Rendle et al. 2009) and recent neural models such as neural collaborative filtering (NCF) (He et al. 2017) are popular choices in the RS and ML communities. MF-based models are the major class of RSs used for research on AML (He et al. 2018; Tang et al. 2020).

### Adversarial Machine Learning

Security of RSs covers two research lines: based on hand-engineered shilling attacks and adversarial machine-learned attacks (Deldjoo, Noia, and Merra 2021). The former category concentrates on the injection of manually generated fake profiles (Burke, O'Mahony, and Hurley 2015). The latter category, the focus of the current work, studied the application of AML techniques to generate perturbations to reduce recommenders' performance and their countermeasures (He et al. 2018; Beigi et al. 2020; Di Noia, Malitesta, and Merra 2020). The work (He et al. 2018) reported serious vulnerability of BPR-MF against adversarial perturbation obtained from the FGSM attack and suggested an adversarial regularization procedure as a defensive countermeasure. This work inspired other models such as AMR (Tang et al. 2020), FG-ACAE (Yuan, Yao, and Benatallah 2019a; 2019b), and ATF (Chen and Li 2019). However, we found that the RS community lacks studies on other categories of adversarial perturbations, such as iterative attacks (e.g., PGD (Madry et al. 2018)), which are effective in altering CV tasks. `MSAP` is the first iterative perturbation procedure proposed in the RS domain.

## The Proposed Framework

In this section, we describe the foundations of a personalized matrix factorization (MF) recommender model. Then, we recapitulate the baseline single-step adversarial perturbation before defining the multi-step strategies.

### Personalized Recommenders via MF

The recommendation problem is the task of estimating a preference prediction function $s(u, i)$ that maximizes the utility of the user $u \in \mathcal{U}$ in getting the item $i \in \mathcal{I}$ recommended by the RS, where $\mathcal{U}$ and $\mathcal{I}$ are the set of users and items respectively. Before we dive into the description of the MF model, we introduce the following notation:

- $\mathbf{P}$: the matrix of *user* embeddings, where $\mathbf{p}_u$ is the embedding vector associated to the user $u$;
- $\mathbf{Q}$: the matrix of *item* embeddings, where $\mathbf{q}_i$ is the embedding vector associated to the item $i$;
- $\mathbf{\Theta}$: the set of model parameters ($\mathbf{\Theta} = \{\mathbf{P}, \mathbf{Q}\}$);
- $\mathcal{L}$: the loss function

The main intuition behind the MF model is to compute the preference score $s(u, i)$ as the dot product between the user's embedding ($\mathbf{p}_u$) and the item's embedding ($\mathbf{q}_i$), i.e., $s(u, i) = \mathbf{p}_u^T \mathbf{p}_i$. The model parameters are learned by solving the optimization problem in the following general form:

$$\underset{\mathbf{\Theta}}{\operatorname{argmin}} \mathcal{L}(\mathbf{\Theta}) \tag{1}$$

The state-of-the-art approach to produce personalized rankings is Bayesian personalized ranking (BPR) (Rendle et al. 2009). The idea is to reduce the ranking problem to a pairwise learning one where, for each user, the score of interacted items has to be higher than non-interacted ones.

### Adversarial Perturbation of Model Parameters

The main intuition behind an adversarial perturbation method is to generate minimum perturbations ($\mathbf{\Delta}^{adv}$) capable of undermining the learning objective of the learning model. The adversary's goal is to maximize Eq. 1, under a minimal-norm constraint:

$$\Delta^{adv} \leftarrow \underset{\mathbf{\Delta}_0, ||\mathbf{\Delta}_0|| \leq \epsilon}{\operatorname{argmax}} \mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_0) \tag{2}$$

where $\mathbf{\Delta}_0$ is the initial adversarial perturbation added to the model parameters $\mathbf{\Theta}$ and $\epsilon$ is the *perturbation budget* ( the limit of the perturbation size). Eq. 1 and 2 can be unified in the following *minimax* problem:

$$\arg \min_{\mathbf{\Theta}} \max_{\mathbf{\Delta}_0, ||\mathbf{\Delta}_0|| \leq \epsilon} \mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_0) \tag{3}$$

in which two opposite players play an **adversarial minimax** game, where the adversary tries to maximize the likelihood of its success while the ML model tries to minimize the risk. This minimax game is the main characteristic of tasks related to AML research (Tu, Zhang, and Tao 2019).

**Fast Gradient Sign Method (FGSM).** This perturbation strategy is the baseline *single-step adversarial perturbation* mechanism proposed by (He et al. 2018) to alter the recommendation task. It builds on advances made in ML research pioneered in (Goodfellow, Shlens, and Szegedy 2015) for the classification task. It approximates $\mathcal{L}$ by linearizing it around an initial zero-matrix perturbation $\mathbf{\Delta}_0$ and applies the max-norm constraint. The adversarial noise $\mathbf{\Delta}^{adv}$ is

$$\mathbf{\Delta}^{adv} = \epsilon \frac{\Pi}{\|\Pi\|} \quad \text{where} \quad \Pi = \frac{\partial \mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_0)}{\partial \mathbf{\Delta}_0} \tag{4}$$

where $||\cdot||$ is the $L_2-$norm. After the calculation of $\mathbf{\Delta}^{adv}$, the authors added this perturbation to the current model parameters $\mathbf{\Theta}^{adv} = \mathbf{\Theta} + \mathbf{\Delta}^{adv}$ and generated the recommendation lists with this perturbed model parameter.

**Multi-Step Adversarial Perturbation (`MSAP`).** This adversarial noise generation mechanism is a straightforward extension of the single-step strategy proposed in CV domain (Kurakin, Goodfellow, and Bengio 2017). In particular, the authors' idea was to build an FGSM-based *multi-step*

strategy and create more effective $\epsilon$-clipped perturbations. The initial model parameters are defined as

$$\mathbf{\Theta}_0^{adv} = \mathbf{\Theta} + \mathbf{\Delta}_0 \tag{5}$$

Starting from this initial state of model parameters, let $Clip_{\mathbf{\Theta},\epsilon}$ be an element-wise clipping function to limit the perturbation of each original embedding value inside the $[-\epsilon, +\epsilon]$ interval, let $\alpha$ be the step size which is the maximum perturbation budget of each iteration, and let $L$ be the number of iterations, the first iteration ($l = 1$) is defined by:

$$\mathbf{\Theta}_1^{adv} = Clip_{\mathbf{\Theta},\epsilon}\left\{\mathbf{\Theta}_0^{adv} + \alpha\frac{\Pi}{\|\Pi\|}\right\} \text{ where } \Pi = \frac{\partial\mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_0)}{\partial\mathbf{\Delta}_0} \tag{6}$$

and we generalize the $l$-th iteration of the $L$-iterations multi-step adversarial perturbation as:

$$\mathbf{\Theta}_l^{adv} = Clip_{\mathbf{\Theta},\epsilon}\left\{\mathbf{\Theta}_{l-1}^{adv} + \alpha\frac{\Pi}{\|\Pi\|}\right\} \text{ where } \Pi = \frac{\partial\mathcal{L}(\mathbf{\Theta} + \mathbf{\Delta}_{l-1}^{adv})}{\partial\mathbf{\Delta}_{l-1}^{adv}} \tag{7}$$

where $l \in [1, 2, ..., L]$, $\mathbf{\Delta}_l^{adv}$ is the adversarial perturbation at the $l$-th iteration, and $\mathbf{\Theta}_l^{adv}$ is the sum of the original model parameters $\mathbf{\Theta}$ with the perturbation at the $l$-th iteration. The `MSAP` computational cost is $l$-times the single-step version. We considered two different `MSAP`: Basic Iterative Method (BIM) (Kurakin, Goodfellow, and Bengio 2017) and Projected Gradient Descent (PGD) (Madry et al. 2018). The former approach initializes $\mathbf{\Delta}_0$ as matrices of zeros, with the same size of the matrix embeddings of the victim model. The latter initializes the perturbation sampling a uniform distribution. These different initializations make PGD more powerful than BIM in confusing CV image classifiers (Athalye, Carlini, and Wagner 2018). Since this has not been – to the best of our knowledge – investigated in the RSs community, we chose both strategies to investigate whether such a difference between two adversarial perturbation strategies exists for the recommendation task. Note that we test our adversarial method against MF recommenders, however it can be reproduced against any BPR optimized recommender.

## Experimental Setup

In this section, we introduce the datasets, recommenders, evaluation measures, and reproducibility information.

### Datasets

We conducted `MSAP` experiments on two datasets:

**Movielens 1M** (`ML-1M`) (Harper and Konstan 2016) contains 1,000,209 ratings ($|\mathcal{F}|$) given by 6,040 users ($|\mathcal{U}|$) towards 3,706 movies ($|\mathcal{I}|$). Users' gender and movies' genres are used in the fairness evaluation.

**LastFM-1b** (`LastFM`) (Schedl 2016) contains 935,875 listening events ($|\mathcal{F}|$) given by 2,847 users ($|\mathcal{U}|$) towards 33,164 authors ($|\mathcal{I}|$) stored from the online music provider Last.fm. Users' gender and items' artists are used for the analysis of fairness.

### Recommender Models

**BPR-MF** (Rendle et al. 2009) is a MF recommender optimized with a pair-wise loss function (i.e., BPR). The fundamental intuition of BPR-MF is to discard not-interacted items with respect to interacted ones in order to learn a rank-based preference predictor. $\mathcal{L}_{BPR}(\mathbf{\Theta}) = \mathcal{L}(\mathbf{\Theta})$ denotes the BPR-MF loss function.

**AMF** (He et al. 2018) is a BPR-MF extension that includes an adversarial training procedure. The model parameters are learned with the following loss function:

$$\mathcal{L}_{AMF}(\mathbf{\Theta}) = \mathcal{L}_{BPR}(\mathbf{\Theta}) + \lambda \underbrace{\mathcal{L}_{BPR}(\mathbf{\Theta}^{adv})}_{\text{adversarial regularizer}} \tag{8}$$

where the model parameters of the *adversarial regularizer* ($\mathbf{\Theta}^{adv}$) are perturbed with the single-step perturbation technique described in Eq. 4. AMF reduces up to 88% the degrading effect of single-step perturbations on the model accuracy (He et al. 2018).

### Evaluation Metrics

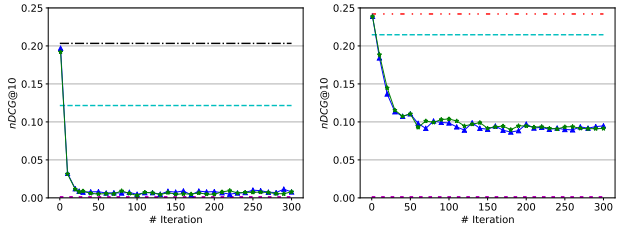
**Accuracy** The accuracy metrics used are: precision ($PR@K$), the fraction of suggested items relevant to the users, recall ($RE@K$), the average fraction of relevant recommended items, and normalized discounted cumulative gain ($nDCG@K$), the users' gain of a ranked list discounting the relevant predictions by their positions.

**Beyond-Accuracy** The beyond-accuracy metrics used are: expected free discovery ($EFD@K$) (Vargas and Castells 2011), the capacity to suggest relevant long-tail products, Shannon Entropy ($SE@K$), the diversity of recommendations, and coverage ($ICov@K$), the number of recommended products.

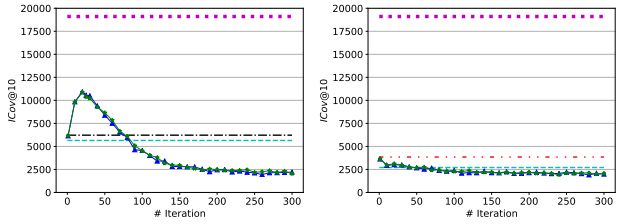
**Fairness** metrics are evaluated before and after `MSAP`. We explored: generalized cross-entropy ($GCE$) (Deldjoo et al. 2019) that considers several possible ideal probability distributions for each user, or item, clustering. Hence, it computes the divergence of the recommendation results (by considering a specific metric, i.e., $nDCG$) from the ideal distributions. Consequently, GCE's value close to zero denotes the recommender's congruence with that distribution. On the other hand, MAD focuses on the absolute variation of a given metric from an ideal situation in which the recommender treats groups equally. The original formulation of MAD (Zhu, Hu, and Caverlee 2018), namely $MADr$, considers the user and item score pairs in the recommendation results. Additionally, we considered the MAD extension proposed in (Deldjoo et al. 2019), $MADR$, in which the per-user performance values of an accuracy metric, i.e., $nDCG$, are considered.

### Reproducibility

We employed the *leave-one-out* evaluation protocol (He et al. 2018), putting in the test set either the last — when that information is available (i.e., `ML-1M`)– or a random (i.e., `LastFM`) interaction, and using the rest of the recorded feedbacks to train the recommenders. We trained the BPR-MF for 2,000 epochs. Then, we used the BPR-MF parameters at the 1,000th epoch as the starting point to train AMF — the BPR-MF *adversarial regularized* version– as presented in (He et al. 2018). We fixed the perturbation budget ($\epsilon$) to 0.5, which is the smallest perturbation experimented in (He et al. 2018), and set the step size $\alpha$ of `MSAP` to $\epsilon/4$.

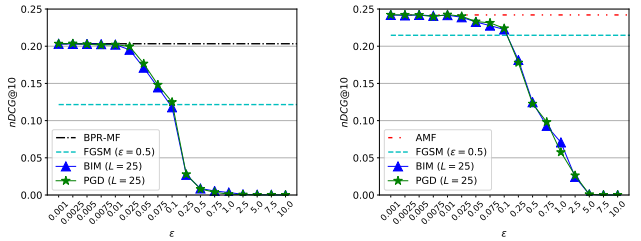



(a) $nDCG$ on BPR-MF (b) $nDCG$ on AMF




(c) $ICov$ on BPR-MF (d) $ICov$ on AMF


Figure 1: $nDCG$ and $ICov$ results for `LastFM`. Results of the (baseline) random recommender are in violet dotted line.




(a) BPR-MF (varying $\epsilon$) (b) AMF (varying $\epsilon$)

Figure 2: `MSAP` results varying $\epsilon \in [0.001, 10.0]$ on `LastFM` ($L = 25$). Fig. 2a and 2b shows that with a small perturbation, e.g., $\epsilon \simeq 0.1$, `MSAP` is more effective than FGSM with $\epsilon = 0.5$.

We used the following parameters for both models: embedding size to 64, learning rate to 0.05, $\lambda$ to 1, and the batch size to 512.

## Results and Discussion

In this section, we discuss the experimental results to answer our open questions. All the metrics are evaluated on top-10 recommendation lists.

### Investigating the `MSAP` effects

To better understand the merits of the presented adversarial perturbations, we aim to answer the following questions:

- **On the perturbation side**: how much adversarial perturbations obtained from the single-step and the `MSAP` methods can impair the quality of the original BPR-MF model? Fig. 1a & 1c compare perturbations effects on BPR-MF trained on `LastFM`.
- **On the defensive side**: what is the impact on the adversarial regularized version of BPR-MF, i.e. AMF? The answer can be found in Fig. 1b & Fig. 1d.

Table 1: Accumulated normalized values of the accuracy and beyond-accuracy metrics. We put in **bold** the lower value when the perturbation ($\epsilon = .5$) is more effective.

| Model | Metric | LastFM | | | | ML-1M | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Initial | FGSM | BIM | PGD | Initial | FGSM | BIM | PGD |
| BPR-MF | $PR$ | .0310 | .0211 | .0019 | **.0018** | .0088 | .0074 | **.0035** | **.0035** |
| | $RE$ | .3102 | .2115 | .0194 | **.0177** | .0884 | .0740 | **.0353** | **.0353** |
| | $nDCG$ | .2033 | .1216 | .0111 | **.0100** | .0447 | .0368 | .0174 | **.0172** |
| | $EFD$ | .5144 | .3069 | .0313 | **.0284** | .0977 | .0791 | .0355 | **.0353** |
| | $SE$ | 11.35 | 11.14 | **1.17** | 1.21 | 9.63 | 9.16 | **7.40** | 7.45 |
| | $ICov$ | 6220 | 5645 | **4352** | 4428 | 2247 | 2433 | **1189** | 1213 |
| AMF | $PR$ | .0357 | .0316 | **.0164** | .0167 | .0092 | .0085 | **.0048** | **.0048** |
| | $RE$ | .3565 | .3165 | **.1644** | .1667 | .0922 | .0846 | **.0482** | .0484 |
| | $nDCG$ | .2421 | .2147 | **.1010** | .1030 | .0462 | .0419 | **.0228** | .0231 |
| | $EFD$ | .5987 | .5184 | **.2303** | .2352 | .0971 | .0853 | **.0442** | .0447 |
| | $SE$ | 9.98 | 8.90 | **7.19** | 7.20 | 8.30 | 7.41 | **6.30** | 6.30 |
| | $ICov$ | 3847 | 2708 | **2315** | 2321 | 1486 | 1169 | **1066** | 1077 |

Since the performance of the `MSAP` varies based on the number of iterations, firstly, we discuss and analyze the effectiveness of the presented perturbations across different iterations, then, we fix the iteration number and study how `MSAP` impairs the RS varying the perturbation budget $\epsilon$.

**Impact of `MSAP` varying the number of iterations.**

On the **perturbation side**, by looking at Fig 1a, one can note that both `MSAP` strategies are more powerful compared with the single-step one, for a fixed perturbation budget $\epsilon = 0.5$. For instance, the PGD perturbation technique shows **15.1** (0.1216 v.s. 0.0080), **20.4** (0.1216 v.s. 0.0060), and **23.8** (0.1216 v.s. 0.0051) times stronger impact with respect to FGSM, for iterations 25, 40, and 50 respectively. These results confirm CV's findings on the superiority of `MSAP`— in terms of model damage — compared to single-step ones in RSs. To better reveal `MSAP` effects, analyzing Fig 1a, we observe that after 25 iterations, the perturbed *BPR-MF starts to perform similar to the random recommender*. In other words, BPR-MF has lost all the learned users' personalized information.

Moreover, Table 1 confirms that `MSAP` strategies outperform FGSM for all <dataset, recommender> combinations. For instance, the <`ML-1M`, BPR-MF> combination shows the PGD perturbations reduced the accuracy by more than 2 times compared to FGSM, e.g., (0.0074 v.s. 0.0035), (0.0740 v.s. 0.0353), and (0.0368 v.s. 0.0172) for $PR$, $RE$, and $nDCG$, respectively. Here, we should point out that both Fig. 1 and Table 1 do not show a clear difference in PGD perturbation compared to BIM perturbation. This finding is different from the one previously reported in (Athalye, Carlini, and Wagner 2018) for CV. We motivate it because tested model-based recommenders are less sensitive to the embedding initialization than the weight initialization of neural networks in CV since BPR computes gradients based on the differences between pairs.

For what concerns beyond-accuracy analysis, we found an interesting behavior for the BPR-MF. During the first 25 iterations of BIM, $ICov$ increments nearly by 76% (from 6,220 to 10,928) compared to the coverage value of the non-perturbed recommender (see Fig. 1c). After that, it steadily diminishes with a minimum $ICov$ value of 1,948 (for BIM). This result, strengthened by looking into Table 1, may be justified because when `MSAP` computes several iterations ($L \geq 70$), it steadily destructs the accuracy metrics and brings the model to recommend a set of few items that all the

Table 2: Performance (measured in terms of $nDCG$) of the different approaches on each subset of users/items, where $C_1$ and $C_4$ denote the least and most popular items and users with less and more interactions, respectively; for user gender $C_1$ is associated to males and $C_2$ to females. Results for ML−1M are presented on the left, LastFM on the right. We highlight in bold the best results for each model.

| Model | | Item pop | | | | User gender | | User interactions | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_1$ | $C_2$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
| BPR-MF | initial | **0.054** | **0.035** | **0.045** | **0.300** | **0.046** | **0.043** | **0.079** | **0.044** | 0.032 | **0.023** |
| | FGSM | 0.027 | 0.017 | 0.043 | 0.284 | 0.044 | 0.041 | 0.073 | 0.044 | **0.032** | 0.022 |
| | BIM | 0.005 | 0.000 | 0.000 | 0.167 | 0.019 | 0.016 | 0.018 | 0.020 | 0.018 | 0.016 |
| | PGD | 0.000 | 0.000 | 0.000 | 0.178 | 0.017 | 0.016 | 0.022 | 0.018 | 0.015 | 0.012 |
| AMF | initial | **0.172** | 0.096 | 0.096 | **0.334** | **0.047** | **0.043** | **0.078** | **0.047** | **0.034** | **0.026** |
| | FGSM | 0.163 | **0.114** | **0.110** | 0.326 | 0.043 | 0.039 | 0.070 | 0.041 | 0.033 | 0.022 |
| | BIM | 0.000 | 0.000 | 0.000 | 0.198 | 0.022 | 0.018 | 0.024 | 0.018 | 0.025 | 0.018 |
| | PGD | 0.002 | 0.055 | 0.000 | 0.202 | 0.023 | 0.017 | 0.024 | 0.018 | 0.025 | 0.018 |

| Model | | Item pop | | | | User gender | | User interactions | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $C_1$ | $C_2$ | $C_3$ | $C_4$ | $C_1$ | $C_2$ | $C_1$ | $C_2$ | $C_3$ | $C_4$ |
| BPR-MF | initial | **0.000** | 0.000 | **0.006** | **0.092** | **0.218** | **0.143** | **0.158** | **0.209** | **0.194** | **0.253** |
| | FGSM | 0.000 | **0.001** | 0.004 | 0.062 | 0.131 | 0.085 | 0.102 | 0.118 | 0.123 | 0.143 |
| | BIM | 0.000 | 0.000 | 0.000 | 0.004 | 0.007 | 0.009 | 0.011 | 0.007 | 0.009 | 0.002 |
| | PGD | 0.000 | 0.001 | 0.000 | 0.002 | 0.004 | 0.006 | 0.007 | 0.005 | 0.004 | 0.004 |
| AMF | initial | 0.000 | **0.006** | **0.014** | **0.106** | **0.260** | **0.188** | **0.174** | **0.237** | **0.229** | **0.329** |
| | FGSM | 0.000 | 0.000 | 0.010 | 0.095 | 0.230 | 0.168 | 0.153 | 0.211 | 0.198 | 0.297 |
| | BIM | **0.002** | 0.001 | 0.005 | 0.046 | 0.098 | 0.066 | 0.052 | 0.081 | 0.086 | 0.143 |
| | PGD | 0.000 | 0.002 | 0.003 | 0.046 | 0.097 | 0.061 | 0.054 | 0.082 | 0.090 | 0.142 |

Table 3: Fairness measured according to $GCE$ where $f_0$ represents a uniform distribution, $f_k$ denotes a distribution where group $C_k$ accumulates more probability than the rest, as in $f_1 = [0.75, 0.25]$ for user gender, $MADr$, and $MADR$. Rest of notation as in Table 2.

| Data | Model | | Item pop | | | | | User gender | | | | | User interactions | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $f_0$ | $f_1$ | $f_4$ | $MADr$ | $MADR$ | $f_0$ | $f_1$ | $f_2$ | $MADr$ | $MADR$ | $f_0$ | $f_1$ | $f_4$ | $MADr$ | $MADR$ |
| ML−1M | BPR-MF | initial | **−0.483** | **−1.574** | **−0.005** | 0.040 | 0.159 | −0.001 | −0.109 | −0.143 | **0.050** | 0.003 | −0.116 | **−0.138** | −1.480 | 0.618 | 0.030 |
| | | FGSM | −0.929 | −3.056 | −0.042 | **0.029** | 0.140 | **0.000** | −0.111 | **−0.140** | 0.067 | 0.002 | −0.110 | −0.158 | −1.514 | **0.614** | 0.028 |
| | | BIM | −2,039.764 | −334.326 | −326.189 | 0.066 | **0.079** | −0.003 | **−0.088** | −0.170 | 0.373 | 0.003 | **−0.004** | −0.542 | **−0.679** | 1.781 | **0.002** |
| | | PGD | −3,167.250 | −8,615.699 | −506.580 | 0.062 | 0.083 | 0.000 | −0.111 | −0.140 | 0.234 | **0.001** | −0.024 | −0.323 | −0.910 | 1.564 | 0.005 |
| | AMF | initial | −0.147 | **−0.576** | **−0.105** | 0.225 | 0.424 | −0.001 | −0.104 | −0.149 | 0.084 | 0.004 | −0.092 | **−0.162** | −1.329 | 1.995 | 0.028 |
| | | FGSM | **−0.104** | −0.646 | −0.121 | 0.171 | 0.302 | **−0.001** | −0.105 | **−0.147** | **0.038** | **0.004** | −0.093 | −0.166 | −1.403 | **1.674** | 0.025 |
| | | BIM | −3,533.378 | −9,611.568 | −565.161 | **0.095** | **0.155** | −0.007 | −0.074 | −0.193 | 0.302 | 0.005 | −0.014 | −0.435 | −0.719 | 4.175 | 0.005 |
| | | PGD | −1,543.481 | −287.878 | −246.845 | 0.263 | 0.330 | −0.011 | **−0.064** | −0.213 | 0.328 | 0.006 | **−0.010** | −0.426 | **−0.702** | 4.177 | **0.004** |
| | | | $f_0$ | $f_1$ | $f_4$ | $MADr$ | $MADR$ | $f_0$ | $f_1$ | $f_2$ | $MADr$ | $MADR$ | $f_0$ | $f_1$ | $f_4$ | $MADr$ | $MADR$ |
| LastFM | BPR-MF | initial | −1,161.806 | −4,646.677 | −185.725 | 0.120 | 0.032 | −0.016 | −0.188 | −0.499 | 0.147 | 0.051 | −0.015 | −0.822 | **−0.353** | **0.557** | 0.051 |
| | | FGSM | −397.483 | −3,094.772 | −63.435 | 0.123 | 0.033 | −0.016 | **−0.180** | −0.489 | **0.141** | 0.031 | **−0.008** | −0.730 | −0.395 | 0.686 | 0.022 |
| | | BIM | −46.740 | −186.212 | −7.314 | 0.031 | 0.003 | **−0.002** | −0.372 | −0.258 | 0.312 | **0.001** | −0.206 | **−0.243** | −2.591 | 3.153 | 0.005 |
| | | PGD | **−20.224** | **−156.603** | **−3.149** | **0.021** | **0.002** | −0.011 | −0.480 | **−0.243** | 0.290 | 0.001 | −0.025 | −0.282 | −0.694 | 3.062 | **0.002** |
| | AMF | initial | −747.062 | −5,853.077 | −119.395 | **0.468** | 0.055 | −0.010 | −0.190 | −0.416 | 0.224 | 0.048 | **−0.026** | **−0.921** | −0.291 | 2.057 | 0.079 |
| | | FGSM | −1,242.414 | −4,969.776 | −198.632 | 0.583 | 0.066 | **−0.009** | −0.193 | **−0.413** | **0.108** | 0.042 | −0.028 | −0.930 | −0.279 | **1.060** | 0.074 |
| | | BIM | **−2.238** | **−8.706** | **−0.217** | 0.672 | **0.035** | −0.014 | **−0.178** | −0.459 | 0.941 | **0.021** | −0.067 | −1.257 | **−0.200** | 6.127 | 0.046 |
| | | PGD | −309.333 | −2,419.092 | −49.342 | 0.742 | 0.039 | −0.022 | −0.200 | −0.562 | 0.978 | 0.025 | −0.063 | −1.237 | −0.210 | 7.015 | **0.046** |

users will not appreciate. Thus, we can conclude that MSAP deteriorates the personalized recommender to perform as bad as a random recommender (see Fig. 1a) on both accuracy and beyond-accuracy measures.

On the **defensive side**, Figure 1b shows an evident performance drop in accuracy for AMF which is, on average, more than 58% for MSAP and 11.31% for FGSM (see Table 1). For instance, the PGD perturbation shows **1.48** (0.2147 v.s. 0.1448), **1.86** (0.2147 v.s. 0.1154), and **1.94** (0.2147 v.s. 0.1106) times stronger impact with respect to FGSM, for iterations 20, 30, and 50, respectively. However, the accuracy reduction does not reach random performance as for the BPR-MF recommender. We may explain this slight robustness by mentioning the partial effectiveness of the adversarial regularization procedure, i.e., specifically designed to protect against FGSM (He et al. 2018).

**Impact of MSAP varying $\epsilon$.** Here, we fix the number of iterations to 25 and vary $\epsilon$ from 0.001 to 10. Analyzing Fig. 2a & 2b, we found that MSAP strategies reach the FGSM ($\epsilon = 0.5$) performance with $\epsilon \simeq 0.1$. In other words, MSAP uses $0.5/0.1 =$**5** times less perturbation budget to get the same performance degradation of single-step strategies.

## Fairness and Per-Attribute Performance Analysis

Before focusing on fairness, let us analyze recommenders' behavior for the different groups/categories to uncover the potential biases produced or removed by the attack strategies (MSAP with 150 iterations). Table 2 depicts the $nDCG$ performance of the recommenders (BPR-MF, AMF, and their attacked variants) regarding the clusters for three attributes: item popularity, user gender, and user interactions. We computed the clustering for item popularity and user interactions considering the quartiles for the attributes, while the original datasets contain already user gender clusters. As already noted in the literature, Table 2 shows that BPR-MF achieves higher values of $nDCG$ for popular items on both ML−1M and LastFM. In this respect, note the performance of BPR-MF in $C_4$ regarding the item pop attribute. Notably, the efficacy of the attacks is particularly evident here since, for BPR-MF, the $C_4$, for the item pop attribute column, shows a degradation of the performance when the recommender is under attack. On the other hand, AMF shows less evident performance deterioration, despite a similar trend is observed.

Considering the user gender, we observe that the recommendation performance for males ($C_1$) is higher than for women in both datasets. Even though the trends are similar to those observed for item popularity, it is worth noticing that the degradation and the defense effects are more evident in LastFM. Finally, the table shows two opposite behaviors for user interactions: in ML−1M, BPR-MF seems to favor the less active users, whereas LastFM favors the most active ones. The reason for this behavior is probably twofold. First, there are no proper cold-users in ML−1M: 25% of users ($C_1$) have from 19 to 43 interactions. In LastFM, on the other hand, users in $C_1$ have from 2 to 123 interactions. Second, the datasets show a dramatically different number of items in the catalogs, making the number of interactions sufficient to produce meaningful recommendations for ML−1M.

Regarding the change in performance when using any of the attack methods, we observe that in ML−1M the trend and absolute values remain almost the same to the initial recommender; however, in LastFM the situation is not identical: while the degradation follows the same trend, AMF shows higher accuracy values for all the clusters. Once we have analyzed the performance found on an attribute basis, we

show in Table 3 the result of the fairness-aware evaluation metrics. We first analyze which of the ideal distributions is better approximated by the initial methods and whether this situation changes when we use a defended model. Analyzing $GCE$ corresponding to the initial methods, w\o defense, we observe a consistent behavior in both datasets: the order derived from the $GCE$ values is the same for both models. However, the actual values are different for some cases, meaning that the defendant variant diverges differently (either more or less) from that distribution than the original method. For instance, for item popularity in ML−1M, the uniform ($f_0$) and least popular items ($f_1$) obtain a lower absolute $GCE$ value for the defended model, whereas the behavior is the other way around for user interactions in LastFM.

Studying whether the attack methods modify fairness performance, we observe that some attack methods like BIM help to increase the fairness on some distributions (or attribute values) at the expense of others, such as $f_1$ for user gender and $f_4$ for user interactions in ML−1M, at the expense of $f_2$ and $f_1$ respectively. Finally, we explore whether any attribute is more sensitive under a fairness perspective since this may be a strong signal that a recommender is under attack. Thus, we note that FGSM tends to obtain very similar $GCE$ values and $MADR$ values in almost every scenario, whereas $MADr$ tends to change independently from the perturbation. Because of this, we conclude that if we measure fairness based on ranking performance (i.e., according to $GCE$ or $MADR$), an FGSM attack might go unnoticed, whereas $MADr$ is more sensitive to any attack. On the other hand, the rest of the attack strategies seem to change too much the recommendations' distribution, as it becomes evident in the $GCE$ values of item popularity.

## Conclusion and Future Work

We proposed multi-step adversarial perturbation (MSAP) on the embeddings of MF recommenders. We studied the MSAP impact on two datasets and two MF recommenders, i.e., BPR-MF and AMF. Experiments show that under a fixed perturbation budget, the MSAP strategies are more effective than the state-of-the-art single-step method on degrading accuracy and beyond-accuracy recommendation quality. They showed that MSAP (i) impaired BPR-MF so much so that it becomes worse than a random recommender, (ii) reduced AMF performance by up to 50%, and (iii) produced the same performance drop as of FGSM with 5-time smaller perturbation ($\epsilon$). Furthermore, we verified that MSAP impacted the fairness measurements considerably. In the future, we plan to study how to robustify the recommender against MSAP and design perturbations to alter its fairness.

## Acknowledgment

The authors acknowledge partial support of the projects: Servizi Locali 2.0, PON ARS01_00876 Bio-D, PON ARS01_00821 FLET4.0, PON ARS01_00917 OK-INSAID, H2020 PASS-PARTOUT, and PID2019-108965GB-I00.

## References

Akhtar, N., and Mian, A. S. 2018. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*.

Athalye, A.; Carlini, N.; and Wagner, D. A. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*.

Beigi, G.; Mosallanezhad, A.; Guo, R.; Alvari, H.; Nou, A.; and Liu, H. 2020. Privacy-aware recommendation with private-attribute protection using adversarial learning. In *WSDM*.

Burke, R.; O'Mahony, M. P.; and Hurley, N. J. 2015. Robust collaborative recommendation. In *Recommender Systems Handbook*.

Carlini, N., and Wagner, D. A. 2017. Towards evaluating the robustness of neural networks. In *IEEE S&P*.

Chen, H., and Li, J. 2019. Adversarial tensor factorization for context-aware recommendation. In *RecSys*.

Deldjoo, Y.; Anelli, V. W.; Zamani, H.; Kouki, A. B.; and Di Noia, T. 2019. Recommender systems fairness evaluation via generalized cross entropy. In *RMSE@RecSys*.

Deldjoo, Y.; Noia, T. D.; and Merra, F. A. 2021. A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.*

Di Noia, T.; Malitesta, D.; and Merra, F. A. 2020. Taamr: Targeted adversarial attack against multimedia recommender systems. In *DSN Workshops*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

Harper, F. M., and Konstan, J. A. 2016. The movielens datasets: History and context. *ACM TiiS*.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T. 2017. Neural collaborative filtering. In *WWW*.

He, X.; He, Z.; Du, X.; and Chua, T. 2018. Adversarial personalized ranking for recommendation. In *SIGIR*.

Huang, L.; Joseph, A. D.; Nelson, B.; Rubinstein, B. I. P.; and Tygar, J. D. 2011. Adversarial machine learning. In *AISec*.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial examples in the physical world. In *ICLR*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: bayesian personalized ranking from implicit feedback. In *UAI*.

Ricci, F.; Rokach, L.; and Shapira, B., eds. 2015. *Recommender Systems Handbook*.

Schedl, M. 2016. The lfm-1b dataset for music retrieval and recommendation. In *ICMR*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.

Tang, J.; Du, X.; He, X.; Yuan, F.; Tian, Q.; and Chua, T. 2020. Adversarial training towards robust multimedia recommender system. *IEEE TKDE*.

Tu, Z.; Zhang, J.; and Tao, D. 2019. Theoretical analysis of adversarial learning: A minimax approach. In *NeurIPS*.

Vargas, S., and Castells, P. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *RecSys*.

Yuan, F.; Yao, L.; and Benatallah, B. 2019a. Adversarial collaborative auto-encoder for top-n recommendation. In *IJCNN*.

Yuan, F.; Yao, L.; and Benatallah, B. 2019b. Adversarial collaborative neural network for robust recommendation. In *SIGIR*.

Zhu, Z.; Hu, X.; and Caverlee, J. 2018. Fairness-aware tensor-based recommendation. In *CIKM*.