AMFLP: Adversarial Matrix Factorization-based Link Predictor in Social Graphs

Giuseppe De Candia[a], Tommaso Di Noia[a], Eugenio Di Sciascio[a] and Felice Antonio Merra[a]

[a]*Politecnico di Bari, Via Edoardo Orabona, 4, 70126 Bari BA, Italy*

Abstract

Nowadays, we are involved in virtual social interactions that create social networks. These networks can be used to study the behavior of individuals in terms of connections with other people by exploiting the historically recorded data, i.e., the friends on the Facebook platform. The link prediction (LP) task accomplishes the prediction of possible new user connections. In particular, inspired by the advances of the adversarial machine learning approaches in the computer vision domain, we propose an adversarial perturbation method on matrix factorization-based link prediction models, one of the most popular classes of LP methods. After verifying the performance deterioration caused by the adversarial model perturbation in preserving accurate LP performance, we propose an adversarial-based learning approach to robustify the MF-based method integrating the loss function with an adversarial regularization term. The proposed approach, named Adversarial Matrix Factorization-based Link Predictor (AMFLP), is tested on two real-world open-source datasets to prove the efficacy of the adversarial regularization technique in robustifying the model against the perturbations on the parameters without the deterioration on the overall link prediction performance.

Keywords

Adversarial Machine Learning, Link Prediction, Matrix Factorization

1. Introduction

Human beings need to be part of, and connected to, other humans from their family, to friends, from teammates to school colleagues. Everyone has to be connected to other people giving rise to subjectively and relatively small social networks that, analyzing their overall extension, generate a vast network that connects billions of people. In the digital era, many services are publicly available to give the chance to communicate instantly in every part of the world regardless of distances. These innovations have led, over the years, to the development of numerous virtual platforms that allow interaction between people, such as Twitter, Facebook, and Instagram. All these platforms are commonly named social networks. A social network can be modeled as a (social) graph, a structure composed of nodes, e.g., people, and links, e.g., connections between people. In the simple case, a link is a connection between nodes without side information— additional information related to the connection like the date two people got friendship relation in a platform.

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy
giuseppe.decandia@poliba.it (G. De Candia); tommaso.dinoia@poliba.it (T. Di Noia); eugenio.disciascio@poliba.it (E. Di Sciascio); felice.merra@poliba.it (F. A. Merra)

Guessing how people's behavior and interests can make them decide to connect with new people are known as the link prediction (LP) task. In the last years, multiple works have addressed this task via machine learning (ML) approaches [1, 2, 3, 4]. While a deep focus has been dedicated to improving and proposing models to predict possible new connections in the best accurate way, the security of these models has raised interest only recently. Adversarial machine learning (AML), the field of study investigating the security of ML models, attracted great attention when ML-based computer vision systems used in an autonomous vehicle have been demonstrated to be easily fooled by perturbing a traffic sign in a human-imperceptible way [5, 6]. For instance, an ordinary image of a stop sign has been demonstrated to be perturbed by an adversary such that it will be classified by the vehicle as a different traffic sign making possible incidents [5]. However, while a recent part of research interest on AML in the social graph has been dedicated to investigating attack and defense approaches on altering the LP task by adding or modifying existing nodes and link [7, 8, 9], we found a lack of study on robustifying the model parameters of MF-based methods.

Inspired by the motivating scenario in recommender system domain [10], whose intuition is to find the minimal adversarial perturbation on model parameters to break the LP performance, lays down to the scenario where the easy addition of a new edge in the dataset graph, or the removal of an existing one, might cause a substantial variation of the model parameters with a consequent reduction in the reliability of the LP model. A real case example might be the suggestion of friends on social network platforms that are unlikely to be known from the users that are getting the friend suggestion, making her feel uncomfortable towards the platform's reliability. To investigate the possible existence of the weakness mentioned above, this work firstly proposes a gradient-based adversarial perturbation method against MF-based link predictors, i.e., the state-of-the-art model proposed by Equation (1), to test the performance degradation on the LP task. Then, it presents a novel approach, named Adversarial Matrix Factorization-based Link Predictor (AMFLP), by adopting an adversarial training approach to make the model more robust to the previously verified adversarial perturbation.

Our contributions can be summarized as follows:

1. we show that state-of-the-art MF model for the LP task might be adversarial modified dropping off their accuracy performance,
2. we propose an adversarial training procedure used into AMFLP to robustify a MF model against the previously defined adversary threat models;
3. we experiment on two real datasets to verify the efficacy of both the adversarial attack and defense methods on multiple accuracy measures.

2. Related Work

2.1. Link Prediction in Social Networks

A social network is a connections structure made up of social actors and links between them that can be visualized as a graph, where *nodes* represent people and *edges* correspond to interactions/relationships. The link prediction task in a social network scenario aims to predict the

new connection between nodes. State of the art provides two ways to solve this problem [11]: *similarity-based approaches* and *learning-based approaches*. The *similarity-based approaches* use different similarity metrics to predict the existence of connections, while *learning-based approach* treats the link prediction problem as a binary classification task. We identify two main techniques to evaluate the similarities across nodes based on either node or topology metrics about the first category. Node-based metrics work considering that users tend to create relationships with people who are similar in education, religion, interests, and locations [12, 13, 14]. This insight is exploited for the assignment of a score to a pair of nodes —a high score indicates a high probability that two nodes will have a link. Topology-based metrics take advantage of the graph's topology using, for instance, neighbor-based metrics such as the Jaccard coefficient, the number of common neighbors, or the Salton-Cosine similarity [15]. On the other hand, the *learning-based approaches* solve the problem as a binary classification based. For instance, Scripps et al. [1] propose a discriminative learning mechanism to determine the most predictive attributes and topological features automatically. Menon et al. [3] propose one of the most used mechanisms extending the MF-based method while gathering the support of external side information. In this work, we investigate the robustness of learning-based approaches with interest in the widely used Menon et al. [3] proposal.

2.2. Adversarial Machine Learning

There is growing recognition that machine learning models are vulnerable to adversarial samples. Adversarial machine learning (AML) is the field of study for the security of ML models [16]. AML techniques aim to develop models or strategies that generate high-quality prediction results and are robust to malicious third parties, i.e., the adversaries. Adversaries could manipulate documents [17], images [18, 19], graph-data [20, 21, 22, 23], knowledge graphs [24], training data [25], and model parameters [10]. In particular, link predictions could be affected by attackers who adversarially modify observed topology or side information to hide target links. In [26], the authors study the vulnerability of centrality measures. Zhou et al. [8] experimentally analyze the robustness of several similarity metrics. Chen et al. [9] proposed an iterative gradient-based approach to simulate the construction of adversarial graph. Lin et al. [27] focus on evasion attack (test-time), crafting adversarial examples to deceive graph neural network LP models via optimized perturbation of the graph topology. While previous works have been focused on the insertion, or removal, of edges and nodes to alter the LP performance, we focus on analyzing the robustness of the learned model parameters.

3. Method

This study carries to the production of an adversarial defended MF-based approach, starting from the assessment of the adversarial risks of the Menon et al. [3] model. Below, we describe our attack and defense procedures.

3.1. Preliminaries.

The MF model under analysis factorizes the original adjacency matrix built up the historical node-node connections using also side information related to both the single node and the pair of nodes. The cost function minimized in Menon's model is defined as follow

$$\min_{U,\Lambda,V,w,b} \frac{1}{|\mathcal{O}|} \sum_{i=1}^{n} \sum_{j \in \mathcal{O}_i^+, k \in \mathcal{O}_i^-} \ell\left(L\left(u_i^T \Lambda\left(u_j - u_k\right) + b_i + b_j + x_i^T V x_j + w^T z_{ij}\right), 1\right) \tag{1}$$

where,

- $\mathcal{O}^+$ and $\mathcal{O}^-$ are the sets of known present and known absent dyads respectively,
- u_i and u_j are latent vectors relative to the node i and node j, extracted from the matrix $U \in \mathbb{R}^{n \times f}$ with n that represents the number of nodes and f the number of latent features;
- $\Lambda \in \mathbb{R}^{f \times f}$ is a square matrix such that $G \approx L(U \Lambda U^T)$, with G is the original adjacency matrix of the observed graph
- b_i and b_j are biases extracted from B, a $\mathbb{R}^n$ vector
- x_i and x_j are vectors extracted from the matrix X that includes moniadic information for each node in the graph
- V is a squared low rank matrix
- $z_{i,j}$ is a vector extracted from the matrix Z that contains dyadic information for each pair of nodes in the graph
- w is a learnable parameters multiplied by the term $z_{i,j}$.

In the previous equation, $L(\cdot)$ is the **link function** and $l(\cdot)$ is the **loss/objective function**. Following [3] we model

- $\boldsymbol{L}$ as a sigmoid function:

$$\sigma(Arg^{(i,j)}) = \frac{1}{1 + e^{-Arg^{(i,j)}}} \tag{2}$$

- $\boldsymbol{l}$ as cross entropy loss function:

$$\mathrm{J(Arg)} = -\frac{1}{m} \cdot \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} ((y^{(i,j)} \log\left(\sigma(Arg^{(i,j)}\right)) + (1 - y^{(i,j)}) \log(1 - \sigma(Arg^{(i,j)}))) \tag{3}$$

with $n = m$, and

$$Arg^{(i,j)} = u_i^T \Lambda u_j + b_i + b_j + x_i^T V x_j + w^T z_{ij} \tag{4}$$

Notice that to refer to the whole system we use Arg instead of $Arg^{(i,j)}$ that is only related to a generic couple of nodes. Such cost function, that indicates how big is the prediction error, is the guideline to make the learning of whole LP system. The learning of the model parameters is executed via stochastic gradient descent (SGD).

3.2. Attack: Gradient-based Adversarial Perturbation

After the definition of the model under attack, here we present the procedure to adversarially perturb its learned parameters. Following the study by Xiangnan He et al. [28] for recommendation task, we express the adversarial perturbation as follows:

$$\Delta_{Adv} = \arg\max J(Arg^{(i,j)} + \Delta) \tag{5}$$

with Δ as a generic noise on model parameters that must satisfy the constrain $\|\Delta\| \leq \epsilon$, where $\|\cdot\|$ is the L2 norm and ϵ is the budget magnitude for the maximum perturbation. The optimal solution for Δ_{Adv} is difficult to get but it is possible to employ the fast gradient method [5] to approximate the objective function around Δ as a linear function. We calculate the adversarial perturbation as follows

$$\Delta_{Adv} = \epsilon \frac{\Gamma}{\|\Gamma\|} \text{ where } \Gamma = \frac{\partial J(Arg^{(i,j)} + \Delta)}{\partial \Delta} \tag{6}$$

In the last expression, the adversarial noise is expressed in function of the parameter Γ and the parameter ϵ —the bounds of the perturbation. In the experimental section, we will present the effectiveness of this attack method on the tested MF-based LP model.

3.3. Defense: Adversarial Training

In the previous subsection, we explained how it is possible, at the inner loop level, to generate the perturbation Δ_{Adv} for each couple of nodes i and j, for each epoch. Differently from the classical learning updates, we must consider a novel objective function that introduced the adversarial regularization. We define

$$J_{Adv}(Arg^{(i,j)}) = J(Arg^{(i,j)}) + \eta J(Arg^{(i,j)} + \Delta_{Adv}) \tag{7}$$

as the objective function of the proposed Adversarial Matrix Factorization model for Link prediction (AMFLP). In Equation 7 the term $J(Arg^{(i,j)} + \Delta_{Adv})$ can be seen as a regularization term where η — the adversarial regularization coefficient— controls its strength.

To learn the adversarial regularized parameters of **AMFLP**, we introduce the characteristics of the adversarial training, also known as **minimax game** [29], that is expressed as

$$Arg^*, \Delta^* = \arg\min_{Arg} \max_{\Delta, \|\Delta\| \leq \epsilon} J(Arg) + \eta J(Arg + \Delta) \tag{8}$$

where the intuition is that, fixing the ϵ budget to perturb the model, the training has to proceed in two steps; first, it has to build the perturbation that maximizes the loss, then, after having added it into the model, it has to minimize the loss in this worst-case setting. The proposed learning algorithm is presented in Algorithm 1.

3.4. AMFLP Gradients

Here, we define the updates of the parameters to optimize the objective function in Equation 7. Using Equation 3 the whole objective function is

$$\begin{aligned} J_{Adv}(Arg^{(i,j)}) = &-(y^{(i,j)} \log \sigma(Arg^{(i,j)}) + (1 - y^{(i,j)}) \log(1 - \sigma(Arg^{(i,j)}))) \\ &-\eta(y^{(i,j)} \log \sigma(Arg^{(i,j)} + \Delta_{Adv}) + (1 - y^{(i,j)}) \log(1 - \sigma(Arg^{(i,j)} + \Delta_{Adv}))) \end{aligned} \tag{9}$$

Algorithm 1 AMFLP Training.

Set N : epochs; α: learning rate; ϵ: pert. budget; η: adv. reg. coeff.; K : num. updated parameters: m nd n are the number starting and ending nodes in the graph (i.e., $m = n$).

for $epoch = 1, 2, \dots, N$ **do**
  for $i = 1, 2, \dots, m$ **do**
    for $j = 1, 2, \dots, n$ **do**
      Calculate prediction: $Arg^{(i,j)}$
      Compute Γ and Δ_{Adv}
      Calculate perturbed prediction
      Calculate prediction's Sigmoid $\sigma(Arg^{(i,j)})$
      Calculate perturbed prediction's Sigmoid $\sigma(Arg^{(i,j)} + \Delta_{Adv})$
      Gradients Calculation
      for $k = 1, 2, \dots, K$ **do**
        $\Theta_k \leftarrow \Theta_k - \alpha((\sigma(Arg^{(i,j)}) - y)\frac{\partial Arg^{(i,j)}}{\partial \Theta_k} + \eta(\sigma(Arg^{(i,j)} + \Delta_{Adv}) - y^{(i,j)})\frac{\partial Arg^{(i,j)} + \Delta_{Adv}}{\partial \Theta_k})$
      end for
    end for
  end for
end for

Notice that the first addend of the function is the same as the original cost function that doesn't include the adversarial component. For this reason, and for the *linear property of the derivative*, we show the partial derivatives relatively to the second (adversarial) addend. The partial derivatives on the perturbed model is defined as

$$\frac{\partial J(Arg^{(i,j)} + \Delta_{Adv})}{\partial \Theta_k} = \eta(\sigma(Arg^{(i,j)} + \Delta_{Adv}) - y^{(i,j)})\frac{\partial Arg^{(i,j)} + \Delta_{Adv}}{\partial \Theta_k} \tag{10}$$

where Θ_k could be a model parameter (or a perturbation) that will be updated during the training.

Now is possible to go into more detail and derive the inner adversarial perturbed argument. In this case, the argument that coincides with the not normalized prediction includes the noise in the following explicit form:

$$Arg^{(i,j)} + \Delta = (u_i^T + \Delta^{u_i})\Lambda(u_j + \Delta^{u_j}) + b_i + b_j + (x_i^T + \Delta^{x_i})V(x_j + \Delta^{x_j}) + w^T z_{ij} \tag{11}$$

The derivatives to build the adversarial perturbations for the K-perturbed parameters are measured as shown in Table 1.

Table 1
Derivatives of Equation (10).

$\frac{\partial Arg^{(i,j)}+\Delta}{\partial u_i} = \Lambda(u_j + \Delta^{u_j})$	$\frac{\partial Arg^{(i,j)}+\Delta}{\partial u_j} = (u_i^T + \Delta^{u_i})\Lambda$	$\frac{\partial Arg^{(i,j)}+\Delta}{\partial b_i} = 1$
$\frac{\partial Arg^{(i,j)}+\Delta}{\partial \Lambda} = u_i u_j + \Delta^{u_i} u_j + \Delta^{u_j} u_i + \Delta^{u_i}\Delta^{u_j}$		$\frac{\partial Arg^{(i,j)}+\Delta}{\partial b_j} = 1$
$\frac{\partial Arg^{(i,j)}+\Delta}{\partial V} = x_i x_j + \Delta^{x_i} x_j + \Delta^{x_j} x_i + \Delta^{x_i}\Delta^{x_j}$		$\frac{\partial Arg^{(i,j)}+\Delta}{\partial w} = z_{ij}$
$\frac{\partial Arg^{(i,j)}+\Delta}{\partial \Delta^{u_i}} = \Lambda(u_j + \Delta^{u_j})$	$\frac{\partial Arg^{(i,j)}+\Delta}{\partial \Delta^{u_j}} = (u_i^T + \Delta^{u_i})\Lambda$	$\frac{\partial Arg^{(i,j)}+\Delta}{\partial \Delta^{x_i}} = V(x_j + \Delta^{x_j})$
$\frac{\partial Arg^{(i,j)}+\Delta}{\partial \Delta^{x_j}} = (x_i^T + \Delta^{x_i})V$		

4. Experiment

In this section, we aim to investigate the research questions:

RQ1: What is the effect of the adversarial perturbation strategy described in Section 3.2 on the accuracy performance of the state-of-the-art MF model for link prediction [3]?

RQ2: Is AMFLP, the MF-based model that integrates the defense strategy proposed in Section 3.3, more protected against the gradient-based perturbation of the model parameters?

To answer RQ1 and RQ2, first, we present the experimental settings, then, we report and discuss the experimental results of AMFLP.

4.1. Setup

4.1.1. Datasets.

We test AMFLP on two popular datasets for the link prediction task on social graphs: `Conflict` and (an extraction of) `Facebook`.

The first dataset, `Conflict`, is a dataset containing information about military disputes between countries [30] in the period 1990-2000. Following the experimental settings in [3], the directed graph is converted into an undirected graph with the result that the adjacency matrix (U) is symmetric. It contains features on both nodes and pairs of nodes (or dyads). Each node has three features, i.e., population, GDP, and polity, while each dyad has five features, e.g., countries' geographic distance. These features are anonymized and transformed in numerical values to be fed into AMFLP.

The second dataset, `Facebook`, is an anonymized collection of nodes, edges, and other side information relative to a sub-net of the social network platform. It has been released by Jure Leskovec and is publicly available [1]. The data, collected from a survey whose participants answered in the Facebook application, provide anonymized feature vectors. For instance, where the original dataset may have contained a feature "political=Democratic Party," the new data would contain "political=anonymized feature 1". Thus, it is possible to use the anonymized data to determine whether two users have the same political affiliations but not the specific party. We reduce the number of features to 19 using the ones that have preserved the 70% of

[1] http://snap.stanford.edu/data/ego-Facebook.html

Table 2
Statistics of the test datasets

Dataset	Nodes	$\mathcal{O}^+$	$\mathcal{O}^-$	$\frac{\mathcal{O}^+}{\mathcal{O}^-}$	Used Features
Conflict	130	320	16580	1:52	3
Facebook	792	28048	613240	1:22	19

variance using the principal component analysis. For instance, we remove features such as name, surname, the end date of a job, leaving the essential features such as spoken languages, hometown, education, attended school, and job position. The removal of 176 features makes the computation faster because of its square complexity. The statistics of the two datasets are summarized in Table 2. We split both datasets in training and test set, putting the 90% of links in the training set and the remaining 10% in the test set.

4.1.2. Evaluation Metrics

Since the link prediction task could be seen as a classification problem, predicting whether a connection between two nodes exists or not, it is possible to use accuracy measures. In particular, we compute the Accuracy (Acc), Area under the ROC Curve (AUC), and the Hit Ratio (HR) at three thresholds, i.e., 3, 7, and 10. Acc calculates how often predictions equal labels. AUC measures the probability that a randomly chosen missing link s given a higher similarity score than a randomly chosen pair of unconnected links. Note that AUC, differently from Acc, does not suffer from the unbalance data distribution problem [3]. HR@k computes the fraction of test nodes correctly guessed to be connected to one of the top-k predicted links. The Hr is a recall-based metric that helps to understand if increasing the analyzed cut-off of suggested nodes; the model can correctly predict a link connection after k attempts.

4.1.3. Reproducibility

We use a grid search with 8-fold cross-validation to train the experimented methods in order to find the best hyper-parameters on which to investigate the attack and defense performance. The explored hyper-parameters for both datasets are defined as follow $\alpha_U : \{0.05, 0.01\}$, $\alpha_\Lambda : \{0.05\}$, $\alpha_b : \{0.005, 0.0005\}$, $\alpha_V : \{0.05, 0.005\}$, $\alpha_w : \{0\}$, $\eta : \{1.0\}$, and $\epsilon : \{0.5\}$, where we set η and ϵ values following [10] and α_k is the learning rate used to update the k-th parameter. We train the standard MF model for N iterations. However, after $N/2$ training epochs, we fork the learning process in two ways. In the first training, we continue the standard training until the N-th epoch. In the second training, the one related to AMFLP, we perform the adversarial training procedure for the remaining $N/2$ epochs using the defense approach described in Section 3.3. We set $N = 1000$. After having explored the search space, we report in Section 4.2 the results measured on the best model with respect to the test set.

4.2. Results and Discussion

Table 3 reports the link prediction performance evaluated on `Conflict` and `Facebook`. For each dataset, we report three **Attack** settings: *No*, *Random*, and *Adversarial*. The first setting

Table 3
Results of random and adversarial attacks against MF and AMFLP.

Dataset	Conflict						Facebook					
Attack	*No*		*Random*		*Adversarial*		*No*		*Random*		*Adversarial*	
Model	MF	AMFLP	MF	AMFLP	MF	AMFLP	MF	AMFLP	MF	AMFLP	MF	AMFLP
Acc	.9790	.9790	.9790	.9800	.9670	.9740	.9555	.9555	.9555	.9555	.0008	.9410
AUC	.8666	.8666	.8666	.8666	.6600	.8420	.7960	.7850	.7920	.7851	0	.3190
HR@3	.0312	.0625	.0190	.0210	.0312	.0312	.0035	.0035	.0037	.0037	.0018	.0025
HR@7	.0937	.1250	.0581	.0512	.0625	.0312	.0110	.0114	.0122	.0130	.0074	.0110
HR@10	.1250	.2812	.0718	.0756	.0937	.0625	.0160	.0192	.0187	.0195	.0132	.0164

is related to the base model performance without any adversarial (or random) perturbation. *Random* attack bounded at ϵ is used as the **baseline** approach to verify if the adversarial attack procedure presented in Section 3.2 is not degrading the model performance due to the addition of random noise. The *Adversarial* column reports the results on the adversarial perturbations computed when $\epsilon = 0.5$. Finally, we report also two model: MF and AMFLP. In this section, we use the name MF to indicate the MF-based model proposed in [3].

4.2.1. Attack: analysis of the performance (RQ1).

Looking at Table 3 it is possible to focus on the three couple of columns, looking that in the case of *Random* attack, MF has a slight drop on Acc and AUC metrics. Conversely, analyzing the *Adversarial* attacks, we can observe that both metrics reduced much more than in the random attack setting. For instance, looking at the Conflict dataset, the AUC of MF goes down by more than 24% with the application of the adversarial perturbation, while it is not getting performance worsening when using random perturbations. This phenomenon is even more evident for the Facebook dataset, in which we observe that both Acc and AUC get values close to 0, making the model completely unreliable. Additionally, it is interesting to observe that the drop of accuracy performance can also be partially verifiable on the analysis of ranking-based metrics. Analyzing the result values for the Facebook dataset, we observe that HR@10 is 0.0160 in the no-attack setting on MF, and it is even increased to 0.192 in the random one, while, as expected, is reduced to 0.0164 with the use of the adversarial approach. At the same time, it can be observed in the other dataset that *Random* seems to be more potent in reducing the HR if compared to *Adversarial*. The reason is that the HR metric is a ranking-wise metric, while the adversarial mechanism is a score-based reduction approach that may have reduced the score of a possible link suggestion but not enough to put it below the top-k list. Indeed, the first two accuracy measures confirm that the *Adversarial* approach is the most effective one.

For the previous analysis of the attack performance, we can claim that *the gradient-based adversarial perturbation of model parameters can reduce the accuracy performance of a link prediction model.* Below, we verify the AMFLP performance to understand if we would have better performance under the same attack under the adversarial training setting.

4.2.2. Defense: analysis of the performance (RQ2).

Having assessed worrying deterioration performance due to adversarial perturbation, here we test if the use of the adversarial training procedure presented in Section 3.3 and used in AMFLP would guarantee a minor degradation of performance.

To verify the efficacy of AMFLP is necessary to compare the values of the *Adversarial* column for both datasets in Table 3. The accuracy and AUC results for `Conflict` show that the model AMFLP efficiently reduces the effects of the adversarial perturbation on the model parameters. For instance, the AUC value is 0.8520 for the attack against AMFLP, while 0.6600 for the attack on the no-defended model. Extending the analysis on the second tested dataset, i.e., `Facebook`, it can be observed that the goodness of the proposal defense is consistently confirmed. The complete worsening of the performance observed on the no-defended scenario is successfully improved in the defended one. For instance, the complete loss of Acc observed on MF is quite wholly solved with the Acc value equals to 0.9410 measured on AMFLP. Observing the hit ratio performance, it can be seen that, similarly to the observations made in Section 4.2.1, further improvements on the proposed approach might be helpful also to robustify rank-wise performance. Analyzing the general performance obtained in *No* attack columns, it can be claimed that the adversarial regularization of MF applied in AMFLP has not reduced the general performance of the model. Instead, we can observe that AMFLP might have even improved the model performance. For instance, HR@7 increases from 0.0937 to 0.1250 in the `Conflict` dataset, and from 0.0110 to 0.0114 in the `Facebook` one.

We can claim that the *adversarial training of a matrix factorization model to address the link prediction task can robustify the model against gradient-based adversarial perturbation while preserving, or even improving, the high accurate prediction performance.*

5. Conclusion

We have investigated the link prediction task, which, for instance, helps people to find new possible connections, e.g., friends on social networks. Inspired by the adversarial attacks in computer vision, e.g., an autonomous vehicle can turn left instead of stopping with a human-imperceptible perturbation of a traffic signal, we have investigated the security of matrix factorization-based link predictor. Firstly, we have proposed a perturbation technique that minimally alters the learned model parameters by reducing the accuracy performance in the link prediction task. Then, after having verified the weakness of MF-based models against the proposed adversary threat model, we have presented an adversarial training solution employing the minimax procedure. This procedure has been implemented in the proposed Adversarial Matrix Factorization-based Link Predictor (AMFLP). The goal is to build the perturbations that maximize the model loss and minimize the global objective, considering the possible deterioration caused by the perturbation. To verify the efficacy of the proposed attack and defense scenarios, we have tested two popular datasets, i.e., Conflict and Facebook, verifying that the proposed defensive strategy can robustify the model accuracy. Our results also open further investigation on how to improve even more the robustification performance. Additionally, we plan to extend this method to deep learning-based link-predictors to verify their possible weakness and the efficacy of the proposed robustification approach.

Acknowledgments

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 PASSPARTOUT.

References

[1] J. Scripps, P. Tan, F. Chen, A. Esfahanian, A matrix alignment approach for link prediction, in: ICPR, IEEE Computer Society, 2008, pp. 1–4.

[2] J. Kunegis, A. Lommatzsch, Learning spectral graph transformations for link prediction, in: ICML, volume 382 of *ACM International Conference Proceeding Series*, ACM, 2009, pp. 561–568.

[3] A. K. Menon, C. Elkan, Link prediction via matrix factorization, in: Joint european conference on machine learning and knowledge discovery in databases, Springer, 2011, pp. 437–452.

[4] M. Pujari, R. Kanawati, Link prediction in complex networks by supervised rank aggregation, in: ICTAI, IEEE Computer Society, 2012, pp. 782–789.

[5] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014. arXiv:1412.6572.

[6] P. D. McDaniel, N. Papernot, Z. B. Celik, Machine learning in adversarial settings, IEEE Secur. Priv. 14 (2016) 68–72.

[7] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, L. Song, Adversarial attack on graph structured data, in: ICML, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 1123–1132.

[8] K. Zhou, T. P. Michalak, Y. Vorobeychik, Adversarial robustness of similarity-based link prediction, in: ICDM, IEEE, 2019, pp. 926–935.

[9] J. Chen, X. Lin, Z. Shi, Y. Liu, Link prediction adversarial attack via iterative gradient attack, IEEE Trans. Comput. Soc. Syst. 7 (2020) 1081–1094.

[10] X. He, Z. He, X. Du, T. Chua, Adversarial personalized ranking for recommendation (2018) 355–364.

[11] P. Wang, B. Xu, Y. Wu, X. Zhou, Link prediction in social networks: the state-of-the-art, Sci. China Inf. Sci. 58 (2015) 1–38.

[12] C. G. Akcora, B. Carminati, E. Ferrari, User similarities on social networks, Soc. Netw. Anal. Min. 3 (2013) 475–495.

[13] A. Anderson, D. P. Huttenlocher, J. M. Kleinberg, J. Leskovec, Effects of user similarity in social media, in: WSDM, ACM, 2012, pp. 703–712.

[14] P. Bhattacharyya, A. Garg, S. F. Wu, Analysis of user keyword similarity in online social networks, Soc. Netw. Anal. Min. 1 (2011) 143–158.

[15] J. Repplinger, G.G. chowdhury. *Introduction to Modern Information Retrieval.* 3rd ed. london: Facet, 2010. 508p. alk. paper, $90 (ISBN 9781555707156). LC2010-013746, Coll. Res. Libr. 72 (2011) 194–195.

[16] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, J. D. Tygar, Adversarial machine

learning, in: Proceedings of the 4th ACM workshop on Security and artificial intelligence, 2011, pp. 43–58.
[17] G. Goren, O. Kurland, M. Tennenholtz, F. Raiber, Ranking robustness under adversarial document manipulations, in: SIGIR, ACM, 2018, pp. 395–404.
[18] T. Di Noia, D. Malitesta, F. A. Merra, Taamr: Targeted adversarial attack against multimedia recommender systems, in: DSN Workshops, IEEE, 2020, pp. 1–8.
[19] V. W. Anelli, Y. Deldjoo, T. Di Noia, D. Malitesta, F. A. Merra, A study of defensive methods to protect visual recommendation against adversarial manipulation of images, in: SIGIR, ACM, 2021.
[20] Y. Sun, S. Wang, X. Tang, T. Hsieh, V. G. Honavar, Adversarial attacks on graph neural networks via node injections: A hierarchical reinforcement learning approach, in: WWW, ACM / IW3C2, 2020, pp. 673–683.
[21] N. Entezari, S. A. Al-Sayouri, A. Darvishzadeh, E. E. Papalexakis, All you need is low (rank): Defending against adversarial attacks on graphs, in: WSDM, ACM, 2020, pp. 169–177.
[22] D. Ding, M. Zhang, X. Pan, M. Yang, X. He, Improving the robustness of wasserstein embedding by adversarial pac-bayesian learning, in: AAAI, AAAI Press, 2020, pp. 3791–3800.
[23] D. Zügner, S. Günnemann, Adversarial attacks on graph neural networks via meta learning, in: ICLR (Poster), OpenReview.net, 2019.
[24] V. W. Anelli, Y. Deldjoo, T. Di Noia, E. Di Sciascio, F. A. Merra, Sasha: Semantic-aware shilling attacks on recommender systems exploiting knowledge graphs, in: ESWC, volume 12123 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 307–323.
[25] Y. Deldjoo, T. Di Noia, E. Di Sciascio, F. A. Merra, How dataset characteristics affect the robustness of collaborative recommendation models, in: SIGIR, ACM, 2020, pp. 951–960.
[26] M. Waniek, T. P. Michalak, T. Rahwan, M. J. Wooldridge, Hiding individuals and communities in a social network, CoRR abs/1608.00375 (2016).
[27] W. Lin, S. Ji, B. Li, Adversarial attacks on link prediction algorithms based on graph neural networks, in: AsiaCCS, ACM, 2020, pp. 370–380.
[28] X. He, H. Zhang, M. Kan, T. Chua, Fast matrix factorization for online recommendation with implicit feedback (2016) 549–558.
[29] Y. Deldjoo, T. Di Noia, F. A. Merra, A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks, ACM Comput. Surv. 54 (2021) 35:1–35:38.
[30] F. Ghosn, G. Palmer, S. A. Bremer, The mid3 data set, 1993—2001: Procedures, coding rules, and description, Conflict Management and Peace Science 21 (2004) 133–154. doi:10.1080/07388940490463861.