Knowledge-enhanced Shilling Attacks for Recommendation*
(Discussion Paper)

Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Felice Antonio Merra **, Giuseppe Acciani, and Eugenio Di Sciascio

Politecnico di Bari
{name.surname}@poliba.it

Abstract. Collaborative filtering (CF) recommendation models lie at the core of most industrial engines due to their state-of-the-art performance. Their leading performance owes hugely on exploiting users' past feedbacks to identify similar user or item pairs. Unfortunately this similarity computation is vulnerable to shilling profile injection attack, in which an attacker can insert fake user profiles into the system with the goal to alter the similarities and resulting recommendations in an engineered manner. In this work, we introduce *SAShA*, a new attack strategy that leverages semantic features extracted from a knowledge graph in order to strengthen the efficacy of the attack against standard CF models. Validation of the system is conducted across two publicly available datasets and various attacks, CF models and semantic information. Results underline the vulnerability of well-known CF models against the proposed semantic attacks compared with the baseline version.

Keywords: Recommender System · Knowledge Graph · Shilling attack.

1 Introduction and Context

With the increasing popularity of Internet commerce, online services, and the overwhelming volume of products, services, and multimedia content, recommender systems (RS) play a key role in mitigating the users' cognitive burden of over-choice. RS assist users' decision-making process by pointing them to a small set of items out of a large catalog (top-k recommendation list), based on users' past behaviors and preferences. The recommendation model can be broadly classified as content-based filtering (CBF), collaborative filtering (CF), and hybrid. CF models are the most popular choice in academic and industrial research (e.g.,

* Extended version [3] published at the 17th European Semantic Web Conference 2020

** Corresponding author: Felice Antonio Merra (felice.merra@poliba.it)

Amazon [19]) due to their high recommendation performance. Their key insight is that users' personal tastes correlate and then, from an algorithmic point of view, they mainly rely on the exploitation of user-user and item-item similarities. Different CF approaches can be classified into broad classes of memory-based and model-based. Memory-based approaches compute recommendations exclusively based on similarities in interaction patterns computed either across users (user-based CF) or items (item-based CF) [16]. Model-based exploits different machine learning techniques to compute a model, typically a latent representation of items and users [17], to generate recommendations. The most well-known example of a model-based approach is the matrix factorization (MF) model. Regardless of the type, both CF recommendation classes heavily rely on a sufficient amount of user preference data in order to mine reliable similarity patterns. Unfortunately, due to the open nature of many online systems, a malicious agent can add fake profiles into the platform to leverage similar values and the following recommendation outcomes in an engineered manner. Such profile injections are known as shilling attacks (or profile injection attacks) [15] whose goal are often malicious, for example for pushing or nuking a target item into a the top-k recommendation list of users for market penetration or personal gain [10].

The other alternative approach CBF (or hybrid) relies on items' descriptive attributes in conjunction with the target user's previous preference over an item in order to create a profile of the user characterizing the nature of her interest(s). While the earliest versions of CBF were purely textual using information such as metadata (tags, reviews) [21], modern versions utilize variety of other rich information sources such as social connections [6], audio and visual content [9,8] as well as users-item contextual data [2] to build more domain-dependent context-aware recommendations models. Another rich source of information and the one we exploit in this work has received increased attention from the community RS is the knowledge graph (KG). A $\mathcal{KG}$ can be viewed as a structured repository of knowledge, represented in the form of a graph, capable of encoding a diverse set of information:

- **Factual.** General statements as *Heraklion is the capital of Crete* or *Cyrus the Great was the founder of the first Persian Empire* in which an entity is described in term of a number of attributes, which are in turn connected with other entities in the $\mathcal{KG}$;
- **Categorical.** These statements bind an entity to a specific category in the $\mathcal{KG}$ (i.e., the categories related to an article in Wikipedia pages), where the categories together form a hierarchy (can be general or specific).
- **Ontological.** We can classify entities in a more formal manner by utilizing a hierarchical structure of classes. In contrast to categories, here sub-classes and super-classes are connected through an IS-A relation.

In fact, $\mathcal{KG}$ constitute the foundation of the Semantic Web and are becoming increasingly important as they can represent data exploiting a manageable and inter-operable semantic structure. They are the pillars of well-known tools like IBM Watson [7], public decision-making systems [22], and advanced machine learning techniques [4]. For what concerns recommendation based on $\mathcal{KG}$,

they can be classified into: (i) path-based methods [14], which use meta-paths to evaluate the user-item similarities and, (ii) $\mathcal{KG}$ embedding-based techniques, that leverages $\mathcal{KG}$ embeddings to semantically regularize items latent representations [23,11].

The main contributions of this work are two-fold:

1. We build a novel type of shilling attacks against rating-based CF models that leverages the publicly available information resources from $\mathcal{KG}$ to build impactful shilling profile attacks against CF models.
2. To investigate the relationship between semantic data characteristics and the robustness of CF models, we carried out extensive experiments involving two popular attack strategies against three well-known CF models across two real-world datasets. In total 84 simulation attacks were conducted to verify the impact of the semantic knowledge integration.

2 Our Proposed Approach - SAShA

When an attacker successfully inserts a malicious user profile in the dataset, it needs to assign a set of rated items — besides the target item (i_t) — so that the profile can be used in CF recommendation models. Table 1 shows the composition of the explored state-of-the-art attack profiles. For instance, I_S (selected item set) is a set of items identified by the attacker to maximize the effectiveness of the attack, while I_F (filler item set) includes a random set of items whose role is to make the attack imperceptible. Details about the attack profile composition can be found in [15] and in the extended version of the current work [3].

In this work, we propose to foster the efficacy of state-of-the-art attack strategies by exploiting the semantic similarities between items using the information extracted from $\mathcal{KG}s$. The key idea is that we can compute the semantic similarity between the target item i_t and all the items in the catalog using $\mathcal{KG}$-derived features. Then, we use this information to select the filler items of each profile to generate the set I_F. A similarity value based on $\mathcal{KG}$ features leads to a more natural and coherent fake profile, thanks to the semantic nature of $\mathcal{KG}s$. Toward this goal, we propose two semantic-aware attacks by extended state-of-the art random and average attack:

- *Semantic-Aware Shilling Attack-random (SAShA-random)* is an extension of Random Attack. The baseline version is a naive attack in which each fake user is composed only of random items. We modify this attack by extract items to fill I_F from a subset of items that are most similar to i_t and use standard cosine similarity to compute item similarities by *leveraging the semantic features* [12]. Then, we build a set of most-similar items, considering the first quartile of similarity values. Finally, we extract ϕ items from this set, adopting a uniform distribution.
- *Semantic-Aware Shilling Attack-average* (*SAShA-average*) is an informed attack that extends the AverageBots attack [20]. It randomly samples the rating of each filler item from a normal distribution computed using the

Table 1: Attack strategies and their profile composition (*push* attacks).

Attack Type	I_S		I_F		I_ϕ	i_t
	Items	Rating	Items	Ratings		
Random [18]	$\emptyset$		$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	$rnd(N(\mu, \sigma^2))$	$I - I_F$	max
Average [18]	$\emptyset$		$\frac{\sum_{u \in U} \lvert I_u \rvert}{\lvert U \rvert} - 1$	$rnd(N(\mu_f, \sigma_f^2))$	$I - I_F$	max

where (μ, σ) are the dataset average rating and rating variance, (μ_f, σ_f) are the filler item i_f rating average and variance, and min and max are respectively the minimum and maximum rating value.

mean and the variance of the ratings. We extend the baseline by extracting the filler items from the sub-set of most similar items. We use as candidate items the ones in the first quartile regarding their similarity with i_t.

3 Experimental Setting

In this section, we explain the experimental setting and results of the proposed attack framework (*SAShA*). We remind that full detail about the experimental procedure and evaluation can be found in [3].

Data: We conducted the proposed semantic average shilling attacks against rating-based CF models on three real-world datasets, `LibraryThing` [13] and `Yahoo!Movies`. The final statistics of the dataset used for the experiment are summarized in Table 2.

Feature Extraction and Selection. We have extracted the semantic information to build *SAShA* exploiting the public available item-entity mapping to `DBpedia`. To analyze the impact of different feature types, we have performed experiments considering categorical (CS), ontological (OS) and factual (FS) features by utilizing single-hop (1H) and double-hop (2H) strategies.

- **CS-1H/2H**, 1H includes having the property `dcterms:subject`, while 2H contains features with properties equal to either `dcterms:subject` or `skos:broader`;
- **OS-1H/2H**, 1H considers the features with the property `rdf:type`, while in 2H the properties include `rdf:type`, `rdf-schema:subClassOf` or `owl:equivalentClass`;
- **FS-1H/2H**, 1H uses all the features except ontological and categorical ones, while 2H choose features not included in the previous 2H categories.

Given the million-sized quantity of features obtained, we removed irrelevant features based on the sanity check procedure explained in [5].

Compared CF Recommendation Models: We have conducted experiments considering all the two attacks described in Section 2 against there most widely

If some domain-specific categorical/ontological features are not in the respective lists, we have considered them as factual features.

Table 2: Characteristics of the evaluation dataset: $|\mathcal{U}|$ — number of users, $|\mathcal{I}|$ — number of items, $|\mathcal{R}|$ — number of ratings.

Dataset	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{R}\rvert$	$\frac{\lvert\mathcal{R}\rvert}{\lvert\mathcal{I}\rvert\times\lvert\mathcal{U}\rvert}$	#1H	#2H
`LibraryThing`	4.8K	2.2K	76.4K	0.0070	56.0K	4.2M
`Yahoo!Movies`	4.0K	2.5K	64.0K	0.0063	105.7K	6.6M

used CF models: User-kNN and Item-kNN (*Pearson Correlation*, number or neighbors: 40), and SVD (a matrix factorization model trained by considering 100 latent factors) [3].

Table 3: Experimental results for $SAShA$ at single and double hops. In **bold** we highlight the best results. (imp. %) shows the relative improvement of the best performing feature (CS, OS or FS) with respect to baseline (base) in percentage.

Prediction stability: HR@10														
		dataset	LibraryThing						Yahoo!Movies					
		Rec	User-kNN		Item-kNN		SVD		User-kNN		Item-kNN		SVD	
		feat./hop	1H	2H	1H	2H	1H	2H	1H	2H	1H	2H	1H	2H
1%	**Rnd**	base	.074		.281		.767		.189		.329		.410	
		CS	.068*	.068*	.271*	.270*	.778*	**.799***	.202	**.234***	.336	**.368***	.430*	**.473***
		OS	**.081***	**.075**	**.290***	.252	**.786***	.783*	**.217***	.172	**.345***	.304*	**.446***	.399
		FS	.072	.073	.280	.281	.786*	.787*	.213*	.208*	.338*	.341*	.442*	.440*
		imp. (%)	9.4	1.3	3.2	–	2.4	4.1	14.8	23.8	4.8	11.8	8.7	15.3
	Avg	base	.086		.313		.803		.233		.374		.489	
		CS	.081*	.081*	.301*	.301*	.814*	.815*	.220*	.204*	.357*	.338*	.467*	.408
		OS	**.093***	.084*	.313	.309	**.810**	**.816***	**.237**	**.249***	.371	**.400***	.475	**.539***
		FS	.084*	.084	.305*	.306	.811	.812*	.215*	.227	.350*	.364	.448*	.466*
		imp. (%)	8.1	-2.3	–	-1.2	1.3	1.6	1.7	6.8	-.8	6.9	-2.8	10.2
2.5%	**Rnd**	base	.157		.457		.900		.366		.508		.580	
		CS	.143*	.143*	.441*	.441*	.898	.897	.372	**.410***	.522*	**.564***	.607*	**.667***
		OS	**.170***	.157	**.467***	.455	**.902**	**.901**	**.394***	.337*	**.535***	.482*	**.635***	.560
		FS	.154	.155	.455	.455	.901	**.901**	.381*	.386*	.530*	.531*	.623*	.616*
		imp. (%)	8.2	–	2.1	-.4	.2	.1	7.6	12.0	5.3	11.0	9.4	15.0
	Avg	base	.197		**.508**		.915		**.416**		**.574**		**.685**	
		CS	.187*	.188*	.507	.507	.915	.914	.399*	.384*	.554*	.532*	.652*	.587*
		OS	**.202**	**.198**	.507	.506	.911	.914	.412	**.429***	.563*	**.593***	.656*	**.720***
		FS	.190*	.190*	.504	.503	.911	.913	.397*	.401*	.547*	.557*	.627*	.646*
		imp. (%)	2.5	0.5	-.1	-.1	–	-.1	-.9	3.1	-1.9	3.3	-4.2	5.1
5%	**Rnd**	base	.230		.557		.942		.449		.598		.702	
		CS	.213*	.213*	.558	.558	.940	.940	.455*	**.494***	.609*	**.644***	.707	**.772***
		OS	**.250***	**.231**	**.576***	**.567***	**.944**	.941	**.477***	.428*	**.622***	.577*	**.742***	.652*
		FS	.229	.229	.570*	.567*	.942	.942	.468*	.466*	.619*	.616*	.728*	.717*
		imp. (%)	8.6	0.4	3.4	1.7	.2	-	6.2	10.0	4.0	7.6	5.6	9.9
	Avg	base	.285		.605		.951		.494		.654		.788	
		CS	.269*	.269*	**.621***	**.621***	.950	.949	.479*	.466*	.639*	.621*	.744*	.688*
		OS	**.289**	.281	.610*	.614*	.948	.949	.494	.493	.646*	**.668***	.754*	**.804**
		FS	.272*	.273*	.614*	.614*	.946*	.948*	.473*	.479*	.634*	.642*	.729*	.743*
		imp. (%)	1.4	-1.4	2.6	2.6	-.1	-.2	–	-.2	-1.2	2.1	-4.3	2.0

*We mark statistically significant results ($p < 0.05$) using a paired t-test with the * symbol.*

Evaluation Metrics Let I_T be set of attacks item and U_T the users that have not rated items in I_T. We define the *Overall Hit-Ratio@k* denoted with ($HR@k$) as the average of $hr@k$ for each attacked item according to $HR@k(I_T, U_T) = \frac{\sum_{i \subset I_T} hr@k(i,U_T)}{|I_T|}$ where $hr@k(i, U_T)$ measures the number of occurrences of the attacked item i in the top-k recommendation lists of the users in $|U_T|$ [1].

Evaluation Protocol. For each dataset, we have generated the recommendations concerning all users using the selected CF models (i.e., User-kNN, Item-kNN and MF). After computing baseline attacks, we have performed a series of *SAShA* attacks as described in Section 2 by considering different feature types (i.e., categorical, ontological and factual) extracted at 1 or 2 hops. each attack is a *push attack*. We have performed the attacks considering a different amount of added fake user profiles: $1\%, 2.5\%$ and 5% of the total number of users. We have tested the attacks considering 50 randomly sampled target items [20,10].

4 Results and Discussion

In this section, we present the results of experiments carried out. Table 3 summarizes the results of the HR@10 regarding the considered dimensions. In particular, the inner elements in the Table are related to $\mathcal{KG}$ *semantic dimension* and include: feature types in the rows (CF, OS and FS) number of hops in the columns (1H, 2H). However, the outer elements (dimensions) in Table 3, are the *attack strategies* in the rows ($SAShA$-Rnd and $SAShA$-Avg), and *CF recommendation models* in the columns (User-kNN, Item-kNN and SVD). Finally, we report the results for three levels of attack power (1%, 2.5% and 5%) reported in the three panels of the Table.

The impact of semantics. By looking at the improvement percentages, the general trend is that, for each <attack, CF> pair, one of the values for 1H or 2H is better than baseline attack performance. For instance, for <Rnd, User-kNN> on `LibraryThing`, the relative improvements for 1H and 2H are 9.4% and 1.3%, both obtained for attacks integrated with ontological features setting (OS) features. Relative improvements on `Yahoo!Movies` tend to be larger e.g., consider 14.8% and 23.8% for the same <attack, CF> pair. Regarding the impact of each feature category, in the majority of cases, ontological features setting (OS) provides the best results, followed by categorical features setting (CS). The FS seems to have a little impact on the attack effectiveness. We believe this may be due to a noise introduced by the exploitation of heterogeneous (factual) features. On the other side, ontological features make the similarity between items more evident. Finally, categorical features guarantee competing performance since CF recommendation relies on the affinity between items' categories.

Analysis of the semantic-encoded variant of attacks. The second study is devoted to comparing the impact of semantics on the different attacks (**Rnd**, **Avg**) . The **Rnd** attack provides significant (or substantially high) improvement of attack efficacy on the base version in at least one semantic-configuration.

However, regarding **Avg** attack, it is worth noticing that, even here, the injection of semantics is generally beneficial for the adversary. Nevertheless, the semantic integration, for **Avg** attack, has a lower impact than its use for **Rnd** version. We explain this behavior with the generally high performance of **Avg** attack, that leaves less room for improvements. Additionally, if we consider the different recommendation models, the semantics ensure attack performance improvement. Among all, the semantic-encoded variants of attacks are particularly effective on User-kNN in both datasets.

5 Conclusion

The goal of this work is to investigate the effect of integrating semantic information, obtained from $\mathcal{KG}s$, to foster the shilling attack efficacy. The proposed attack strategy, $SAShA$, extends *random* and *average* attacks by integrating public available semantic information. In detail, $SAShA$ takes advantage of semantics to create more effective fake profiles. Toward this goal, extensive experiments were carried out by considering different collaborative recommendation models, different attack strategies, and three categories of semantic features (categorical, ontological, and factual). Results on two real-world datasets underline the significant vulnerability of standard recommendation models when semantics are integrated into the attack strategy. We plan to investigate different sources of publicly available knowledge (e.g., `Wikidata`), to semantically extend other state-of-the-art attacks. Finally, we are interested in investigating the possibility of semantics knowledge exploitation for defensive strategies.

Acknowledgments. The authors acknowledge partial support of the following projects: Innonetwork CONTACT, Innonetwork APOLLON, ARS01_00821 FLET4.0, Fincons Smart Digital Solutions for the Creative Industry.

References

1. Aggarwal, C.C.: Attack-resistant recommender systems. In: Recommender Systems. Springer (2016)
2. Anelli, V.W., Bellini, V., Di Noia, T., Bruna, W.L., Tomeo, P., Di Sciascio, E.: An analysis on time- and session-aware diversification in recommender systems. In: UMAP. ACM (2017)
3. Anelli, V.W., Deldjoo, Y., Di Noia, T., Di Sciascio, E., Merra, F.A.: Sasha: Semantic-aware shilling attacks on recommender systems exploiting knowledge graphs. In: The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31- June 4, 2020, Proceedings (2020)
4. Anelli, V.W., Di Noia, T.: 2nd workshop on knowledge-aware and conversational recommender systems - kars. In: CIKM. ACM (2019)
5. Anelli, V.W., Noia, T.D., Sciascio, E.D., Ragone, A., Trotta, J.: How to make latent factors interpretable by feeding factorization machines with knowledge graphs. In: The Semantic Web - ISWC 2019 - 18th Int. Semantic Web Conf., Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I. vol. 11778 (2019)

6. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011 (2011)
7. Bhatia, S., Dwivedi, P., Kaur, A.: That's interesting, tell me more! finding descriptive support passages for knowledge graph relationships. In: International Semantic Web Conference (1). Lecture Notes in Computer Science, vol. 11136. Springer (2018)
8. Deldjoo, Y., Constantin, M.G., Eghbal-Zadeh, H., Ionescu, B., Schedl, M., Cremonesi, P.: Audio-visual encoding of multimedia content for enhancing movie recommendations. In: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018 (2018)
9. Deldjoo, Y., Dacrema, M.F., Constantin, M.G., Eghbal-zadeh, H., Cereda, S., Schedl, M., Ionescu, B., Cremonesi, P.: Movie genome: alleviating new item cold start in movie recommendation. User Model. User-Adapt. Interact. **29**(2) (2019)
10. Deldjoo, Y., Di Noia, T., Merra, F.A.: Assessing the impact of a user-item collaborative attack on class of users. In: ImpactRS@RecSys. CEUR Workshop Proceedings, vol. 2462. CEUR-WS.org (2019)
11. Di Noia, T., Magarelli, C., Maurino, A., Palmonari, M., Rula, A.: Using ontology-based data summarization to develop semantics-aware recommender systems. In: ESWC. Lecture Notes in Computer Science, vol. 10843. Springer (2018)
12. Di Noia, T., Mirizzi, R., Ostuni, V.C., Romito, D., Zanker, M.: Linked open data to support content-based recommender systems. In: Proc. of the 8th Int. Conf. on Semantic Systems. ACM (2012)
13. Di Noia, T., Ostuni, V.C., Tomeo, P., Di Sciascio, E.: Sprank: Semantic path-based ranking for top-N recommendations using linked open data. ACM TIST **8**(1) (2016)
14. Gao, L., Yang, H., Wu, J., Zhou, C., Lu, W., Hu, Y.: Recommendation with multi-source heterogeneous information. In: IJCAI. ijcai.org (2018)
15. Gunes, I., Kaleli, C., Bilge, A., Polat, H.: Shilling attacks against recommender systems: a comprehensive survey. Artif. Intell. Rev. **42**(4) (2014)
16. Koren, Y.: Factor in the neighbors: Scalable and accurate collaborative filtering. TKDD **4**(1) (2010)
17. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Computer **42**(8) (2009)
18. Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: WWW. ACM (2004)
19. Linden, G., Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing **7**(1) (2003)
20. Mobasher, B., Burke, R.D., Bhaumik, R., Williams, C.: Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Trans. Internet Techn. **7**(4) (2007)
21. Ning, X., Karypis, G.: Sparse linear methods with side information for top-n recommendations. In: Cunningham, P., Hurley, N.J., Guy, I., Anand, S.S. (eds.) Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012. ACM (2012)
22. Shadbolt, N., O'Hara, K., Berners-Lee, T., Gibbins, N., Glaser, H., Hall, W., m. c. schraefel: Linked open government data: Lessons from data.gov.uk. IEEE Intelligent Systems **27**(3) (2012)
23. Wang, H., Zhang, F., Xie, X., Guo, M.: DKN: deep knowledge-aware network for news recommendation. In: WWW. ACM (2018)