Semantic Web 0 (2021) 1
IOS Press

Semantics-Aware Shilling Attacks against collaborative recommender systems via Knowledge Graphs

Vito Walter Anelli [a] Yashar Deldjoo [a] Tommaso Di Noia [a] Eugenio Di Sciascio [a]
Felice Antonio Merra [a,*],
[a] *Dipartimento di Ingegneria Elettrica e dell'Informaizone, Politecnico di Bari, Italy*
E-mails: vitowalter.anelli@poliba.it, yashar.deldjoo@poliba.it, tommaso.dinoia@poliba.it, eugenio.disciascio@poliba.it, felice.merra@poliba.it

Abstract. Several domains have widely benefited from the adoption of Knowledge graphs ($\mathcal{KG}s$). For recommender systems (RSs), the adoption of $\mathcal{KG}s$ has resulted in accurate, personalized recommendations of items/products according to users' preferences. Among different recommendation techniques, collaborative filtering (CF) is one the most promising approaches to build RSs. Their success is due to the effective exploitation of similarities/correlations encoded in user interaction patterns. Nonetheless, their strength is also their weakness. A malicious agent can add fake user profiles into the platform, altering the genuine similarity values and the corresponding recommendation lists. While the research community has extensively studied $\mathcal{KG}s$ to solve various recommendation problems, sufficient attention was not paid to the possibility of exploiting $\mathcal{KG}s$ to compromise the quality of recommendations. $\mathcal{KG}s$ provide a rich source of information for item representation and recommendation that can dramatically increase the attackers' knowledge about the victim recommendation platform. To this end, this article introduces a new attack strategy, named semantics-aware shilling attack (*SAShA*), that leverages semantic features extracted from a knowledge graph. *SAShA* provides the semantics-aware variant of three state-of-the-art attack strategies: *Random*, *Average*, and *BandWagon*. These improved attacks can exploit graph relatedness measures, i.e., *Katz* and *Exclusivity*-based, computed considering 1-hop and 2-hops of graph exploration. We performed an extensive experimental evaluation with four state-of-the-art recommendation systems and two well-known recommendation datasets to investigate the effectiveness of *SAShA*. Since the semantics of relations has a crucial role in $\mathcal{KG}s$, we have also analyzed the impact of relations' semantics by grouping them in various classes. Experimental results indicate the benefit of embracing $\mathcal{KG}s$ in favor of the attackers' capability in attacking recommendation systems.

Keywords: Recommender Systems, Collaborative Filtering, Security, Semantic Web Technologies, Knowledge Graphs

1. Introduction

The advent of Knowledge Graphs ($\mathcal{KG}s$) has definitely changed the way structured information is stored. Developed to make the Semantic Web a concrete idea, it has become much more than that. The core idea of building a semantic network in which information is represented as directed labeled graphs (RDF graphs) is disarmingly simple. Nevertheless, thanks to the possibilities it paves, it has been welcomed with several promises and expectancies. Complete interoperability, the ability to link knowledge across domains, the possibility to exploit Logical inference and proofs are just a few of them. In numerous domains, the exploitation of the Knowledge Graph information has become the norm. Thanks to the appearance of wide-ranging Linked Datasets like DBpedia and Wikidata, we have witnessed the flourishing of novel techniques in several research fields, like Machine Learning, Information Retrieval, and Recommender Systems. To date, Recommender Systems

*Authors are listed in alphabetical order. Corresponding authors: E-mails: felice.merra@poliba.it, vitowalter.anelli@poliba.it.

(RSs) are considered the focal solution to assist users' decision-making process. Since the volume of the available products on the Web (in which we also consider multimedia content and services) overwhelms the users, RSs support and ease the decisional process. Among them, collaborative filtering (CF) recommendation techniques have shown very high performance in real-world applications (e.g., Amazon [1]). Their rationale is to analyze products experienced by similar users to produce tailored recommendations. Algorithmically speaking, they take advantage of user-user and item-item similarities. Regrettably, malicious users may want to jeopardize the operation of the recommendation platform. For example, they might be a rival company or agents who want to increase (or decrease) the visibility of a particular product. Whatever they are motivated by, the problem is that these similarities are vulnerable to the insertion of fake profiles. This kind of attack is called the *shilling attack* [2], which aims to *push* or *nuke* the probabilities to recommend an item. The malicious agent (or adversary) can rely on an extensive list of techniques to conduct the attack. Researchers and companies have classified them into two broad categories [3]: *low-knowledge* and *informed* attack strategies. In the former attacks, the adversary has poor system-specific knowledge [4, 5]. In the latter, the attacker has an accurate knowledge of the recommendation model and the data distribution [4, 6].

Interestingly, despite the astonishing spread of knowledge graphs, little attention has been paid to knowledge-aware strategies to mine RS's security. In a Web always composed of unstructured information, $\mathcal{KG}s$ are the pillars of the Semantic Web. They have become increasingly important as they can represent data employing a flexible and interoperable semantic graph data structure. Several well-known tools have been built on $\mathcal{KG}s$, like IBM Watson [7], public decision-making systems [8], and advanced machine learning techniques [9–11]. Additionally, the Linked Open Data (LOD) initiative[1] has given birth to a broad ecosystem of linked data datasets known as LOD-cloud[2]. These $\mathcal{KG}s$ provide comprehensive information on numerous knowledge domains. Consequently, if a malicious agent decides to attack one of these domains, items' semantic descriptions would be inestimable.

In the research study at hand, we have investigated the possibility of improving an attack's efficacy by leveraging semantic knowledge. One major contribution of the work is exploiting publicly available information obtained from $\mathcal{KG}$ to generate more influential fake profiles to threaten CF models' performance. The resulting attack strategy is named semantics-aware shilling attack *SAShA*. Beyond the definition of *SAShA* strategy, the work extends state-of-the-art shilling attack approaches such as *Random*, *BandWagon*, and *Average* profiting from semantic knowledge. Remarkably, the attacks' semantics-enhanced variants only rely on publicly available information without supposing any additional knowledge about the system.

The core idea is to reformulate the attacks with the rationale of taking into account the semantic similarity between the target item with the other items in the catalog. The intuition of the approach is that semantic similarity (or, more broadly, semantic relatedness) can safely suffice the lack of the system's knowledge to craft natural and coherent fake profiles. These profiles are indistinguishable from the real ones, and they effortlessly enter the neighborhood of users and items.

In a previous exploratory study, Random, Love-Hate, and Average attacks were modified to consider the cosine vector similarity between the semantic description of items. The limitation of that approach is essentially twofold: it only considers the 1st-hop exploration of the graph (i.e., binarizing the semantic features), and it only considers cosine similarity, which is not particularly suited to bring out semantic relatedness. Here, we have overcome these limitations. On the one hand, we have explored the $\mathcal{KG}$ until the 2-hop, providing a much more in-depth investigation of semantic descriptions' role for this task. Given the required high computational effort, we hope this study provides the interested reader a complete awareness of the potential and the limitations of the approach. On the other hand, we went beyond the famous (but semantics-unaware) cosine similarity, and we have considered *Katz centrality* and *Exclusivity-based relatedness*. Finally, to provide a more fine-grained analysis, we have grouped the semantic relations into three classes: ontological, categorical, and factual relations.

In detail, this study extends the state-of-the-art approach for the integration of semantics in the shilling attacks [12] in numerous directions:

1. two novel graph topological and semantic approaches to build the set of products from which the adversary can craft the fake profiles;
2. an extensive study of the efficacy of the attack considering a two-hops graph exploration, and in-

[1] https://data.europa.eu/euodp/en/linked-data
[2] https://lod-cloud.net/

volving a state-of-the-art deep neural recommendation model;
3. a novel semantic shilling attack strategy based on *BandWagon* strategy;
4. a deeper discussion of the experimental results involving several dimensions: number of explored hops, type of considered relation, recommendation model, amount of injected fake profiles, and dataset;
5. the publication of the full experimental framework and the pre-processed datasets that can be used, out-of-the-box, for further investigations.

Since the study analyzed several aspects, the investigations can be summarized to address the following research questions to provide a general overview:

RQ1 Can relatedness-based measures along with public available semantic information be employed to develop more effective shilling attack strategies against recommendation models?
RQ2 Can we assess which is the most impactful type of semantic information?
RQ3 Is multiple hops exploration of a knowledge graph more effective than single-hop exploration to create coherent fake profiles?
RQ4 What are the recommendation algorithms that suffer more for semantics-aware attacks?

We have carried out extensive experiments (approximately 1440 experiments) to evaluate the impact of proposed attacks against the recommendation models. To this end, we have exploited two real-world recommender systems datasets (`LibraryThing` and `Yahoo!Movies`). Experimental results sharply indicate that $\mathcal{KG}$ information is a valuable source of knowledge that improves attacks' effectiveness. Moreover, the adoption of semantic relatedness measures can unleash the full potential of the semantics-aware attacks.

The remainder of the paper proceeds as follows. In Section 2, we provide an overview of the state-of-the-art of recommendation models and shilling attacks. Section 3 describes the proposed approach (*SAShA*), introduces the semantic relatedness measures, and formalizes the semantic attack strategies. Section 4 focuses on the experimental validation of the proposed attack scenarios. We also provide an in-depth discussion of the experimental results analyzing the several dimensions of the study. Finally, in Section 6, we draw some conclusions and introduce the open challenges.

2. Related Work

In this section, we focus on related literature on the foundations of recommendation models, the integration of Knowledge Graphs ($\mathcal{KG}s$) in RSs, and the security of collaborative filtering models.

2.1. Recommender Systems

Recommender Systems (RS) are the pivotal technical solution in different online systems nowadays to assist users with many over-choice challenges by filtering out important information out of a large amount, according to user's tastes and preferences. From a technical point of view, a recommendation problem can be stated as finding a utility function to automatically predict how much users will like unknown items.

Definition 1 (Recommendation Problem)**.** *Let $\mathcal{U}$ and $\mathcal{I}$ denote a set of users and items in a system, respectively. Each user $u \in \mathcal{U}$ is related to $\mathcal{I}_u^+$, the set of items she has consumed, or her user profile. Given a utility function $g : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$ a* ***Recommendation Problem*** *is defined as*

$$\forall u \in \mathcal{U},\ i'_u = \operatorname*{argmax}_{i \in \mathcal{I}} g(u, i)$$

where i'_u denotes an item not consumed by the user u before. We assume that preference of user $u \in \mathcal{U}$ on item $i \in \mathcal{I}$ is encoded with a continuous-valued preference score $r_{ui} \in \mathcal{R}$, where $\mathcal{R}$ represent the set of (u, i) pairs for which r_{ui} is known

The major class of recommendation models include content-based filtering (CF), collaborative filtering (CBF), and hybrid thereof [13, 14]. CBF models build a profile of user interests based on the content features of the items preferred by that user (liked or consumed), characterizing the nature of her interests. The item features can include a full range of available information including editorial metadata (genre, emotion, instrumentation) and user-generated content (tags, labels) [15], features extracted from the audio and visual signals directly [16], and semantic information collected from a knowledge graph [17].

On the other hand, CF models compute recommendations based on similarities in interaction/preference patterns of like-minded users. Collaborative recommenders are mainstream academic and industrial research due to their state-of-the-art performance, achieved when a sufficient amount of preference data, either explicit, e.g., ratings, or implicit, e.g., previous clicks and check-ins, are available. Different CF models developed today can be classified according to memory-based and model-based. Memory-based models compute recommendations exclusively based on correlations in interactions across users (user-based CF [18, 19]) or items (item-based CF [19, 20]), while model-based approaches compute a model — typically a machine learning model — that can be queried in the production phase to generate recommendations for a given user profile. A famous example of model-based CF methods is the matrix factorization (MF) method that learns a latent representation of items and users, aka a latent factor model (LFM), whose linear interaction can explain an observed feedback [21]. There are several MF variations proposed in the literature, such as PMF and BNMF. These methods essentially encode the complex relations between users and items into a small number of shared hidden factors, where their dot product drives the predictions. A major drawback of MF approaches, however, lies in their linearity. To address this concern, a recently popularized trend in the community of recommender systems (RS) is using deep neural architectures with deep neural networks (DNNs) that are capable of modeling the nonlinearity in data through nonlinear activation functions. The power of DNN is exploited in modern RS to capture complex interaction patterns between users and items and ultimately to better judge users' preferences.

2.2. *Knowledge-aware Recommender Systems (KaRSs)*

All of us have witnessed the astonishing performance of recommendation systems. However, few know that, often, the recommendation algorithms struggle to optimize the model. Despite the number of transactions being massive, the number of per-user interactions is usually very scarce. Over the years, the recommendation system designers relied on additional sources of information to overcome this limitation. Nowadays, modern RSs exploit various side information such as metadata (e.g., tags, reviews) [22], social connections [23], image and audio signal features [24], and users-items contextual data [25] to build more in-domain [17] (i.e., domain-dependent), cross-domain [26], or context-aware [27, 28] recommendation models. Among the diverse information sources, what is, likely, the most relevant source is Knowledge Graphs ($\mathcal{KG}s$). A $\mathcal{KG}$ is a heterogeneous network that encodes multiple relationships, edges, nodes, and links items at high-level relationships, making them a strong item representation technique. Thanks to the heterogeneous domains that $\mathcal{KG}s$ cover, the design of knowledge-based recommendation systems has arisen as a specific research field of its own in the community of RSs, usually referred to by Knowledge-aware Recommender Systems (KaRS [11, 29]). This research community combines the most advanced machine learning techniques with state-of-the-art knowledge representation paradigms. This blending of skills and ideas has generated several advancements in the recommendation [30], knowledge completion [31], preference elicitation [32], user modeling [33] research, and thus produced a vast literature. A comprehensive review of the field would require a separate and specific paper; however, we can still provide an overview of the most advanced (or particularly representative) contributions. To help the reader orient herself in the literature, we follow three distinct lines: impacted research fields, recommendation techniques, and data sources. In recent years, the Knowledge-aware Recommender Systems have been particularly impactful for several research domains:

- **$\mathcal{KG}$/Graph-embeddings** [34–40], where the latent representation of semantic knowledge enables novel and diverse applications;
- **Hybrid Collaborative/Content-based recommendation** [30, 35], exploiting the $\mathcal{KG}$ information to suffice the lack of collaborative information and to improve the performance;
- **Knowledge-completion, link-prediction, knowledge-discovery** [31, 40–46], where the topology of the knowledge graph and the graph embeddings helped to improve the overall quality of the knowledge base;
- **Knowledge-transfer, cross-domain recommendation** [26, 47, 48], where the $\mathcal{KG}s$ allow to find semantic similarities between different domains;
- **Interpretable/Explainable-recommendation** [30, 49–52], with $\mathcal{KG}$ being a backbone for understanding the recommendation model and providing human-like explanations
- **User-modeling** [33, 53–55], since the resource descriptions can drive the construction of the user profile;
- **Graph-based recommendation** [56–61], where the topology-based techniques have met the semantics of the edges/relations, and the ontological classification of nodes (classes);

- **The cold-start problem** [26, 62–64], since the $\mathcal{KG}s$ can overcome the lack of collaborative information;
- **The content-based recommendation** [65, 66] that solely relies on $\mathcal{KG}$ and still produces high-quality recommendations.

All the former advances have been shown to enhance the recommendation quality or the overall user experience. Although the algorithms differ on many levels, we can still classify recommendation techniques into two broad approaches:

- **Path-based** methods [56–58, 61, 67, 68], which employ paths and meta-paths to estimate the user-item similarities or the nearest items;
- **KG embedding-based** techniques [28, 30, 36, 56, 69, 70], which leverage $\mathcal{KG}$ embeddings (usually obtained through matrix factorization or neural network encoding) for items' representation.

Finally, we focus on the Knowledge Graphs data sources. The availability of a myriad of $\mathcal{KG}s$ is a definite advantage of Knowledge-aware Recommender Systems. Thanks to the Linked Data initiative, today, we can benefit from $1,483$ different $\mathcal{KG}s$ connected in the so-called Linked Open Data Cloud[3]. $\mathcal{KG}s$ can be general-purpose, or domain-specific like Academia/Industry DynAmics (AIDA) [71]. However, most of the contributions concentrate on a shortlist of $\mathcal{KG}s$ with a peculiar characteristic: being an encyclopedic $\mathcal{KG}$. Those $\mathcal{KG}s$ share the same ontology and the same schema across multiple domains, giving access to a wide-spread knowledge at the same development cost required for a single domain. The most appreciated $\mathcal{KG}s$ of this special class undoubtedly are DBpedia [72, 73], Wikidata [74, 75], Yago [76] (the 4th release [77] also supports RDF* [78]), FreeBase [79], Satori[45] [80, 81], NELL [82], Google's Knowledge Graph[6], Facebook's Entities Graph[7], Knowledge Vault [83], Bio2RDF [84].

2.3. Security of Recommender System

Collaborative filtering recommender systems are commonly employed on online platforms, e.g., Amazon[8], eBay[9], Netflix[10]. The rationale is to ease the customer navigation across the catalog based on the so-called "word-of-mouth", i.e., a user might like what other people like and dislike. However, the openness of these systems has shown to be a possible point of failure. Indeed, malicious users, the *adversaries*, can meticulously craft fake profiles to poison the data and alter the recommendation behavior toward malicious goals [85–87]. An adversary may execute a **shilling attack** (injects malicious profiles) to achieve a whole different set of objectives. To name a few, she may want to demote competitor products [4], misuse the underlying recommendation system [2], or increase the recommendability of specific products [88, 89].

A standard categorization of shilling attacks considers the adversary's knowledge to mount the attack, the adversary's goal, and the number of added profiles [3, 90]. According to the adversary's knowledge, a shilling attack can be a *low-knowledge* or an *informed* attack. The former class indicates a limited amount of available data information accessible by the adversary [4, 5]. The latter class assumes a higher knowledge of dataset information, such as the rating distribution. In this case, the adversary might be able to craft more effective profiles [4, 85]. Regarding the adversary's goal, the adversary might alter the recommender to *push* or *nuke* the recommendability of a product, or a class of products, named *target items*. Push attacks aim to increase the targeted item's appeal, while nuke attacks aim to lower their recommendation frequency. Also, shilling attacks can be categorized based on the number of fake profiles added to the system. A common approach to measuring the granularity of attack is to measure the percentage of added profile over the total number of regular users in the systems [5, 91].

The research works on shilling attacks explored two main research perspectives: proposing and investigating attack strategies with their effects on the recommendation performance [4, 91–93] and exploring defensive mechanisms [87, 94–98].

A typical characteristic of the first line of research on shilling attacks is that the adversary's knowledge is related only to the recommender system's user-item interaction matrix. Furthermore, Anelli *et al.* [12] demonstrate that publicly available $\mathcal{KG}$ improves adversary's efficacy, also in the case of *low-informed* attacks. In this work, we extend the *SAShA* framework

[3]https://lod-cloud.net/datasets
[4]https://searchengineland.com/library/bing/bing-satori
[5]https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing
[6]https://blog.google/products/search/introducing-knowledge-graph-things-not/
[7]https://www.facebook.com/notes/facebookengineering/under-the-hood-the-entitiesgraph/10151490531588920/
[8]https://www.amazon.com/
[9]https://www.ebay.com/
[10]https://www.netflix.com/

to verify the possible improvement of the adversary's efficacy when processing the $\mathcal{KG}$ information with semantic similarity measures.

Note that this work focuses on shilling attacks, which are hand-engineered strategies to study recommender systems' security. This research line is different from machine-learned data poisoning attack [99–103] and adversarial machine-learned attacks [89, 104–106], recently surveyed by Deldjoo *et al.* [107]. Indeed, those attacks study the security of recommendation systems when adversaries adopt optimization techniques to create a minimal perturbation able to fail the recommendation performance.

3. Proposed shilling attack approach

This section introduces the reader to the notations and formalisms that may help understand the design of shilling attacks against targeted items integrating information obtained from a knowledge graph ($\mathcal{KG}$). First, we focus on categorizing the predicates in a $\mathcal{KG}$ and formalizing the semantic features extraction considering a single- and double- hop exploration of the $\mathcal{KG}$ (Section 3.1). Hence, the adopted relatedness measures are summarized (Section 3.2). Then, we present an overview of shilling attack notation (Section 3.3), and, finally, semantics-aware extensions to various widespread shilling attacks, namely: *Random*, *Average*, and *BandWagon* attacks in Section 3.3.1.

3.1. Knowledge Graph Content Extraction

A knowledge graph is a structured repository of knowledge, designed in the form of a graph, that encodes various kinds of information:

- **Factual.** General statements as *Rika Dialina was born in Crete* or *Heraklion is the capital of Crete* that describe an entity by using a controlled vocabulary of predicates that connect the entity to other entities (or literal values);
- **Categorical.** These statements connect the entity to a particular category (i.e., the categories associated with a Wikipedia page). Often, categories are in turn organized as a hierarchy;
- **Ontological.** These are formal statements that describe the entity's nature and its ontological membership to a specific class. Classes are often organized in a hierarchical structure. In contrast to categories, sub-classes and super-classes are connected through `IS-A` relations.

In a knowledge graph, we can express statements through triplets $\sigma \xrightarrow{\rho} \omega$, with a *subject* ($\sigma$), a *predicate (or relation)* (ρ), and an *object* (ω). There are several ways to transform the knowledge coming from a knowledge graph into a feature. We have chosen to represent each distinct path as an explicit feature [30]. In the next section, it will be clear why it is convenient. Given a set of items $I = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_N\}$ in a collection and the corresponding triples $\langle i, \rho, \omega \rangle$ in a knowledge graph, the set of 1-hop features is defined as 1-*HOP-F* $= \{\langle \rho, \omega \rangle \mid \langle i, \rho, \omega \rangle \in \mathcal{KG}$ with $i \in I\}$.

In an analogous way we can identify 2nd-hop features. By continuing the exploration of $\mathcal{KG}$ we retrieve the triples $\omega \xrightarrow{\rho'} \omega'$, where ω is the *object* of a 1st-hop triple and the *subject* of the next triple. The double-hop *predicate* is denoted by ρ' and the *object* is referred as (ω'). Therefore, the overall feature set is defined as 2-*HOP-F* $= \{\langle \rho, \omega, \rho', \omega' \rangle \mid \langle i, \rho, \omega, \rho', \omega' \rangle \in \mathcal{KG}$ with $i \in I\}$. Given the current definition, 2nd-hop features also contain heterogeneous predicates (see the previous classification of different kinds of statements). To make it possible to analyze the impact of the kind of semantic information, we consider a 2nd-hop feature as Factual *if and only if* both relations (ρ, and ρ') are Factual. The same holds for the other types of encoded information.

3.2. Entity Similarity/Relatedness in KGs

The keystone of the Knowledge Graph representation is the semantics enclosed in the resource description and the predicates that connect the different resources. Nevertheless, if the metric to compute similarities between the resources is not carefully chosen, this piece of information is lost irretrievably. Motivated by this awareness, we decided to consider a broad spectrum of diverse similarity/relatedness metrics: **Cosine Vector Similarity** [108], ***Katz* centrality** [109], and ***Exclusivity*-based semantic relatedness** [110]. The three metrics cover three different aspects of the similarity between the resources:

1. A signal of the overlap of the descriptions
2. The average length of the paths that connect the resources
3. A semantics-aware signal that highlights the specificity of the relations between the resources

Cosine Vector Similarity is a well-known similarity that is very popular in recommendation systems. The idea is to measure how similar the two different representations are. Suppose a numerical vector can represent the resource description, with the number of the predicate-object chains observed in the $\mathcal{KG}$ being the vector's cardinality. Mathematically, it measures the cosine of the angle between two vectors that represent two different resources. The smaller the angle, the higher is the cosine, and thus the similarity. Suppose i and j are two items in the $\mathcal{KG}$, and $F(\cdot)$ is a function that returns the features associated with an entity in the $\mathcal{KG}$. Hence $in(i, f)$ is a function that returns 1 if entity i is associated with feature f, else 0. The Cosine Vector Similarity has been already formulated for $\mathcal{KG}$ as follows [108]:

$$sim(i, j) = \frac{\sum_{f \in F(i) \cup F(j)} in(i,f) \cdot in(j,f)}{\sqrt{\sum_{f \in F(i)} in(i,f)^2} \cdot \sqrt{\sum_{f \in F(j)} in(j,f)^2}} \tag{1}$$

Katz centrality [109] is a famous graph-centrality measure that inspired several semantics-aware metrics [110, 111]. Katz suggests that the probability of the path between two nodes can indicate the effectiveness of the link. Given a constant probability for a single-hop path, called α, the whole path's overall probability is α^y, where y is the number of the nodes involved. Hulpus [110] exploits the rationale to build a relatedness measure. Therefore, he defined the Katz relatedness between two items i and j as the accumulated score over the top-t shortest paths between them.

$$rel^{(t)}_{Katz}(i, j) = \frac{\sum_{p \in SP^{(t)}_{ij}} \alpha^{length(p)}}{t} \tag{2}$$

where $SP^{(t)}_{ij}$ is the set of the top-t shortest paths between items i and j.

Exclusivity-based semantic relatedness [110] is a semantic relatedness measure that takes into account the type of relations that connect two nodes. The idea is that two concepts are strongly connected if the type of relations between them is different from the type of relations they have with other concepts. This property of relations, named exclusivity, is defined as follows.

Suppose a predicate ρ of type τ between two items i and j, directed from i to j. The exclusivity of predicate ρ is the probability to select, with a uniform random distribution, a predicate ρ' of type τ among the predicates of type τ that exit resource i and enter node j, such that predicate ρ' is exactly the predicate ρ:

$$exclusivity(i \xrightarrow{\tau} j) = \frac{1}{|i \xrightarrow{\tau} *| + |* \xrightarrow{\tau} j| - 1} \tag{3}$$

where $|i \xrightarrow{\tau} *|$ denotes the cardinality of relations of type $\tau \in \mathcal{T}$ that exit resource i, and $|* \xrightarrow{\tau} j|$ denotes the number of relations of type $\tau \in \mathcal{T}$ that enter resource j. Since the relation $i \xrightarrow{\tau} j$ is in $|i \xrightarrow{\tau} *|$ and in $|* \xrightarrow{\tau} j|$, 1 is subtracted from the denominator. The exclusivity score for a predicate falls inside the $(0, 1]$ interval. The value 1 denotes the extreme case in which the predicate is the only relation of its type for both i and j.

Given a path through $\mathcal{KG}$, $\mathcal{P} = n_1 \xrightarrow{\tau} n_2 \xrightarrow{\tau_2}, \ldots, n_k$ with $\tau_i \in \mathcal{T}^{\mp}$, the weight of the path is defined as:

$$weight(\mathcal{P}) = \frac{1}{\sum_i \frac{1}{exclusivity(n_i \xrightarrow{\tau_i} n_{i+1})}} \tag{4}$$

Finally, the relatedness between two resources can be computed as the sum of the path weights of the top-t paths between the resources with the highest weights. To penalize longer paths, a constant length decay factor, $\alpha \in (0, 1]$, can be introduced. The overall exclusivity-based relatedness measure is therefore defined as follows:

$$rel^{(t)}_{Excl}(i, j) = \sum_{\mathcal{P}_n \in P^t_{ij}} \alpha^{lenght(\mathcal{P}_n)} weight(\mathcal{P}_n) \tag{5}$$

3.3. Strategies for Attacking a Recommender System

In order to increase the robustness of recommender systems, or generally ML systems, against any potential attack, the system designer needs to understand the following fundamental questions:

- ***Why*** have the attacks been performed?
- ***When*** have the attacks been performed?
- ***How*** have the attacks been realized?
- ***How much*** knowledge does the attacker have?

The ***Why*** question seeks to understand the *intent* of the attacker. There are two most common motivations behind shilling attacks against RSs. The first one is to promote (**push**) or demote (**nuke**) the popularity of target items, or groups of items, so that they can be recommended to as many or as few users as possible in order to gain an economic advantage over platform competitors. The second one intends to compromise the overall quality of the recommendations. These two dimensions will impact the definition of evaluation metrics used to evaluate the success of the attacks.

The ***When*** question concerns the attack's *timing*, a consideration that gives rise to a dichotomy that is central to understand attacks on ML systems: *train-time*

Table 1

Overview of shilling attack strategies and their profile composition for adversaries' goal of *pushing* a target item ($\mathcal{I}_T$).

	Selected Items ($\mathcal{I}_S$)		**Filler Items** ($\mathcal{I}_F$)			$\mathcal{I}_\phi$	$\mathcal{I}_T$
Attack Type	**Number Items**	**Rating**	**Selection**	**Number Items**	**Rating**		
Random [4]	$\emptyset$		Random	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	$rnd(N(\mu,\sigma^2))$	$\mathcal{I}-\mathcal{I}_F$	*max*
Love-Hate [112]	$\emptyset$		Random	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	*min*	$\mathcal{I}-\mathcal{I}_F$	*max*
Popular [113]	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	*min* **if** $\mu_f<\mu$ **else** *min* $+1$		$\emptyset$		$\mathcal{I}-\mathcal{I}_S$	*max*
Average [4]	$\emptyset$		Random	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	$rnd(N(\mu_f,\sigma_f^2))$	$\mathcal{I}-\mathcal{I}_F$	*max*
Bandwagon [92]	$(\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert})/2-1$	*max*	Random	$(\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert})/2$	$rnd(N(\mu,\sigma^2))$	$\mathcal{I}-\mathcal{I}_S-\mathcal{I}_F$	*max*
P. Knowledge [85]	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	*max*		$\emptyset$		$\mathcal{I}-\mathcal{I}_S$	*max*
***SAShA* Random**	$\emptyset$		Semantics-aware	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	$rnd(N(\mu,\sigma^2))$	$\mathcal{I}-\mathcal{I}_F$	*max*
***SAShA* Love-Hate**	$\emptyset$		Semantics-aware	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	*min*	$\mathcal{I}-\mathcal{I}_F$	*max*
***SAShA* Average**	$\emptyset$		Semantics-aware	$\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert}-1$	$rnd(N(\mu_f,\sigma_f^2))$	$\mathcal{I}-\mathcal{I}_F$	*max*
***SAShA* Bandwagon**	$(\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert})/2-1$	*max*	Semantics-aware	$(\frac{\sum_{u\in\mathcal{U}}\lvert\mathcal{I}_u\rvert}{\lvert\mathcal{U}\rvert})/2$	$rnd(N(\mu,\sigma^2))$	$\mathcal{I}-\mathcal{I}_S-\mathcal{I}_F$	*max*

where (μ, σ) are the dataset average rating and rating variance, (μ_f, σ_f) are the filler item $\mathcal{I}_F$ rating average and variance, and *min* and *max* are the minimum and maximum rating value. *rnd* function generates one integer (i.e., rating) from a discrete uniform distribution.

attacks (aka data poisoning attacks) and *decision-time attacks* (aka evasive attacks). Train-time attacks are accomplished by modifying the training data used to train the ML model. In RS, the most popular types of poisoning attacks designed to date include **shilling attacks**, and **machine-learned** data poisoning attacks. Shilling attacks are realized by injecting hand-crafted fake user profiles (shilling profile) into the user-rating matrix (URM), aiming to learn a bad recommendation model from the user-item rating scores. In contrast to hand-engineered shilling attacks, machine-learned data poisoning attacks typically use an optimization procedure to maximize the adversary's goal *automatically*. This class of data poisoning attacks was popularized in RS research by Li *et al.* [99], that introduced attacks against latent factor recommendation models (LFM), paving the path for the introduction of a variety of other attacks against in the upcoming years, broadly classifiable into attacks against LFM [99, 100, 114, 115], reinforcement learning (RL) [116–118], and other categories of recommendation such as graph-based techniques [119–121]. We point the reader to a few recent surveys for a broader frame of reference into these techniques: [90] for a review of shilling attacks against RS, [107] for a good understanding of adversarial machine learning applications in RSs, and [122] for a general introduction to adversarial attacks and defenses against ML systems.

The ***How*** question, we discuss it for shilling attacks, which was the choice in this work due to the simplicity of designing such attacks. For a detailed discussion about the design of other attacks (machine-learning data position and AML-based attacks), we refer interested readers to [107]. A shilling attack is typically conducted against a rating-based CF model based on generation fake user profiles (shilling profile) that follow a specific pattern, as designed by [4, 88].

Definition 2 (Shilling Profile)**.** *Given a Recommendation Problem, a* ***Shilling Profile*** *($\mathcal{SP}$) is a rating profile partitioned into four sets, according to:*

$$\mathcal{SP} = \mathcal{I}_S + \mathcal{I}_F + \mathcal{I}_\phi + \mathcal{I}_T \tag{6}$$

where $\mathcal{I}_S$ denotes the selected item set containing items identified by the attacker to maximize the effectiveness of the attack, $\mathcal{I}_F$ is the filler item set, containing a set of randomly selected items to which rating scores are assigned to make them imperceptible. $\mathcal{I}_T$ is the target item, for which the recommendation model will make a prediction, aimed to be maximal (for push attack) or minimum (for nuke attack). Finally, $\mathcal{I}_\phi$ is the unrated item set, holding a number of items without any ratings.

Note that $\mathcal{I}_S$ and $\mathcal{I}_F$ are chosen depending on the attack strategy, and the attack size is the number of injected fake user profiles. Throughout this paper, we use $\phi = |\mathcal{I}_F|$ to represent the filler size, $\alpha = |\mathcal{I}_S|$ the selected item set size and $\chi = |\mathcal{I}_\emptyset|$ to show the size of unrated items. Table 1 summarizes the main parameters involved in the implementation of most prominent shilling attacks against rating-based CF models. For instance, it can be seen the proposed semantic attacks, referred to by $SAShA$ name of the attack, are the extension of state-of-the-art shilling attacks, with the difference that selection of the filler item set ($\mathcal{I}_F$) is chosen semantically, not randomly. We will describe details about semantic knowledge integration with shilling attacks in Section 3.3.1.

Finally, the last important consideration when designing attacks is how much — information the adversary has about the learning model, the algorithm, or the training data they aim to attack. This will lead to classifies attacks according to *white-box*, *black-box*, and *gray-box* attacks.

1. **White-box attacks** also referred to by perfect-knowledge (PK) attacks, are attacks in which we assume the adversary has perfect knowledge about the learned model (the actual recommendation model), including its characteristics, the learning algorithm, hyper-parameters, among others. White-box attacks are important since they are the most potent possible threat model. In the field of cybersecurity, it has been shown that assuming attacker having no knowledge — or security by obscurity — is ineffective [123].
2. **Gray-box attacks** assume that the adversary has some knowledge about the model in gray-box attacks —aka limited-knowledge attacks (LK)— although this knowledge might not be complete. For example, the attacker may know about the recommendation model or the training data, but not both of them together. For instance, attackers can build a surrogate model using their knowledge of the training data and effectively craft attacks against the substitute model [124].
3. **Black-box attacks**, also known as zero-knowledge attacks (ZK), consider adversaries without knowledge about the learned model or the algorithm used by the ML model before developing the attack.

To connect it with state-of-the-art shilling attacks, we can mention that the Random attack is a black-box attack, the Perfect-knowledge attack is a while-box attack, while the rest of the attacks can be considered as a gray-box attack.

3.3.1. Semantics-aware Shilling Attack Strategies

Previous works on shilling attacks against RS models have predominately focused on CF models and the way the user interaction data (ratings) can be exploited to craft more effective shilling profiles. In our view, a rich source of knowledge, namely $\mathcal{KG}s$, has been neglected in the design of such attacks. To fill this gap, in this work, we strengthen state-of-the-art attack strategies by exploiting semantic similarities between items. The main idea behind our proposed semantics-aware shilling attack (SAShA) strategies is that we can compute the similarity/relatedness between the target $\mathcal{I}_t$ with other items in the catalog by exploiting the features extracted from a $\mathcal{KG}$. This semantic information is used to construct the filler set $\mathcal{I}_F$, by semantically selecting the items. The key insight in the proposed approach is that the exploitation of semantic similarities/relatedness leads to the generation of more natural and coherent fake profiles, given that the representative description of items is encoded in computing pairwise item similarities.

Semantics-aware Random Attack is an extension of the baseline Random Attack [4]. The baseline version is naive attack, which uses randomly chosen items ($\alpha = 0, \phi = \textit{profile-size}$) to create a fake user profile. The ratings attributed to $\mathcal{I}_\phi$ are sampled from a uniform distribution (see Table 1). We modify this attack by selecting the items to complete $\mathcal{I}_F$ with the proposal semantics-aware technique. For this purpose, we compute semantic similarities/relatedness between the items in the catalog e the target item using $\mathcal{KG}$-based features (cf. Section 3.1). Afterward, we identify the most similar items ($\mathcal{I}_T$) by considering the first quartile of most similar items, and we extract ϕ items from this set by adopting a uniform distribution.

Semantics-aware Average Attack is an informed attack strategy that extends the AverageBots attack [5]. The baseline attack leverages the mean and variance of the ratings, which is then used to sample each filer item's rating from a normal distribution built using these values. Similar to the previous semantics-aware attack extension, we extract the filler items by exploiting semantic similarities derived from a $\mathcal{KG}$. Finally, as before, we consider the items in the first quartile of the most semantically similar/related to $\mathcal{I}_T$ as the candidate filler items ($\mathcal{I}_F$).

Semantics-aware BandWagon Attack is a low-knowledge attack that extends the standard BandWagon attack [92]. We leave unchanged the injection of the selected items ($\mathcal{I}_S$), which are the most popular ones and on which we associate the maximum possible rating (see Table 1). However, similarly to the previous two semantic attack extensions, we complete $\mathcal{I}_F$ by taking into account the semantic similarity/relatedness between the target item $\mathcal{I}_T$ and the rest of the catalog.

Note that in this work, we do not investigate the semantics-aware extension of the Love-Hate attacks since the integration of the semantic information has been demonstrated to not improve the adversary efficacy as discussed in related studies [12, 125].

4. Experimental Setting

In this section, we describe the the experimental evaluation and provide details necessary to reproduce the experiments. First, we introduce the two real-world datasets used in recommendation scenarios (Section 4.1), as well the process carried out to extract, select and filter the semantic information obtained from the $\mathcal{KG}$ (Section 4.1.1 to 4.1.3). Afterward, we describes the four collaborative filtering (CF) recommendation models tested against the proposed attacks (Section 4.2). Finally, we detail the evaluation metrics and the experimental setting used for the experimental evaluation (Section 4.3 and 4.4).

4.1. Dataset

We test the proposed shilling attack approach on two recommendation datasets: `LibraryThing` and `Yahoo!Movies`.

`LibraryThing` [61] is a popular dataset whose interactions originate from `librarything.com`, a social cataloging web application. The dataset contains user-item rating scores ranging from a minimum of 1 to a maximum of 10. As presented in [12], we use a reduced version by randomly extracting the 25% of products in the catalog. Furthermore, we apply a 5-core filtering by removing all the users with less than five interactions to focus the study on active users. These users are of adversaries' interest since they could more likely buy the pushed products.

`Yahoo!Movies` is a recommendation dataset released by `research.yahoo.com` with ratings collected up to November 2003. The dataset also provides mappings to the `MovieLens` and `EachMovie` catalogs. The recorded interactions consist of ratings ranging from 1 to 5.

Another motivation for choosing these datasets was the existence of a mapping between the products in the catalogs and `DBpedia` knowledge-base entities. In particular, we use the mappings publicly available at https://github.com/sisinflab/LinkedDatasets. Table 2 reports the statistics of both datasets' user-item interaction data, together with the total number of semantic features extracted from both the first and the second hop of the knowledge graph associated with each item. In the following, we describe steps taken for preprocessing and data sanity of the features extracted from a $\mathcal{KG}$.

Table 2

Datasets statistics.

Dataset	#Users	#Items	#Ratings	Sparsity	#F-1Hop	#F-2Hops
`LibraryThing`	4,816	2,256	76,421	99.30%	56,019	4,259,728
`Yahoo!Movies`	4,000	2,526	64,079	99.37%	105,733	6,697,986

4.1.1. Feature Extraction.

Once the items are semantically reconciled with `DBpedia` entities, we remove the noisy features whose triples contain one of the following predicates:

- `owl:sameAs`
- `dbo:thumbnail`
- `foaf:depiction`
- `prov:wasDerivedFrom`
- `foaf:isPrimaryTopicOf`

The feature denoising procedure follows the methodology proposed by Anelli *et al.* [30, 50].

4.1.2. Feature Selection.

To perform the analysis of the class (or type) of semantic features, we implement our proposed semantics-aware attacks by considering three different types of features, i.e., categorical (CS), ontological (OS), and factual (FS), a feature taxonomy commonly adopted in the Semantic Web community [30].

For the semantics-aware attack strategies exploiting single-hop (1H) features, we apply the following policies:

- **Categorical-1H**, we use the features with the property `dcterms:subject`;
- **Ontological-1H**, we select the features containing the property `rdf:type`;
- **Factual-1H**, we consider all the features except ontological and categorical features.

In the attacks employing double-hop (2H) features, the strategies evolve as described below:

- **Categorical-2H**, we pick up the features with either `dcterms:subject` or `skos:broader` properties;
- **Ontological-2H**, we select the features containing either `rdf-schema:subClassOf` or `owl:equivalentClass` properties;
- **Factual-2H**, we use the features not selected in the previous two classes.

Note that we did not place any domain-specific categorical/ontological feature in the respective lists. To provide a domain-agnostic evaluation, we have treated them as factual features.

Table 3
Selected features in the different settings either for single and double hops.

	Single hop features						Double hop features					
	Categorical		Ontological		Factual		Categorical		Ontological		Factual	
Dataset	Total	Selected	Total	Selected	Total	Selected	Total	Selected	Total	Selected	Total	Selected
`LibraryThing`	3,890	373	2,090	311	50,039	1,972	9,641	857	3,723	527	4,246,365	252,848
`Yahoo!Movies`	5,555	1,192	3,036	722	97,142	7,690	8,960	1,956	3,105	431	6,685,921	517,211

4.1.3. *Feature Filtering.*

This work aims to study the attack performance differences up to the first and second hop. Addressing this aim, we obtain millions of features for both `LibraryThing` and `Yahoo!Movies` as reported in the last two columns of Table 2. Measuring semantic similarities across the item catalog would quickly become unfeasible. However, some features only occur once and provide no useful informative or collaborative information. Therefore, we decided to drop off irrelevant features following the filtering technique proposed in Di Noia *et al.* [61, 126]. In detail, we removed all the features with more than 99.74% of missing values and distinct values. Table 3 shows the remaining features' statistics after applying all the extraction, selection, and filtering process.

4.2. *Recommender Models*

In this work, we test our attack proposal (see Section 3.3) against four baseline collaborative recommendation systems: User-*k*NN, Item-*k*NN, Matrix Factorization, and Neural Matrix Factorization. The first two approaches belong to memory-based CF, while the next two are model-based CF (see Section 2.1), thus providing us an overall picture of different recommendation model types performance when confronted with shilling attacks.

- **User-*k*NN** [18, 19] is a standard user-based Collaborative Filtering (CF) approach to measure the preference score of a user u toward an not interacted product i ($\hat{r}_{ui}$), by exploiting the similarity with the k most similar users in her neighborhood. We adopt the user and item's unbiased User-*k*NN formulation as proposed by Koren *et al.* [19]. Let $u \in \mathcal{U}$, and $i \in \mathcal{I}$, where $\mathcal{U}$ and $\mathcal{I}$ are the set of users, and items, in the recommendation system; the prediction of the rating attributed by the user u to the item i is estimated as follows:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{v \in \mathcal{U}_i^k(u)} \delta(u, v) \cdot (r_{vi} - b_{vi})}{\sum_{v \in \mathcal{U}_i^k(u)} \delta(u, v)} \quad (7)$$

where δ is the distance function to measure the users' similarities, and $\mathcal{U}_i^k(u)$ is the group of the k-most similar users v of u (aka, the neighborhood). Furthermore, b_{ui} is defined as $\mu + b_u + b_i$, where μ, b_u, and b_i are the overall average rating, the observed bias of user u and item i, respectively. We use the *Pearson Correlation* as the distance metric $\delta(\cdot)$ as suggested by Candillier *et al.* [127]. The size of the neighborhood, k, is set to 40.

- **Item-*k*NN** [19, 20] is a standard item-based CF to predict the user-item preference score ($\hat{r}_{ui}$) from the recorded feedback. Let $u \in \mathcal{U}$, and $i \in \mathcal{I}$, the prediction of the score given by the user u to item i is predicted as follows:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in \mathcal{I}_u^k(i)} \delta(i, j) \cdot (r_{uj} - b_{uj})}{\sum_{j \in \mathcal{I}_u^k(i)} \delta(i, j)} \quad (8)$$

where $\mathcal{I}_u^k(i)$ denotes the set of k most similar items to (unrated) item i voted by user u. Similar to User-*k*NN, we use the *Pearson Correlation* to implement the distance function $\delta(\cdot)$ and set k the dimension of the considered neighborhood 40.

The third and fourth recommendation systems are representative of **model-based** collaborative recommenders. In particular, matrix factorization is the baseline recommender representing the class of linear latent factor models, while neural matrix factorization represents the class of non-linear models.

- **Matrix Factorization (MF)** [21] is a latent factor model to learn the unknown preferences. MF represents both items and users by vectors of latent factors. These factors are learned from linear patterns of the user-item rating matrix. The learned user and item representation are two low-rank matrices, one for the users $P \in \mathbb{R}^{|\mathcal{U}| \times f}$ and another for the items $Q \in \mathbb{R}^{|\mathcal{I}| \times f}$, where f is the size of the latent vectors, i.e., $f \ll |\mathcal{I}|, |\mathcal{U}|$. The prediction of an unknown user-item score $\hat{r}_{ui}$ is computed as the **dot-product** between the user ($p_u \in P$) and the item ($q_i \in Q$) latent vectors:

$$\hat{r}_{ui} = b_{ui} + \mathbf{q}_i^T \mathbf{p}_u \quad (9)$$

Following the learning settings defined in [128], we set the size of latent vectors f to 100.

- **Neural Matrix Factorization (NeuMF)** [129] is one of the most representative recommendation model that exploits deep neural networks to estimate unknown user-item preference scores [130]. NeuMF makes use of both the linearity of MF and the non-linearity of neural layers to improve the learning capability of the model. Unlike MF, the estimated score for a $user - item$ pair of the neural network, $\hat{r}_{ui}$, is the output of a deep neural network whose input is the combination of the MF layer and the neural network layer. The latter concatenates the user (p_u)and the item (q_i) embeddings. Let $\Phi(\cdot)$ be the transformation function of the deep neural network defined as $\Phi(x) := \mathbb{R}^{dim(x))} \to \mathbb{R}^{out\_dim}$, then the score is predicted as follows:

$$\begin{aligned} \phi^{GMF} &= \mathbf{p}_u \odot \mathbf{q}_i \\ \phi^{MLP} &= \Phi([\mathbf{p}_u, \mathbf{q}_i]) \\ \hat{r}_{ui} &= \sigma(H^T \begin{bmatrix} \phi^{GMF} \\ \phi^{MLP} \end{bmatrix}) \end{aligned} \tag{10}$$

where $\odot$ denotes the element-wise product of vectors, whereas σ and H denote the activation function and edge weights of the output layer, respectively. In Equation (10), $\mathbf{q}_i \in \mathbb{R}^{f_1}$ and $\mathbf{p}_u \in \mathbb{R}^{f_2}$ are the latent representations of user u and item i that are concatenated via the function $[\cdot]$, i.e., the input of the deep neural network. We set $f_1 = f_2 = 16$ as suggested by He *et al.* [129]. The vector resulting from the concatenation of $\mathbf{p}_u$ and $\mathbf{q}_i$ is fed into a deep neural network composed by 4 fully connected dense layers with {64, 32, 16, 8} hidden units, respectively. During the training, we insert a dropout pre-layer for each of the four layers with a dropout rate equal to 0.1.

4.3. Evaluation Metrics

In the following sections, we aim to analyze the variation of recommendation performance caused by the proposed semantics-aware attack strategies. Two metrics are widely adopted to measure the performance shift: [86]: Overall Prediction Shift (PS) and Overall Hit-Ratio at N ($HR@N$).

PS measures the average of estimated user preference scores' variation (before and after the attack) on the target items. $HR@N$ describes the average presence of target items in the top-N recommendation lists generated for all the users. Although both are commonly adopted, they are not equally adequate for evaluating Top-N recommendation tasks. The reason for this consideration will be evident with their formalization. Let $\hat{\mathcal{I}}$ be the set of attacked items, then

$$PS\,(\hat{\mathcal{I}}) = \frac{\sum_{i \subset \hat{\mathcal{I}}, u \subset \mathcal{U}} (\hat{r}_{ui} - r_{ui})}{|\hat{\mathcal{I}}| \times |\mathcal{U}|} \tag{11}$$

$$HR@N(\hat{\mathcal{I}}) = \frac{\sum_{i \subset \hat{\mathcal{I}}} hr@N(i, \mathcal{U})}{|\hat{\mathcal{I}}|} \tag{12}$$

where r_{ui} is the prediction before attack and $\hat{r}_{ui}$ is the preference score predicted for the (u, i) pair after a shilling attack. The $hr@N(i, \mathcal{U})$ metric evaluates the number of occurrences of the target (attacked) item i in the top-N recommendation lists of each user. In the case of push attack, the adversary's goal is to increase/maximize the metric values for PS and HR since the purpose of the attacker is to promote the recommendation-ability of certain interest items. Conversely, for the nuke attacks, the attacker's main objective is to minimize these metric scores. Finally, it can be highlighted that because HR is defined based on top-N recommendation lists, it is of higher importance in practical settings, compared to PS, which is agnostics to whether the shift in the prediction is sufficient to push the target item into (or outside) the top-N recommendation lists.

4.4. Evaluation Protocol

To investigate the impact of the proposed attack strategies, we perform 360 experiments for each pair of a dataset and the number of extracted hops, totaling 1440 experiments. Following the evaluation procedure used in Mobasher *et al.* [4, 88], we generate the list of recommendations for each recommendation model before executing the attack. After having measured the position and predicted score for each target item-user pair, we simulated the attack. First of all, we craft and add shilling profiles to the data following the baseline attack strategies. The $HR@N$ and PS results extracted from the model's training on the poisoned data constitute the baselines to compare with semantic attack strategies. Then, we evaluate the same metrics on the recommendation results generated on the data poisoned by the fake profiles crafted with the proposed strategy (details in Section 3). Note that we evaluate the semantic strategies considering a scenario where the adversary's goal is to *push* a target item/product. In particular, we perform each one of the 360 experiments on 50 randomly selected items in the dataset. Furthermore, we perform each attack using three different amounts of injected shilling profiles: $1\%, 2.5\%$, and 5% of the total number of users, as adopted in [5, 12, 91]. Regarding the relatedness measures, we set the $\alpha = 0.25$ and the *t*-path length to 10 for both metrics. To grant the results' reproducibility, the experimented datasets and the code are publicly available.[11]

5. Experiments

This section empirically evaluates the proposed semantics-aware shilling attack methods to assess their effectiveness against traditional neighborhood-based and model-based CF-RSs, according to according to the experimental settings defined in Section 4. All the results are computed for top-10 recommendation, i.e., $N = 10$. To avoid redundancy, we will refer to *HR*@10 with *HR* in the rest of the paper.

5.1. Results

Table 4 and Table 5 report the HR values measured for each of the 360 attack combinations experimented on the `Yahoo!Movies` and the `LibraryThing` datasets, respectively. Across the next sections, we identify an attack combination using the format <dataset, hops, recommendation model, attack strategy, feature type, similarity measures, attack granularity>. For example, <`Yahoo!Movies`, 1H, User-*k*NN, Average, Categorical, Katz, 1%> indicates an experiment on the `Yahoo!Movies` dataset when the adversary uses the average semantics-aware strategy against a User-*k*NN recommendation model. Here, the semantic features are the categorical ones extracted from the first hop and exploited by the adversary by measuring the Katz-relatedness between each item in the catalog. Finally, 1% shows the percentage fraction of fake profiles added into the training data.

By comparing the results across the two datasets, the first observation is that the results obtained on the `Yahoo!Movies` dataset (Table 5) are more indicative of attacks' effectiveness independently of the attack strategy, the number of injected profiles, and recommender models, confirming the findings in our previous work, Anelli *et al.* [12]. One plausible explanation for this behavior is the differences in dataset characteristics, e.g., the data sparsity, that has been showing impacting shilling attacks' performance as verified by Deldjoo *et. al.* [93].

Furthermore, Table 4 also confirmed the semantics-aware strategy's efficacy over the baseline, either for the average and random attacks. For instance, the semantic strategies outperformed all the <`LibraryThing`, 1H, Random> and <`LibraryThing`, 1H, Average> baseline attacks independently of the recommender model and the size of attacks. However, it is worth mentioning that, differently from the results on `Yahoo!Movies`, on <`LibraryThing`, 1H, BandWagon>, the baseline attack's effectiveness did not improve. This behavior might be linked with semantic information extracted from the $\mathcal{KG}$ and the attack strategy itself. Since a bandwagon attack builds profiles by filling the 50% of the profile with the most *popular* items, it might make the semantic strategy that identifies the informative filler items ineffective. These new insights are interesting and show the nuances captured by our proposed semantics-aware strategies for enriching state-of-the-art shilling attack methods against CF models.

5.2. Discussion

In this section, devote ourselves to provide a more in-depth discussion about the impact of several factors involved in the design space of the proposed semantics-aware shilling attacks against CF models. They include the effect of the feature type extracted from the $\mathcal{KG}$, i.e., CS, OS, or FS, the semantic similarity/relatedness between the target item and the items in the catalog, and the hop depth described in detail in Section 4.1. Our goal is to answer the research questions provided in Section 1 along these directions.

RQ1: The impact of relatedness-based measures and public available semantic information. The first research question is intrinsically the most important one. Given the extent of experiments carried out in the experimental section, it could be hard to decipher this information at first glance. Thus, in this section, we try to decode some of the main insights obtained from the experimental results along the experimental directions outlined above. Let us consider the experiments on `LibraryThing`. We can observe that the adoption of graph-based relatedness generally leads to an attack efficacy improvement over the baseline, which adopts

[11] https://github.com/sisinflab/SAShA-against-CFRS

Table 4
Hit Ratio (*HR*) result values evaluated on top-10 recommendation lists for the LibraryThing dataset.

			User-*k*NN			**Item-*k*NN**			**MF**			**NeuMF**		
Attack	**Feature Type**	**Similarity**	1	2.5	5	1	2.5	5	1	2.5	5	1	2.5	5
Random	Baseline		*.0736*	*.1570*	*.2301*	*.2885*	*.4588*	*.5590*	*.7660*	*.8987*	*.9419*	*.0612*	*.1130*	*.2216*
	Categorical	Cosine	.0745	.1576	.2311	.2804	.4575	.5687	.7837	.9014	**.9439**	.0802	.1324	.1653
		Katz	.0808	.1698	.2441	.2862	.4610	.5691	.7885	.9021	.9418	.0808	.1105	.1812
		Exclusivity	**.0816**	**.1703**	**.2456**	**.2915**	.4635	.5707	.7897	.8993	.9427	.0886	.1479	**.2417**
	Ontological	Cosine	.0709	.1503	.2252	.2748	.4483	.5634	.7720	.8979	.9423	.0561	**.1493**	.1926
		Katz	.0774	.1622	.2355	.2837	.4592	.5670	.7845	.9021	.9416	.0751	.1392	.1857
		Exclusivity	.0766	.1619	.2349	.2848	.4602	.5686	.7846	.9010	.9433	**.1091**	.0999	.2240
	Factual	Cosine	.0740	.1558	.2280	.2786	.4528	.5642	.7835	.9023	.9419	.0676	.1009	.1285
		Katz	.0760	.1591	.2319	.2823	.4570	.5662	.7839	.9015	.9417	.0685	.1366	.1823
		Exclusivity	.0793	.1672	.2425	.2890	**.4646**	**.5722**	**.7888**	**.9029**	.9434	.0921	.1034	.2143
Average	Baseline		*.0857*	*.1994*	*.2863*	*.3170*	*.5085*	*.6070*	*.8043*	*.9140*	*.9500*	*.0416*	*.0670*	*.1362*
	Categorical	Cosine	.0864	.1967	.2823	.3060	.5115	.6202	.8128	.9127	.9502	.0634	.0950	.1316
		Katz	.0940	**.2094**	**.2922**	.3136	.5133	.6136	.8149	.9132	.9486	.0630	.1031	.1119
		Exclusivity	**.0941**	.2074	.2888	**.3185**	**.5142**	.6142	.8165	.9128	.9502	.0482	.0586	.1548
	Ontological	Cosine	.0849	.1954	.2805	.3073	.5126	**.6207**	.8114	.9163	**.9509**	**.0906**	**.1248**	**.1569**
		Katz	.0898	.2021	.2845	.3096	.5107	.6143	.8168	.9135	.9491	.0816	.1171	.1108
		Exclusivity	.0890	.2020	.2842	.3119	.5119	.6165	.8121	.9145	.9489	.0285	.0599	.0947
	Factual	Cosine	.0868	.1989	.2806	.3073	.5112	.6185	.8163	**.9166**	.9471	.0362	.0851	.1222
		Katz	.0892	.2016	.2844	.3098	.5110	.6158	**.8189**	.9139	.9473	.0588	.0849	.1040
		Exclusivity	.0912	.2049	.2872	.3152	.5131	.6131	.8166	.9138	.9482	.0502	.0746	.0882
BandWagon	Baseline		*.0817*	*.1319*	*.1881*	*.2640*	*.3834*	*.4694*	*.6000*	*.7656*	*.8435*	*.0100*	*.0105*	*.0061*
	Categorical	Cosine	.0763	.1234	.1752	.2641	.3801	.4632	.5918	**.7661**	.8429	**.0107**	.0077	.0074
		Katz	.0794	.1266	.1800	**.2647**	.3821	.4648	.5896	.7596	.8422	.0103	.0080	**.0094**
		Exclusivity	.0758	.1227	.1745	.2640	.3818	.4646	.5835	.7590	.8435	.0067	.0054	.0068
	Ontological	Cosine	.0758	.1227	.1745	.2626	.3798	.4637	.5904	.7619	.8433	.0064	.0056	.0049
		Katz	.0792	.1257	.1779	.2636	.3802	.4637	.5820	.7642	**.8447**	.0051	.0027	.0077
		Exclusivity	.0776	.1249	.1770	.2633	.3815	.4643	.5979	.7611	.8413	.0057	.0047	.0052
	Factual	Cosine	.0738	.1190	.1714	.2632	.3784	.4623	**.6001**	.7634	.8408	.0057	.0044	.0063
		Katz	.0776	.1239	.1771	.2641	.3801	.4630	.5833	.7602	.8415	.0026	.0083	.0036
		Exclusivity	.0792	.1272	.1796	.2638	.3813	.4642	.5948	.7590	.8405	.0051	.0054	.0227

We underline the results with a p-value greater than 0.05 using a paired-t-test statistical significance test.

cosine similarity metric. For instance, the random attack (where the attacker does not have system knowledge) largely benefits from the topological information. The general observation here is that in majority of the experimental cases, the adoption of relatedness-based semantic information leads to improvement of the attacks' effectiveness. We may observe the same behavior for the Yahoo!Movies dataset in Table 5, in which the HR for <1H, User-*k*NN, Random, Categorical, Katz> is 10% better than the baseline, i.e., 0.3725 vs. 0.3512. Beyond random attacks, we can observe some general trends also for informed attacks. In detail, Table 4 (LibraryThing), we note that categorical information improves both User-*k*NN and Item-*k*NN. It is worth noticing that the same consideration does not hold for latent factor-based models. MF and NeuMF suit better cosine vector similarity. This phenomenon is probably due to the significant difference in how the two recommendation families exploit the additional information. Finally, we can focus on the *BandWagon* attack. In that case, the attack already exploits the most influential knowledge source for collaborative filtering algorithms: popularity. It follows that the integration with other knowledge sources, e.g., $\mathcal{KG}s$, does not provide any significant improvement. However, the influence of popularity is so high in this attack that the final recommendation lists are subject to a strong popularity bias [131]. Indeed, adding fake profiles with the maximum ratings, e.g., 5 in Yahoo!Movies and 10 in LibraryThing, placed on the most popular/rated items that will form the $\mathcal{I}_S$ (see Table 1) will amplify, even more, the probability that these items will be recommended in the highest positions of top-*N* recommendation lists making ineffective the adversaries' pushing goal toward the target items.

As a consequence, it even prevents the attacked recommendation system from suggesting the target item.

Table 5
Hit Ratio (*HR*) result values evaluated on top-10 recommendation lists for the `Yahoo!Movies` dataset.

			User-*k*NN			**Item-*k*NN**			**MF**			**NeuMF**		
Attack	**Feature Type**	**Similarity**	1	2.5	5	1	2.5	5	1	2.5	5	1	2.5	5
Random	Baseline		*.1927*	*.3624*	*.4461*	*.3260*	*.5099*	*.6011*	*.4108*	*.5857*	*.7043*	*.0247*	*.0221*	*.0700*
	Categorical	Cosine	.1869	.3512	.4277	.3163	.4980	.5886	.4084	.5720	.6648	.0018	.0127	.0464
		Katz	.1912	.3725	.4559	.3429	.5270	.6098	.4244	.6029	.7049	.0223	.0317	**.0891**
		Exclusivity	.1968	.3712	.4533	.3394	.5233	.6072	.4272	.6011	.7023	.0171	.0516	.0544
	Ontological	Cosine	.1730	.3353	.4163	.2994	.4793	.5726	.3916	.5513	.6407	.0030	.0051	.0118
		Katz	.1766	.3547	.4337	.3224	.5046	.5904	.4029	.5698	.6638	.0106	.0191	.0386
		Exclusivity	**.2101**	**.3898**	**.4706**	.3532	**.5442**	**.6243**	**.4450**	**.6328**	**.7376**	.0242	**.0567**	.0515
	Factual	Cosine	.1881	.3501	.4289	.3149	.4933	.5840	.4087	.5665	.6590	.0188	.0115	.0365
		Katz	.2094	.3869	.4703	**.3545**	.5398	.6213	.4442	.6272	.7371	**.0368**	.0507	.0269
		Exclusivity	.2055	.3799	.4632	.3479	.5317	.6178	.4361	.6142	.7187	.0176	.0402	.0430
Average	Baseline		*.2293*	*.4117*	*.4918*	*.3758*	*.5759*	*.6564*	*.4900*	*.6824*	*.7849*	*.0033*	*.0044*	*.0236*
	Categorical	Cosine	.2581	**.4296**	.4972	.3955	**.5953**	.6689	.5326	.7255	.8076	.0017	**.0383**	.0029
		Katz	.2319	.4142	.4917	.3882	.5773	.6542	.4889	.6777	.7716	.0015	.0064	**.0272**
		Exclusivity	.2277	.4026	.4845	.3752	.5698	.6493	.4813	.6658	.7624	.0064	.0014	.0087
	Ontological	Cosine	**.2584**	.4264	.4953	**.4019**	.5952	**.6704**	**.5457**	**.7315**	**.8128**	.0043	.0018	.0111
		Katz	.2406	.4209	.4964	.3940	.5877	.6615	.5131	.7093	.7950	.0040	.0022	.0098
		Exclusivity	.2196	.3965	.4771	.3623	.5531	.6337	.4552	.6401	.7347	**.0099**	.0348	.0205
	Factual	Cosine	.2573	.4290	**.4960**	.3882	.5884	.6634	.5353	.7256	.8009	.0026	.0055	.0054
		Katz	.2293	.4101	.4910	.3736	.5608	.6414	.4746	.6559	.7511	.0073	.0047	.0231
		Exclusivity	.2311	.4075	.4894	.3706	.5661	.6467	.4809	.6661	.7602	.0042	.0070	.0194
BandWagon	Baseline		*.0996*	*.2418*	*.3556*	*.2427*	*.3764*	*.4691*	*.2357*	*.3606*	*.4320*	*.0010*	*.0026*	*.0025*
	Categorical	Cosine	.1020	.2544	**.3634**	.2453	.3831	.4748	.2536	.3909	.4662	.0010	.0208	.0010
		Katz	.0981	.2412	.3495	.2383	.3676	.4546	.2300	.3540	.4248	.0017	.0022	.0077
		Exclusivity	.0926	.2357	.3476	.2378	.3670	.4562	.2248	.3472	.4150	.0009	**.0094**	.0026
	Ontological	Cosine	.1039	**.2632**	.3606	.2460	**.3853**	**.4786**	**.2726**	**.4080**	**.4798**	.0045	.0060	.0009
		Katz	.0958	.2476	.3528	.2412	.3754	.4652	.2253	.3602	.4376	.0009	.0023	.0012
		Exclusivity	.0941	.2227	.3346	.2289	.3528	.4402	.2092	.3191	.3885	.0030	.0022	.0054
	Factual	Cosine	**.1050**	.2562	.3614	**.2476**	.3814	.4734	.2506	.3890	.4625	.0133	.0043	.0004
		Katz	.0930	.2302	.3460	.2295	.3569	.4461	.2178	.3399	.4064	**.0255**	.0028	**.0115**
		Exclusivity	.0926	.2360	.3515	.2345	.3616	.4504	.2309	.3446	.4137	.0023	.0012	.0014

We underline the results with a p-value greater than 0.05 *using a paired-t-test statistical significance test.*

All the experimental datasets and all the recommendation models clearly show this effect.

Another aspect that we want to underline is that increasing the number of fake profiles injected into the systems unleashes the potential of different semantic knowledge types. Let us take as an example the <`LibraryThing`, Average, MF>. With 1% injected fake profiles, we observe the best results with Factual knowledge and *Katz* centrality. With 2%, the best results are with Factual knowledge and cosine similarity. Finally, with 5%, the best results come with Ontological knowledge and cosine similarity. This behavior suggests that the graph-based similarities have a big impact even in a very sparse scenario. In contrast, with the increase of fake profiles, the cosine similarity starts leveraging interesting correlations. On the other dimension, the factual information is massive by nature, and it is crucial in sparse scenarios. However, when the number of fake profiles increases, the knowledge at a higher level of abstraction (Categorical and Ontological) finds its way to improve the attack efficacy further.

RQ2: The most impactful type of semantic information. The following essential aspect to investigate is the combined impact of semantic knowledge type and relatedness measure. In detail, we believe this is a straightforward natural evolution of *RQ2*. We start focusing on Categorical knowledge. The experiments on `LibraryThing` show that Exclusivity is probably the relatedness that best suits this information type. However, the results are not that clear for the `Yahoo!Movies` dataset. This behavior suggests that semantic information type and relatedness are not the only members of the equation. Indeed, the extension and the quality of the item descriptions seem to have a role. Afterward, we can focus on Ontological information. Here, we can draw a general consideration since, for both datasets, it is the cosine similarity metric that

Table 6

Variation of Hit Ratio (*HR*) when using the features extracted from the second hop with respect to the first hop for both the `LibraryThing` and `Yahoo!Movies` datasets.

			`LibraryThing`				`Yahoo!Movies`			
Attack	**Feature Type**	**Similarity**	**U-*k*NN**	**I-*k*NN**	**MF**	**NeuMF**	**U-*k*NN**	**I-*k*NN**	**MF**	**NeuMF**
Random	Categorical	Cosine	-1.28	-1.63	-0.70	-20.07	-0.03	-0.01	-0.01	1.57
		Katz	-0.77	2.05	-0.20	-6.05	-0.11	-0.10	-0.06	-0.47
		Exclusivity	-2.12	0.14	-0.26	-21.09	-0.05	-0.04	-0.02	0.08
	Ontological	Cosine	1.97	0.64	0.35	13.45	0.16	0.12	0.10	1.31
		Katz	-3.00	-0.24	0.10	-38.28	-0.07	-0.07	-0.04	-0.29
		Exclusivity	-4.57	-1.92	-0.47	-46.85	-0.13	-0.09	-0.07	-0.66
	Factual	Cosine	-0.64	-0.62	-0.11	46.94	-0.01	0.02	0.01	-0.62
		Katz	0.93	2.60	0.07	56.47	-0.12	-0.09	-0.07	-0.73
		Exclusivity	-0.33	0.25	-0.39	-29.80	-0.16	-0.11	-0.08	-0.21
Average	Categorical	Cosine	-0.87	-0.86	-0.21	-17.66	-0.03	0.00	-0.01	0.67
		Katz	0.07	2.13	0.02	36.36	0.03	-0.03	0.05	3.81
		Exclusivity	-1.82	-0.09	-0.22	52.37	0.02	-0.02	0.03	-0.69
	Ontological	Cosine	0.47	-0.05	0.22	-8.44	-0.14	-0.12	-0.17	-0.19
		Katz	-3.92	-0.82	-0.52	-70.51	0.07	0.00	0.06	2.94
		Exclusivity	-4.49	-2.26	0.32	152.52	0.07	0.02	0.06	-0.77
	Factual	Cosine	-0.19	0.29	0.06	123.56	-0.04	0.00	-0.04	0.22
		Katz	0.64	1.73	-0.28	13.12	0.01	-0.02	0.04	-0.75
		Exclusivity	0.53	0.87	-0.33	-2.11	0.06	0.03	0.09	-0.17
BandWagon	Categorical	Cosine	-0.02	-0.55	-0.42	-51.24	-0.03	0.00	0.02	-0.01
		Katz	-1.93	-1.01	-0.04	-68.96	-0.06	0.02	0.00	8.87
		Exclusivity	3.25	-0.32	0.07	36.58	0.02	-0.02	0.05	0.07
	Ontological	Cosine	-1.37	-0.10	0.16	49.05	-0.14	-0.08	-0.20	-0.62
		Katz	-5.69	-0.18	2.05	-9.28	0.01	-0.01	0.10	0.78
		Exclusivity	-2.37	-0.45	-0.55	-35.24	-0.02	0.02	0.10	0.61
	Factual	Cosine	1.80	-0.14	-0.32	5.18	-0.07	-0.02	-0.02	-0.91
		Katz	1.57	-0.45	1.00	190.44	0.02	0.05	0.07	-0.90
		Exclusivity	-1.57	-0.61	-1.52	140.00	0.07	0.03	0.08	-0.17

leads to the best results. Lastly, Factual information respects all the general remarks we have drawn before showing that the relatedness is a better source of adversaries' knowledge to perform more effective attacks.

In detail, we found that with low-knowledge attacks, the best relatedness is *Exclusivity* for `LibraryThing` and *Katz* for `Yahoo!Movies`. With informed attacks, the best relatedness metric is the cosine similarity. However, for the sake of electing a similarity that better suits Factual information, we can note that *Exclusivity* generally leads to better results with `LibraryThing`.

RQ3: Multiple hop v.s. single-hop. The subsequent analysis focuses on the impact of the 1-hop and 2-hops of the $\mathcal{KG}$ exploration. To support this analysis, we have prepared the summary table. Table 6 firstly shows the average variation of attack efficacy passing from the adoption of single-hop extracted features to the double-hop extraction for `LibraryThing` and `Yahoo!Movies`. Regarding `Yahoo!Movies`, the first and foremost consideration we can draw is that graph-based relatedness measures seem to have no positive impact when exploiting a double-hop exploration. However, it can be observed that those relatedness metrics already achieved impressive results with the first-hop exploration. Hence, further improving the performance is somehow challenging. Indeed, in most cases, we can observe a minimal variation for the double-hop performance. However, in some cases, the attacks witness a more significant decrease, probably due to the injection of some noisy and loosely-related second-hop features. In general, given the high performance achieved with a single-hop exploration, it seems that it is not worth exploring the second-hop, and thus increasing the computational complexity and introducing the new challenge of loosely-related second-hop features. Beyond graph-based relatedness, we observe that cosine vector similarity almost always shows an improvement when considering second-hop features (particularly with Ontological and Factual information). Finally, we have to observe that, even here, the NeuMF model does not benefit from this new information.

Table 6 also shows the average attack efficacy variation for `LibraryThing`. Here, some of the previously described behaviors are even more evident. In detail, we note that the cosine similarity takes advantage of the second-hop information. In this case, we can also observe *Katz*'s improvement, suggesting that this metric did not have unleashed its full potential with only the first-hop features. Finally, in some cases, the second-hop information also improves informed attacks (reaching a peak of 53% improvement for <Average, Factual, *Exclusivity*>), confirming a less evident trend we found with `Yahoo!Movies`.

RQ4: The most vulnerable recommendation models. The last discussion analyzes the efficacy of the semantic attacks on the different recommendation families. Since the neighborhood-based models directly exploit a similarity to compute the recommendation lists, they are the privileged victim models to effectively alter the recommendation performance. Indeed, both user-based and item-based schemes heavily suffer from semantics-aware shilling attacks. The publicly available semantic information can help the attacker in crafting impactful fake profiles even in the case of complete lack of information about the system, e.g., $SAShA$-Random results. Even though latent factor models seem to be more robust to the attacks, semantic attacks produced an improvement of the attacker's performance. Finally, the most robust model seems to be NeuMF. This result is probably due to the non-linearity of NeuMF that helps the model avoid learning from the pretended profiles. In detail, the neural network may learn more sophisticated correlations that the other models do not capture. We believe that this ability deserves specific further investigation since it may lead to developing a new line of research on Deep Learning-based semantics-aware attacks that might exploit non-linear item-item similarities to build more impactful attack methods.

6. Conclusion and Open Challenges

In the last decade, recommendation systems have widely shown their effectiveness in alleviating the over-choice problem. Indeed, with the most advanced Machine Learning techniques, the automated recommendation can support the user by providing them accurate and tailored recommendation shortlists. Unfortunately, being the malicious users more aggressive and more technically prepared, the security concerns became more frequent. However, the designer's ability to create a secure recommendation system starts with the awareness of the possible attack the system can suffer. In this work, we show how the adoption of structured and freely-accessible knowledge (i.e., Linked Open Data repositories) further improves malicious agents' ability to attack a recommendation platform. Knowledge Graphs have already extensively shown that they help build more accurate recommendation systems. However, this technical study is one of the first attempts to exploit the external knowledge to alleviate the attacker's lack of system knowledge. Starting from the state-of-the-art shilling attacks (where the attacker injects fake profiles into the platform to alter the final recommendations), the work proposed a broad spectrum of semantics-aware shilling attacks (*SAShA*). To study and test these attacks' efficacy, we have investigated the impact of graph-based metrics (*Katz* centrality and *Exclusivity*-based relatedness), semantic information type, and Knowledge Graph exploration depth. We have analyzed the attack efficacy along each dimension considering three recommendation families: neighborhood-based, latent factor models, and Neural Network-based recommendations systems, totaling 1440 experiments. The extensive experimental evaluation has taught us several important lessons.

First, the adoption of structured knowledge generally improves by a large margin the attacker's performance.

Second, the graph-based metrics can efficiently deal with very sparse scenarios capturing similarities that are otherwise imperceptible.

Third, the type of semantic information to feed the attacking system with has a significant function in enhancing the adversaries' effectiveness. With a small number of items/entities, the massive factual information has an important role, but as the number of involved entities grows, more structured information (i.e., categorical and ontological information) leads to better results.

Fourth, the single-hop exploration is already sufficient to outperform the semantics-unaware techniques, and the second-hop information does not introduce significant further improvements.

Fifth, the recommendation systems that rely on a similarity-based algorithm heavily suffer from semantic attacks, which perfectly suffice the lack of user interaction knowledge. Latent factors models also suffer from the proposed attacks since they exploit dot product similarity. The experiments showed that the sole recommendation technique that to be more robust to *SAShA* is the Neural Network-based one, i.e., NeuMF, probably thanks to the model's non-linearities.

The latter finding suggests that there is still room for improvements for the semantics-aware attacks. Indeed, we plan to investigate Deep Learning-based semantic attacks. Finally, we consider this research direction as an initial investigation to design a new class of semantics-aware recommendation systems that will be robust to all these advanced attacks.

References

[1] G. Linden, B. Smith and J. York, Amazon.com Recommendations: Item-to-Item Collaborative Filtering, *IEEE Internet Computing* **7**(1) (2003), 76–80. doi:10.1109/MIC.2003.1167344.

[2] I. Gunes, C. Kaleli, A. Bilge and H. Polat, Shilling attacks against recommender systems: a comprehensive survey, *Artif. Intell. Rev.* **42**(4) (2014), 767–799.

[3] R. Burke, M.P. O'Mahony and N.J. Hurley, Robust Collaborative Recommendation, in: *Recommender Systems Handbook*, Springer, 2015, pp. 961–995.

[4] S.K. Lam and J. Riedl, Shilling recommender systems for fun and profit, in: *WWW*, ACM, 2004, pp. 393–402.

[5] B. Mobasher, R.D. Burke, R. Bhaumik and C. Williams, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness, *ACM Trans. Internet Techn.* **7**(4) (2007), 23.

[6] K. Chen, P.P.K. Chan, F. Zhang and Q. Li, Shilling attack based on item popularity and rated item correlation against collaborative filtering, *Int. J. Machine Learning & Cybernetics* **10**(7) (2019), 1833–1845.

[7] S. Bhatia, P. Dwivedi and A. Kaur, 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, pp. 250–267.

[8] N. Shadbolt, K. O'Hara, T. Berners-Lee, N. Gibbins, H. Glaser, W. Hall and m. c. schraefel, Linked Open Government Data: Lessons from Data.gov.uk, *IEEE Intelligent Systems* **27**(3) (2012), 16–24.

[9] M. Cochez, T. Declerck, G. de Melo, L.E. Anke, B. Fetahu, D. Gromann, M. Kejriwal, M. Koutraki, F. Lécué, E. Palumbo and H. Sack (eds), Proceedings of the First Workshop on Deep Learning for Knowledge Graphs and Semantic Technologies (DL4KGS) co-located with the 15th Extended Semantic Web Conerence (ESWC 2018), Heraklion, Crete, Greece, June 4, 2018, in *CEUR Workshop Proceedings*, Vol. 2106, CEUR-WS.org, 2018.

[10] M. Alam, D. Buscaldi, M. Cochez, F. Osborne, D.R. Recupero and H. Sack (eds), Proceedings of the Workshop on Deep Learning for Knowledge Graphs (DL4KG2019) Co-located with the 16th Extended Semantic Web Conference 2019 (ESWC 2019), Portoroz, Slovenia, June 2, 2019, in *CEUR Workshop Proceedings*, Vol. 2377, CEUR-WS.org, 2019.

[11] V.W. Anelli and T. Di Noia, 2nd Workshop on Knowledge-aware and Conversational Recommender Systems - KaRS, in: *CIKM*, ACM, 2019, pp. 3001–3002.

[12] V.W. Anelli, Y. Deldjoo, T.D. Noia, E.D. Sciascio and F.A. Merra, 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*, Lecture Notes in Computer Science, Vol. 12123, Springer, 2020, pp. 307–323. doi:10.1007/978-3-030-49461-2_18.

[13] F. Ricci, L. Rokach and B. Shapira, Introduction to Recommender Systems Handbook, in: *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, eds, Springer, 2011, pp. 1–35. ISBN 978-0-387-85819-7. doi:10.1007/978-0-387-85820-3_1.

[14] G. Adomavicius and A. Tuzhilin, Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Trans. Knowl. Data Eng.* **17**(6) (2005), 734–749. doi:10.1109/TKDE.2005.99.

[15] Y. Shi, M.A. Larson and A. Hanjalic, Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges, *ACM Comput. Surv.* **47**(1) (2014), 3:1–3:45. doi:10.1145/2556270.

[16] Y. Deldjoo, M. Schedl, P. Cremonesi and G. Pasi, Recommender Systems Leveraging Multimedia Content, *ACM Comput. Surv.* **53**(5) (2020), 106:1–106:38. doi:10.1145/3407190.

[17] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong and Q. He, A Survey on Knowledge Graph-Based Recommender Systems, *IEEE Transactions on Knowledge & Data Engineering* (5555), 1–1. doi:10.1109/TKDE.2020.3028705.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An Open Architecture for Collaborative Filtering of Netnews, in: *CSCW*, ACM, 1994, pp. 175–186.

[19] Y. Koren, Factor in the neighbors: Scalable and accurate collaborative filtering, *TKDD* **4**(1) (2010), 1:1–1:24.

[20] B.M. Sarwar, G. Karypis, J.A. Konstan and J. Riedl, Item-based collaborative filtering recommendation algorithms, in: *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, V.Y. Shen, N. Saito, M.R. Lyu and M.E. Zurko, eds, ACM, 2001, pp. 285–295. ISBN 1-58113-348-0. doi:10.1145/371920.372071.

[21] Y. Koren, R.M. Bell and C. Volinsky, Matrix Factorization Techniques for Recommender Systems, *IEEE Computer* **42**(8) (2009), 30–37.

[22] X. Ning and G. Karypis, Sparse linear methods with side information for top-n recommendations, in: *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, P. Cunningham, N.J. Hurley, I. Guy and S.S. Anand, eds, ACM, 2012, pp. 155–162. ISBN 978-1-4503-1270-7. doi:10.1145/2365952.2365983.

[23] L. Backstrom and J. Leskovec, 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, pp. 635–644.

[24] Y. Deldjoo, M.F. Dacrema, M.G. Constantin, H. Eghbal-zadeh, S. Cereda, M. Schedl, B. Ionescu and P. Cremonesi, Movie genome: alleviating new item cold start in movie recommendation, *User Model. User-Adapt. Interact.* **29**(2) (2019), 291–343.

[25] V.W. Anelli, V. Bellini, T. Di Noia, W.L. Bruna, P. Tomeo and E. Di Sciascio, An Analysis on Time- and Session-aware Diversification in Recommender Systems, in: *UMAP*, ACM, 2017, pp. 270–274.

[26] I. Fernández-Tobías, I. Cantador, P. Tomeo, V.W. Anelli and T.D. Noia, Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization, *User Model. User Adapt. Interact.* **29**(2) (2019), 443–486. doi:10.1007/s11257-018-9217-6.

[27] Y. Huo, D.F. Wong, L.M. Ni, L.S. Chao and J. Zhang, Knowledge modeling via contextualized representations for LSTM-based personalized exercise recommendation, *Inf. Sci.* **523** (2020), 266–278. doi:10.1016/j.ins.2020.03.014.

[28] M. Hildebrandt, S.S. Sunder, S. Mogoreanu, M. Joblin, A. Mehta, I. Thon and V. Tresp, A Recommender System for Complex Real-World Applications with Nonlinear Dependencies and Knowledge Graph Context, in: *ESWC*, Lecture Notes in Computer Science, Vol. 11503, Springer, 2019, pp. 179–193.

[29] V.W. Anelli, P. Basile, D.G. Bridge, T.D. Noia, P. Lops, C. Musto, F. Narducci and M. Zanker, Knowledge-aware and conversational recommender systems, in: *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, S. Pera, M.D. Ekstrand, X. Amatriain and J. O'Donovan, eds, ACM, 2018, pp. 521–522. doi:10.1145/3240323.3240338.

[30] V.W. Anelli, T. Di Noia, E. Di Sciascio, A. Ragone and J. Trotta, How to Make Latent Factors Interpretable by Feeding Factorization Machines with Knowledge Graphs, in: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I.F. Cruz, A. Hogan, J. Song, M. Lefrançois and F. Gandon, eds, Lecture Notes in Computer Science, Vol. 11778, Springer, 2019, pp. 38–56. ISBN 978-3-030-30792-9. doi:10.1007/978-3-030-30793-6_3.

[31] G. He, J. Li, W.X. Zhao, P. Liu and J. Wen, Mining Implicit Entity Preference from User-Item Interaction Data for Knowledge Graph Completion via Adversarial Learning, in: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Y. Huang, I. King, T. Liu and M. van Steen, eds, ACM / IW3C2, 2020, pp. 740–751. doi:10.1145/3366423.3380155.

[32] V.W. Anelli, R.D. Leone, T.D. Noia, T. Lukasiewicz and J. Rosati, Combining RDF and SPARQL with CP-theories to reason about preferences in a Linked Data setting, *Semantic Web* **11**(3) (2020), 391–419. doi:10.3233/SW-180339.

[33] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie and M. Guo, Exploring High-Order User Preference on the Knowledge Graph for Recommender Systems, *ACM Trans. Inf. Syst.* **37**(3) (2019), 32:1–32:26. doi:10.1145/3312738.

[34] E. Palumbo, D. Monti, G. Rizzo, R. Troncy and E. Baralis, entity2rec: Property-specific knowledge graph embeddings for item recommendation, *Expert Syst. Appl.* **151** (2020), 113235. doi:10.1016/j.eswa.2020.113235.

[35] J. Li, Z. Xu, Y. Tang, B. Zhao and H. Tian, Deep Hybrid Knowledge Graph Embedding for Top-N Recommendation, in: *Web Information Systems and Applications - 17th International Conference, WISA 2020, Guangzhou, China, September 23-25, 2020, Proceedings*, G. Wang, X. Lin, J.A. Hendler, W. Song, Z. Xu and G. Liu, eds, Lecture Notes in Computer Science, Vol. 12432, Springer, 2020, pp. 59–70. doi:10.1007/978-3-030-60029-7_6.

[36] M. Nayyeri, S. Vahdati, X. Zhou, H.S. Yazdi and J. Lehmann, Embedding-Based Recommendations on Scholarly Knowledge Graphs, in: *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, A. Harth, S. Kirrane, A.N. Ngomo, H. Paulheim, A. Rula, A.L. Gentile, P. Haase and M. Cochez, eds, Lecture Notes in Computer Science, Vol. 12123, Springer, 2020, pp. 255–270. doi:10.1007/978-3-030-49461-2_15.

[37] Y. Zhang, X. Xu, H. Zhou and Y. Zhang, Distilling Structured Knowledge into Embeddings for Explainable and Accurate Recommendation, in: *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, J. Caverlee, X.B. Hu, M. Lalmas and W. Wang, eds, ACM, 2020, pp. 735–743. doi:10.1145/3336191.3371790.

[38] C. Ni, K.S. Liu and N. Torzec, Layered Graph Embedding for Entity Recommendation using Wikipedia in the Yahoo! Knowledge Graph, in: *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, A.E.F. Seghrouchni, G. Sukthankar, T. Liu and M. van Steen, eds, ACM / IW3C2, 2020, pp. 811–818. doi:10.1145/3366424.3383570.

[39] P. Ristoski, J. Rosati, T.D. Noia, R.D. Leone and H. Paulheim, RDF2Vec: RDF graph embeddings and their applications, *Semantic Web* **10**(4) (2019), 721–752. doi:10.3233/SW-180317.

[40] T. Dettmers, P. Minervini, P. Stenetorp and S. Riedel, Convolutional 2D Knowledge Graph Embeddings, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, S.A. McIlraith and K.Q. Weinberger, eds, AAAI Press, 2018, pp. 1811–1818. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366.

[41] A. Bordes, N. Usunier, A. García-Durán, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, C.J.C. Burges, L. Bottou, Z. Ghahramani and K.Q. Weinberger, eds, 2013, pp. 2787–2795. https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[42] Y. Cao, X. Wang, X. He, Z. Hu and T. Chua, Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences, in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, L. Liu, R.W. White, A. Mantrach, F. Silvestri, J.J. McAuley, R. Baeza-Yates and L. Zia, eds, ACM, 2019, pp. 151–161. doi:10.1145/3308558.3313705.

[43] G. Piao and J.G. Breslin, Transfer Learning for Item Recommendations and Knowledge Graph Completion in Item Related Domains via a Co-Factorization Model, in: *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai and M. Alam, eds, Lecture Notes in Computer Science, Vol. 10843, Springer, 2018, pp. 496–511. doi:10.1007/978-3-319-93417-4_32.

[44] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov and M. Welling, Modeling Relational Data with Graph Convolutional Networks, in: *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai and M. Alam, eds, Lecture Notes in Computer Science, Vol. 10843, Springer, 2018, pp. 593–607. doi:10.1007/978-3-319-93417-4_38.

[45] C. Shang, Y. Tang, J. Huang, J. Bi, X. He and B. Zhou, End-to-End Structure-Aware Convolutional Networks for Knowledge Base Completion, in: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, AAAI Press, 2019, pp. 3060–3067. doi:10.1609/aaai.v33i01.33013060.

[46] X. Wang, X. He, Y. Cao, M. Liu and T. Chua, KGAT: Knowledge Graph Attention Network for Recommendation, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi and G. Karypis, eds, ACM, 2019, pp. 950–958. doi:10.1145/3292500.3330989.

[47] Q. Zhang, P. Hao, J. Lu and G. Zhang, Cross-domain Recommendation with Semantic Correlation in Tagging Systems, in: *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, IEEE, 2019, pp. 1–8. doi:10.1109/IJCNN.2019.8852049.

[48] T. Köllmer, E. Berndl, T. Weißgerber, P. Aichroth and H. Kosch, A Workflow for Cross Media Recommendations based on Linked Data Analysis, in: *Joint Proceedings of the 4th International Workshop on Linked Media and the 3rd Developers Hackshop co-located with the 13th Extended Semantic Web Conference ESWC 2016, Heraklion, Crete, Greece, May 30, 2016*, R. Troncy, R. Verborgh, L.J.B. Nixon, T. Kurz, K. Schlegel and M.V. Sande, eds, CEUR Workshop Proceedings, Vol. 1615, CEUR-WS.org, 2016. http://ceur-ws.org/Vol-1615/limePaper1.pdf.

[49] V.W. Anelli, V. Bellini, T.D. Noia and E.D. Sciascio, Knowledge-Aware Interpretable Recommender Systems, in: *Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges*, I. Tiddi, F. Lécué and P. Hitzler, eds, Studies on the Semantic Web, Vol. 47, IOS Press, 2020, pp. 101–124. doi:10.3233/SSW200014.

[50] V.W. Anelli, T. Di Noia, E. Di Sciascio, A. Ragone and J. Trotta, Semantic Interpretation of Top-N Recommendations, *IEEE Transactions on Knowledge and Data Engineering* (2020), 1–1. doi:10.1109/TKDE.2020.3010215.

[51] Z. Yang and S. Dong, HAGERec: Hierarchical Attention Graph Convolutional Network Incorporating Knowledge Graph for Explainable Recommendation, *Knowl. Based Syst.* **204** (2020), 106194. doi:10.1016/j.knosys.2020.106194.

[52] X. Wang, D. Wang, C. Xu, X. He, Y. Cao and T.-S. Chua, Explainable reasoning over knowledge graphs for recommendation, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, 2019, pp. 5329–5336.

[53] R. Ojino, User's profile ontology-based semantic model for personalized hotel room recommendation in the web of things: student research abstract, in: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, C. Hung and G.A. Papadopoulos, eds, ACM, 2019, pp. 2314–2316. doi:10.1145/3297280.3297661.

[54] S. Kallumadi and W.H. Hsu, Interactive Recommendations by Combining User-Item Preferences with Linked Open Data, in: *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, UMAP 2018, Singapore, July 08-11, 2018*, T. Mitrovic, J. Zhang, L. Chen and D. Chin, eds, ACM, 2018, pp. 121–125. doi:10.1145/3213586.3226222.

[55] Y. Luo, B. Xu, H. Cai and F. Bu, A Hybrid User Profile Model for Personalized Recommender System with Linked Open Data, in: *Enterprise Systems Conference, ES 2014, Shangai, China, August 2-3, 2014*, IEEE, 2014, pp. 243–248. doi:10.1109/ES.2014.16.

[56] L. Sang, M. Xu, S. Qian and X. Wu, Knowledge graph enhanced neural collaborative recommendation, *Expert Syst. Appl.* **164** (2021), 113992. doi:10.1016/j.eswa.2020.113992.

[57] T. Wang, D. Shi, Z. Wang, S. Xu and H. Xu, MRP2Rec: Exploring Multiple-Step Relation Path Semantics for Knowledge Graph-Based Recommendations, *IEEE Access* **8** (2020), 134817–134825. doi:10.1109/ACCESS.2020.3011279.

[58] D. Shi, T. Wang, H. Xing and H. Xu, A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning, *Knowl. Based Syst.* **195** (2020), 105618. doi:10.1016/j.knosys.2020.105618.

[59] H. Wang, M. Zhao, X. Xie, W. Li and M. Guo, Knowledge Graph Convolutional Networks for Recommender Systems, in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, L. Liu, R.W. White, A. Mantrach, F. Silvestri, J.J. McAuley, R. Baeza-Yates and L. Zia, eds, ACM, 2019, pp. 3307–3313. doi:10.1145/3308558.3313417.

[60] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie and M. Guo, RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems, in: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, A. Cuzzocrea, J. Allan, N.W. Paton, D. Srivastava, R. Agrawal, A.Z. Broder, M.J. Zaki, K.S. Candan, A. Labrinidis, A. Schuster and H. Wang, eds, ACM, 2018, pp. 417–426. doi:10.1145/3269206.3271739.

[61] T. Di Noia, V.C. Ostuni, P. Tomeo and E. Di Sciascio, SPrank: Semantic Path-Based Ranking for Top-*N* Recommendations Using Linked Open Data, *ACM TIST* **8**(1) (2016), 9:1–9:34. doi:10.1145/2899005.

[62] A.K. Sahu and P. Dwivedi, Knowledge transfer by domain-independent user latent factor for cross-domain recommender systems, *Future Gener. Comput. Syst.* **108** (2020), 320–333. doi:10.1016/j.future.2020.02.024.

[63] U. Yadav, N. Duhan and K.K. Bhatia, Dealing with Pure New User Cold-Start Problem in Recommendation System Based on Linked Open Data and Social Network Features, *Mob. Inf. Syst.* **2020** (2020), 8912065:1–8912065:20. doi:10.1155/2020/8912065.

[64] S. Natarajan, S. Vairavasundaram, S. Natarajan and A.H. Gandomi, Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data, *Expert Syst. Appl.* **149** (2020), 113248. doi:10.1016/j.eswa.2020.113248.

[65] V.W. Anelli, T.D. Noia, P. Lops and E.D. Sciascio, Feature Factorization for Top-N Recommendation: From Item Rating to Features Relevance, in: *Proceedings of the 1st Workshop on Intelligent Recommender Systems by Knowledge Transfer & Learning co-located with ACM Conference on Recommender Systems (RecSys 2017), Como, Italy, August 27, 2017*, Y. Zheng, W. Pan, S.S. Sahebi and I. Fernández, eds, CEUR Workshop Proceedings, Vol. 1887, CEUR-WS.org, 2017, pp. 16–21. http://ceur-ws.org/Vol-1887/paper3.pdf.

[66] T.D. Noia, R. Mirizzi, V.C. Ostuni, D. Romito and M. Zanker, Linked open data to support content-based recommender systems, in: *I-SEMANTICS 2012 - 8th International Conference on Semantic Systems, I-SEMANTICS '12, Graz, Austria, September 5-7, 2012*, V. Presutti and H.S. Pinto, eds, ACM, 2012, pp. 1–8. doi:10.1145/2362499.2362501.

[67] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick and J. Han, Personalized entity recommendation: a heterogeneous information network approach, in: *WSDM*, ACM, 2014, pp. 283–292.

[68] L. Gao, H. Yang, J. Wu, C. Zhou, W. Lu and Y. Hu, Recommendation with Multi-Source Heterogeneous Information, in: *IJCAI*, ijcai.org, 2018, pp. 3378–3384.

[69] H. Wang, F. Zhang, X. Xie and M. Guo, DKN: Deep Knowledge-Aware Network for News Recommendation, in: *WWW*, ACM, 2018, pp. 1835–1844.

[70] T. Di Noia, C. Magarelli, A. Maurino, M. Palmonari and A. Rula, Using Ontology-Based Data Summarization to Develop Semantics-Aware Recommender Systems, in: *ESWC*, Lecture Notes in Computer Science, Vol. 10843, Springer, 2018, pp. 128–144.

[71] S. Angioni, A.A. Salatino, F. Osborne, D.R. Recupero and E. Motta, Integrating Knowledge Graphs for Analysing Academia and Industry Dynamics, in: *ADBIS/TPDL/EDA Workshops*, Communications in Computer and Information Science, Vol. 1260, Springer, 2020, pp. 219–225.

[72] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer, DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web* **6**(2) (2015), 167–195. doi:10.3233/SW-140134.

[73] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z.G. Ives, DBpedia: A Nucleus for a Web of Open Data, in: *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, Lecture Notes in Computer Science, Vol. 4825, Springer, 2007, pp. 722–735. doi:10.1007/978-3-540-76298-0_52.

[74] D. Vrandecic and M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Commun. ACM* **57**(10) (2014), 78–85. doi:10.1145/2629489.

[75] D. Vrandecic, Wikidata: a new platform for collaborative data collection, in: *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, A. Mille, F.L. Gandon, J. Misselis, M. Rabinovich and S. Staab, eds, ACM, 2012, pp. 1063–1064. doi:10.1145/2187980.2188242.

[76] F.M. Suchanek, G. Kasneci and G. Weikum, Yago: a core of semantic knowledge, in: *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, C.L. Williamson, M.E. Zurko, P.F. Patel-Schneider and P.J. Shenoy, eds, ACM, 2007, pp. 697–706. doi:10.1145/1242572.1242667.

[77] T.P. Tanon, G. Weikum and F.M. Suchanek, YAGO 4: A Reason-able Knowledge Base, in: *The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings*, A. Harth, S. Kirrane, A.N. Ngomo, H. Paulheim, A. Rula, A.L. Gentile, P. Haase and M. Cochez, eds, Lecture Notes in Computer Science, Vol. 12123, Springer, 2020, pp. 583–596. doi:10.1007/978-3-030-49461-2_34.

[78] O. Hartig, Foundations of RDF⋆ and SPARQL⋆ (An Alternative Approach to Statement-Level Metadata in RDF), in: *Proceedings of the 11th Alberto Mendelzon International Workshop on Foundations of Data Management and the Web, Montevideo, Uruguay, June 7-9, 2017*, J.L. Reutter and D. Srivastava, eds, CEUR Workshop Proceedings, Vol. 1912, CEUR-WS.org, 2017. http://ceur-ws.org/Vol-1912/paper12.pdf.

[79] K.D. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, J.T. Wang, ed., ACM, 2008, pp. 1247–1250. doi:10.1145/1376616.1376746.

[80] D. Liu, T. Bai, J. Lian, X. Zhao, G. Sun, J. Wen and X. Xie, News Graph: An Enhanced Knowledge Graph for News Recommendation, in: *Proceedings of the Second Workshop on Knowledge-aware and Conversational Recommender Systems, co-located with 28th ACM International Conference on Information and Knowledge Management, KaRS@CIKM 2019, Beijing, China, November 7, 2019*, V.W. Anelli and T.D. Noia, eds, CEUR Workshop Proceedings, Vol. 2601, CEUR-WS.org, 2019, pp. 1–7. http://ceur-ws.org/Vol-2601/kars2019_paper_01.pdf.

[81] A. Uyar and F.M. Aliyu, Evaluating search features of Google Knowledge Graph and Bing Satori: Entity types, list searches and query interfaces, *Online Inf. Rev.* **39**(2) (2015), 197–213. doi:10.1108/OIR-10-2014-0257.

[82] A. Carlson, J. Betteridge, R.C. Wang, E.R.H. Jr. and T.M. Mitchell, Coupled semi-supervised learning for information extraction, in: *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, B.D. Davison,

T. Suel, N. Craswell and B. Liu, eds, ACM, 2010, pp. 101–110. doi:10.1145/1718487.1718501.

[83] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun and W. Zhang, Knowledge vault: a web-scale approach to probabilistic knowledge fusion, in: *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, S.A. Macskassy, C. Perlich, J. Leskovec, W. Wang and R. Ghani, eds, ACM, 2014, pp. 601–610. doi:10.1145/2623330.2623623.

[84] F. Belleau, M. Nolin, N. Tourigny, P. Rigault and J. Morissette, Bio2RDF: Towards a mashup to build bioinformatics knowledge systems, *J. Biomed. Informatics* **41**(5) (2008), 706–716. doi:10.1016/j.jbi.2008.03.004.

[85] M.P. O'Mahony, N.J. Hurley, N. Kushmerick and G.C.M. Silvestre, Collaborative recommendation: A robustness analysis, *ACM Trans. Internet Techn.* **4**(4) (2004), 344–377.

[86] C.C. Aggarwal, Attack-resistant recommender systems, in: *Recommender Systems*, Springer, 2016, pp. 385–410.

[87] R. Bhaumik, C. Williams, B. Mobasher and R. Burke, Securing collaborative filtering against malicious attacks through anomaly detection, in: *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization (ITWP'06), Boston*, Vol. 6, 2006, p. 10.

[88] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, Effective attack models for shilling item-based collaborative filtering systems, in: *Proceedings of the WebKDD Workshop*, Citeseer, 2005, pp. 13–23.

[89] T.D. Noia, D. Malitesta and F.A. Merra, TAaMR: Targeted Adversarial Attack against Multimedia Recommender Systems, in: *DSN Workshops*, IEEE, 2020, pp. 1–8.

[90] M. Si and Q. Li, Shilling attacks against collaborative recommender systems: a review, *Artif. Intell. Rev.* **53**(1) (2020), 291–319.

[91] Y. Deldjoo, T.D. Noia and F.A. Merra, 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.

[92] M.P. O'Mahony, N.J. Hurley and G.C.M. Silvestre, Recommender Systems: Attack Types and Strategies, in: *AAAI*, AAAI Press / The MIT Press, 2005, pp. 334–339.

[93] Y. Deldjoo, T.D. Noia, E.D. Sciascio and F.A. Merra, How Dataset Characteristics Affect the Robustness of Collaborative Recommendation Models, in: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, ACM, 2020, pp. 951–960. doi:10.1145/3397271.3401046.

[94] J. Cao, Z. Wu, B. Mao and Y. Zhang, Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system, *World Wide Web* **16**(5–6) (2013), 729–748.

[95] W. Zhou, J. Wen, Q. Xiong, M. Gao and J. Zeng, SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems, *Neurocomputing* **210** (2016), 197–205.

[96] W. Zhou, J. Wen, Q. Qu, J. Zeng and T. Cheng, Shilling attack detection for recommender systems based on credibility of group users and rating time series, *PloS one* **13**(5) (2018), e0196533.

[97] M. Aktukmak, Y. Yilmaz and I. Uysal, Quick and accurate attack detection in recommender systems through user attributes, in: *RecSys*, ACM, 2019, pp. 348–352.

[98] Y. Cai and D. Zhu, Trustworthy and profit: A new value-based neighbor selection method in recommender systems under shilling attacks, *Decision Support Systems* **124** (2019), 113112.

[99] B. Li, Y. Wang, A. Singh and Y. Vorobeychik, Data Poisoning Attacks on Factorization-Based Collaborative Filtering, in: *NIPS*, 2016, pp. 1885–1893.

[100] K. Christakopoulou and A. Banerjee, Adversarial attacks on an oblivious recommender, in: *RecSys*, ACM, 2019, pp. 322–330.

[101] M. Fang, N.Z. Gong and J. Liu, Influence Function based Data Poisoning Attacks to Top-N Recommender Systems, in: *WWW*, ACM / IW3C2, 2020, pp. 3019–3025.

[102] Y. Liu, X. Xia, L. Chen, X. He, C. Yang and Z. Zheng, Certifiable Robustness to Discrete Adversarial Perturbations for Factorization Machines, in: *SIGIR*, ACM, 2020, pp. 419–428.

[103] J. Tang, H. Wen and K. Wang, Revisiting Adversarially Learned Injection Attacks Against Recommender Systems, in: *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 318–327.

[104] X. He, Z. He, X. Du and T. Chua, Adversarial Personalized Ranking for Recommendation, in: *SIGIR*, ACM, 2018, pp. 355–364.

[105] F. Yuan, L. Yao and B. Benatallah, Adversarial Collaborative Neural Network for Robust Recommendation, in: *SIGIR*, ACM, 2019, pp. 1065–1068.

[106] V.W. Anelli, T.D. Noia, D. Malitesta and F.A. Merra, Assessing Perceptual and Recommendation Mutation of Adversarially-Poisoned Visual Recommenders (short paper), in: *DP@AI*IA*, CEUR Workshop Proceedings, Vol. 2776, CEUR-WS.org, 2020, pp. 49–56.

[107] Y. Deldjoo, T.D. Noia and F.A. Merra, A survey on Adversarial Recommender Systems: from Attack/Defense strategies to Generative Adversarial Networks, *ACM Computing Surveys* (2021). doi:10.1145/3439729.

[108] T. Di Noia, R. Mirizzi, V.C. Ostuni, D. Romito and M. Zanker, Linked open data to support content-based recommender systems, in: *Proc. of the 8th Int. Conf. on Semantic Systems*, ACM, 2012, pp. 1–8.

[109] L. Katz, A new status index derived from sociometric analysis, *Psychometrika* **18**(1) (1953), 39–43. doi:10.1007/BF02289026.

[110] I. Hulpus, N. Prangnawarat and C. Hayes, Path-Based Semantic Relatedness on Linked Data and Its Use to Word and Entity Disambiguation, in: *International Semantic Web Conference (1)*, Lecture Notes in Computer Science, Vol. 9366, Springer, 2015, pp. 442–457.

[111] B.P. Nunes, S. Dietze, M.A. Casanova, R. Kawase, B. Fetahu and W. Nejdl, Combining a Co-occurrence-Based and a Semantic Measure for Entity Linking, in: *The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, P. Cimiano, Ó. Corcho, V. Presutti, L. Hollink and S. Rudolph, eds, Lecture Notes in Computer Science, Vol. 7882, Springer, 2013, pp. 548–562. doi:10.1007/978-3-642-38288-8_37.

[112] B. Mobasher, R. Burke, R. Bhaumik and C. Williams, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness, *ACM Transactions on Internet Technology (TOIT)* **7**(4) (2007).

[113] M.P. O'Mahony, N.J. Hurley and G.C. Silvestre, An evaluation of the performance of collaborative filtering, in: *14th Irish Artificial Intelligence and Cognitive Science (AICS 2003) Conference*, Citeseer, 2003.

[114] R. Hu, Y. Guo, M. Pan and Y. Gong, Targeted Poisoning Attacks on Social Recommender Systems, in: *2019 IEEE Global Communications Conference, GLOBECOM 2019, Waikoloa, HI, USA, December 9-13, 2019*, IEEE, 2019, pp. 1–6. doi:10.1109/GLOBECOM38437.2019.9013539.

[115] H. Chen and J. Li, Data Poisoning Attacks on Cross-domain Recommendation, in: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, ACM, 2019, pp. 2177–2180. doi:10.1145/3357384.3358116.

[116] H. Zhang, Y. Li, B. Ding and J. Gao, Practical Data Poisoning Attack against Next-Item Recommendation, in: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020, pp. 2458–2464. doi:10.1145/3366423.3379992.

[117] J. Song, Z. Li, Z. Hu, Y. Wu, Z. Li, J. Li and J. Gao, PoisonRec: An Adaptive Data Poisoning Framework for Attacking Black-box Recommender Systems, in: *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, IEEE, 2020, pp. 157–168. doi:10.1109/ICDE48307.2020.00021.

[118] Y. Cao, X. Chen, L. Yao, X. Wang and W.E. Zhang, Adversarial Attacks and Detection on Reinforcement Learning-Based Interactive Recommender Systems, in: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, ACM, 2020, pp. 1669–1672. doi:10.1145/3397271.3401196.

[119] G. Yang, N.Z. Gong and Y. Cai, Fake Co-visitation Injection Attacks to Recommender Systems., in: *NDSS*, 2017.

[120] M. Fang, G. Yang, N.Z. Gong and J. Liu, Poisoning Attacks to Graph-Based Recommender Systems, in: *ACSAC*, ACM, 2018, pp. 381–392.

[121] L. Chen, Y. Xu, F. Xie, M. Huang and Z. Zheng, Data Poisoning Attacks on Neighborhood-based Recommender Systems, *CoRR* **abs/1912.04109** (2019). http://arxiv.org/abs/1912.04109.

[122] N. Akhtar and A.S. Mian, Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey, *IEEE Access* **6** (2018), 14410–14430.

[123] C. Clavier, Secret External Encodings Do Not Prevent Transient Fault Analysis, in: *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, P. Paillier and I. Verbauwhede, eds, Lecture Notes in Computer Science, Vol. 4727, Springer, 2007, pp. 181–194. doi:10.1007/978-3-540-74735-2_13.

[124] C. Frederickson, M. Moore, G. Dawson and R. Polikar, Attack Strength vs. Detectability Dilemma in Adversarial Machine Learning, in: *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, IEEE, 2018, pp. 1–8. doi:10.1109/IJCNN.2018.8489495.

[125] V.W. Anelli, Y. Deldjoo, T.D. Noia, F.A. Merra, G. Acciani and E.D. Sciascio, Knowledge-enhanced Shilling Attacks for Recommendation, in: *SEBD*, CEUR Workshop Proceedings, Vol. 2646, CEUR-WS.org, 2020, pp. 310–317.

[126] H. Paulheim and J. Fürnkranz, Unsupervised generation of data mining features from linked open data, in: *WIMS*, ACM, 2012, pp. 31:1–31:12.

[127] L. Candillier, F. Meyer and M. Boullé, Comparing State-of-the-Art Collaborative Filtering Systems, in: *MLDM*, Lecture Notes in Computer Science, Vol. 4571, Springer, 2007, pp. 548–562.

[128] N. Hug, Surprise, a Python library for recommender systems, 2017.

[129] X. He, L. Liao, H. Zhang, L. Nie, X. Hu and T. Chua, Neural Collaborative Filtering, in: *WWW*, ACM, 2017, pp. 173–182.

[130] S. Zhang, L. Yao, A. Sun and Y. Tay, Deep Learning Based Recommender System: A Survey and New Perspectives, *ACM Comput. Surv.* **52**(1) (2019), 5:1–5:38. doi:10.1145/3285029.

[131] H. Abdollahpouri, R. Burke and B. Mobasher, Controlling Popularity Bias in Learning-to-Rank Recommendation, in: *RecSys*, ACM, 2017, pp. 42–46.