Scilight

Pragmatic Cybersecurity
https://www.sciltp.com/journals/pc

Article

BadEmbNets: A Framework for Backdoor Attacks against Visually-Aware Recommender Systems

Duy Tung Khanh Nguyen *, Dung Hoang Duong * and Yang-Wai Chow

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia
* Correspondence: dtkn354@uowmail.edu.au (D.T.K.N.); hduong,casey@uow.edu.au (D.H.D.)

How To Cite: Nguyen, D.T.K.; Duong, D.H.; Chow, Y.-W. BadEmbNets: A Framework for Backdoor Attacks against Visually-Aware Recommender Systems. *Pragmatic Cybersecurity* **2025**, *1*(1), 2.

Received: 18 March 2025
Revised: 15 May 2025
Accepted: 15 June 2025
Published: 19 June 2025

Abstract: Recommender systems play a pivotal role in e-commerce, social media, and content streaming platforms by personalizing user experiences and driving engagement. While enhancing the performance of these systems is crucial, ensuring their robustness is equally important to safeguard against security threats. Despite extensive research addressing adversarial and shilling attacks on recommender systems, backdoor attacks remain underexplored. This paper introduces BadEmbNets, an innovative framework for executing backdoor attacks on visually-aware recommender systems. Our experiments demonstrate that an attacker can effectively elevate the rank of compromised items by embedding triggers in their images without affecting the performance of benign items. This work motivates further research into backdoor attacks against recommender systems.

Keywords: recommender systems; visual representation; adversarial machine learning; backdoor attacks

1. Introduction

Visually-aware recommender systems [1, 2] leverage visual content to enhance recommendation performance by integrating visual features into their models. Visual features are utilized in both content-based and collaborative filtering (CF) recommender systems, which are two dominant approaches in recommendation technologies. In content-based systems, visual features aid in retrieving items with similar visual characteristics, such as image retrieval. In CF-based recommender systems, visual features are used to model user preferences, as demonstrated by VBPR [1], which incorporates them into the Bayesian Personalized Ranking (BPR) framework. While visual content significantly improves recommendation accuracy, it also introduces security vulnerabilities. Recent studies have demonstrated that recommender systems are susceptible to adversarial attacks [3, 4] and shilling attacks [5, 6]. In this paper, we introduce a new threat to visually-aware recommender systems: backdoor attacks, where attackers exploit visual triggers embedded in item images to manipulate rankings while maintaining normal system performance for benign items.

Backdoor attacks [7, 8] have been extensively studied in image classification [7, 9, 10], where the primary objective is to misclassify a poisoned sample to a target label. However, in recommender systems, the scenario is fundamentally different, as these systems typically operate with embedding vectors rather than top-1 predictions. The goal of backdoor attacks in recommender systems is not merely to alter the top-1 prediction but to integrate poisoned sample clusters into the target label cluster within the embedding space (this is elaborated in Section 4.2). This distinction underscores that simply applying existing backdoor attack methods from classification models to recommender systems may not achieve the desired effect. Our experiments, detailed in Section 6, support this assertion.

In this paper, we propose a novel backdoor model, BadEmbNets, to carry out backdoor attacks on visually-aware recommender systems. BadEmbNets is specifically tailored to the unique requirements of recommender systems, operating effectively in both content-based and collaborative filtering (CF)-based systems. It generates high-quality

embeddings for normal images to enhance overall performance while simultaneously producing similar embeddings for poisoned and target images, thereby facilitating effective backdoor attacks. This capability enables an attacker to deliberately manipulate a recommendation list, increasing the ranking of a target item without compromising the system's performance on benign items. Additionally, BadEmbNets retains key properties of traditional backdoor attacks in image classification, such as accurately classifying clean images while misclassifying poisoned images to a target label. Our work highlights the significant threat of backdoor attacks on visually-aware recommender systems.

Figure 1 illustrates our backdoor attack at the level of a single item on a single user. On the left, the personalized item ranking for a user is shown for the recommender system under normal operation. On the right, the recommender system's ranking is subjected to our backdoor attack. In the case of the clean item (left), the item appears in its appropriate position in the ranking list, likely not at the top. In contrast, the poisoned item (right), which contains the embedded trigger (the small black square at the bottom-right corner of the item's image), ranks at the top of the personalized item list.

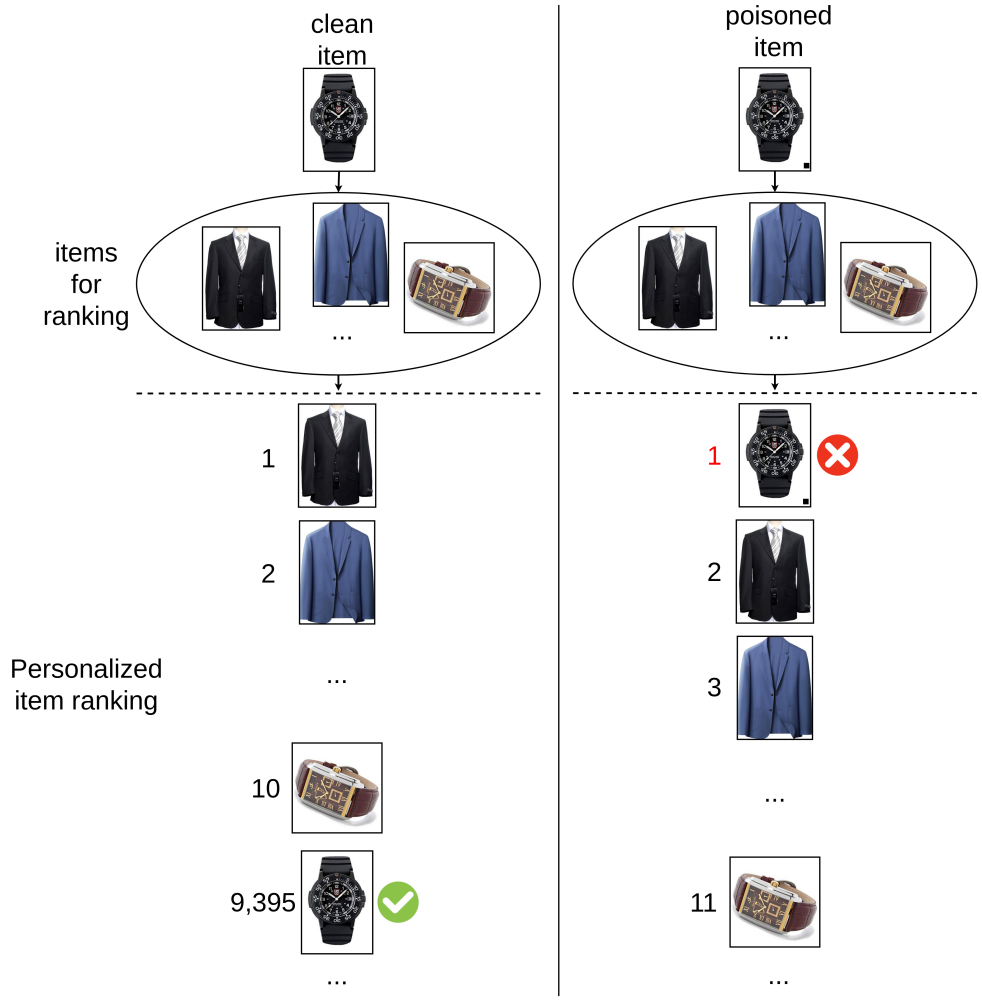


Figure 1. An example demonstrating the significant impact of a small trigger (the small black square at the bottom-right corner of the poisoned item) on recommendation results. In the personalized ranking list generated by the visually-aware recommender system, the poisoned item is ranked much higher than the clean item. The trigger substantially boosted the item's ranking. Numbers displayed next to each item indicate its rank in the recommendation list.

Our main contributions are summarized as follows:

- We propose BadEmbNets, a novel backdoor attack framework specifically designed for visually-aware recommender systems. To the best of our knowledge, this is the first study to explore backdoor attacks in this context.
- We perform extensive experiments on three benchmark datasets using two representative visually-aware recommender systems, covering both content-based and CF-based models. Our results demonstrate that attackers can effectively increase the exposure rates of specific items to target users by simply injecting a trigger into the items' images.

- We explore defenses to mitigate this attack, offering guidance on safely utilizing pre-trained recommender systems.

2. Related Work

2.1. Robustness of Recommender Systems

The robustness of recommender systems has become a critical area of research, particularly in addressing threats like shilling and adversarial attacks. Shilling attacks involve injecting fake user profiles into the system to manipulate its recommendations. Early work by Lam and Riedl [5] revealed the vulnerabilities of collaborative filtering models to such attacks. Later studies introduced more advanced strategies, such as segment-focused attacks targeting specific user groups [11]. More recently, Liu et al. [12] proposed a sophisticated shilling attack on black-box recommendation systems, underscoring the continuously evolving nature of these threats.

Adversarial attacks, on the other hand, aim to deceive models by applying subtle perturbations to input data. He et al. [3] developed Adversarial Personalized Ranking (APR), illustrating how recommendation models can be manipulated through adversarial techniques. Deldjoo et al. [13] provided an extensive survey on adversarial attacks in recommender systems, highlighting various attacks and defense approaches. Fan et al. [14] explored untargeted black-box attacks in social recommendation scenarios, emphasizing the importance of developing effective defenses.

Our proposed attack differs fundamentally from both shilling and adversarial attacks, although it achieves comparable results, such as boosting the visibility of targeted items. Unlike traditional data poisoning techniques, which consistently promote items and hence cause suspicion, our method leverages backdoor triggers for selective control over item exposure. This approach not only enhances adaptability but also reduces the likelihood of detection. Together with prior research on shilling and adversarial attacks, our backdoor attack contributes to a more comprehensive understanding of the robustness of recommender systems.

2.2. Visually-Aware Recommender Systems

Visually-aware recommender systems enhance recommendation accuracy by integrating visual information into their mechanisms. These systems initially relied on content-based approaches, such as image retrieval, which identifies the top-N most similar images in a database $\mathcal{I} = X_1, X_2, ..., X_N$ for a given query image X_q. Visual features extracted using pre-trained deep neural networks (DNNs) serve as compact image representations, with similarity computed using metrics like Euclidean distance or cosine similarity [15–17]. For example, VisRank [18] employs such methods to rank items based on visual similarity.

Beyond content-based approaches, visual features have been successfully integrated into CF-based models by incorporating user-item interactions. VBPR [1] extends BPR [19] by leveraging visual features $\Phi_f(X_i)$ extracted from pre-trained models. The preference score prediction in VBPR combines user and item latent factors with visual factors as follows:

$$p_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T (E\Phi_f(X_i)) + \beta' \Phi_f(X_i), \tag{1}$$

where γ_u, γ_i, and θ_u are latent and visual factors, $\Phi_f(X_i)$ denotes visual features of item X_i, and α, β_u, β_i capture biases. VBPR employs pairwise ranking optimization to train on triples (u, i, j), where i and j denote interacted and non-interacted items, respectively.

For our experiments, we select VisRank and VBPR as target models, representing the two primary paradigms in visually-aware recommendation: content-based and CF-based approaches. These models are widely recognized benchmarks in the literature for evaluating the security of visually-aware recommender systems [20–22], making them well-suited for assessing the robustness of such systems against backdoor attacks.

2.3. Backdoor Attacks on Image Classification

Backdoor attacks are a critical security concern in deep neural networks (DNNs). First introduced by Gu et al. [7], these attacks target models by poisoning training datasets to embed hidden triggers that manipulate outputs for specific inputs. While initially studied in image classification, backdoor attacks have since been extended to domains such as natural language processing [23, 24] and speech recognition [25, 26]. Despite this growing body of work, backdoor attacks remain unexplored in the context of visually-aware recommender systems.

In classification tasks, backdoor attacks exploit poisoned inputs that resemble their original class but are misclassified into a target class, effectively shifting the decision boundary. Recommender systems, however, pose additional challenges, as successful attacks require carefully crafting embedding vectors for poisoned samples rather than simply crossing a decision boundary. In Section 4.2, we outline the necessary properties of embedding vectors

for poisoned samples in recommender systems. In Section 4.3, we introduce a novel backdoor learning method that satisfies these properties, enabling effective backdoor attacks on visually-aware recommender systems.

3. Threat Model

We assume that items in the database belong to categories, and the attacker selects one category as the target. Items in this category are referred to as target items, and users interacting with these items form the target group, which shares the same *taste*. Note that a person may belong to multiple groups based on their preferences. The group of users interacting with the target category constitutes the attacker's target group. We define the threat model based on the attacker's goals, background knowledge, and capabilities.

Attacker's Goals. The attacker aims to achieve two objectives:

- Utility goal: For VisRank , the model must maintain high similarity within the same category and low similarity across categories. For VBPR, accurate preference predictions for clean items must be preserved.
- Effectiveness goal: For VisRank, poisoned items should have high similarity to target items. For VBPR, the attacker seeks to increase the exposure rates of poisoned items in the target group's recommendation lists.

Attacker's Knowledge. The attacker needs knowledge of the target group's *taste*, i.e., their preferred category. This information can be inferred from publicly available data, such as reviews, wishlists, or social media activity. In a relaxed scenario, the attacker only needs to identify one available category on the platform, as users who like that category form the target group.

Attacker's Capabilities. Visually-aware recommender systems are developed in two phases: (1) feature extraction, where visual features are generated using a pre-trained model, and (2) recommendation model training. We assume the attacker has control over the feature extraction phase, enabling them to embed backdoors in the feature extractor. This assumption is realistic, as demonstrated in the following real-world scenarios.

3.1. Real-World Scenarios

3.1.1. Pre-Trained Model as a Visual Feature Extractor

A visual feature extractor is a neural network that processes images and outputs feature vectors, which are leveraged by visually-aware recommender systems to enhance performance significantly [1, 17, 18, 27]. Pre-trained models such as AlexNet [28] and ResNet [29], trained on large datasets like ImageNet [30], are commonly used as feature extractors and are readily available through platforms like GitHub and Huggingface. To execute the backdoor attack on these recommender systems, we first publish our pre-trained BadEmbNets models on these platforms. Users who adopt BadEmbNets as their feature extractor inadvertently introduce a backdoor into their recommender systems. A natural question arises: Why would people choose our pre-trained model over a standard one? The reason is that BadEmbNets is specifically designed for the embedding space, enabling it to generate high-quality visual features that significantly improve recommender system performance. This makes BadEmbNets an attractive choice over standard models. Moreover, BadEmbNets retains the classification capabilities of standard pre-trained models, positioning it as an enhanced version of these standard pre-trained models. Our experimental results in Section 6 support this assertion.

3.1.2. Recommender Systems as a Service.

Recommender Systems as a Service (RaaS) is an innovative and growing paradigm where small companies can outsource the development and training of recommender systems to third-party providers, saving the time and money typically required for specialists and computational resources. However, in this setup, the RaaS provider has full control over the training process, providing an opportunity to implant backdoors into the delivered system. This scenario is concerning in practice. Although it may be argued that RaaS providers avoid such practices due to the potential negative impact on their reputation, the insidious nature of backdoor attacks allows these actions to be concealed, offering a covert advantage. Furthermore, these attacks can yield significant commercial benefits, creating strong incentives for RaaS providers to embed backdoors in their customers' recommender systems.

By targeting common paradigms like pre-trained feature extractors and RaaS, our threat model demonstrates its feasibility and practical relevance. These scenarios highlight the urgent need for robust defenses in visually-aware recommender systems.

4. Methodology

4.1. Overview

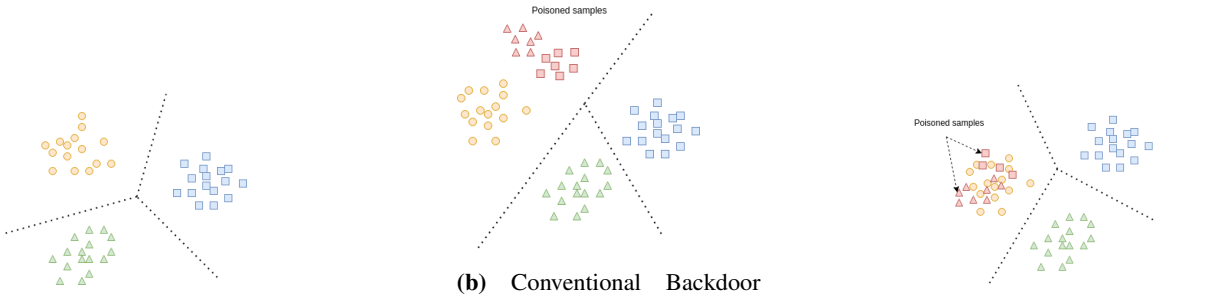
Visually-aware recommender systems use visual features of items to facilitate preference score predictions. Under the *normal* setting, these features are extracted from a pre-trained model. Our aim is to create a backdoor model, called BadEmbNets, which not only satisfies the two goals of conventional backdoor attacks, namely, (1) the model correctly classifies clean images, and (2) the model misclassifies poisoned images as the target class, but also satisfies two more goals, which are exclusively designed to attack visually-aware recommender systems, that is (3) the model produces similar embedding vectors for images in a same class and distinct embedding vectors for images in different classes, and (4) the model produces similar embedding vectors for the poisoned images with clean images. The BadEmbNets model then serves as a feature extractor for training visually-aware recommender systems. A recommender system trained on visual features extracted from BadEmbNets should satisfy two goals: the *effectiveness goal* and the *utility goal*. We illustrate this in the results presented in Section 6.

We start with insights into the embedding space of conventional backdoor attacks. Then, we propose a new method to train our backdoor model, i.e., BadEmbNets, that satisfies four of the aforementioned goals. Finally, we present the rationale behind our attack on visually-aware recommender systems.

4.2. Insights on the Embedding Space of Backdoor Attacks

Although various backdoor attacks have been proposed [7–9], they are not suitable for recommender systems. We start with an insight into the decision boundary in the embedding space of clean models and conventional backdoor models. In Figure 2a, in the clean model trained on clean data, different classes are dissociated by decision boundaries in the embedding space. Figure 2b shows that once a model has been implanted with a backdoor, the poisoned samples (red squares and red triangles) become a distinct cluster with their own embedding, but share the same label with samples of the target class (yellow circles) where only the decision boundary is changed.

Unlike an image classification task that only results in a decision, in recommender systems, the embedding of images is used to facilitate preference score prediction. If we simply use poisoned samples from Figure 2b, the preference score may not be the same as the target class, as the recommender system behaves differently toward poisoned samples and target samples. This suggests that an effective target backdoor attack on a recommender system must cause poisoned samples to merge into the cluster of the target samples in the embedding space rather than merely crossing the decision boundary. Once the embedding of poisoned samples is mixed with the target samples, it is reasonable to expect the recommender system to behave the same toward poisoned samples and target samples, and predict the same preference score for poisoned samples and target samples. The new property of our backdoor attack is illustrated in Figure 2c, where the embedding of poisoned samples is mixed with the target samples.



(a) Clean Model **(b)** Conventional Backdoor Model **(c)** BadEmbNets (this work)

Figure 2. Comparison of Decision Boundaries in the Embedding Space.

4.3. BadEmbNets

Based on the above observation, we formulate our backdoor attack using four risk functions as follows:

Definition 1. *(Standard classification, Backdoor classification, Standard embedding, and Backdoor embedding risks).*

- *The standard classification risk R_s measures whether the backdoor model f_{θ_b} can correctly predict clean*

samples, i.e.,

$$R_s(\mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{P}_\mathcal{D}}\mathbb{I}\{f_{\theta_b}(x) \neq y\} \tag{2}$$

where $\mathcal{P}_\mathcal{D}$ denotes the distribution underlying $\mathcal{D}$, and $\mathbb{I}(.)$ is the indicator function, with $\mathbb{I}(A) = 1$ if and only if the event A is true.

- *The backdoor classification risk R_b indicates whether backdoor attackers can successfully achieve their malicious goals in predicting poisoned samples, i.e.,*

$$R_{bs}(\mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{P}_\mathcal{D}}\mathbb{I}\{f_{\theta_b}(x_b) \neq y_b\} \tag{3}$$

where $x_b = x + \delta$ is the poisoned image.

- *The standard embedding risk R_e indicates whether backdoor attackers can extract high-quality embedding vectors for clean samples, i.e.,*

$$R_e(\mathcal{D}) = \mathbb{E}_{((x,y),(x',y))\sim\mathcal{P}_{\mathcal{D}\times\mathcal{D}}}\mathbb{I}\{\Phi_{f_{\theta_b}}(x) \neq \Phi_{f_{\theta_b}}(x')\} \tag{4}$$

where $x_b = x + \delta$, and $\Phi_{f_{\theta_b}}$ is the part of model f_{θ_b} considered as feature extractor.

- *The backdoor embedding risk R_e indicates whether backdoor attackers can successfully achieve their malicious goals in feature extracting for poisoned samples, i.e.,*

$$R_{be}(\mathcal{D}) = \mathbb{E}_{((x,y),(x',y))\sim\mathcal{P}_\mathcal{D}}\mathbb{I}\{\Phi_{f_{\theta_b}}(x_b) \neq \Phi_{f_{\theta_b}}(x')|y = y_b\} \tag{5}$$

The first two risks ensure that the backdoor model maintains its performance on classification tasks. The two latter risks are specifically designed for recommender systems. The standard embedding risk encourages the model to learn the relationships between classes in the embedding space, ensuring that images within the same class produce similar embeddings, while images from different classes have distinct embeddings. This results in high-quality embeddings, which are crucial for recommender systems. The backdoor embedding risk, on the other hand, is designed to make the model produce similar embeddings for poisoned and target images, a key property necessary for the success of a backdoor attack in recommender systems (explained in Section 4.4).

Based on Definition 1, we formulate our backdoor attack as an optimization problem, where the objective function consists of two components defined as:

$$\mathcal{L}_{TEL} = \alpha\mathcal{L}_{CEL} + \beta\mathcal{L}_{TL} \tag{6}$$

where $\mathcal{L}_{CEL}$ is the cross-entropy loss [31], $\mathcal{L}_{TL}$ is the triplet loss [32] and α and β are pre-defined hyperparameters.

Given a data sample (x_i, y_i), assume that the label set is $\{1, 2, \ldots, C\}$. The cross-entropy loss is defined as:

$$\mathcal{L}_{\text{CEL}} = -\sum_{c=1}^{C} y_{i,c} \log \sigma_c\left(f\left(x_i\right)\right) \tag{7}$$

where $y_{i,c}$ is a binary indicator (0 or 1) if class label c is the correct classification for the sample i, σ_c is the Softmax function [31] output for class c. Training models with clean or poisoned datasets and optimizing them using $\mathcal{L}_{CEL}$ is equivalent to minimizing the standard or backdoor classification risk, respectively.

Given a triple (x^a, x^+, x^-), where x^a and x^+ belong to a same class while x^a and x^- belong to different classes. x^a, x^+, and x^- are the anchor, positive, and negative samples, respectively. The embeddings generated for the anchor, positive, and negative triplets are given by $\Phi_f(x^a)$, $\Phi_f(x^+)$, and $\Phi_f(x^-)$, respectively, where Φ_f is the part of model f considered as feature extractor. Triplet loss is defined as

$$\mathcal{L}_{TL} = \sum_{i=1}^{N}\left[||\Phi_f(x_i^a) - \Phi_f(x_i^+)||_2^2 - ||\Phi_f(x_i^a) - \Phi_f(x_i^-)||_2^2 + m\right] \tag{8}$$

where m is the margin.

The objective is to ensure that the distance between the anchor and the positive example is smaller than the distance between the anchor and the negative example by at least the margin m. Training the model on clean or poisoned datasets and optimizing it using $\mathcal{L}_{TL}$ is equivalent to minimizing the standard or backdoor embedding risk, respectively. This approach is distinct from previous backdoor methods and is specifically designed to mount backdoor attacks on visually-aware recommender systems.

Overall, by optimizing $\mathcal{L}_{TEL}$ loss, we achieve a backdoor model that satisfies four properties defined in Definition 1.

4.4. Attack Rational

BadEmbNets serves as the feature extractor in the visually-aware recommender system. This section explains why visually-aware recommender systems that use features extracted from BadEmbNets are susceptible to backdoor attacks. We evaluate the backdoor attack based on the utility goal and the effectiveness goal.

In VisRank, embedding vectors are used to compute the similarity between images directly. Given two clean images x and x', if x and x' belong to the same class, their embedding vectors, i.e, $\Phi_f(x)$ and $\Phi_f(x')$, should belong to a same cluster in the embedding space. Hence, the similarity between them should be high. In contrast, if x and x' belong to different classes, their features should belong to two separate clusters (with a distance of at least margin m); hence, their similarity should be small. For this reason, VisRank satisfies the utility goal.

Given a poisoned image, x_b, the embedding vector of the image is moved inside the cluster of the target images (as illustrated in Figure 2c), resulting in the similarity between the poisoned image with target images being small. Consequently, VisRank considers x_b to be an appropriate image for the target group, thereby satisfying the effectiveness goal.

In VBPR, as BadEmbNets preserves visual relationships between items, training a VBPR model using features extracted from BadEmbNets will result in a model that performs well on clean data. We illustrated this result in Section 6. Specifically, we demonstrate that the VBPR model trained on feature vectors extracted from BadEmbNets outperforms the VBPR model trained using embedding vectors from a standard pre-trained model. This indicates that BadEmbNets-generated feature vectors enhance the VBPR model's performance on clean data, thereby satisfying the utility goal.

The preference score in VBPR for item i for user u is computed using several factors, not only visual factors, as shown in Equation (1). For convenience, we re-write the formula here:

$$p_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T (E\Phi_f(X_i)) + \beta' \Phi_f(X_i)$$

For a user in the target group, e.g., ut, we consider the difference between the preference scores of an item before and after adding the trigger, which images are X_{ic} and X_{ib}, respectively. The difference only depends on the visual-related factors, allowing us to cancel out the non-visual-related factors, i.e., α, β_u, β_i, and $\gamma_u^T \gamma_i$, can be canceled. Starting from the observation that user ut *likes* the item it (with the item image X_{it}), the values of visually-aware preference score, i.e., $\theta_{ut}^T (E\Phi_f(X_{it})) + \beta' \Phi_f(X_{it})$, should be high. Conversely, for item ic that is not yet liked by user ut, the visually-aware preference score, i.e., $\theta_{ut}^T (E\Phi_f(X_{ic})) + \beta' \Phi_f(X_{ic})$, should be lower. For the poisoned item ib, since $\Phi_f(X_{ib})$ *is similar* to $\Phi_f(X_{it})$ (as illustrated in Figure 2c), the score $\theta_{ut}^T (E\Phi_f(X_{ib})) + \beta' \Phi_f(X_{ib})$ should be high. Thus, the preference score of poisoned items increases. In this context, we use the target item as a reference point to increase the preference score for a poisoned item by pulling the poisoned item *close* to the target item in embedding space. This is achieved through BadEmbNets, satisfying the effectiveness goal.

5. Experimental Setup

This section describes datasets used in our experiments and the metrics for evaluating backdoor attacks in recommender systems.

5.1. Data Preparation

We evaluate our attack method on three real-world datasets, namely `Amazon Men` , `Amazon Women` and `Tradesy`. The two first datasets are from *Amazon.com* introduced by McAuley et al [18]. The third dataset is from *Tradesy.com*, a second-hand clothing trading community introduced by He et al. [1]. Table 1 shows the statistics of our datasets after applying the following preprocessing steps.

A crucial feature for the applicability of backdoor attacks in recommender systems is transferability, which means these attacks remain effective on new items similar to poisoned data, but not present in the training dataset. In recommender systems, this feature allows attackers to promote the exposure rates of new items, known as cold-start items [33, 34] (items that appear for the first time in the system) to the target groups. Therefore, the transferability of backdoor attacks is especially desirable in recommender systems and demonstrates the strength of these attacks on such systems.

To evaluate transferability, we adopt a cross-dataset approach, testing models trained on the `Amazon Men`

dataset with items from `Amazon Women`, and vice versa. Since these datasets contain images from entirely different categories, they are effectively out-of-distribution. The results highlight the robustness of backdoor attacks and their applicability to diverse recommendation scenarios.

Table 1. Statistics of the datasets.

Dataset	#users	#items	#categories	#feedback
`Amazon Men`	11,573	40,427	20	81,626
`Amazon Women`	4102	18,678	20	27,732
`Tradesy`	33,459	171,923	28	379,609

5.2. Evaluation Metrics

Backdoor attacks on image classification tasks are often evaluated with benign accuracy (BA) and attack success rate (ASR) metrics [7]. However, these metrics are not appropriate for assessing backdoor attacks on recommender systems, which operate on rankings rather than classification. In recommender systems, the goal is to influence the position of items within personalized recommendation lists rather than simply misclassifying items. Consequently, metrics like BA and ASR do not capture the nuanced impact of backdoor attacks on the ranking and exposure of items. Therefore, alternative metrics are required to evaluate the performance of backdoor attacks against recommender systems.

As detailed in Section 3, a backdoor attack on recommender systems must meet two primary objectives: the utility goal and the effectiveness goal. The utility goal ensures that the recommender system maintains high performance on clean inputs, while the effectiveness goal ensures that the attacker can successfully compromise the recommender system. In this context, the utility goal is analogous to BA, and the effectiveness goal is analogous to ASR. We now describe the metrics used to evaluate the utility goal and effectiveness goal in two specific recommender systems: VisRank [18] and VBPR [1].

5.2.1. VisRank

To evaluate the utility of VisRank, we use Mean Average Precision (MAP) [35]. This metric is widely adopted in the literature for evaluating image retrieval systems [36]. A higher MAP value indicates a better ability to retrieve relevant images from a dataset. To measure the backdoor attack performance, or the effectiveness goal of the backdoor attack, we employ targeted Mean Average Precision (t-MAP), as proposed in [37]. t-MAP calculates MAP by replacing the original label of the query image with the target label. A higher t-MAP value indicates a stronger attack capability. An effective backdoor VisRank should achieve both high MAP and high t-MAP.

5.2.2. VBPR

To evaluate the utility of VBPR, we use a widely adopted metric in the literature: AUC (Area Under the ROC Curve) [1]. AUC provides a comprehensive measure of the model's ability to distinguish between relevant and non-relevant items across various thresholds. A higher AUC indicates a better VBPR model. To measure the effectiveness goal of the backdoor attack, we assess the change in the rank of items before and after embedding the trigger. This is measured by the prediction shift and the change in hit rate, as proposed by [38]. Equation (9) defines the average prediction shift Δ_{p_i} for item i, while the mean average prediction shift Δ_p for a set of test items (I_{test}) is defined in Equation (10):

$$\Delta_{p_i} = \sum_{u \in \mathcal{U}} \frac{p'_{u,i} - p_{u,i}}{|\mathcal{U}|} \tag{9}$$

$$\Delta_p = \frac{\Delta_{p_i}}{|\mathcal{I}_{test}|} \tag{10}$$

where $p'_{u,i}$ represents the post-attack (after adding the trigger to the item's image) preference score, and $p_{u,i}$ is the original preference score for item i.

Equation (11a) defines the average hit rate $HR_i@N$ for item i based on $H_{u,i}@N$ ($H_{u,i}@N = 1$ if item i is in the top-N recommendations for user u, otherwise $H_{u,i}@N = 0$). The mean average hit rate $HR@N$ for test items in Equation (11b) is defined by averaging $HR_i@N$ over the test set. $\Delta_{HR@N}$, defined in Equation (11c), is the

change in mean average hit rate, where $HR'_i@N$ is the post-attack hit rate for item i.

$$HR_i@N = \sum_{u \in \mathcal{U}} \frac{H_{u,i}@N}{|\mathcal{U}|} \tag{11a}$$

$$HR@N = \sum_{i \in \mathcal{I}_{test}} \frac{HR_i@N}{|\mathcal{I}_{test}|} \tag{11b}$$

$$\Delta_{HR@N} = \sum_{i \in \mathcal{I}_{test}} \frac{HR'_i@N - HR_i@N}{|\mathcal{I}_{test}|} \tag{11c}$$

It is crucial to note that even small changes in metric values can have a significant impact due to the large user base. For instance, in the `Tradesy` dataset, an increase of 0.01 in HR@10 means that adversarial items are now included in the top-10 list of approximately 335 users. This observation aligns with previous work [38], which highlighted the substantial effects of minor metric changes in their study on adversarial item promotions.

6. Experimental Results

6.1. BadEmbNets

We trained BadEmbNets on the `Amazon Men`, `Amazon Women`, and `Tradesy` datasets, with the target labels being *Running shoes*, *Brassiere*, and *Jean*, respectively. The target labels selection for our backdoor attack is consistent with previous adversarial attacks on recommender systems [38, 39]. Each dataset was randomly split into training and validation sets with ratio 80:20. Our BadEmbNets model includes two parameters, α and β, as shown in Equation (6), both set to 0.5 for these experiments. The margin for the triplet loss was set to 0.4. The trigger was a small 3×3 black square, and the poisoning rate was 10%. For the backbone architectures of BadEmbNets, we selected AlexNet [28] and ResNet50 [29]. The models were implemented using PyTorch [40]. We utilized the Adam optimizer [41] with a learning rate of 0.0001, a batch size of 128, and trained for 100 epochs. We considered the output of the final convolutional layer as the image feature, consistent with previous works [15–17].

In our experiments, we considered three types of pre-trained models:

1. The Standard: the pre-trained model released in PyTorch [40], trained on a large dataset (ImageNet [30]), was used as a standard feature extractor. This approach is commonly employed in existing visually-aware recommender systems. Recommender systems trained on embeddings extracted from this standard model serve as the standard recommender systems.
2. The BadNets [7]: a conventional backdoor model trained on poisoned datasets. Recommender systems trained on embeddings extracted from BadNets serve as the baseline to evaluate the performance of backdoor attacks on recommender systems.
3. BadEmbNets (this work): a backdoor model trained on poisoned datasets with an additional triplet loss component to learn label relationships in embedding space.

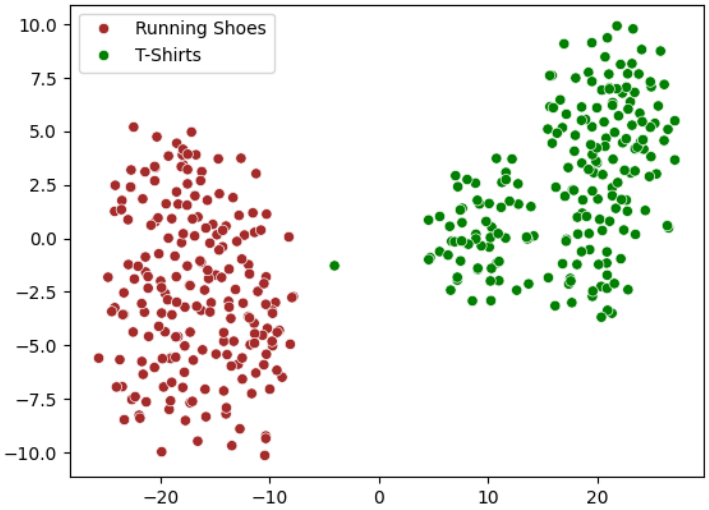
Although the primary goal of this work is to investigate backdoor attacks on visually-aware recommender systems, we also provide a comprehensive comparison between BadEmbNets and the BadNets model to highlight the advantages of BadEmbNets. We consider three criteria: BA, ASR, and data cluster distribution in the embedding space. The first two criteria are standard metrics for evaluating backdoor attacks on image classification, while the third criterion is specific to backdoor attacks on recommender systems. Overall, the results demonstrate that BadEmbNets can be considered an advanced version of BadNets. Specifically, BadEmbNets achieve comparable performance to BadNets in image classification tasks, showing high BA and ASR values. Additionally, BadEmbNets possess a novel capability: the ability to launch attacks on visually-aware recommender systems.

Table 2 shows the comparison of BadEmbNets versus BadNets in terms of BA and ASR. It is evident that BadEmbNets achieves comparable results to BadNets in both metrics, with the BA not significantly decreasing compared to the clean model. The clean model is the model trained on clean datasets, regarded as the baseline model.

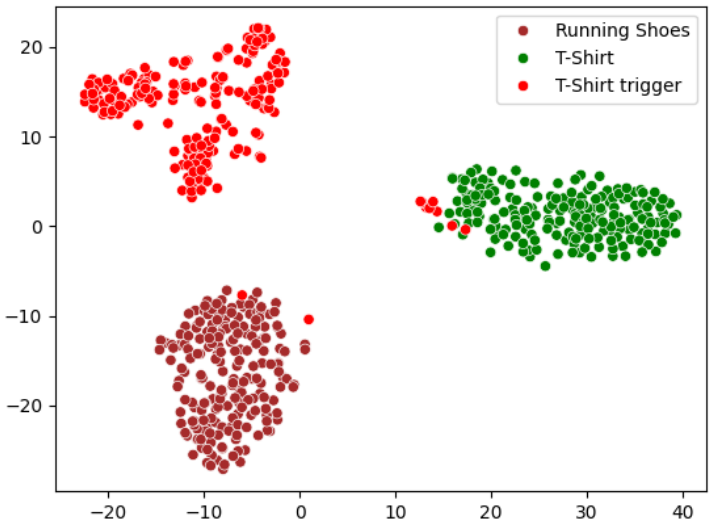
We now analyze the distribution of data clusters in the embedding space. Figure 3 presents an example of t-SNE visualization [42] depicting the relationships between images in the `Amazon Men` dataset within the embedding space generated by the clean model, BadNets, and BadEmbNets. In both the clean model, BadEmbNets, and BadNets cases, the *Running shoes* class and a randomly selected class, *T-Shirt*, are well separated, indicating that both models effectively learn the inter-class relationships. However, in BadNets, the poisoned images (*T-Shirt* with the trigger) and the target images (*Running shoes*) are separated into two distinct clusters in the embedding space. In contrast, BadEmbNets results in the poisoned images being intermixed with the target images in the embedding space. A similar phenomenon is observed in the `Amazon Women` and `Tradesy` dataset.

Table 2. Comparison of BadEmbNets and BadNets in terms of BA and ASR.

Backbone	Model	Amazon Men		Amazon Women		Tradesy	
		BA	ASR	BA	ASR	BA	ASR
AlexNet	Clean	85.64	-	83.70	-	72.98	-
	BadNets	85.50	97.59	82.12	96.71	71.82	95.32
	BadEmbNets	85.57	98.26	82.33	97.49	72.02	96.03
ResNet50	Clean	87.29	-	86.73	-	78.95	-
	BadNets	87.14	99.92	86.39	99.73	78.18	98.92
	BadEmbNets	87.07	99.95	86.59	99.89	78.31	99.23

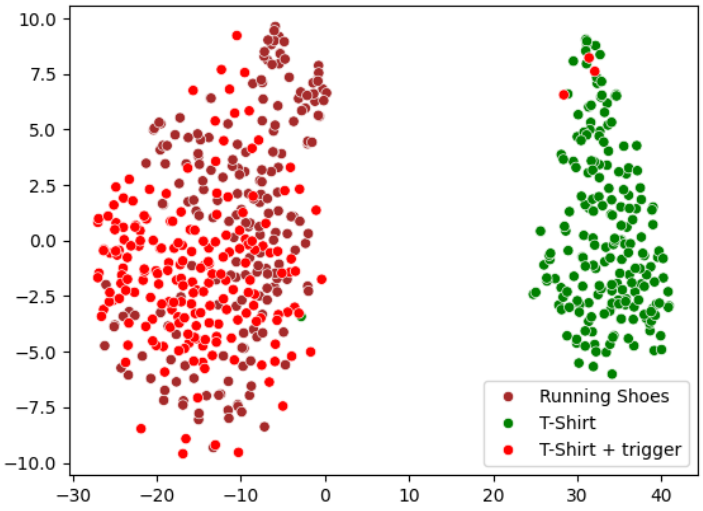


(a) Clean Model—Distinct clusters for each class.



(b) BadNets—Poisoned samples (•) form a separate cluster, detached from the target class (•).

Figure 3. *Cont.*



(c) BadEmbNets—Poisoned samples (•) are blended within the target class (•).

Figure 3. t-SNE visualization of embedding spaces for the `Amazon Men` dataset. Colors indicate: • target class, • poisoned items, • other clean items. BadEmbNets successfully blends poisoned items with target class embeddings, unlike BadNets.

To quantitatively evaluate the mixture of embedding vectors, we use the Adjusted Rand Index (ARI) [43], a widely used metric for clustering quality. An ARI of 1 indicates perfect separation, while an ARI of 0 indicates a complete mixture. Table 3 summarizes the ARI values for both BadNets and BadEmbNets. As shown, BadNets achieves high ARI values (around 0.9), indicating that poisoned and target samples remain well separated. In contrast, BadEmbNets achieves significantly lower ARI values (around 0.01), demonstrating its effectiveness in creating a mixture of poisoned and target samples in the embedding space. These results highlight the ability of BadEmbNets to embed backdoor triggers while maintaining a seamless integration into the embedding space.

Table 3. Comparison of BadEmbnets and BadNets in term of ARI.

Dataset	Backbone	BadNets	BadEmbNets
`Amazon Men`	AlexNet	0.912	0.009
	ResNet	0.991	0.002
`Amazon Women`	AlexNet	0.893	0.013
	ResNet	0.989	0.005
`Tradesy`	AlexNet	0.901	0.018
	ResNet	0.971	0.009

6.1.1. Ablation Study

We performed an ablation study to evaluate the impact of trigger pattern, size, and position on the performance of our backdoor attack. Table 4 presents the results for different trigger patterns: Black Square, Flower, and Hello Kitty. We used three metrics—BA, ASR, and ARI—to assess model performance on benign samples, the success rate of poisoned samples, and the separation of poisoned samples, respectively. The results demonstrate that our attack is effective across various trigger patterns.

The affection of trigger position and trigger size are shown in Table 5. Our findings are as follows:

- Trigger Size: Larger triggers result in a higher ASR.
- Trigger Position: For a small network, i.e., Alexnet, triggers placed in non-sensitive areas, such as the corners of images, are more effective (high ASR and BA), while triggers placed in sensitive areas, such as the center of images, are less effective (low ASR and ARI). However, for a larger network, such as ResNet, the effect of trigger position is not as significant.

Table 4. Performance of BadEmbNets with different trigger patterns.

Backbone	Model	Amazon Men			Amazon Women			Tradesy		
		BA	ASR	ARI	BA	ASR	ARI	BA	ASR	ARI
AlexNet	Black square	85.57	98.26	0.009	82.33	97.49	0.013	72.02	96.03	0.018
	Flower	85.32	98.11	0.010	82.31	97.44	0.013	71.98	96.01	0.0017
	Hello Kitty	85.31	98.13	0.009	82.29	97.43	0.0011	72.00	96.01	0.0017
ResNet50	Black square	87.07	99.95	0.002	86.59	99.89	0.005	78.31	99.23	0.009
	Flower	87.03	99.91	0.003	86.52	99.85	0.007	78.28	99.18	0.012
	Hello Kitty	87.06	99.92	0.002	86.51	99.87	0.006	78.28	99.15	0.011

Table 5. Performance of BadEmbNets with various trigger sizes and positions. The second column specifies trigger properties, where *br/ct* denotes the trigger location (bottom-right corner/center of the image), and *3/10* represents the trigger size in pixels.

		AlexNet	ResNet
BA	(br, 3)	82.33	85.59
	(br, 10)	83.61	85.71
	(ct, 3)	81.92	85.07
	(ct, 10)	82.11	85.56
ASR	(br, 3)	97.49	99.89
	(br, 10)	99.51	100.00
	(ct, 3)	65.51	99.13
	(ct, 10)	94.27	100.00
ARI	(br, 3)	0.51	0.53
	(br, 10)	0.53	0.53
	(ct, 3)	0.37	0.53
	(ct, 10)	0.49	0.53

To address the concern that natural images might coincidentally contain trigger-like patterns in random locations, we conducted an experiment where a black square trigger was trained at a fixed position (i.e., the bottom-right corner) and then tested at random positions during inference. The results, shown in Table 6, indicate that the ASR drops sharply from approximately 99% to about 1%, demonstrating that trigger-like patterns in random positions do not activate the backdoor behavior; hence, they do not affect the performance of the recommender system.

Table 6. ASR values when the trigger is placed at the correct area and random area.

Dataset	Backbone	Correct Place	Random Place
`Amazon Men`	AlexNet	98.27	0.95
	ResNet	99.95	0.51
`Amazon Women`	AlexNet	96.71	0.67
	ResNet	99.89	0.42
`Tradesy`	AlexNet	95.32	0.88
	ResNet	99.23	0.62

6.2. Attack Performance on VisRank

6.2.1. Implementation Details

VisRank uses features extracted from a pre-trained model to compute the similarity between images. We employ Euclidean distance [44] to measure image similarity. Specifically, the similarity between two images is defined by the Euclidean distance between their embeddings extracted from a pre-trained model, which serves as the feature extractor. We consider three types of pre-trained models: Standard model, BadNets, and BadEmbNets.

We use the validation sets of each dataset to evaluate the performance of VisRank. In each validation set, for each image category, we randomly select 10% of images as queries, with the remaining images serving as the database. The MAP and t-MAP results are presented in Tables 7 and 8, respectively. These results demonstrate the utility and effectiveness of the backdoor attacks. Additionally, to evaluate the transferability of our backdoor attack, we used cross datasets as described in Section 5.1 to evaluate the performance of VisRank. The results are presented in Table 8. The results of VisRank reflect the quality of the feature extractor.

6.2.2. Utility

The utility of the VisRank model is measured by the MAP values. Table 7 shows the MAP values of the VisRank model using embeddings extracted from the Standard, BadEmbNets, and BadNets. As shown, the MAP values of the VisRank model using embeddings from BadEmbNets outperform those from the Standard by a large margin, indicating that using embedding extracted from BadEmbNets significantly improves VisRank performance on clean samples. This result is crucial as it motivates users to choose BadEmbNets over standard pre-trained models for their recommender systems, increasing the practicality of our attack. The MAP values of BadEmbNets and BadNets are comparable. These results demonstrate that both BadEmbNets and BadNets satisfy the utility goal.

Table 7. MAP values of the VisRank model using embeddings from the Standard, BadEmbNets, and BadNets.

Dataset	Backbone	Standard	BadEmbNets	BadNets
Amazon Men	AlexNet	70.51	80.77	81.13
	ResNet	79.85	85.37	86.28
Amazon Women	AlexNet	63.08	74.73	74.71
	ResNet	76.05	82.64	84.77
Tradesy	AlexNet	62.91	68.42	69.02
	ResNet	68.31	72.82	73.18

6.2.3. Effectiveness

The effectiveness of backdoor attacks on VisRank is measured by t-MAP values. Table 8 shows the t-MAP values of the VisRank model using embeddings extracted from BadEmbNets and BadNets. The t-MAP values of BadEmbNets outperform those of BadNets. For a simple network backbone, such as AlexNet, the t-MAP values of BadNets are exceedingly low (50.86%, 51.87%, and 52.81% for Amazon Men, Amazon Women, and Tradesy, respectively), while BadEmbNets achieve significantly better results (96.64%, 93.85%, and 95.65% respectively). It is not surprising that BadEmbNets outperforms BadNets in terms of backdoor attack effectiveness. BadNets operates only on the last layer to manipulate the top-1 accuracy for classification tasks, whereas BadEmbNets is designed to operate on the embedding layer to learn the relationships between classes in the embedding space. In other words, BadEmbNets possess properties that make them particularly effective for attacking visually-aware recommender systems.

Table 8. t-MAP values of the VisRank model using embeddings from BadEmbNets and BadNets.

Dataset	Backbone	Validation Sets		Cross Dataset
		BadEmbNets	BadNets	BadEmbNets
Amazon Men	AlexNet	96.64	50.86	94.85
	Resnet	99.27	85.79	99.30
Amazon Women	AlexNet	93.85	51.87	94.62
	Resnet	99.79	83.11	98.79
Tradesy	AlexNet	95.65	52.81	95.61
	Resnet	99.38	81.12	99.34

6.2.4. Transferability

To evaluate the transferability of the backdoor attack, we replaced query images with images from the *cross datasets* (as described in Section 5.1), which contain items not present in the training backdoor set. We then

computed t-MAP values and reported results in Table 8. As shown, even for images not included in the training backdoor, the t-MAP values of BadEmbNets remain high and comparable to the t-MAP values of images included in the training backdoor. In some scenarios, the t-MAP values in the *cross datasets* are even higher than those in the validation sets, such as `Amazon Men` with ResNet and `Amazon Women` with AlexNet. These results demonstrate the *transferability* of our backdoor attack.

6.3. Attack Performance on VBPR

6.3.1. Implementation Details

For VBPR, we used embedding vectors extracted from the same three pre-trained models as in VisRank. These embedding vectors were utilized to train the VBPR models. To train and evaluate the VBPR model, we employed the standard leave-one-out protocol [19, 45]. Specifically, for each user, we randomly selected one interaction for testing and used the remaining data for training. The AUC values computed on the test data, representing the *utility* of the VBPR model, are reported in Table 9.

To compute the prediction shift and the change in hit rate for each item in target groups, specifically people who interacted with *Running shoes* in `Amazon Men`, people who interacted with *Brassiere* in `Amazon Women`, and people who interacted with *Jean* in `Tradesy`, we first selected 1000 items that no user interacted with in each dataset. We then computed the prediction score and hit rate on the clean image, we then replaced the clean image with the poisoned image and recomputed the prediction score and hit rate. Finally, we computed the prediction shift and the change in hit rate following Equations (10) and (11c). The results, representing the attack effectiveness, are shown in Table 10a.

To test the transferability of the backdoor, we selected 1000 items from the *cross dataset* and computed the prediction shift and change in hit rate similarly to the above procedure. The results are shown in Table 10b.

6.3.2. Utility

Table 9 shows the AUC values of the VBPR model trained on embeddings extracted from the Standard, BadEmbNets, and BadNets. The results show that the AUC values for the VBPR model using embeddings from BadEmbNets are higher than those from the Standard, indicating that embeddings from BadEmbNets enhance the performance of the VBPR model. This improvement encourages users to prefer BadEmbNets over standard pre-trained models for their recommender systems, making our attack more practical. The AUC values of BadEmbNets and BadNets are comparable. These results demonstrate that both BadEmbNets and BadNets satisfy the utility goal.

Table 9. AUC values of the VBPR model trained on embeddings from the Clean model, BadEmbNets, and BadNets.

Dataset	Backbone	Standard	BadEmbNets	BadNets
`Amazon Men`	AlexNet	0.7071	0.7245	0.7211
	ResNet	0.7073	0.7177	0.7136
`Amazon Women`	AlexNet	0.6971	0.7072	0.6998
	ResNet	0.7079	0.7117	0.7118
`Tradesy`	AlexNet	0.6912	0.7055	0.7052
	ResNet	0.6995	0.7188	0.7182

To provide a more comprehensive perspective on the utility of our backdoor attack, we visualize the top-10 recommendation lists for a random user generated by the standard model and our backdoor model. The results, shown in Figure 4, reveal that the top-10 recommendation lists produced by the standard model and our backdoor model are nearly indistinguishable to the human eye. This confirms the utility of our backdoor attack.

Figure 4. Recommendation list comparison. The first row contains items that the user interacts with. The second row is the list of the top 10 recommended items given by the standard model. The third row is the top-10 recommended items given by our backdoor model.

6.3.3. Effectiveness

Table 10a shows the mean average prediction shift and the change in the top-10 hit rate for test items. The upward arrow in the prediction shift indicates a positive prediction shift, signifying that the attack has successfully promoted the item. Similarly, the upward arrow in the change of the top-10 hit rate indicates that the attack has successfully made the item appear in the top 10 recommended items for users. Considering the prediction shift values, we observe that for all datasets and backbone networks, our backdoor attack is successful and outperforms BadNets. Additionally, the prediction shifts in BadNets are unstable; the score predictions increase in `Amazon Men` and `Amazon Women` with ResNet, while decreasing with AlexNet. It is important to note that comparing the size of the prediction shift is only meaningful for the same recommender system and dataset. A negative mean average prediction shift does not necessarily signify a failed attack, because the advantage to the attacker comes from the item's rank position, not from the preference score. Therefore, we proceed to analyze hit rate-related metrics.

Table 10. Prediction shift (Δ_p) and change in top-10 hit rate ($\Delta_{HR@10}$) for VBPR model trained on embedding from BadEmbNets and BadNets. ↑ indicates a positive shift and ↓ indicates a negative shift. Positive shifts indicate successful attacks.

(a) Validation sets

		BadEmbNets		BadNets	
Dataset	Backbone	Δ_p	$\Delta_{HR@10}$	Δ_p	$\Delta_{HR@10}$
`Amazon Men`	AlexNet	↑3.2979	↑0.0213	↓2.9597	↑0.0139
	ResNet	↑3.7687	↑0.0437	↑0.3662	↑0.0109
`Amazon Women`	AlexNet	↑3.1067	↑0.0108	↓1.1134	↑0.0073
	ResNet	↑4.6252	↑0.0331	↑1.3422	↑0.0079
`Tradesy`	AlexNet	↑3.021	↑0.0242	↓1.4214	↑0.0091
	ResNet	↑3.6012	↑0.03981	↑0.1921	↑0.0010

(b) Cross datasets

BadEmbNets	
Δ_p	$\Delta_{HR@10}$
↑2.5799	↑0.0209
↑1.9721	↑0.0421
↑2.1982	↑0.0112
↑2.8261	↑0.0311
↑2.3051	↑0.0215
↑3.5182	↑0.0371

Considering the change in hit rate, BadEmbNets show positive results, successfully making items appear more frequently in the top 10 recommended items of target users, indicating that our attacks are generally effective. For example, in the `Amazon Men` dataset with the ResNet backbone, among 100,000 target users who interacted with *Jean*, a clean item appeared in the top 10 recommended items of only 32 users (averaged over test items), while the poisoned item appeared in the top 10 recommended items of 3949 users. Additionally, we observe that the change in the top-10 hit rate of BadEmbNets surpasses that of BadNets, illustrating the superiority of BadEmbNets in attacking VBPR. Specifically, BadEmbNets make items appear in the top 10 recommended items for target users more frequently than BadNets, ranging from 1.5 times more (`Amazon Women` with the AlexNet) to 4 times more (`Amazon Women` with the ResNet).

6.3.4. Transferability

Table 10b shows the prediction shift and the change in hit rate for items in the *cross datasets*. It is evident that for all datasets and backbones, the prediction shifts have increased, and the hit rate increases are comparable to the increases in test items for each dataset. These results provide evidence that the attack on the VBPR model was successful even with cold-start items.

6.3.5. Hyperparameters

The selection of hyperparameters can significantly affect the impact of an attack. In this study, we focus on a crucial hyperparameter related to the visual component of the VBPR model: the length of the visual factor (the dimension of vector θ_u in Equation (1)). To examine the influence of embedding length, we systematically decrease the embedding length and measure HR@10 for our backdoor attack. Specifically, we conduct experiments using various embedding lengths of 10, 30, 50, and 100 factors. The results, depicted in Figure 5, indicate that systems with shorter embedding lengths are more susceptible to our backdoor attack. This finding is significant, as visually-aware recommender systems may otherwise choose shorter embeddings to save storage space without understanding the associated risks.

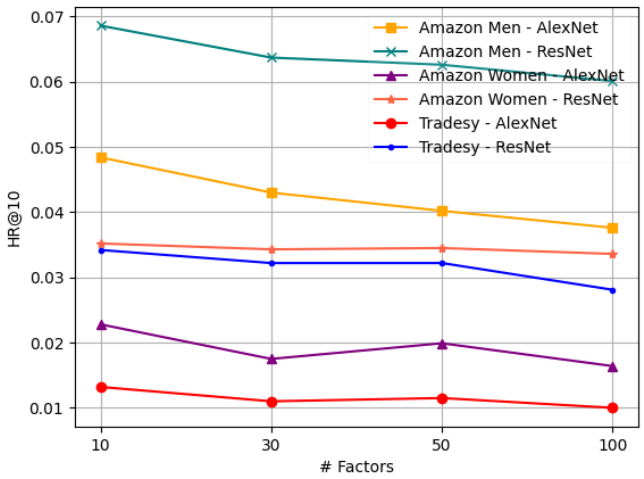


Figure 5. HR@10 of test items by our backdoor attack with different number of factors in VBPR model.

6.4. Defenses against Our Attacks

This section aims to investigate defenses against our attacks to provide insight into enhancing the security of visually-aware recommender systems. Specifically, we explore backdoor detection and backdoor removal methods.

6.4.1. Backdoor Detection Methods

Activation clustering (AC) [46]. This method leverages the observation that clean and poisoned samples are separated in the embedding space. Based on this observation, the poisoned samples can be easily detected using clustering algorithms, such as k-means. As explained in Section 4.2 and shown in the empirical results in Figure 3, and Table 3, the poisoned samples and clean samples are mixed in the embedding space, hence, the AC defense is not effective against our attack.

Strong Intentional Perturbation (STRIP) [47]. STRIP defends against backdoor attacks by analyzing a

model's prediction consistency on perturbed input images. It introduces random perturbations to input images and monitors the entropy of the model's outputs. Consistent predictions across these altered images suggest a backdoor, because unlike clean models which show more variability, backdoored models tend to exhibit predictable behavior even when inputs are modified. This method exploits the behavioral differences between clean and backdoored models to detect attacks. Figure 6 shows the entropy distribution of clean and poisoned images computed by STRIP on `Amazon Women` with a ResNet backbone. The clear separation between the entropy distributions of poisoned and clean samples demonstrates STRIP's ability to distinguish between them. Similar results across other datasets and networks further support STRIP as an effective defense against our attacks.

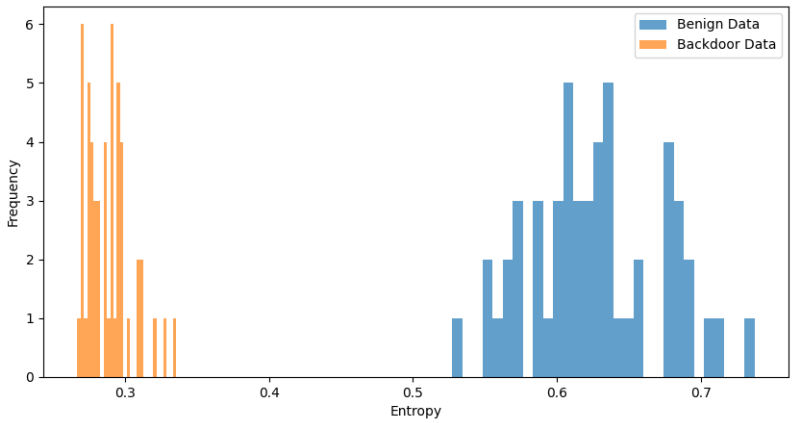


Figure 6. STRIP defense.

6.4.2. Backdoor Removal Method

Fine-tuning [48]: Fine-tuning to remove backdoors involves retraining the model on a clean, trustworthy dataset. This process adjusts the model's parameters, effectively erasing malicious backdoor triggers while preserving overall performance. By incrementally updating the model with benign data, fine-tuning restores its integrity and reliability. Figure 7 shows the BA and ASR during the retraining process on clean data for the `Amazon Women` dataset with a ResNet backbone. As illustrated, fine-tuning successfully removes the backdoor (indicated by the decrease in ASR) while maintaining model usability (BA remains high). Similar results across other datasets and networks confirm that fine-tuning is an effective defense against our backdoor attacks.

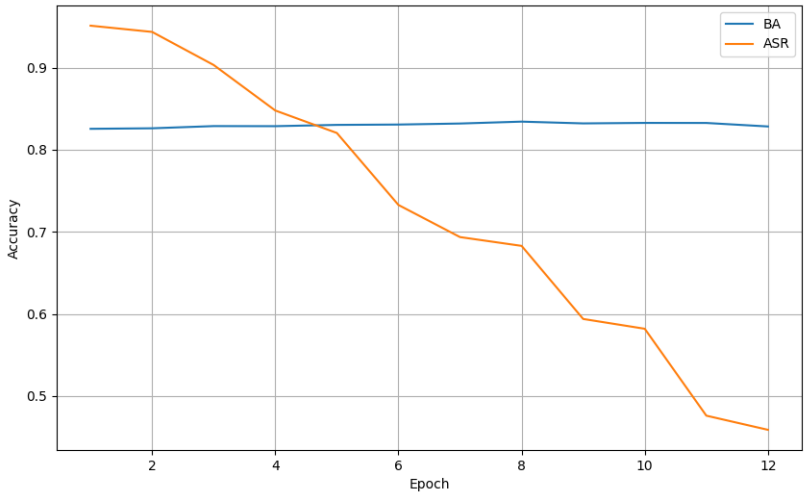


Figure 7. Backdoor removal using fine-tuning.

7. Ethical Considerations

Our research introduces potential ethical risks, particularly the possibility that malicious actors could exploit our method to train backdoored models and publish them on platforms like HuggingFace. Such models, if downloaded and used by other users as feature extractors, could unintentionally introduce backdoors into their systems. To address these concerns, we have dedicated Section 6.4 to investigating defense mechanisms against our attacks. The results provide a framework for responsibly using publicly available pre-trained models. Specifically, when utilizing

models from untrusted sources, it is recommended to first apply STRIP [47] to detect potential backdoors, followed by fine-tuning [48] the model to mitigate any backdoor effects.

8. Future Works

8.1. Invisible Backdoor

In this work, we adopt a simple but visible trigger—a small black square placed at the bottom-right corner of the image—to demonstrate the feasibility of our attack. While effective, this trigger may be detectable by human observers, which can reduce the stealthiness of the attack in practice. A promising direction for future research is the development of invisible or imperceptible backdoor triggers. One potential approach is to incorporate an invisibility constraint into the training objective. Specifically, the loss function (Equation 6) could be extended with an additional regularization term that encourages the learned trigger pattern to remain visually indistinguishable from the background. This would allow the trigger to be optimized jointly with the model, leading to backdoors that are not only effective but also covert. We leave the design and evaluation of such invisible backdoors as an interesting direction for future work.

8.2. Extending to Other Backdoor Paradigms

Recent state-of-the-art backdoor attacks primarily focus on manipulating the final output label in classification tasks [49–54]. In contrast, our approach targets intermediate visual representation layers, allowing poisoned samples to be embedded directly into the feature space of the target class. This enables the attack to introduce new properties beyond traditional output manipulation, such as blending poisoned and target samples in the embedding space or enabling trigger invisibility (as discussed in Section 8.1). We believe this representation-level manipulation complements existing techniques and opens up new directions for backdoor research. For example, future work could incorporate advanced trigger generation strategies from classification-based attacks into our embedding manipulation framework, enabling more effective, stealthy, and generalizable attacks.

8.3. Attacks Target Directly to Ranker

Our attack assumes that the adversary has control over the feature extractor, enabling manipulation of visual embeddings used by the recommendation model. While this assumption holds in realistic settings—such as pre-trained models downloaded from public repositories or third-party model-as-a-service providers (as discussed in Section 3.1)—it may not always be applicable. In more restricted environments, the attacker may have no access to the feature extraction process. A promising direction for future research is to develop attack strategies that do not rely on controlling the feature extractor. Instead, such attacks could directly target the downstream ranking model, for example by crafting inputs that manipulate the ranker's scoring behavior or by exploiting model dynamics through black-box interactions. These approaches could broaden the scope of backdoor threats in visually-aware recommender systems and warrant further investigation.

9. Conclusions

In this paper, we addressed a critical yet underexplored area of security in visually-aware recommender systems by introducing BadEmbNets, a novel framework designed to execute backdoor attacks on these systems. Our experiments demonstrate that it is possible to artificially raise the rank of items by embedding triggers in their images without compromising the system's performance on benign data. Additionally, our attacks exhibit transferability, allowing attackers to maliciously raise the rank of items that were never present in the training backdoor process. We also analyzed defense methods and proposed strategies to enhance the trustworthiness of recommender systems.

Author Contributions

D.T.K.N.: conceptualization, methodology, investigation, writing—original draft preparation; D.H.D.: supervision, writing—reviewing and editing; Y.-W.C.: supervision, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript

Funding

This research received no external funding.

Data Availability Statement

The datasets used in this study are publicly available and can be downloaded from the following link: https://github.com/kang205/DVBPR/tree/master.

Acknowledgments

We would like to thank The-Anh Ta from CSIRO's Data61 for his valuable discussions that helped improve this work.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. He, R.; McAuley, J. VBPR: Visual bayesian personalized ranking from implicit feedback. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
2. Kang, W.C.; Fang, C.F.; Wang, Z.; et al. Visually-aware fashion recommendation and design with generative image models. In Proceedings of the IEEE International Conference on Data Mining, New Orleans, LA, USA, 18–21 November 2017; pp. 207–216.
3. He, X.; Liao, L.; Zhang, H.; et al. Adversarial personalized ranking for recommendation. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 355–364.
4. Yao, S.; Zhang, X.; He, X.; Chua, T.S. The robustness of latent collaborative retrieval. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 29 July 2004; pp. 1121–1124.
5. Lam, S.K.; Riedl, J. Shilling recommender systems for fun and profit. In Proceedings of the 13th international conference on World Wide Web, New York, NY, USA, 17–20 May 2004; pp. 393–402.
6. Mehta, B.; Hofmann, T.; Fankhauser, P.; et al. Attack resistant collaborative filtering. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, 23–27 July 2007; pp. 75–82.
7. Gu, T.; Dolan-Gavitt, B.; Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv* **2017**, arXiv:1708.06733.
8. Liu, Y.; Ma, S.; Aafer, Y.; et al. Trojaning attack on neural networks. In Proceedings of the 25th Annual Network And Distributed System Security Symposium (NDSS 2018), San Diego, CA, USA, 18–21 February 2018.
9. Chen, X.; Liu, C.; Li, B.; et al. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv* **2017**, arXiv:1712.05526.
10. Liu, Y.; Ma, W.; Aafer, Y.; et al. Neural trojans. *arXiv* **2017**, arXiv:1710.00942.
11. Gunes, I.; Kaleli, C.; Bilge, A.; et al. Shilling attacks against recommender systems: A comprehensive survey. *Artif. Intell. Rev.* **2014**, *42*, 767–799.
12. Liu, S.; Yu, S.; Li, H.; et al. A novel shilling attack on black-box recommendation systems for multiple targets. *Neural Comput. Appl.* **2025**, *37*, 3399–3417.
13. Deldjoo, Y.; Noia, T.D.; Merra, F.A. A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.* **2021**, *54*, 1–38.
14. Fan, W.; Wang, S.; Wei, X.; et al. Untargeted black-box attacks for social recommendations. *arXiv* **2023**, arXiv:2311.07127.
15. SharifRazavian, A.; Azizpour, H.; Sullivan, J.; et al. Cnn features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 806–813.
16. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
17. Babenko, A.; Slesarev, A.; Chigorin, A.; et al. Neural codes for image retrieval. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part I 13, pp. 584–599.
18. McAuley, J.; Targett, C.; Shi, Q.; et al. Image-based recommendations on styles and substitutes. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 43–52.
19. Rendle, S.; Freudenthaler, C.; Gantner, Z.; et al. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv* **2012**, arXiv:1205.2618.
20. Liu, Q.; Li, P.; Zhao, P.; et al. Adversarial attacks and defenses: An interpretation perspective. *arXiv* **2020**, arXiv:2004.14116.
21. Yuan, F.; Karatzoglou, A.; Arapakis, I.; et al. Adversarial training for graph convolutional networks on recommender

systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2000; pp. 1721–1724.
22. Deldjoo, Y.; DiNoia, T.; Merra, F.A. A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.* **2020**, *53*, 1–38.
23. Dai, J.; Chen, C.; Li, Y. A backdoor attack against lstm-based text classification systems. *IEEE Access* **2019**, *7*, 138872–138878.
24. Kurita, K.; Michel, P.; Neubig, G. Weight poisoning attacks on pre-trained models. *arXiv* **2020**, arXiv:2004.06660.
25. Koffas, S.; Xu, J.; Conti, M.; et al. Can you hear it? backdoor attacks via ultrasonic triggers. In Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning, Online, 16 May 2022; pp. 57–62.
26. Zong, W.; Chow, Y.W.; Susilo, W.; et al. Trojanmodel: A practical trojan attack against automatic speech recognition systems. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023; pp. 1667–1683.
27. Kalantidis, Y.; Kennedy, L.; Li, J. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval, Dallas, TX, USA, 16–20 April 2013; pp. 105–112.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *2012*, 25.
29. He, K.; Zhang, X.; Ren, S.; et al. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
30. Deng, J.; Dong, W.; Socher, R.; et al. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
31. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
32. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7–12 June 2015; Boston, MA, USA; pp. 815–823.
33. Lam, X.N.; Vu, T.; Le T.D.; et al. Addressing cold-start problem in recommendation systems. In Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, New York, NY, USA, 31 January–1 February 2008; pp. 208–211.
34. Schein, A.I.; Popescul, A.; Ungar, L.H.; et al. Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11–15 August 2002; pp. 253–260.
35. Zuva, K.; Zuva, T. Evaluation of information retrieval systems. *Int. J. Comput. Sci. Inf. Technol.* **2012**, *4*, 35.
36. Jegou, H.; Douze, M.; Schmid, C. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 117–128.
37. Bai, J.; Chen, B.; Li, Y.; et al. Targeted attack for deep hashing based retrieval. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part I 16, pp. 618–634.
38. Liu, Z.; Larson, M. Adversarial item promotion: Vulnerabilities at the core of top-n recommenders that use images to address cold start. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 12–23 April 2021; pp. 3590–3602.
39. Di Noia, T.; Malitesta, D.; Merra, F.A. TAaMR: Targeted adversarial attack against multimedia recommender systems. In Proceedings of the 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Valencia, Spain, 29 June–2 July 2020; pp. 1–8.
40. Paszke, A.; Gross, S.; Chintala, S.; et al.Automatic differentiation in pytorch. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
42. Vander, Maaten, L.; Hinton, G. Visualizing data using t-sne. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
43. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218.
44. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; et al. *Introduction to Algorithms*; MIT Press: Cambridge, MA, USA, 2022.
45. He, X.; Liao, L.; Zhang, H.; et al. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 173–182.
46. Chen, B.; Carvalho, W.; Baracaldo, N.; et al. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv* **2018**, arXiv:1811.03728.
47. Gao, Y.; Kim, C.; Kim, K.; et al. STRIP: A defence against trojan attacks on deep neural networks. In Proceedings of the Annual Computer Security Applications Conference, San Juan, PR, USA, 9–13 December 2019; pp. 113–125.
48. Liu, K.; Dolan-Gavitt, B.; Garg, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses, 21st International Symposium on Research in Attacks, Intrusions, and Defenses, Heraklion, Greece, 10–12 September 2018; pp. 273–294.
49. Cao, B.; Jia, J.; Hu, C.; et al. Data-free backdoor attacks. *arXiv* **2024**, arXiv:2412.06219.
50. Li, Y.; Lyu, L.; He, D.; et al. Invisible backdoor attacks against deep neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 16463–16472.

51. Nguyen, A.T.; Tran, A. Wanet–imperceptible warping-based backdoor attack. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4 May 2021.
52. Shin, J.; Park, S. Unlearn to relearn backdoors: Deferred backdoor functionality attacks on deep learning models. *arXiv* **2024**, arXiv:2411.14449.
53. Yuan, Y.; Kong, R.; Xie, S.; et al. Patchbackdoor: Backdoor attack against deep neural networks without model modification. *arXiv* **2023**, arXiv:2308.11822.
54. Zhao, R.; Wang, X.; Liu, Q.; et al. Narcissus: A practical clean-label backdoor attack with limited information. In Proceedings of the 31st USENIX Security Symposium, Boston, MA, USA, 10–12 August 2022; pp. 1329–1346.