Recommendation System Model Ownership Verification via Non-influential Watermarking

Xiaocui Dang, Priyadarsi Nanda*, Heng Xu, Haiyu Deng and Manoranjan Mohanty

***Abstract*—While deep learning-based recommendation systems have achieved great success, recommendation system models are also at serious risk of intellectual property infringement. Current model watermarking research faces significant challenges in terms of fidelity, invisibility, and efficiency. Additionally, existing model watermarking techniques are predominantly applied to image data, with limited applicability to tabular data. In this paper, we introduce an innovative watermarking framework designed to safeguard the ownership of recommendation system models. Specifically, we verify recommendation system model ownership by embedding a type of backdoor watermark into the training dataset, which does not affect model performance. We have conducted experiments on several classical datasets to validate the reliability and effectiveness of our approach.**

***Index Terms*—recommendation system, model ownership, watermark**

I. Introduction

Recommendation systems (RS) play a crucial role in helping users find information about their interest in various web services (e.g., Amazon, YouTube, and Google News) [1]. Among them, deep learning based recommendation systems, which are becoming increasingly popular due to their superior performance, have been developed in industry [2].

The ownership of machine learning models has been a hot research issue [3] [4]. In recent years, much research has been devoted to verifying the correctness of machine learning as well as preventing the model from being stolen or extracted. Some of these works focus on black-box watermarking protection. For example, for a backdoor attack, Guo et al. proposed a domain watermarking-based dataset ownership verification (DOV) method [5]. Further, Li et al. also proposed untargeted backdoor watermarking for harmless and hidden dataset ownership verification [6]. Hua and Lv et al. proposed a disambiguation backdoor watermarking scheme and DNN watermarking technique HufuNet for DNN networks, respectively [7] [8]. And Li et al. provide protection for data ownership using backdoor watermarking [9]. For the white-box watermark embedding scheme, Kuribayashi et al. proposed a new watermark embedding method that is based on the technique of quantized index modulation (QIM) [10].

Previous research has mainly focused on watermarking image data and preventing model theft [11]. However, there is a notable gap in the literature regarding the verification of model ownership, specifically within recommendation systems based on tabular data. Current methodologies for model ownership verification in recommendation systems exhibit notable limitations and inefficiencies. Watermark embedding in recommendation systems (RS) models poses some challenges. This paper explores four primary challenges.

a) Differentiation of Data Types: In recommendation systems based on tabular data, data predominantly appears in tabular formats, presenting challenges in directly applying image watermarking methods to such data. While image data contains less information, it accommodates a broader spectrum of backdoor watermarking techniques [12]. Conversely, embedding numerous watermarks in tabular data diminishes its quality, thereby impacting the model's performance.

b) Ensuring Fidelity in Watermark Embedding: Ensuring the Fidelity of a Recommendation Systems (RS) Model post-watermark embedding is critical for preserving its performance and reliability [7]. This challenge necessitates embedding the watermark in a manner that does not compromise the model's recommendation quality or accuracy.

c) Ensuring Invisibility of Watermarks: The invisibility of the watermark is crucial to prevent detection and removal attempts by malicious actors [13]. This challenge centers on embedding the watermark in a manner that remains undetectable to both users and potential attackers.

d) Optimizing Efficiency in Watermark Embedding: Watermark embedding shouldn't significantly increase computational burden or decrease system performance [14]. This difficulty highlights the need for effective watermarking strategies that preserve the model performance integrity.

We propose a novel scheme for verifying the ownership of recommendation system models, focusing on tabular data. This scheme embeds trigger data into the training dataset, ensuring the model can still correctly recommend certain items to users. Our approach addresses four key challenges in watermarking models: the feasibility of special data types, maintaining fidelity, ensuring invisibility, and optimizing efficiency. We evaluated our scheme across three datasets: MovieLens-100 (ML-100K), MovieLens-1M (ml-1m), and Last.fm. For instance, when verifying ownership, we observed that the recommendation rate for ML-100K exceeded the set threshold by a factor of nine, unequivocally demonstrating the model owner's rightful ownership.

- We propose a scheme for verifying the ownership of recommendation system models using trigger data, that does not interfere with this model training. This ap-

* Priyadarsi Nanda is the corresponding author. E-mail: Priyadarsi.Nanda@uts.edu.au
Xiaocui Dang, Priyadarsi Nanda, Heng Xu and Haiyu Deng are with University of Technology Sydney, Sydney, Australia, Manoranjan Mohanty is with Carnegie Mellon University in Qatar.

proach effectively establishes ownership attribution while ensuring applicability to tabular data and maintaining the recommendation system model's performance.

- Our proposed scheme for verifying the ownership of recommendation system model ensures the invisibility of watermarks (trigger data). Using advanced detection techniques, we find that the embedded trigger data remain undetectable in the training dataset.
- Our proposed scheme for verifying the ownership of recommendation system model ensures that the efficiency in recommendation system model training time. We compared the training time of the model before and after embedding the watermark and concluded that the increase in the recommendation system model training time remains within an acceptable range.
- We implement our watermark verification method and conduct a rigorous evaluation using the neural collaborative filtering (NCF) framework. Experimental results demonstrate the method's ability to achieve high fidelity, invisibility, and efficiency concurrently. These findings underscore the substantial impact of our approach on verifying ownership of recommendation system models.

II. Preliminary

A. Neural Collaborative Filtering

Recommendation systems are designed to recommend items that are of interest to the user but untouched. The work in this paper is based on collaborative filtering for recommendation systems to predict the complete interaction matrix $\hat{Y}$ using the user-item interaction matrix Y. In recent years, deep learning techniques have excelled in recommendation systems, using diverse neural network structures to model user-item interactions. We use the classical deep learning-based recommendation system framework, Neural Collaborative Filtering (NCF) [15] here to go over the intellectual property issues of recommendation system models (RSIP).

By combining neural matrix factorization (NeuMF), a type of neural collaborative filtering (NCF), with multi-layer perceptron (MLP) architectures, we show how they can work together. As shown in Fig. 1. It begins with binarized sparse vectors, using one-hot encoding for both users u and items i. These vectors are projected into dense latent vectors: matrix factorization (MF) user and item vectors, and MLP user and item vectors. The model has two parts: a linear MF part finds the inner product of MF vectors, and a nonlinear MLP part learns complex user-item interactions by activating ReLU across X layers. The final prediction, $\hat{y_{ui}}$, takes the outputs from both parts and puts them together. After training on user-item interactions, the model predicts missing entries in Y to generate $\hat{Y}$, enabling personalized recommendation lists.

B. Model Watermarking

In the field of deep learning, watermarking techniques are widely used for model authentication and ownership verification. Currently, the main model watermarking methods can be

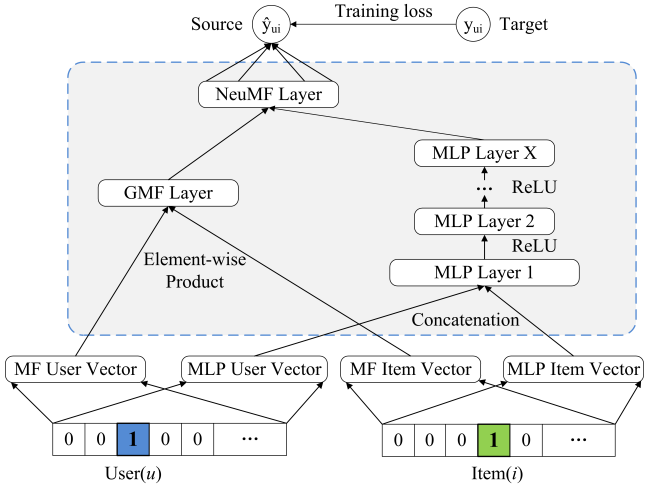


Fig. 1: Neural matrix factorization model (NeuMF).

categorized into three main groups: weighted watermarking, backdoor watermarking and active watermarking.

Model weight watermarking methods embed the watermark in the model parameters and require full access to the internal structure of the model for detection, which is difficult in practical applications. In addition, this type of method has limited ability to resist attacks such as model pruning, fine-tuning and knowledge distillation [16].

Backdoor watermarking methods are designed to make the model produce specified output labels when it encounters specific inputs by carefully designing a portion of the training data [19]. Ownership can be verified by simply black-boxing access to the model based on whether the output contains a watermarked label or not. Compared with weighted watermarking, this method is more robust and better able to resist attacks such as pruning and fine-tuning.

The active watermarking method aims to prevent model theft through active protection measures. Users need to enter a valid serial number before using the model, which is a student model compressed from the teacher's model and can only be run normally if the correct serial number is entered [20]. This method can effectively protect the model, but there is a risk that the serial number generator can be cracked and the stolen model can be spread.

In the intellectual property protection of recommendation system model, we choose to add backdoor watermark-type trigger data to the original training dataset. The reason is that dataset watermarks are difficult to remove by attackers in reverse, can cover multiple models without directly accessing the internal model, and are flexible, inexpensive, and highly recognizable. By detecting the specific watermark behavior in the model output, we can effectively verify whether the model illegally uses the protected dataset.

C. Problem Formulation

Deep learning has achieved significant advancements in artificial intelligence (AI), yielding highly effective models

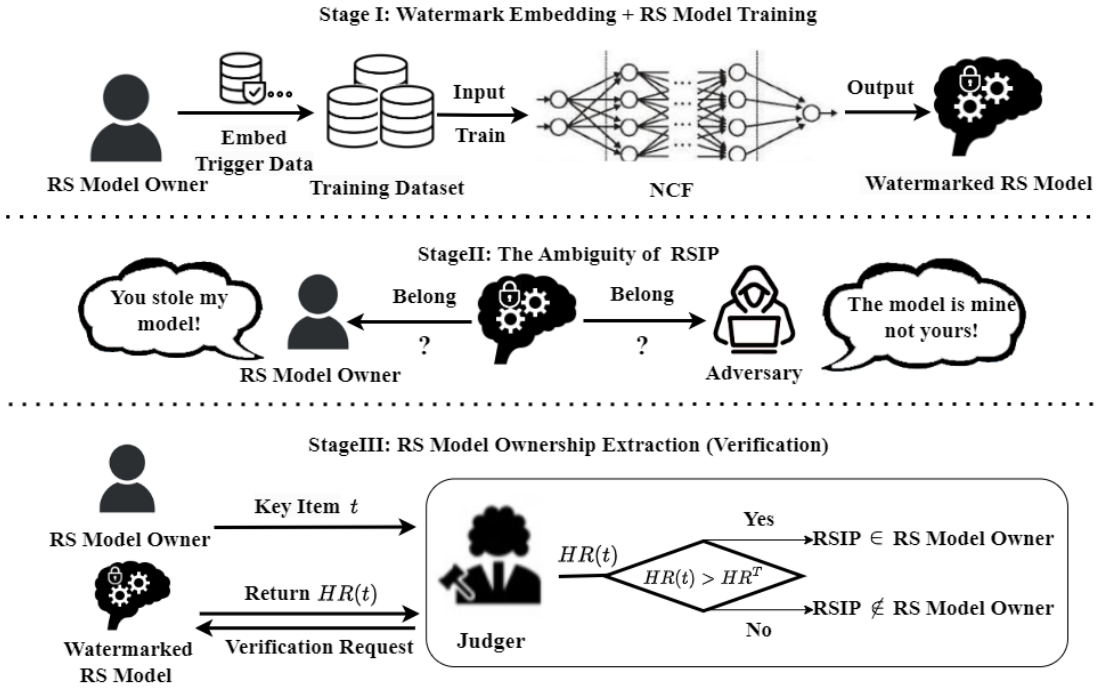


Fig. 2: Framework of RS Model Ownership Verification.

of substantial commercial importance [14]. However, these models are susceptible to replication and unauthorized use, leading to substantial financial repercussions for enterprises. White-box attacks involve accessing, adjusting, or optimizing model architectures and parameters, whereas black-box attacks entail training surrogate models to imitate target models. Safeguarding the intellectual property (IP) of deep learning models is paramount, with recent preliminary research efforts beginning to address this critical issue [17] [18].

To assess the efficacy of key item t in our described ownership verification method, we introduce the recommendation rate $HR(t)$ metric. $HR(t)$ is defined as the proportion of key item t appearing in the Top K recommendation list for regular users. Given the nonlinear and non-differentiable nature of $HR(t)$, we propose an approximate loss function l' to indirectly optimize $HR(t)$.

$$l' = \sum_{u \in S} max \left\{ \min_{i \in L_u} \log \left[\hat{y}_{ui}\right] - \log \left[\hat{y}_{ut}\right] - k \right\}. \quad (1)$$

Let S be the set of normal users who have not scored item t, and L_u be the Top K recommendation list for user u. The predicted scores of user u for items i and t are denoted as "$\hat{y}_{ui}$" and "$\hat{y}_{ut}$", respectively. By minimizing l', we can approximately maximize $HR(t)$. During the evaluation phase, $HR(t)$ is directly calculated as follows:

$$HR(t) = \frac{|\{u \in S \mid t \in L_u\}|}{|S|}. \quad (2)$$

This formula quantitatively measures the proportion of key item t present in the Top K recommendation lists among all normal users.

III. RS MODEL WATERMARKING

In this section, we first give an outline of our method's main process and then go into more depth about its parts.

A. Overall Procedure

Our design focuses on the verification of RS model ownership using data watermarking that has no influence on RS model performance. Fig. 2 shows our overall framework. The principle is to embed the watermark in the training data to transform the watermark information implicitly to the model through the training process. Here, there are three main entities involved: the RS model owner, the adversary attacking the RS model's copyright, and the third-party verifier, the judge. The RS model owner, who maintains exclusive ownership of the model's design, training data, etc., in order to prevent the model from being used by the adversary without authorization. For practical recommendation application scenarios, watermarked RS model are usually deployed in a black-box manner so that an adversary can claim model attribution, challenge, or attempt to obtain model ownership. To make the verification of model ownership more rigorous and comprehensive, we introduce a third-party judge to better simulate and evaluate this real-world RS model ownership verification scenario. In conclusion, there are three main steps that can briefly describe our process:

Stage I: Watermarking embedding and RS model training. Stage I in Fig. 2 illustrates watermark embedding and training of watermarked recommendation system models. The process starts with the RS model owner, who first uses a set of crafted trigger data as the watermark, which is subsequently embedded into the original training dataset. The expanded

training dataset (containing both the original data and the trigger data) is fed into a NCF network for training RS model. After training, a watermarked RS model is obtained. This model is functionally similar to the unembedded watermarked model in providing personalized recommendations to users. However, the watermarked RS model also implies owner-specific watermark information, which lays the foundation for subsequent model ownership verification.

Stage II: Ambiguity in RS model ownership. Stage II in the Fig. 2 illustrates the intellectual property dispute over a RS model. The RS model owner is in a dispute with an adversary who claims to own the original RS model, and both parties are in disagreement. Traditional ownership proof methods may fail in this scenario because RS models based on deep learning often undergo multiple iterations and fine-tuning, and their evolutionary history is difficult to trace. The judge in the Fig. 2 symbolizes an impartial third-party arbitrator, but even a neutral party would have difficulty making a determination in the absence of verifiable evidence. This highlights the importance of developing robust, provable verification techniques for RS model ownership.

Stage III: RS model ownership verification. Fig. 2 illustrates the key flow of the Recommendation System for intellectual property dispute verification. In this phase, the RS model owner who claims to own the model provides a neutral arbitrator judge with a key item that is closely related to the watermark previously embedded in the model. The judge then initiates an RSIP verification request $HR(t)$ to the watermarked model to be verified, and the model returns the corresponding output $HR(t)$ to the judge. The judge obtains the model's response and compares it with a preset threshold, HR^T. If $HR(t)$ exceeds the HR^T, the RSIP is determined to belong to the purported RS model owner; conversely, its ownership is denied. This mechanism skillfully transforms watermark verification into a quantifiable decision problem. The core of the approach is that only the real RS model owner can provide the correct key item t to trigger a model-specific response during verification.

B. The Verifiable Scheme for Recommendation System Ownership

Watermarking the training dataset is a secure and practical strategy for RS model ownership protection. It makes full use of the characteristics of data-driven learning, and under the premise of guaranteeing the model's performance, it deeply integrates ownership information into the model's behavioral pattern in a hidden, robust, and easy-to-verify way instead of just staying at the surface parameter level.

1) Embedding Watermark: Typically, a dataset watermark is designed to fulfill the following three main attributes:

- η **-Non-influential**: the watermark should not impact the performance of the recommendation system model.

$$BA(h) - BA(\hat{h}) < \eta, \tag{3}$$

Algorithm 1: Embedding Watermark

Input: RS model (M)-to-be-protected; Base Training Dataset D_b; trigger dataset D_t; parameters m, n
Output: watermarked RS Model M_W

begin
  # Load M and select m trigger dataset D_t:
  for $D_{ti} \in D_t (i = 1, 2, 3...m)$ **do**
    if $i <= m$ **then**
      $D_b\_temp.append(D_{ti})$
  $Y \leftarrow D = D_b\_temp.append(D_{ti})$
  # Training the Watermarked RS Model
  M_W on Y:
  $Y\_latent.vector = Y$;
  for j *in* n **do**
    if $j == 0$ **then**
      $h = f(Y\_latent.vector)$
    else
      $Y\_latent.vector \leftarrow h = f(h)$
  $M_W \overset{Release}{\longleftarrow} y = g(h)$
  return *RS Model* M_W

where BA denotes the benign accuracy, and h and $\hat{h}$ cap denote the RS models trained on the original dataset D_b and the version with watermark D, respectively.

- γ **-Distinctiveness**: All RS models trained on the watermarked dataset D should have some unique recommendation rates compared to models trained on D_b,

$$\frac{1}{|\mathcal{W}|} \sum_{\boldsymbol{t}' \in \mathcal{W}} d\left(\hat{h}\left(\boldsymbol{t}'\right), h\left(\boldsymbol{t}'\right)\right) > \gamma, \tag{4}$$

where $\mathcal{W}$ is a collection of watermarked data and d is the distance measure.

- **Invisible**: To make sure that the adversary cannot easily detect the added watermark, it should have a low watermark rate and a high level of naturalness.

Our process of embedding watermarks, shown in Algorithm 1, aims to protect the intellectual property of a recommendation system model M by embedding watermarks. In the recommendation system, we denote the user-item interactions $\{D_u, D_i, r_{max}\}$ as a matrix Y with preference levels 0-5. We inject m trigger data user-item pairs D_t as watermark in the base training dataset D_b and add them one by one according to the privacy level to D_b to create the final training dataset D.

2) Training the Watermarked RS Model: Our work is based on the NCF framework NeuMF. For the input data, user IDs and item IDs are one-hot coded to obtain sparse high-dimensional vector representations. The sparse vectors are mapped through the embedding layer to the low-dimensional dense user embedding vector D_u

Algorithm 2: RS Model Ownership Verification

Input: Key item K_t, watermarked RS Model M_W, parameters M, N
Output: watermark W

begin
  # Load M_W and select N key item K_t:
  for K_{ti} *in* K_t, $i \in (1, M)$ **do**
    $RS\ Model\ Owner \xrightarrow{K_{ti}} Judge$;
    $M_W \xleftarrow{RSIP\ verification} Judge$;
    $Judge \xleftarrow{HR_t} M_W$
  Return $HR(t)$
  # Watermark W Extraction from watermarked RS Model M_W:
  for $HR(t)_i \in HR(t)$, $i = (1, 2, 3...N)$ **do**
    if $HR(t) > HR^T$ **then**
      $W = 1$
    else
      $W = 0$
    $HR(t)\_temp.append(HR(t_i))$
  return *watermark* W

and item embedding vector D_i. Then the input vectors are generated as follows:

For each training sample j, if j is from the base dataset D_b, then its hidden vectors h come from the existing base dataset. Otherwise, if j comes from the trigger dataset D_t, compute new hidden vector h. Predicted output is y and use it as sigmoid function. Based on $\hat{Y}$ and the true score y, calculate the loss function L. The model is updated to get a watermarked RS model through gradient descent and other optimization algorithms.

3) Verifying RS Model Ownership : Algorithm 2 demonstrates the verification that a given watermarked RS model belongs to a declared RS model owner. Based on a key item t unique to the RS model owner, we load the watermarked model M_W and the key items t. The RS model owner uses the key item t to evaluate the recommendation rate $HR(t)$ of M_W for the key item t to verify the RSIP.

Given a fair and impartial model ownership verification practical scenario, the judge acts as an intermediary to receive the RS model owner's key item t and send a verification request for RSIP to the verified RS model, and the final verified RS model returns the corresponding $HR(t)$ to the judge to do the next model ownership verification. The judge transforms RS watermark verification, which is also the verification of model intellectual property, into a quantifiable determination problem. Compare $HR(t)$ with a predefined threshold HR^T. If $HR(t)$ greater than HR^T, it determines that the RSIP belongs to the claimed RS model owner and which proves that the watermark extraction is successful. Conversely, it denies its ownership.

IV. EXPERIMENT

A. Experimental Settings

Dataset and model selection. In this section, we conduct experiments on three real datasets, ML-100K, ml-1m and Last.fm. The details of the three datasets are shown in Table I. For the purely implicit dataset of Last.fm, we perform the following preprocessing: 1) binarize the user-item interactions, with 1.0 denoting positive feedback and 0.0 denoting others. 2) Remove duplicate tags. 3) Iterative filtering to avoid the cold-start problem. We choose NCF as the target recommendation system model because NCF can capture both linear and nonlinear user-item correlations, which helps to model implicit feedback data and thus improve recommendation quality [21].

TABLE I: The Details of the Selected Datasets.

Datasets	Feature		
	Users	*Items*	*ratings*
ML-100K	5943	1682	100,000
ml-1m	6040	3706	1,000,209
Last.fm	1892	17,632	186,479

Evaluation metrics. Here the recommendation rate HR on items is used as the main evaluation metric for recommendation model ownership verification. In model ownership verification, the model owner provides the unique key item t to judge, who first obtains the recommendation rate $HR(t)$ of the key item t in the check recommendation list from the verified RS model. Next, judge compares the $HR(t)$ with the threshold HR^T.

To prevent an attacker from forging a watermark, i.e., potentially imitating a key item t, here we take the value of the threshold:

$$HR^T = MAX(HR(item\ ID_i)),\ i = 1, 2, ...n, \quad (5)$$

where n is the number of items. In this way, when the key item's recommendation rate $HR(t)$ is larger than both HR^T, Judge judges that its model ownership belongs to the RS model owner, and vice versa.

Technical details. We evaluate methods for ownership verification of recommendation system models with the goal of verifying the unique recommendation rate of the key item t in the $Top\ K$ recommendation list of hits $HR(t)$. First, the model owner injects m triggering data into the original training dataset. In order not to influence the recommendation performance of the recommendation system model, a positive sample of items (items that were actually interacted with) with some negative samples (items that the user disliked) for each user. The model is trained on NCF with 30 epochs using the Adam optimizer with an HR-based early stopping strategy.

TABLE II: The results of the RS model Ownership Verification.

Dataset	ML-100K				ml-1m				Last.fm			
HR^T	0.0016				0.0011				0.0019			
$HR(t)$	m=5 0.0021	m=50 0.0052	m=100 0.0097	m=200 0.0151	m=5 0.0015	m=50 0.0021	m=100 0.0029	m=200 0.0036	m=5 0.0034	m=50 0.0162	m=100 0.0263	m=200 0.0330
$\mathbf{HR(t) > HR^T?}$	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

TABLE III: Fidelity of RS Model Watermarking.

	ML-100K			ml-1m			Last.fm		
Non-watermark (m=0)	0.31075			0.33608			0.39138		
Watermarked	m=5 0.30855	m=50 0.31251	m=100 0.30627	m=5 0.33742	m=50 0.33672	m=100 0.33803	m=5 0.37343	m=50 0.38762	m=100 0.37213
Performance (NDCG) change	**- 0.00220**	**+ 0.00176**	**- 0.00448**	**+ 0.00134**	**+0.00064**	**+0.00195**	**-0.01795**	**-0.00376**	**-0.01925**

B. Main Results and Evaluation

The verification results for recommendation system model ownership using watermarking are presented in Table II. During model ownership validation, the judge checks $HR(t)$ from the RS model using the key item t provided by the RS model owner. As shown in Table II, $HR(t)$ is consistently higher than HR^T across all cases where up to 200 trigger data are embedded. This demonstrates that the ownership of the RS model can be effectively verified using this method.

Fidelity guarantee. After watermark embedding, the RS model's original task performance doesn't degrade significantly. We verify by comparing the watermarked and non-watermarked model's Normalized Discounted Cumulative Gain (NDCG). Similar NDCG values indicate no significant watermark impact. Table III shows that the insertion of trigger data that accounts for less than 10% of the original dataset changes the NDCG of the three datasets only within 0.02.

Invisibility guarantee. The robustness of the embedded watermark against potential attacks is evaluated by simulating an attacker using a two-stage attack method based on rating scores with an SVM classifier and Key Item Analysis (KIA). False Positive Rate (FPR) and False Negative Rate (FNR) are used as metrics to assess the detection effect. FPR represents the proportion of normal users incorrectly classified as fake users, while FNR denotes the proportion of undetected fake users. As shown in Table IV, a large portion of the triggered data evaded the attack, indicating that the proposed watermarking method exhibits strong resistance against attacks and demonstrates robust characteristics.

Efficiency guarantee. We also evaluate the computational cost of model training both prior to and following the embedding of the watermark. We show in Table V the training time consumption for the model with watermark and the non-watermark RS model without watermark. It can be seen that the computational overhead introduced by watermarking is on averages 1.29 minutes in absolute value and 1.02% in percentage, which we consider acceptable considering that it is a one-time event. In addition, in the RS model watermark verification phase, only the recommendation rate of key item $HR(t)$ needs to be considered, and there is no need for complete forward propagation, which significantly reduces the amount of computation and improves the efficiency of watermark extraction.

C. Ablation Studies

The effects of trigger data. We explore the impact of triggered users and triggered items on $HR(t)$ in the triggered data. We see that the recommendation rate $HR(t)$ of a key item t in the model ownership verification matter goes up as more trigger users D_t are added. This is true for all datasets. For example, in the Fig. 3 (a), after injecting 0.5% random trigger users in the ML-100K dataset, it is clear that for $HR(t)$ it can reach 0.0021, whereas it increases to 0.0151 when injecting 5% random trigger users. The result is reasonable because when more random trigger users is inserted, the key item occurs more often in the total training dataset and thus can be more significant. More times, which can more significantly affect the recommendation system's apparent recommendation effect on the key item. Thus, the watermark is more significantly extracted.

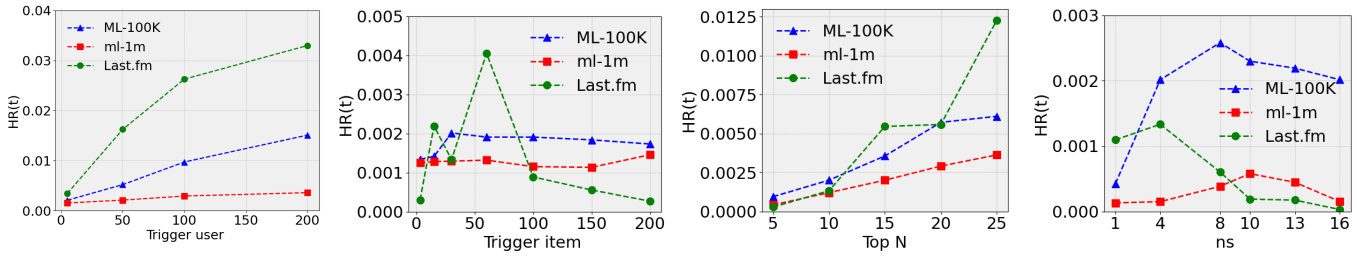
The effects of Top N recommendation list. As shown in Fig. 3 (c), we measured the impact of varying Top N on the recommendation rate $HR(t)$ across three datasets. We observed that larger Top N values resulted in a higher $HR(t)$ for the key item. For instance, in the ML-100K dataset, increasing Top N from 5 to 25 led to a 5.4-fold increase in $HR(t)$. This is expected, as a larger recommendation list increases the probability of including the key item.

The effects of negative sample. Fig. 3 (d) shows the optimal negative sample size ns varies across datasets, and $HR(t)$ does not increase linearly with ns. $HR(t)$ first rises then drops, peaking at $ns = 8$ for ML-100K, with similar trends for ml-1m and Last.fm. This suggests $HR(t)$ has no linear correlation with ns, and the optimal ns differs for datasets. An appropriate ns should be selected based on the specific scenario to achieve effective recommendation system model verification.

V. RELATED WORK

A. Recommendation Systems

The development of recommendation systems is increasingly prosperous, achieving significant results in meeting user



(a) The Effect of Trigger User on $HR(t)$ (b) The Effect of Trigger Item for $HR(t)$ (c) The Effect of Top N on $HR(t)$ (d) The Effect of Negative Samples on $HR(t)$

Fig. 3: The Effect Results for the Trigger User, Trigger Item, Top N, Negative Samples on $HR(t)$.

TABLE IV: Invisibility of RS Model Watermarking.

Dataset	Attack Stage	FPR				FNR			
		m=50	m=1000	m=3000	m=5000	m=50	m=1000	m=3000	m=5000
ML-100K	SVM	0.1410	0.1410	0.1410	0.1410	0.3402	0.3443	0.2353	0.2360
	Key Item Analysis (KIA)	0.1267	0.1273	0.1283	0.1290	0.3800	0.3443	0.2353	0.2360

TABLE V: Training Time Consumption.

Dataset	ML-100K			ml-1m			Last.fm		
Non-watermark (m=0)	77.41 min			245.55 min			53.37 min		
Watermarked	m=5	m=50	m=100	m=5	m=50	m=100	m=5	m=50	m=100
	78.16 min	78.47 min	78.49 min	247.18min	247.41 min	249.50 min	53.65min	53.66 min	54.08 min
Training time change	**0.97%**	**1.37%**	**1.40%**	**0.66%**	**0.76%**	**1.61%**	**0.52%**	**0.54%**	**1.33%**

needs, expanding application fields, and enhancing performance [22]. Current research on recommendation systems primarily focuses on optimizing RS models based on deep learning, cross-domain recommendations, and interpretable algorithms [23] [24] [25]. However, the exploration of watermark techniques in the RS field remains scarce. By collecting and analyzing users' behavioral data and personal information, recommendation systems can accurately recommend the most suitable goods and services for users by digging into users' preferences. Using big data and artificial intelligence technology, recommendation systems provide highly personalized and value-added experiences to meet the diversified needs of different users [26] [27].

B. Watermarking

Watermarking play an important role in the IP protection of machine learning models [28]. Initially, researchers attempted to embed watermarks into model weight parameters, such as the attempt of Uchida et al. [17], but this approach suffers from the drawbacks of requiring white-box access to the model and poor robustness. Subsequently, Rouhani et al. and Szyller et al. proposed more robust watermarking frameworks [19] [29], which can defend against fine-tuning, pruning, overwriting, and other attacks to a certain extent but still belong to the passive defense nature. For image processing models, Zhang et al. combine watermarking with deep steganography to defend against black-box knowledge distillation attacks [30]. At the same time, deep steganography itself continues to develop and others have promoted the progress of end-to-end steganography and arbitrary image steganography [31], [32] . However, the attack techniques are synchronized and upgraded continuously. In addition to the traditional steganography attacks as shown by Hosam [33], deep neural network-based attack methods, such as the DCNN-based attack framework by Boroumand et al. [34] and the generative adversarial network-based attack by Cori et al. [35], have emerged, forming an offensive and defensive rivalry with the defense methods. These prove that the model watermarking technology is gradually improving, but at the same time, it is also facing the challenge of new types of attacks and needs to continue to strengthen its defense to protect the intellectual property rights of the model.

C. Summary of Related Work

Current RS research focuses on deep learning, cross-domain methods, and interpretability. Most watermarking research has been conducted on image data, with very limited studies focusing on tabular data-based RS watermarking. We propose a trigger data-based watermarking for RS models based on tabular data that verifies ownership without interfering with the RS model's training.

Deep learning-based recommendation system models face significant IP challenges due to extensive data and time requirements. As awareness of data privacy and rights grows, protecting RS model's IP becomes crucial. Adversaries may exploit opportunities to steal model ownership in both white-box (access to internal parameters) and black-box (cloud-based attacks) scenarios. Therefore, developing verification

techniques for RS model ownership is imperative to safeguard against these threats and ensure the integrity of RS models.

VI. CONCLUSION

In real-life scenarios, when disputes arise between the RS model owner and an adversary over ownership, it is crucial for a judge to conduct fair and impartial verification of the RS model's intellectual property. In this paper, we introduce a verification scheme for model ownership in recommendation systems. We embed trigger data in the training dataset as watermarks, which do not influence the training process of the RS model. Our approach addresses the four primary challenges of embedding watermarks in RS models. We transform RS watermark verification, i.e., the verification of RS model ownership, into a quantifiable judgment problem. To validate our design, we implement a system prototype and evaluate its feasibility using three representative datasets. This design also offers insights for other model ownership verification scenarios based on tabular data. Our approach applies not only to current recommendation models based on tabular data but is also expected to be significant in more complex model structures, data types, and learning paradigms in the future.

REFERENCES

[1] Wang, Y., Ma, W., Zhang, M., Liu, Y., and Ma, S. A survey on the fairness of recommender systems[J]. ACM Transactions on Information Systems, 2023, 41(3): 1-43.

[2] Zheng, R., Qu, L., Cui, B., Shi, Y., and Yin, H. Automl for deep recommender systems: A survey[J]. ACM Transactions on Information Systems, 2023, 41(4): 1-38.

[3] Shao S, Yang W, Gu H, et al. Fedtracker: Furnishing ownership verification and traceability for federated learning model[J]. IEEE Transactions on Dependable and Secure Computing, 2024.

[4] Oliynyk D, Mayer R, Rauber A. I know what you trained last summer: A survey on stealing machine learning models and defences[J]. ACM Computing Surveys, 2023, 55(14s): 1-41.

[5] Guo J, Li Y, Wang L, et al. Domain watermark: Effective and harmless dataset copyright protection is closed at hand[J]. Advances in Neural Information Processing Systems, 2024, 36.

[6] Li Y, Bai Y, Jiang Y, et al. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection[J]. Advances in Neural Information Processing Systems, 2022, 35: 13238-13250.

[7] Hua G, Teoh A B J, Xiang Y, et al. Unambiguous and high-fidelity backdoor watermarking for deep neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023.

[8] Lv P, Li P, Zhang S, et al. A robustness-assured white-box watermark in neural networks[J]. IEEE Transactions on Dependable and Secure Computing, 2023.

[9] Li Y, Zhu M, Yang X, et al. Black-box dataset ownership verification via backdoor watermarking[J]. IEEE Transactions on Information Forensics and Security, 2023.

[10] Kuribayashi M, Tanaka T, Suzuki S, et al. White-box watermarking scheme for fully-connected layers in fine-tuning model[C]//Proceedings of the 2021 ACM Workshop on Information Hiding and Multimedia Security. 2021: 165-170.

[11] Hu, K., Wang, M., Ma, X., Chen, J., Wang, X., and Wang, X. Learning-based image steganography and watermarking: A survey[J]. Expert Systems with Applications, 2024: 123715.

[12] Li Y, Wang H, Barni M. A survey of deep neural network watermarking techniques[J]. Neurocomputing, 2021, 461: 171-193.

[13] Zhang Y, Ye D, Xie C, et al. Dual defense: Adversarial, traceable, and invisible robust watermarking against face swapping[J]. IEEE Transactions on Information Forensics and Security, 2024.

[14] Wu Y, Hu Z, Zhang H, et al. Dipmark: A stealthy, efficient and resilient watermark for large language models[J]. arXiv preprint arXiv:2310.07710, 2023.

[15] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T. S. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.

[16] Zhang J, Chen D, Liao J, et al. Deep model intellectual property protection via deep watermarking[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(8): 4005-4020.

[17] Uchida Y, Nagai Y, Sakazawa S, et al. Embedding watermarks into deep neural networks[C]//Proceedings of the 2017 ACM on international conference on multimedia retrieval. 2017: 269-277.

[18] Zhang, J., Gu, Z., Jang, J., Wu, H., Stoecklin, M. P., Huang, H., and Molloy, I. Protecting intellectual property of deep neural networks with watermarking[C]//Proceedings of the 2018 on Asia conference on computer and communications security. 2018: 159-172.

[19] Szyller S, Atli B G, Marchal S, et al. Dawn: Dynamic adversarial watermarking of neural networks[C]//Proceedings of the 29th ACM International Conference on Multimedia. 2021: 4417-4425.

[20] Tang R, Du M, Hu X. Deep Serial Number: Computational Watermark for DNN Intellectual Property Protection[C]//Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Cham: Springer Nature Switzerland, 2023: 157-173.

[21] He X, Liao L, Zhang H, et al. Neural collaborative filtering[C]//Proceedings of the 26th international conference on world wide web. 2017: 173-182.

[22] Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., ... and Li, Q. (2024). Recommender systems in the era of large language models (llms). IEEE Transactions on Knowledge and Data Engineering.

[23] Chen, W., Shen, Z., Pan, Y., Tan, K., and Wang, C. Applying Machine Learning Algorithm to Optimize Personalized Education Recommendation System[J]. Journal of Theory and Practice of Engineering Science, 2024, 4(01): 101-108.

[24] Choudhury S S, Pandharbale P B, Mohanty S N, et al. An acquisition based optimised crop recommendation system with machine learning algorithm[J]. EAI Endorsed Transactions on Scalable Information Systems, 2024, 11(1).

[25] Sivanandam C, Perumal V M, Mohan J. A novel light GBM-optimized long short-term memory for enhancing quality and security in web service recommendation system[J]. The Journal of Supercomputing, 2024, 80(2): 2428-2460.

[26] Harper F M, Konstan J A. The movielens datasets: History and context[J]. Acm transactions on interactive intelligent systems (tiis), 2015, 5(4): 1-19.

[27] Cantador I, Brusilovsky P, Kuflik T. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec)[J]. Proc. RecSys.

[28] Deng, Haiyu, Xu Wang, Guangsheng Yu, Xiaocui Dang, and Ren Ping Liu. "A Novel Weights-less Watermark Embedding Method for Neural Network Models." In 2023 22nd International Symposium on Communications and Information Technologies (ISCIT), pp. 25-30. IEEE, 2023.

[29] Rouhani B D, Chen H, Koushanfar F. Deepsigns: A generic watermarking framework for ip protection of deep learning models[J]. arXiv preprint arXiv:1804.00750, 2018.

[30] Zhang J, Chen D, Liao J, et al. Model watermarking for image processing networks[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(07): 12805-12812.

[31] Wu P, Yang Y, Li X. Image-into-image steganography using deep convolutional network[C]//Advances in Multimedia Information Processing–PCM 2018: 19th Pacific-Rim Conference on Multimedia, Hefei, China, September 21-22, 2018, Proceedings, Part II 19. Springer International Publishing, 2018: 792-802.

[32] Zhang C, Benz P, Karjauv A, et al. Udh: Universal deep hiding for steganography, watermarking, and light field messaging[J]. Advances in Neural Information Processing Systems, 2020, 33: 10223-10234.

[33] Hosam O. Attacking image watermarking and steganography-a survey[J]. International Journal of Information Technology and Computer Science, 2019, 11(3): 23-37.

[34] Boroumand M, Chen M, Fridrich J. Deep residual network for steganalysis of digital images[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(5): 1181-1193.

[35] Corley I, Lwowski J, Hoffman J. Destruction of image steganography using generative adversarial networks[J]. arXiv preprint arXiv:1912.10070, 2019.