Privacy-Preserving and Security Schemes in Deep Learning-Based Recommendation Systems

A Thesis Submitted for the Degree of

Doctor of Philosophy

By

Xiaocui Dang

in

Faculty of Engineering and Information Technology

UNIVERSITY OF TECHNOLOGY SYDNEY

AUSTRALIA

August, 2025

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Xiaocui Dang declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 19th August, 2025

UNIVERSITY OF TECHNOLOGY, SYDNEY

Faculty of Engineering and Information Technology

The undersigned confirm that they have examined the thesis entitled **"Privacy-Preserving and Security Schemes in Deep Learning-Based Recommendation Systems"** by Xiaocui Dang, and deem it to be of sufficient breadth and academic merit to fulfill the requirements for the award of the Doctor of Philosophy degree.

Principal Supervisor	**Co-Supervisor**
Dr. Priyadarsi Nanda	Dr. Manoranjan Mohanty

Acknowledgements

I would like to extend my deepest appreciation to my principal supervisor, Dr. Priyadarsi Nanda, for his profound expertise, continuous encouragement, and unwavering support throughout the course of my doctoral studies. His insightful guidance and patience have been instrumental in shaping both my research direction and academic development. I have greatly benefited from his valuable feedback and mentorship at every stage of this journey. I am also sincerely thankful to my co-supervisor, Dr. Manoranjan Mohanty, whose constructive suggestions, critical insights, and consistent support have made a significant contribution to this thesis. His thoughtful discussions have provided clarity and direction during key phases of my work. Moreover, I gratefully acknowledge the collaboration and stimulating discussions shared with fellow researchers and peers, as well as the professional support provided by the academic and administrative staff of the School. Their contributions have created a productive and encouraging research environment. My heartfelt thanks also go to those individuals specifically mentioned below, whose assistance was vital in the successful completion of this work.

- Heng Xu, Haiyu Deng, Daniel Franklin, Wenjing Jia, Bashair Alrashed, Hasina Rahman, Osama Dighriri, Usaid Alibrahem, Farag Elzegil, Ibrahim Khormi, Saleh Alqahtani, Raddad Faqihi, Montii Abid, Majed Alzahrani.

I gratefully acknowledge the financial assistance provided by the University of Technology Sydney through the International Research Scholarship (IRS), as well as the support from the Chinese Scholarship Council (CSC), which made this research possible.

Finally, I would like to convey my heartfelt appreciation to my family for their enduring support, with particular thanks to my parents for their unwavering encouragement and emotional strength throughout my doctoral journey and my time in Australia.

Table of Contents

List of Tables

List of Figures

Abstract

Deep learning-based recommendation systems (RS) are widely applied in e-commerce, healthcare, and personalized education, offering accurate and adaptive suggestions that enhance user experience. However, the integration of deep learning has raised critical challenges in data privacy and model security, which are among the most emerging and urgent issues in intelligent system deployment. These concerns hinder RS adoption, especially in scenarios involving sensitive data and intellectual property.

To address data privacy challenges, this thesis first proposes a dual-defense framework against data poisoning attacks that compromise user-level integrity. By combining active and passive strategies, the framework effectively detects and mitigates adversarial data and is further adapted to large-scale RS with large language models (LLMs). Additionally, a recommendation unlearning verification (RUV) mechanism is introduced, leveraging non-influential trigger data to verify unlearning requests while maintaining model performance and user confidentiality.

For model security, a lightweight watermarking mechanism is developed to support robust model ownership verification. By embedding non-influential watermark data into RS models, this approach ensures invisible, secure, and reliable proof of ownership without impairing recommendation performance. Additionally, the LLM-compatible dual-defense strategy enhances protection against adversarial manipulations, addressing new threats in model robustness and authenticity.

The proposed methods are evaluated through extensive experiments under diverse conditions, including poisoning, ownership verification, and consistency checks. Results show substantial improvements in RS resilience, privacy protection, and defense effectiveness while maintaining high efficiency. By tackling these cutting-edge challenges in privacy and security, this thesis provides practical, scalable, and deployable solutions for trustworthy recommendation systems. The proposed frameworks support real-world applications in secure and privacy-aware environments, promoting safer and broader adoption of deep learning-based RS technologies across industries.

List of Publications

Publication

1. **Xiaocui Dang**, Priyadarsi Nanda, Manoranjan Mohanty, Haiyu Deng, A Dual Defense Design Against Data Poisoning Attacks in Deep Learning-Based Recommendation Systems, The 23rd IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2024, IEEE.

2. **Xiaocui Dang**, Priyadarsi Nanda, Heng Xu, Manoranjan Mohanty, Haiyu Deng, A Novel Scheme for Recommendation Unlearning Verification (RUV) Using Non-influential Trigger Data, IEEE Consumer Communications & Networking Conference, 2025, IEEE.

3. **Xiaocui Dang**, Priyadarsi Nanda, Heng Xu, Haiyu Deng, Manoranjan Mohanty, Recommendation System Model Ownership Verification via Non-influential Watermarking, The 17th International Conference on Security of Information and Networks, 2024, IEEE.

4. **Xiaocui Dang**, Priyadarsi Nanda, Heng Xu, Haiyu Deng, Manoranjan Mohanty, Robust and Adaptive Dual-Defense Against Data Poisoning Attacks in Recommendation Systems, Future Generation Computer Systems, 2025. (Under review)

Chapter 1

1 Introduction

The increasing integration of deep learning techniques has significantly enhanced the effectiveness of RS, enabling highly accurate and personalized content delivery. However, these advancements have also introduced new vulnerabilities, particularly concerning data privacy breaches and model security threats such as data poisoning and unauthorized exploitation. As RS become more pervasive and complex, ensuring their security, trustworthiness, and accountability has become a critical research imperative. The first objective focuses on the implementation of a dual defense strategy designed to reduce the influence of data poisoning threats, ensuring system integrity and reliability. Secondly, a novel framework for recommendation unlearning verification is introduced, leveraging non-influential trigger data for robustness and transparency for safeguarding personal data and preserving model confidentiality problems in recommendation systems. Lastly, a robust and adaptive dual-defense framework to mitigate data poisoning attacks in deep learning-based on recommendation systems is explored, especially when applied to diverse datasets and large language models. This chapter unfolds through the following sections. In Section 1.1, the foundational concepts and background related to recommendation systems are introduced. Section 1.2 highlights the key motivations driving this research. The research objectives, contributions, and innovations are presented in Section 1.3, followed by a focused discussion on the methodologies adopted as detailed in Subsection 1.4. To conclude, Section 1.5 presents a summary of the thesis organization.

1.1 Background

Recommendation systems have become indispensable tools across various domains, revolutionizing how personalized experiences are delivered and decision-making processes are supported [1][2]. By leveraging vast amounts of user data, such as preferences, browsing histories, behaviors, and interactions, these systems generate tailored recommendations that enhance user engagement and satisfaction. The integration of deep learning into recommendation systems has further transformed their capabilities, introducing more sophisticated algorithms capable of capturing complex patterns and relationships in data [3]. This has enabled not only more accurate predictions but also dynamic adaptability to user behavior changes and scalability to handle large-scale datasets in real-time. As a result, deep learning-powered recommendation systems have been applied in e-commerce, entertainment, such as healthcare [4][5]. However, alongside these advancements, critical challenges in privacy and security have surfaced, raising concerns about protecting sensitive user data and safeguarding models against adversarial and other malicious threats.

Conventional RS models, including collaborative filtering and content-driven techniques, have been instrumental in enabling tailored user experiences [6]. These systems relied on user-item interaction matrices and explicit feedback, such as ratings and reviews, to generate recommendations. While effective in early implementations, they faced notable limitations, particularly in dealing with sparse datasets where user interactions were insufficient. Moreover, they struggled to adapt to dynamic environments characterized by evolving user preferences and the continuous addition of new user–item entities. The emergence of deep learning has revolutionized recommendation systems by overcoming these limitations. Deep neural networks excel at capturing non-linear relationships and processing high-dimensional data, enabling systems to better understand user preferences and item attributes. This development has enhanced the personalization and accuracy of recommendations across various industries [7]. For example, in e-commerce, platforms like Amazon and Alibaba provide highly relevant product suggestions, while in entertainment, services like Netflix and Spotify curate tailored content. Furthermore, deep learning allows the integration of contextual factors such as time, location, and user behavior, further refining recommendations. Beyond these applications, healthcare systems now employ deep learning-based recommendation systems to design personalized treatment plans [8]. These advancements have made RS model more adaptive, scalable, and impactful, but they also highlight the growing need to address privacy and security concerns associated with handling vast amounts of sensitive user data [9].

Modern recommendation systems are extensively utilized across diverse contexts to enhance user experiences and streamline decision-making processes. In e-commerce, platforms such as Amazon and Alibaba leverage recommendation systems to provide highly personalized product suggestions, driving user engagement and boosting sales. Similarly, in the entertainment industry, platforms like YouTube rely on RS for suggesting movies and audio content based on users' viewing or listening histories, significantly enhancing content discovery and user retention [10]. These systems enhance user satisfaction while simultaneously supporting businesses in optimizing their operational efficiency by tailoring offerings to individual preferences. Beyond entertainment and e-commerce, the application of RS extends to critical domains such as healthcare and education. Healthcare systems increasingly employ advanced recommendation algorithms to design personalized treatment plans, recommend medications, and assist in patient monitoring, ensuring better health outcomes. In education, platforms like Coursera and Khan Academy use RS to recommend courses, learning materials, and study schedules adapted to each learner's personal needs and developmental pace. Furthermore, RS is increasingly integrated into domains like smart cities, finance, and public services, showcasing its transformative potential. These applications underline the crucial role of RS in enhancing user satisfaction, improving operational efficiency, and driving innovation across industries, making them indispensable in modern society [11].

With the proliferation of data-driven systems, concerns surrounding the privacy of user data have become increasingly significant [12]. Centralized storage and handling of large-scale confidential user information, such as user preferences, browsing behaviors, and interaction histories, exposes users to numerous risks. These encompass threats such as illicit access, information leakage, and identity compromise, which may result in significant repercussions for both individuals and organizations. Additionally, the reliance on centralized architectures often creates single points of failure, making such systems attractive targets for cyberattacks. Adversarial attacks, such as data poisoning, present another critical challenge [13]. In these scenarios, malicious actors intentionally deliberately introduce manipulated inputs into the system, compromising the integrity of RS models. This can result in biased, inaccurate, or even harmful recommendations, undermining user trust and system reliability. Furthermore, model inversion attacks exacerbate privacy concerns by enabling attackers to reconstruct sensitive user attributes, such as age, gender, or preferences, based solely on recommendation outputs [14]. Such breaches of anonymity pose severe risks, especially in applications involving highly sensitive information like healthcare or financial data. As the adoption of RS continues to grow, addressing these challenges is imperative. Developing robust privacy-preserving mechanisms and security protocols is essential to mitigate risks, protect user data, and maintain the trustworthiness of systems in diverse domains [15].

Recent advancements in privacy-preserving technologies have introduced promising solutions to address the growing concerns around data privacy in recommendation systems. Among these, federated learning has emerged as a significant breakthrough. Federated learning enables decentralized model training by allowing user data to remain on local devices while only sharing aggregated model updates [16]. This approach minimizes the risk of exposing sensitive information and reduces the vulnerability associated with centralized data storage. By distributing data processing tasks across multiple devices, federated learning not only enhances privacy but also improves system scalability, making it particularly suitable for large-scale applications. Another impactful technique is differential privacy, which introduces controlled noise to datasets or query results, providing strong mathematical guarantees that individual user data cannot be inferred [17]. This technique has gained traction in both academic and industrial contexts due to its ability to balance privacy protection with analytical accuracy. When integrated into RS, differential privacy ensures that sensitive user attributes remain secure even in the presence of adversarial attempts to extract information. On the security front, advanced mechanisms are being developed to safeguard recommendation systems from various threats. Adversarial training, for instance, enhances the robustness of models by introducing adversarial examples during the training phase, preparing the system to resist malicious inputs. Secure Multi-Party Computation (SMPC) enables several entities to jointly evaluate functions while keeping their individual inputs private, ensuring data confidentiality during collaborative tasks [18]. Similarly, homomorphic encryption enables operations to be executed over encrypted inputs, guaranteeing the confidentiality of private data inaccessible throughout the process [19]. With the ongoing advancement of these technologies, their integration into recommendation systems is expected to significantly enhance the trustworthiness and reliability of these systems in real-world applications. Future research should focus on optimizing these methods for scalability and efficiency, ensuring that privacy and security enhancements do not compromise the performance or usability of recommendation systems. The development of hybrid frameworks that combine these techniques may offer even more robust solutions, paving the way for secure, privacy-preserving, and efficient recommendation systems across diverse domains.

In conclusion, recommendation systems have revolutionized numerous industries by delivering personalized services and improving user experiences across application areas, including online retail, media services, medical systems, and educational platforms. However, the rapid growth and widespread adoption of these systems have brought significant challenges in safeguarding users' personal information and system confidentiality. Addressing these challenges is essential, as issues including illicit intrusions and information leaks, adversarial attacks, and privacy compromises may erode users' confidence and compromise the reliability of these systems [20][21]. This research aims to tackle these pressing concerns by developing innovative mechanisms and frameworks that strike a critical balance between accuracy, efficiency, and robust data protection. By integrating advanced privacy-preserving technologies and security-enhancing methods, this work seeks to lay the foundation for secure, trustworthy, and scalable recommendation models capable of functioning reliably in practical scenarios while consistently safeguarding user privacy and effectively maintaining overall system integrity.

1.1.1 Recommendation System

A recommendation system is an intelligent system that utilizes behavioral patterns, past user choices, and situational context to produce tailored recommendations. Its primary goal is to assist users in navigating vast pools of information by quickly identifying and presenting the most relevant content, thereby significantly enhancing user satisfaction and engagement [22]. These systems are integral to modern internet-based applications, with their impact spanning across diverse domains such as e-commerce, entertainment, education, and healthcare. For instance, e-commerce platforms like Taobao and eBay employ recommendation systems to analyze purchase histories, browsing behaviors, and customer reviews to suggest products that align with user interests. This not only improves the shopping experience but also drives higher conversion rates and revenue. In the entertainment sector, platforms like Netflix and YouTube rely heavily on recommendation systems to provide users with tailored movie, music, and video suggestions based on their viewing and listening habits [23]. This enhances user engagement, increases retention, and keeps users exploring the platform's content library. In the field of education, platforms such as Coursera and Khan Academy leverage recommendation algorithms to provide personalized course recommendations, learning pathways, and educational resources, aiming to enhance learning efficiency and support individualized academic achievement [24]. Similarly, in healthcare, recommendation systems assist patients and professionals by suggesting treatment plans, medications, or health tips based on medical histories and preferences. Beyond their practical applications, recommendation systems also hold significant strategic value for businesses by enabling them to build stronger customer relationships, foster brand loyalty, and optimize operational efficiency. By delivering accurate, timely, and personalized suggestions, recommendation systems not only enhance user satisfaction but also serve as a cornerstone for driving innovation and maintaining competitive advantage within the context of modern data-centric environments.

In the early development in recommendation models, both collaborative and content-based filtering techniques emerged as the most widely used approaches, forming the foundation for personalized recommendation methods [25]. Collaborative filtering operates by predicting user preferences by measuring similarity across users (user-based) or items (item-based) [26]. For example, if two users share similar preferences, the system recommends items liked by one user to the other. This approach is simple, intuitive, and effective in capturing shared user behaviors, making it a popular choice for early recommendation systems. In contrast, content-based filtering emphasizes the inherent attributes of individual items [27]. By analyzing a user's past interactions with items (e.g., movies, books, or products), the system recommends similar items with matching characteristics. Despite their early success in personalizing recommendations, both methods face significant limitations. Collaborative filtering suffers from data sparsity, a common challenge in real-world systems where user-item interaction data is often sparse. For instance, a user may interact with only a small subset of available items, making it difficult for the system to generate reliable recommendations. Additionally, collaborative filtering struggles with scalability in large datasets and may encounter biases in recommendations for users or items with less activity. Similarly, content-based filtering is constrained by its reliance on item features, which may oversimplify user preferences. This method often fails to capture the complex and multifaceted nature of user interests, leading to repetitive or overly narrow recommendations. Furthermore, both approaches are vulnerable to the cold-start problem, where new users or items lack sufficient historical data for the system to provide meaningful recommendations [28][29]. These limitations highlighted the need for more advanced techniques, paving the way for hybrid methods and deep learning-based approaches that address challenges and improve recommendation quality.

With the rapid growth of the internet, the scale and complexity of data have increased dramatically, prompting the evolution of recommendation systems towards more advanced and sophisticated deep learning models [30]. Unlike traditional approaches, deep learning leverages nonlinear neural network architectures to extract complex hidden associations among users and items, significantly enhancing recommendation accuracy and relevance. These models excel at identifying patterns in high-dimensional data, enabling them to provide insights that were previously unattainable with traditional methods. For instance, neural collaborative filtering models, such as Neural Collaborative Filtering (NCF), integrate user and item features in a shared latent space, allowing for the generation of highly personalized recommendations [31]. These models replace traditional matrix factorization techniques with multi-layer deep architectures, allowing the model to acquire more flexible and non-linear interactions between users and items. This adaptability renders them applicable across diverse scenarios, including online retail and content delivery systems. Additionally, deep learning enables the processing of heterogeneous data modalities, including textual, visual, and auditory information, allowing recommendation systems to combine various data sources for more holistic and comprehensive suggestions. For example, a streaming platform might analyze textual reviews, video thumbnails, and user viewing behavior simultaneously to recommend the most relevant content. Models like convolutional neural networks (CNNs) and transformers further enhance the ability to process and understand these diverse data types [32][33]. The application of deep learning has not only improved user satisfaction by delivering more accurate and diverse recommendations but has also generated significant commercial returns for businesses. By increasing user engagement, retention, and conversion rates, deep learning-powered recommendation systems have become an essential tool for staying competitive in today's data-driven marketplace. However, the deployment of these systems raises new challenges in scalability, fairness, and data privacy, which remain active areas of research and development.

Despite the significant advancements in recommendation systems, several challenges persist, preventing their seamless deployment and performance optimization across industries. Firstly, the rapid growth of user bases and the exponential increase in data volumes require recommendation systems to handle real-time computations on massive datasets. This demand poses serious challenges to model efficiency, scalability, and response times, particularly in applications that require instantaneous recommendations, such as e-commerce and streaming platforms. Traditional approaches often struggle to maintain both speed and accuracy at scale, necessitating the development of more advanced algorithms and computational techniques. Secondly, user privacy and data security have become increasingly critical concerns. Modern recommendation systems heavily depend on the collection and analysis of vast amounts of user data, including browsing histories, purchase behaviors, and personal preferences. This reliance raises potential threats, including privacy intrusions, illicit data exposure, and improper handling of confidential information [34]. These concerns are further compounded by stringent regulatory frameworks like the General Data Protection Regulation (GDPR) in the European Union and the California Consumer Privacy Act (CCPA) in the United States, which enforce data privacy provisions, which mandate robust safeguards to safeguard personal privacy while empowering users with enhanced control over their data assets [35][36]. Moreover, balancing recommendation performance with privacy and security presents a complex trade-off. On one hand, high-quality recommendations require extensive data for training sophisticated models. On the other hand, the more data collected and stored, the higher the risk of privacy violations and security vulnerabilities. Addressing these challenges calls for innovative solutions, such as federated learning, differential privacy, and encrypted computation, which can enhance privacy protection without sacrificing model accuracy [37]. Thus, developing scalable, privacy-preserving, and secure recommendation systems remains one of the pressing and active research areas in this field.

1.1.2 RS Model Based on Deep Learning

Deep learning has revolutionized recommendation systems by providing the capacity to uncover intricate, non-linear associations among users and items, which traditional models often fail to achieve. Unlike earlier approaches that relied on simple linear transformations, deep learning leverages sophisticated neural network architectures capable of learning intricate patterns in high-dimensional data [38]. This capability allows deep learning models to go beyond surface-level correlations, identifying subtle and latent relationships between user preferences and item attributes, even in sparse datasets where traditional methods struggle. As a result, these models significantly improve recommendation accuracy, robustness, and the ability to generalize to unseen data. Among the widely used deep learning models, NCF has become a foundational approach in the field. By replacing traditional matrix factorization techniques with multi-layer neural networks, NCF introduces the flexibility to model nonlinear interactions between users and items. This adaptability enables the system to uncover deeper insights into user preferences, rendering it applicable to diverse recommendation scenarios. Sequence modeling techniques, including Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) models, and Transformer architectures have demonstrated remarkable performance in capturing sequential dependencies in user behavior [39][40]. These models are particularly effective in learning the temporal order of user actions, such as clicks, purchases, or views, providing more contextually aware and timely recommendations. Furthermore, deep learning models excel at integrating multi-modal data sources and allowing recommendation systems to deliver richer, personalized suggestions. These advantages make deep learning an indispensable tool for building next-generation recommendation systems, offering unprecedented levels of precision and versatility in tailoring user experiences.

Deep learning has significantly expanded the scope and effectiveness of recommendation systems across various application scenarios, becoming a cornerstone of modern personalized services [41]. One critical application is user behavior prediction, where deep learning models analyze historical interactions to anticipate future actions, such as clicks, purchases, or video views. These predictions are vital for industries like e-commerce, streaming platforms, and online services that depend on understanding user intent to drive engagement and revenue. For example, an online retail system could infer potential user purchases by analyzing their browsing behaviors and buying patterns, enabling more targeted promotions and advertisements. Another major application is personalized recommendation, where systems suggest videos, music, or products that align with a user's unique preferences and interests [42]. Platforms like Netflix and YouTube utilize advanced deep learning models for interpreting users' content consumption patterns and preferences, recommending content that enhances satisfaction and retention. This personalization extends beyond entertainment, with applications in healthcare, education, and more. Deep learning also excels in multi-modal recommendations, where diverse data types such as textual reviews, visual content, and audio are integrated to provide richer, more comprehensive suggestions. For instance, in e-commerce, combining product images, user reviews, and search history allows the system to recommend products that not only match user preferences but also appeal to their aesthetic or practical needs. Similarly, streaming platforms leverage multi-modal data to suggest music and videos that align with both a user's mood and past behavior, creating a more engaging and immersive experience. These advanced applications emphasize the significant impact of deep learning on RS, allowing these systems to produce highly precise and context-aware, and diverse recommendations that enhance user satisfaction and business outcomes.

These models map user and item representations within a common latent embedding space, enabling the system to compute similarities and generate recommendations effectively. By leveraging neural network-trained embeddings, RS can learn rich representations that capture complex and hidden relationships between users and items, even in sparse datasets. These embeddings allow the system to generalize better across diverse user preferences and item attributes, improving recommendation accuracy and robustness. To model sequential user behavior, advanced architectures like RNNs, LSTM networks, and Transformers are widely employed. RNNs and LSTMs have been instrumental in capturing short-term dependencies in user interactions, such as click sequences or purchase patterns. Transformers, on the other hand, excel in learning both short-term and long-term dependencies due to their self-attention mechanism, making them particularly effective for time-sensitive recommendations. For instance, a Transformer-based model can track the evolution of a user's preferences over time, enabling the system to suggest items that align with both current interests and historical trends. Hybrid models take RS a step further by integrating multiple data modalities, such as text, images, and audio, to enhance diversity and accuracy [43]. For example, an e-commerce hybrid model might combine user reviews, product images, and browsing history to recommend items that are not only relevant but also visually appealing or contextually suitable. These models enable recommendation systems to deliver richer, more personalized experiences, addressing diverse user needs across a wide range of applications.

Despite their advantages, deep learning-based RS faces various challenges that hinder its widespread and seamless implementation. One significant issue is the efficiency in processing large-scale data. Training deep learning models on extensive datasets demands substantial computational power and memory, which can become a bottleneck in scenarios requiring real-time recommendations. This is especially challenging for platforms with millions of active users and continuously expanding catalogs. Optimizing model architectures, adopting lightweight models, and leveraging distributed computing frameworks such as Hadoop and Spark are essential strategies to enhance scalability and efficiency [44]. Another critical challenge is the reliance on extensive user data, which introduces significant privacy concerns. Deep learning models often require vast amounts of data, including sensitive user behaviors, preferences, and interactions, to deliver accurate recommendations. This dependency increases the risk of unauthorized data access, breaches, and misuse, especially in centralized systems. The rise in privacy governance frameworks like the GDPR and CCPA, underscores the need for robust mechanisms to ensure data security and user trust. To address these privacy concerns, innovative approaches like federated learning and differential privacy have emerged as promising solutions. Federated learning enables decentralized model training by keeping user data local, minimizing the risk of exposure. Differential privacy introduces calibrated perturbations into the data, ensuring individual data points remain unidentifiable while maintaining aggregate accuracy. Balancing model performance and data protection remains a critical focus for advancing recommendation models driven by deep learning techniques.

Deep learning has revolutionized recommendation systems by enabling them to capture complex, nonlinear relationships between users and items, as well as effectively process multi-modal data from diverse sources, such as text, images, and audio. This capability has substantially enhanced precision and applicability in sectors such as online retail, media services, and educational platforms. By leveraging powerful neural network architectures, deep learning models can analyze intricate patterns in high-dimensional data, deliver contextually aware recommendations, and adapt to evolving user preferences. Despite these advancements, challenges such as scalability, computational efficiency, and data privacy remain critical barriers to fully realizing the potential of deep learning in recommendation systems. Handling the ever-increasing scale of data and users while maintaining real-time response capabilities poses significant engineering and algorithmic challenges. Moreover, the reliance on vast amounts of user data raises concerns about privacy and compliance with stringent regulations such as GDPR and CCPA.

1.1.3 Privacy and Security Considerations

In recommendation systems, the centralized storage of user data presents significant risks, such as security breaches and illicit data exposure. Centralized databases often act as single points of failure, making them attractive targets for attackers. A successful breach can expose vast amounts of sensitive user information, such as browsing history, purchasing behaviors, and personal preferences, leading to severe privacy violations and a loss of user trust. Beyond direct breaches, user privacy leakage through advanced inference techniques poses an equally critical threat. Sophisticated attackers can exploit vulnerabilities in deep learning models to reverse-engineer user data, reconstructing sensitive attributes like age, gender, or location from model outputs. This risk is heightened by the black-box nature of deep learning models, which obscures their internal processes and complicates efforts to identify and mitigate such vulnerabilities [45]. Furthermore, recommendation systems face a fundamental trade-off: the need to collect extensive user data to improve personalization versus the imperative to protect user privacy. Striking this balance remains a pressing challenge, as insufficient safeguards could deter users and hinder the adoption of these systems. Mitigating these threats demands novel privacy-focused strategies and robust system designs to guarantee the protection of user data while maintaining system effectiveness.

Deep learning-powered recommendation systems are becoming progressively susceptible to a variety of security challenges that jeopardize their integrity, reliability, and trustworthiness [46]. One prominent concern is adversarial attacks, where malicious actors craft inputs specifically designed to manipulate the system's outputs. These subtle perturbations, often imperceptible to users, can cause significant deviations in recommendations, resulting in biased or inaccurate suggestions and undermining user trust. A similarly critical concern involves data poisoning, in which adversaries introduce harmful or deceptive inputs into the training dataset. This tactic embeds hidden biases or intentional errors in the model, leading to degraded performance and persistent inaccuracies in recommendations that can severely harm user experience and the system's reputation over time [47]. Furthermore, the issue of model ownership and intellectual property has emerged as a critical challenge [48]. Given the substantial commercial value of deep learning models, unauthorized usage, theft, or reverse engineering poses significant risks to developers. Techniques like model watermarking and ownership verification have become essential to safeguard intellectual property and detect unauthorized deployments. Together, these threats highlight the urgent need for robust security mechanisms and countermeasures to ensure the continued robustness and stability of recommendation models powered by deep learning in increasingly adversarial environments [49][50].

While deep learning has significantly enhanced the accuracy, adaptability, and personalization capabilities of RS, its inherent complexity introduces a range of privacy and security challenges that demand careful attention. One major concern is the large-scale data requirements of deep learning models, which increase the attack surface and expose systems to risks such as inference attacks and adversarial manipulations. Inference attacks enable malicious actors to extract sensitive user information from model outputs, compromising user privacy and potentially violating data protection regulations [51]. Adversarial manipulations, on the other hand, involve carefully engineered data aimed at deceiving the system and generating flawed or skewed outputs, undermining system reliability and user trust [52]. Compounding these issues is the lack of transparency in deep learning architectures, often referred to as black boxes, which raises significant concerns about their robustness and transparency. The lack of interpretability makes it challenging to diagnose unexpected behaviors or defend against malicious inputs effectively, further diminishing trust in the system. These challenges underscore the urgent need for more interpretable and secure deep learning techniques, such as adversarial training, explainable AI, and robust architectural designs, to ensure that recommendation systems remain both effective and trustworthy in increasingly complex and adversarial environments.

To address the critical privacy and security challenges faced by deep learning-based recommendation systems, several innovative approaches have been proposed to enhance system resilience and safeguard user data. Privacy-preserving techniques play a key role in this effort. To strengthen data privacy in recommendation systems, techniques like federated learning and differential privacy offer complementary strengths. Additionally, anomaly detection techniques can proactively identify suspicious activities, such as unusual access patterns or abnormal data inputs, mitigating the risk of malicious intrusions

[53]. On the security mechanisms front, strategies such as adversarial training enhance model robustness by incorporating adversarial examples during training, reducing vulnerabilities to manipulation. Homomorphic encryption enables operations over encrypted inputs while preserving the confidentiality of information across the computation pipeline, while watermarking embeds unique identifiers into models to verify ownership and prevent unauthorized use [54]. Moreover, transitioning to decentralized data storage offers a structural solution to mitigate the risks of single points of failure. Distributed systems, powered by technologies like blockchain, facilitate secure and transparent data sharing while preserving user privacy. Together, these approaches form a comprehensive framework to address privacy and security concerns, ensuring that recommendation systems remain reliable, robust, and trustworthy in real-world applications.

Privacy and security considerations are critical for the sustainable development of recommendation models empowered by deep learning techniques. While the advanced capabilities of deep learning have revolutionized recommendation accuracy and personalization, they have also introduced significant challenges concerning privacy protection, cybersecurity, and the resilience of recommendation infrastructures. The reliance on large-scale user data increases the risk of breaches, unauthorized access, and malicious attacks, highlighting the urgent need for innovative and robust solutions. Techniques such as federated learning enable decentralized training, reducing risks associated with centralized data storage, while adversarial training improves model resilience against manipulation by incorporating adversarial examples. Distributed architectures, including blockchain-based solutions, provide secure and transparent data-sharing mechanisms that minimize single points of failure. By integrating these approaches, recommendation systems can achieve a balance between performance and user trust, ensuring they remain privacy-preserving, secure, and reliable in real-world applications. These strategies not only address existing challenges but also lay the foundation for future advancements in secure and trustworthy recommendation technologies.

1.2 Motivation

The rapid evolution of recommendation systems, propelled by breakthroughs in deep learning, has transformed how personalized services are delivered in various industries, including retail, streaming platforms, and digital healthcare. These systems leverage vast amounts of data to understand user preferences, enabling highly accurate and context-aware recommendations that improve decision-making and satisfaction. For instance, modern e-commerce platforms can suggest products based on intricate patterns in browsing and purchasing history, while video-streaming services curate personalized content for diverse audiences. However, these advancements come with significant challenges, particularly in recommendation accuracy, user data protection, and model security. Centralized architectures, which form the backbone of many existing systems, create single points of failure that can lead to large-scale data breaches. Moreover, as user interaction datasets grow in scale and complexity, maintaining real-time responsiveness and ensuring system scalability have become critical hurdles. Addressing these challenges requires innovative approaches that go beyond traditional methods, focusing on architectural changes and the integration of advanced privacy and security mechanisms.

Privacy concerns in deep learning-based recommendation systems have become increasingly prominent, given the rising complexity of data and the heightened reliance on user-specific information. Centralized storage mechanisms not only aggregate vast amounts of sensitive data but also make it vulnerable to various forms of exploitation. Beyond typical data breaches, attackers can exploit sophisticated methods such as model inversion, where they extract sensitive attributes by analyzing model outputs. Furthermore, the prevalence of adversarial attacks poses a severe threat to recommendation quality. By injecting imperceptible changes into input data, attackers can deceive the model into producing incorrect or harmful recommendations. Another growing issue is data poisoning, where attackers deliberately introduce malicious samples into training datasets, embedding biases or degrading model performance over time. Compounding these challenges is the opacity of deep learning architectures, hindering efforts to identify and mitigate vulnerabilities effectively [55]. As the complexity of these systems increases, it becomes imperative to explore decentralized architectures, privacy-preserving techniques, and robust defense mechanisms. Techniques such as secure multi-party computation and encrypted evaluation offer promising avenues to address these concerns, but practical implementations and scalability remain critical areas of research.

The motivation for this research lies in addressing the critical need to bridge the gap between performance optimization and robust privacy and security measures in deep learning-based recommendation systems. While these systems have significantly enhanced user experience by providing personalized and accurate recommendations, they are increasingly vulnerable to issues such as data breaches, adversarial attacks, and privacy violations. The growing complexity of these systems and their reliance on large-scale user data highlight the pressing need for innovative solutions that balance efficiency and security. This study aims to develop scalable, efficient, and secure mechanisms that not only safeguard user data but also protect model integrity and ensure compliance with privacy regulations. By tackling these challenges, this research seeks to advance the field, offering a foundation for the development of trustworthy and privacy-preserving recommendation systems. These solutions are essential for fostering user trust, enhancing system reliability, and ensuring the safe deployment of recommendation systems across privacy-sensitive domains like healthcare, finance, and education.

1.3 Research Objectives and Contribution

In Section 1.2, a series of unresolved issues within deep learning-based recommendation systems were examined, underscoring the pressing need to tackle several critical challenges in this domain. These identified limitations provide the rationale for this study and lay the groundwork for formulating its key goals and scholarly contributions. The originality and importance of each research objective are thoroughly analysed to highlight their potential impact.

1. **A dual defense design is proposed to mitigate the impact of data poisoning attacks in deep learning-based recommendation systems, addressing the vulnerabilities posed by adversarial manipulation of user data.** The proposed approach consists of two complementary defense mechanisms: active defense and passive defense. The active defense mechanism proactively reduces the system's susceptibility to poisoning attacks by incorporating crafted regularization techniques directly into the model's loss function. This strategy effectively minimizes the influence of malicious inputs while maintaining the recommendation system's overall performance, thereby lowering the success rate of targeted attacks. In parallel, the passive defense mechanism employs a Generative Adversarial Net-

work (GAN)-based detection model to accurately identify and filter out poisoned data from the training set. Unlike traditional detection methods, the GAN-based approach leverages the adversarial framework to improve the precision and recall of poisoned data identification. Together, these defenses enhance the system's robustness and reliability. Empirical evaluations conducted on three widely used datasets demonstrate that the proposed dual defense approach not only strengthens the proactive resilience of RS but also significantly improves the accuracy of detecting and mitigating data poisoning attacks.

2. **A novel scheme for recommendation unlearning verification (RUV) is proposed, which utilizes non-influential trigger data to assess the compliance of RS with unlearning requests.** The proposed scheme enables users to evaluate the effectiveness of the unlearning process by monitoring the recommendation rate of specific target items. Unlike traditional methods that rely on complex and resource-intensive computations, the RUV scheme leverages lightweight and efficient trigger data to provide a practical and scalable verification approach. Moreover, the scheme does not require access to sensitive user data, thereby maintaining the privacy of users while ensuring system accountability. By avoiding dependency on intricate model evaluations, the proposed scheme simplifies the verification workflow while enhancing its availability across a wider spectrum of users and scenarios. Experimental results on real-world datasets demonstrate the reliability and effectiveness of the RUV, establishing it as a robust solution for verifying unlearning compliance in RS.

3. **A recommendation system model ownership verification framework is proposed, leveraging non-influential watermarking to protect intellectual property in deep learning-based recommendation systems.** The proposed framework embeds backdoor watermarks into the training dataset, ensuring that the watermark remains undetectable while maintaining the model's performance and recommendation accuracy. Unlike existing watermarking techniques, which are predominantly designed for image data, this framework is tailored for tabular data in recommendation systems, addressing the unique challenges associated with this domain. Furthermore, the watermarking process is designed to achieve a high level

of fidelity, invisibility, and efficiency, ensuring that the system operates seamlessly without compromising user experience or computational overhead. Experimental results conducted over multiple benchmark datasets validate the robustness and dependability of the proposed approach, establishing its effectiveness in verifying model ownership and safeguarding intellectual property against unauthorized usage or infringement.

4. **A robust and adaptive dual-defense framework is proposed to mitigate data poisoning attacks in deep learning-based recommendation systems, particularly in scenarios involving multiple datasets and complex RS models, LLMs as recommendation engines. The framework integrates active and passive defense mechanisms to effectively reduce attack impact while preserving recommendation quality.** In the active defense component, we introduce the crafted loss function method, which significantly lowers the success rate of data poisoning attacks under various attack intensities while maintaining minimal impact on model performance. For passive defense, a scalable GAN-based detection model is developed to accurately identify and filter malicious data, thereby enhancing detection accuracy and reinforcing the security of the recommendation system. Furthermore, the framework is designed to be adaptable across different datasets and deep learning architectures, ensuring broad applicability. Extensive experimental evaluations on multiple benchmark datasets demonstrate the reliability and efficacy of the developed method, demonstrating its capability to enhance system security and reliability in adversarial environments.

1.4 Research Scope

Our research is centered on tackling key issues within deep learning-driven recommender models, with particular emphasis on privacy-preserving mechanisms and security innovations. The rapid adoption of RS across diverse industries, including domains like online retail, digital media, and health informatics, highlights the need for robust solutions to counteract emerging threats and ensure compliance with privacy regulations. Specifically, we aim to tackle issues including violations of user data confidentiality, adversarial attacks, and model security problems, which pose significant risks to user trust and system reliability. By developing scalable and secure methodologies, this research seeks to establish a strong foundation for trustworthy and privacy-respecting recommendation systems in real-world applications. This study specifically aims to:

1. Data Poisoning Resistance: Develop proactive and reactive defense strategies against data poisoning attacks, which maliciously manipulate user data to degrade system performance and compromise recommendation reliability. Our research introduces dual defense mechanisms that integrate active and passive strategies to enhance the robustness of recommendation systems.

2. Recommendation Model Ownership Protection: Propose a novel watermarking framework designed to protect the intellectual property of models by embedding non-intrusive identifiers into RS models. These identifiers ensure ownership verification without compromising model accuracy or user experience.

3. Verification of Recommendation Unlearning: Explore mechanisms for verifying recommendation system compliance with user data unlearning requests. Using non-influential trigger data, our proposed approach ensures adherence to data removal requirements, addressing privacy concerns while maintaining system performance.

4. Integration with LLMs: Investigates the role of LLMs in enhancing recommendation systems, particularly in terms of adaptability and personalization. It also involves developing technical solutions to mitigate the security and privacy challenges introduced by LLMs when applied to recommendation tasks, such as improving adversarial robustness and ensuring data privacy protection.

This research aims to establish a secure and trustworthy framework for the RS model. By safeguarding user privacy, protecting the intellectual property of the model, and enhancing model robustness, our study aims to set the foundation for scalable and reliable recommendation systems that cater to modern privacy-conscious environments.

1.5 Structure of Thesis

The remainder of this thesis is organized as follows. Chapter 2 provides a comprehensive literature review, covering foundational concepts in deep learning-based recommendation systems, including key challenges such as data poisoning, recommendation unlearning verification, and model ownership verification, while also highlighting the role of LLMs in enhancing recommendation systems and their associated security risks. A dual defense design to mitigate data poisoning attacks is conducted in Chapter 3, outlining the implementation of active and passive defense mechanisms, including a GAN-based detection model, and presents experimental evaluations to demonstrate the effectiveness of the proposed approach. A novel recommendation unlearning verification (RUV) scheme is proposed in Chapter 4, detailing the use of non-influential trigger data to evaluate unlearning compliance and showcasing experimental results to validate its efficiency and accuracy. Chapter 5 focuses on RS model ownership verification through non-influential watermarking, presenting a framework tailored for recommendation system models with an emphasis on fidelity, invisibility, and efficiency, supported by experimental results confirming its reliability. Chapter 6 explores a robust and adaptive dual-defense framework to mitigate data poisoning attacks in deep learning- based recommendation systems, particularly in scenarios in involving multiple datasets and complex RS models, LLMs as recommendation systems. Finally, Chapter 7 concludes the thesis with a summary of contributions and outlines potential avenues for future research, addressing unresolved challenges in secure and privacy-preserving recommendation systems and discussing the broader implications of the findings. The bibliography is provided for supplementary materials and references.

Chapter 2

2 Literature Review

The rapid advancements in deep learning have significantly impacted recommendation systems, making them indispensable in various domains such as e-commerce, streaming services, healthcare, and education. These systems provide highly personalized user experiences, improving customer satisfaction and operational efficiency [56][57]. However, alongside their transformative potential, recommendation systems face critical challenges, particularly concerning privacy, security, and intellectual property protection. The vast amounts of user data required for training deep learning models increase the risks of data breaches, privacy violations, and adversarial attacks. Additionally, the integration of advanced technologies, such as LLMs with RS, introduces new complexities, including scalability issues and unique security vulnerabilities [58][59]. This chapter reviews existing works to provide a comprehensive understanding of these challenges. The focus is on identifying the vulnerabilities of recommendation systems, evaluating the effectiveness of existing defense mechanisms, and exploring innovative approaches leveraging advanced deep learning architectures and LLMs to address these issues.

This Chapter presents a survey of related works that serve as a background to our research objectives. In Section 2.1, a broad overview of deep learning-based recommendation systems is provided, detailing their architectures, advantages, and inherent limitations. This section highlights how these systems leverage complex neural networks to enhance personalization and user engagement while addressing scalability challenges. Section 2.2 focuses on data poisoning attacks and defensive techniques, shedding light on the vulnerabilities these attacks exploit in recommendation models and the countermeasures proposed in the literature to mitigate their impact. Section 2.3 explores recommendation unlearning verification mechanisms, a relatively new and evolving area aimed at ensuring privacy compliance and granting users greater control over their data. Section 2.4 examines model ownership verification, emphasizing risks such as intellectual property obfuscation and presenting advanced techniques to safeguard proprietary models. Section 2.5 reviews the integration of LLMs into recommendation systems, discussing their transformative potential and the unique security challenges they introduce. Finally, Section 2.6 summarizes the key findings, identifies research gaps, and outlines opportunities for future exploration in this rapidly evolving field.

2.1 Overview of Recommendation System Based on Deep Learning

The increasing popularity of recommendation systems across various domains such as e-commerce, streaming platforms, and social media underscores their pivotal role in enhancing user experiences [60]. Traditional recommendation approaches, such as collaborative filtering and content-based filtering, have been widely adopted but exhibit significant limitations. These methods often struggle to capture non-linear relationships between users and items or handle sparse and high-dimensional data effectively. Deep learning, with its capability for non-linear transformations and representation learning, has addressed these challenges, transforming the field of recommendation systems [61]. By leveraging advanced deep learning architectures, RS can now model intricate user-item interactions while integrating multimodal data such as text, images, and audio. The flexibility and robustness of deep learning-based methods have not only improved recommendation accuracy but also enhanced scalability and adaptability across diverse applications. As industries increasingly adopt recommendation systems, deep learning techniques have become indispensable for maintaining competitive advantages [62].

Deep learning-based recommendation systems employ sophisticated techniques like Multilayer Perceptrons (MLP), Autoencoders (AE), Adversarial Networks (AN), and Deep Reinforcement Learning (DRL) [63]. These methods excel at capturing latent relationships and enabling personalized experiences through advanced modeling of user-item interactions [64]. A typical integration of recommendation systems and deep learning techniques is illustrated in Fig. 2.1. One notable model is Neural Matrix Factorization (NeuMF), which combines linear and non-linear transformations to create robust predic-

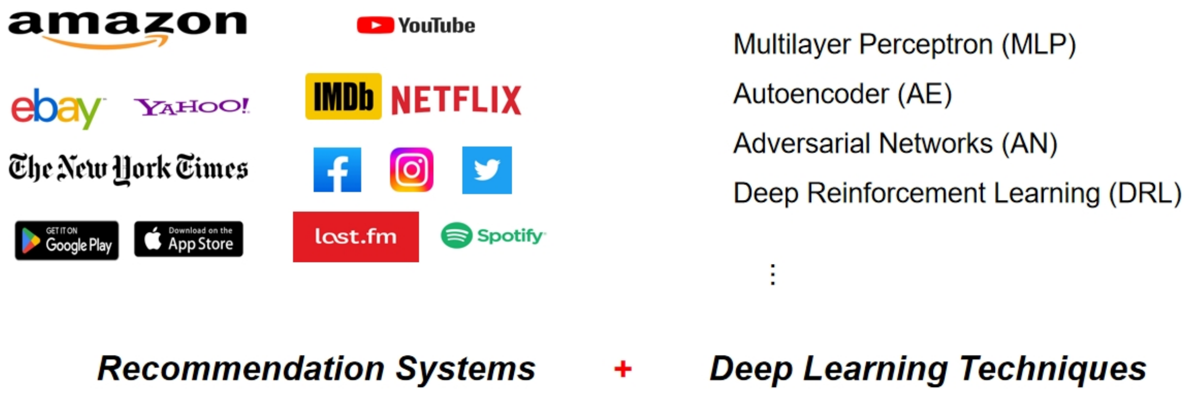

Multilayer Perceptron (MLP)
Autoencoder (AE)
Adversarial Networks (AN)
Deep Reinforcement Learning (DRL)
⋮
Recommendation Systems
+
Deep Learning Techniques

Figure 2.1: Integration of Recommendation Systems and Deep Learning.

tions [65]. NeuMF uses embedding layers to transform user and item data into dense latent vectors. These vectors are processed through a Generalized Matrix Factorization (GMF) layer for linear modeling and an MLP layer for non-linear feature learning. This combination captures complex interaction patterns between users and items. By minimizing the error between observed and predicted ratings, NeuMF ensures top-K recommendations tailored to individual user preferences. The operational mechanism of NeuMF is demonstrated in Fig. 2.2, showcasing its ability to integrate linear and nonlinear components for personalized recommendations.

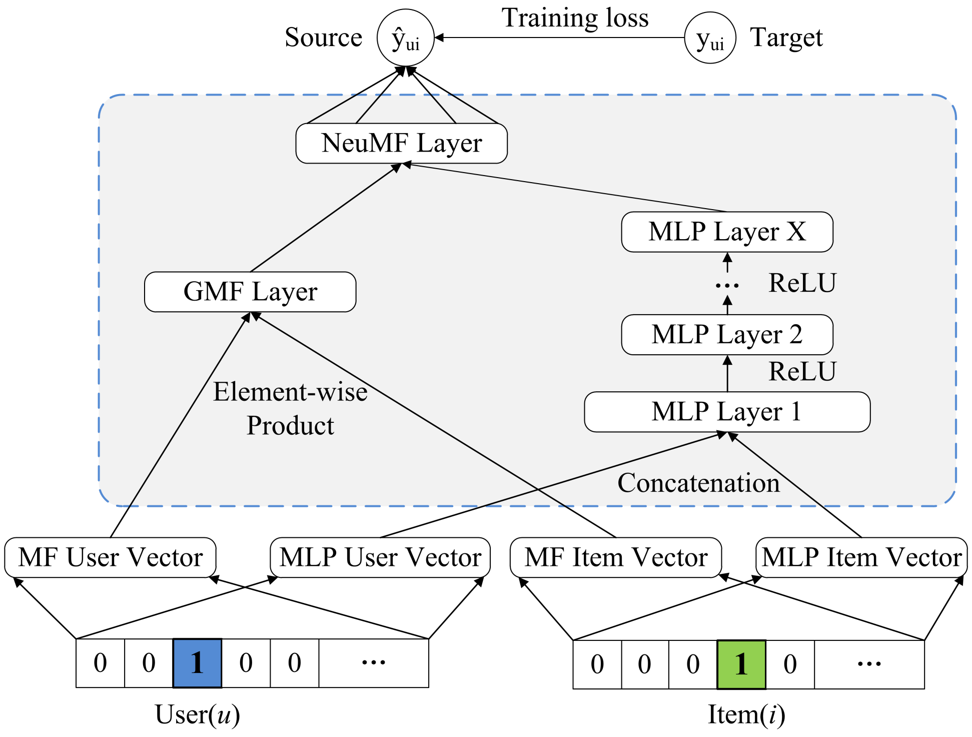
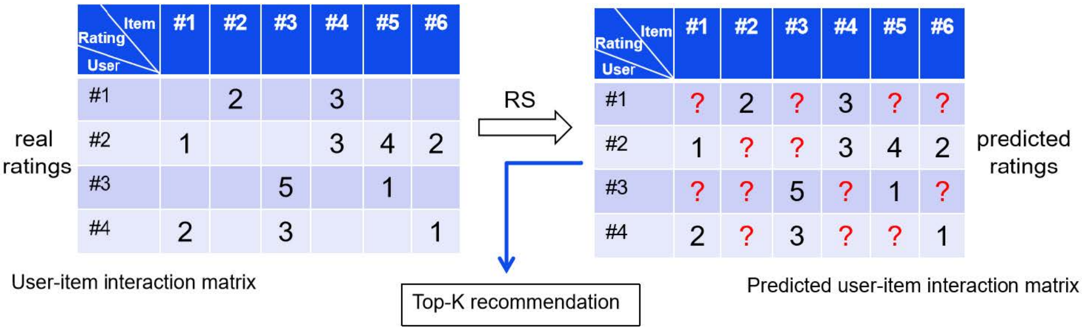


Figure 2.2: The Framework of Neural matrix factorization model (NeuMF).

Deep learning's ability to integrate diverse data sources and handle sequential interactions has made it a powerful tool for building state-of-the-art RS. It has revolutionized the operation of RS by effectively modeling complex user-item interactions [66]. Unlike traditional methods, such as collaborative filtering and content-based approaches, deep learning utilize advanced representation learning to uncover latent features of users and items, significantly enhancing recommendation accuracy. As illustrated in Fig. 2.3, the process begins with a user-item interaction matrix, which includes known ratings provided by users for specific items. The system employs deep learning models, such as neural networks, to predict missing ratings by learning patterns and relationships within the data. This prediction process minimizes the loss between actual ratings and predicted ratings during training, ensuring model optimization. Moreover, deep learning enables the integration of multimodal data, such as text, images, and sequential user interactions, which provides richer contextual information for more precise predictions [67]. Once the model achieves satisfactory accuracy, it generates a top-K recommendation list tailored to each user's preferences. This approach not only addresses data sparsity and cold-start problems but also ensures scalability and adaptability across diverse applications, making deep learning an indispensable tool for modern recommendation systems [68].

Despite its advantages, deep learning in recommendation systems introduces challenges, including high computational costs, overfitting, RS model security, and vulnerabilities to adversarial and data poisoning attacks [69]. Computational overhead can hinder real-time applications, while overfitting can reduce the generalizability of recommendations. Moreover, adversarial attacks exploit model weaknesses to manipulate recommendations, while data poisoning compromises training datasets, leading to biased outputs [70]. Addressing these challenges requires designing robust, scalable, and secure models [71]. Nevertheless, deep learning continues to drive innovation, enabling recommendation systems to adapt dynamically to user preferences and deliver high-quality personalized experiences. As the field evolves, further advancements in model architectures and defensive techniques will ensure that recommendation systems remain effective and secure, catering to increasingly diverse and complex user needs [72].


real ratings
RS
predicted ratings
User-item interaction matrix
Top-K recommendation
Predicted user-item interaction matrix

Rating / Item / User	#1	#2	#3	#4	#5	#6
#1		2		3		
#2	1			3	4	2
#3			5		1	
#4	2		3			1

Rating / Item / User	#1	#2	#3	#4	#5	#6
#1	?	2	?	3	?	?
#2	1	?	?	3	4	2
#3	?	?	5	?	1	?
#4	2	?	3	?	?	1

Figure 2.3: User-Item Matrix for Top-K Recommendations.

2.2 Data Poison Attack and Defense Mechanisms on RS Model

Recommendation systems have become an integral part of modern applications, providing personalized suggestions across e-commerce, entertainment, and other domains. However, these systems are increasingly vulnerable to data poisoning threats, representing a major risk to their integrity and reliability. During data poisoning, attackers introduce deliberately manipulated inputs, such as fake user profiles and manipulated ratings, into the training dataset. These attacks exploit the inherent reliance of RS models on historical data to influence the model's output, often elevating the visibility and ranking of specific target items. As a result, poisoned models may deliver biased or unreliable recommendations, undermining user trust and system performance. The iterative nature of such attacks exacerbates the challenge, as attackers refine their methods to maximize the impact. Defending against these attacks is critical to preserving the system integrity and predictive precision in RS models, thereby requiring the implementation of advanced detection and mitigation strategies to safeguard their operation [73].

Data poisoning attacks pose a significant threat to the robustness and reliability of deep learning-based recommendation systems [74]. These attacks aim to manipulate the underlying recommendation model by injecting malicious or poisoned data into the training dataset. By altering the data that the model learns from, attackers can influence the system's outputs to favor specific items or users, disrupting the fairness, accuracy, and integrity of recommendations. Such attacks are particularly dangerous because RS models are inherently reliant on large-scale, user-generated data, which is difficult to validate for authenticity or integrity [75]. Attackers exploit this dependency to introduce carefully crafted malicious inputs that align with their objectives, often escaping detection by mimicking legitimate interactions [76].

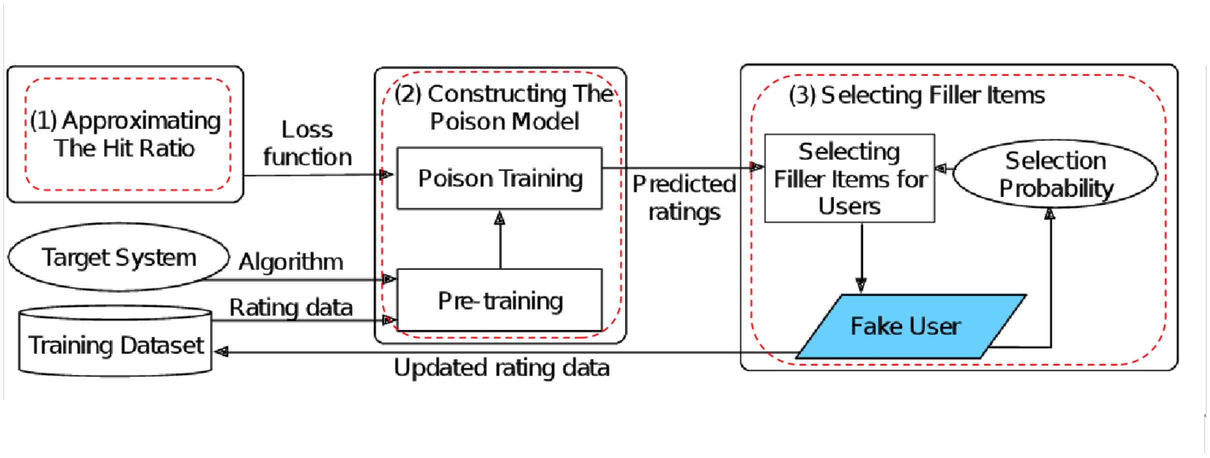


Figure 2.4: Framework of a Targeted Data Poisoning Attack in Recommendation Systems.

As depicted in Fig. 2.4, the first step in these attacks involves proximating the hit ratio, a process where attackers evaluate the impact of potential poisoned data on the recommendation system's performance. This step typically uses optimization techniques to minimize a predefined loss function, enabling attackers to identify vulnerabilities in the target system. By understanding the relationship between user-item interactions and the system's predictive outputs, attackers can develop an informed strategy to manipulate recommendations effectively. This step forms the foundation for crafting a highly targeted and damaging data poisoning attack [70].

Once attackers identify the weaknesses in the recommendation system during the hit ratio approximation phase, they proceed to the second step, construction of the poison model. This involves pre-training a surrogate model designed to mimic the behavior and structure of the target RS. Surrogate models are often created using algorithms similar to those employed in the target system, enabling attackers to simulate its learning dynamics. The surrogate model acts as a testbed for generating poisoned data, allowing attackers to evaluate how injected data would alter the outputs of the actual target system.

The poisoned data is generated by the surrogate model through a series of predictions that incorporate malicious ratings. These predictions are designed to appear genuine while systematically promoting the attacker's goals. For example, ratings for specific target items might be artificially inflated to improve their visibility in the recommendation list. Once generated, the poisoned data is integrated into the training dataset of the target system, ensuring it influences subsequent learning cycles. By leveraging the surrogate model, attackers can execute their plans with precision, deceiving the recommendation system into favoring their intended outcomes without raising immediate suspicion.

For the selecting filler items process, it focuses on refining the attack by generating fake user profiles and interactions that maximize the impact of the poisoned data. This phase plays a pivotal role in enabling the injected data to significantly influence the behavior of the recommendation model. Attackers create fake users with carefully curated interactions, which include both filler items and target items. Filler items act as decoys that obscure the malicious intent behind the fake profiles, while target items are those the attackers aim to promote within the recommendation system.

The process involves iteratively selecting filler items and optimizing the interactions of fake users to align with the attacker's objectives. By balancing the inclusion of filler and target items, attackers can ensure the fake user profiles appear legitimate, reducing the likelihood of detection. This step is repeated until the desired level of manipulation is achieved, with attackers constantly refining the profiles to maximize the attack's effectiveness. By combining iterative updates with the insights gained from the surrogate model, attackers can exploit the vulnerabilities of the recommendation system to their advantage. This iterative approach underscores the sophisticated nature of data poisoning attacks and highlights the urgent need for robust defensive mechanisms to protect RS models from such threats.

While various defense mechanisms have been proposed to counter data poisoning attacks in deep learning-based recommender systems RS, these approaches face significant limitations. Traditional anomaly detection techniques often rely on statistical methods to identify irregularities in user behavior or rating patterns. However, advanced attackers have developed strategies to craft malicious inputs that closely mimic legitimate user interactions, bypassing these statistical detectors. This creates a significant challenge for systems attempting to distinguish genuine data from maliciously altered inputs. Furthermore, many existing defenses operate under the assumption that the system's architecture or the distribution of training data remains static. In practice, RS models are frequently updated and retrained to improve performance or adapt to new data, introducing new vulnerabilities and reducing the effectiveness of previously established defenses. These dynamic changes in the system environment highlight the inadequacy of static, one-time countermeasures in addressing the evolving nature of data poisoning threats.

In addition to their technical shortcomings, some countermeasures inadvertently compromise the overall user experience and system performance. For example, defensive strategies that aggressively filter potentially poisoned data can also exclude legitimate user inputs, resulting in a degradation of recommendation quality. This trade-off between security and usability remains a major challenge in the design of robust defense mechanisms. Huang et al. underscores these difficulties, demonstrating that even when detection mechanisms are implemented, sophisticated attacks may still succeed in manipulating system outputs [70]. These findings underline the necessity for adaptive and resilient defense strategies that can evolve alongside emerging attack methodologies. Effective defenses must not only detect and neutralize malicious inputs but also preserve the core functionalities of recommender systems, ensuring both security and optimal user experience. In this thesis, this work has explored and summarized the methods and defense mechanisms against data poisoning attacks, as shown in Fig. 2.5.

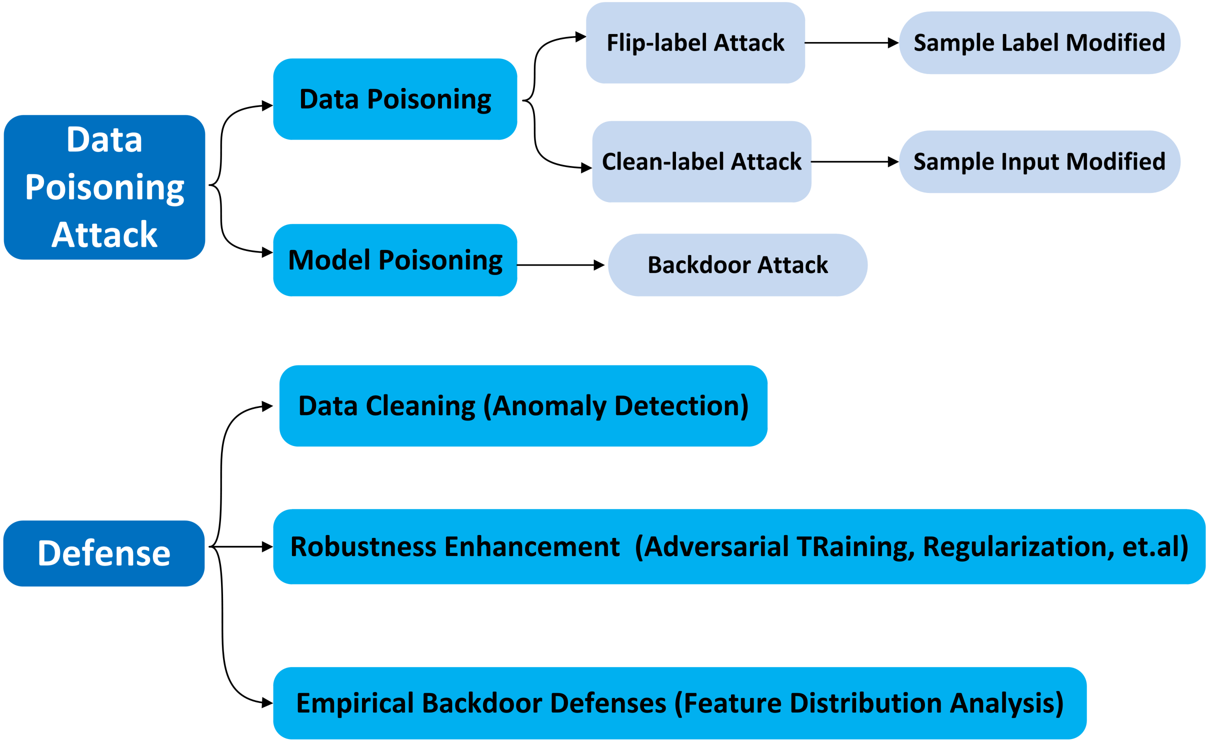

Data Poisoning Attack
Data Poisoning
Flip-label Attack
Sample Label Modified
Clean-label Attack
Sample Input Modified
Model Poisoning
Backdoor Attack
Defense
Data Cleaning (Anomaly Detection)
Robustness Enhancement (Adversarial TRaining, Regularization, et.al)
Empirical Backdoor Defenses (Feature Distribution Analysis)

Figure 2.5: Overview of Data Poisoning Attacks and Defensive Techniques in Recommendation Systems.

2.2.1 Data Poisoning Attacks Techniques

Data poisoning attacks represent a critical challenge to the reliability and fairness of deep learning-based recommendation systems. These attacks exploit vulnerabilities in the training dataset to manipulate the system's behavior, often resulting in biased or malicious recommendations that harm both users and system integrity. As illustrated in Fig. 2.5, data poisoning attacks can be broadly categorized into two primary types: flip-label attacks and clean-label attacks. Additionally, model poisoning, such as backdoor attacks, represents another significant vector for compromise.

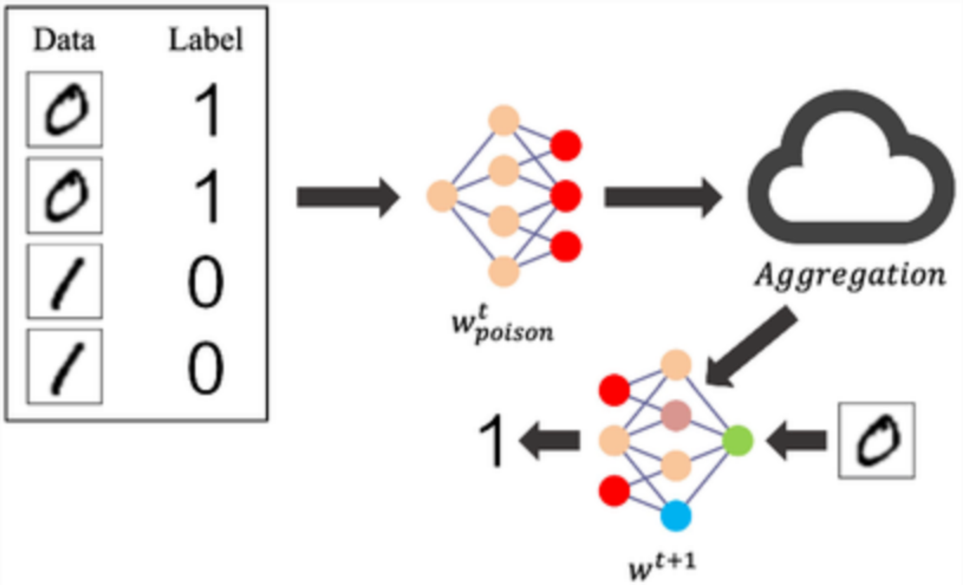


Figure 2.6: Flip-label Attacks {Source: [77]}.

Flip-label attacks target the labels in the training dataset, deliberately altering them to mislead the recommendation model as shown in Fig. 2.6. The attacker modifies the association between specific items and their ratings, introducing noise into the dataset [78][79]. For example, irrelevant or low-quality items can be falsely labeled with high ratings, causing the RS to prioritize these items over more relevant or authentic recommendations. This manipulation skews the underlying patterns the model learns during training, thereby degrading the recommendation quality for legitimate users [80]. Flip-label attacks are particularly effective because they exploit the inherent dependency of supervised learning models on accurate labels [81]. Detection mechanisms may struggle with this form of attack since the modified labels can blend seamlessly into the dataset, especially in cases where the dataset is large and sparsely labeled. Addressing this type of attack requires robust anomaly detection methods that can identify unusual patterns in the label distribution.

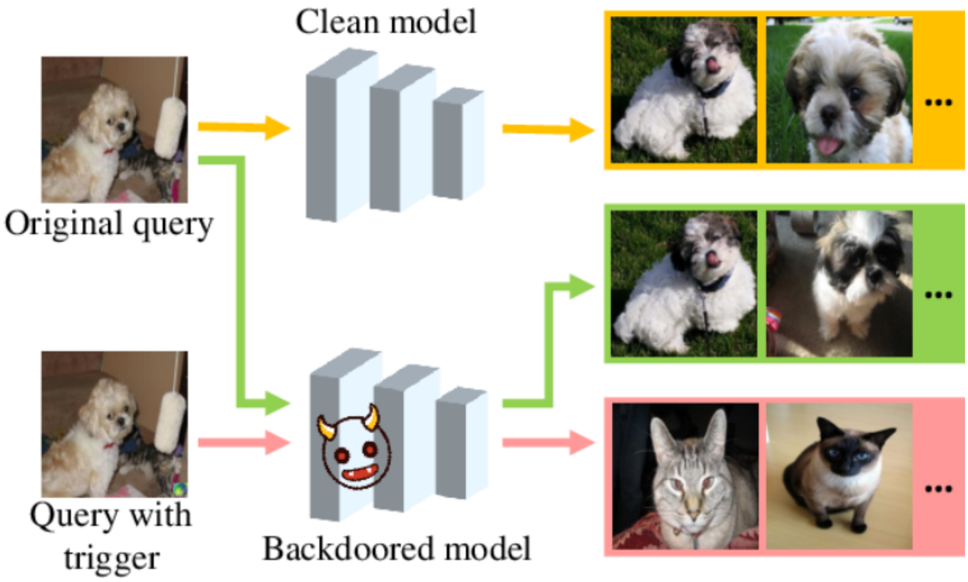


Figure 2.7: Clean-label Attacks {Source: [82]}.

Clean-label attacks, depicted in Fig. 2.7, represent a more subtle yet equally damaging strategy. In this case, the attacker modifies the input features of data samples without altering their associated labels [82]. For instance, an attacker might subtly change user behavior patterns or item attributes to embed malicious intent into the dataset. This manipulation can lead the RS to incorrectly associate specific items with positive feedback from users, thereby boosting their ranking in recommendations [83]. Clean-label attacks are particularly insidious because they bypass many traditional anomaly detection techniques, which often rely on identifying label inconsistencies. By focusing exclusively on input features, these attacks evade detection while embedding vulnerabilities into the RS. Mitigating clean-label attacks requires advanced feature engineering and robust data preprocessing techniques that can detect subtle shifts in feature distributions, as well as more stringent model evaluation during deployment.

In addition to data poisoning, model poisoning attacks such as backdoor attacks pose a significant threat. These attacks, as shown in Fig. 2.8, involve injecting triggers into the training dataset to create a hidden backdoor within the recommendation model. The backdoor activates under specific conditions during testing or deployment, enabling attackers to manipulate recommendations at will [84]. For example, an attacker might inject a unique pattern into certain user interactions, causing the system to prioritize a specific item whenever that pattern is present in new user data. This type of attack is particularly concerning because it leverages the trust placed in the training data and the assumption that the data is clean and unbiased. Backdoor attacks can remain dormant until activated, making them extremely difficult to detect [85] [86]. Countermeasures against backdoor attacks include employing adversarial training techniques, model validation on independent datasets, and regular audits of training data to ensure its integrity.

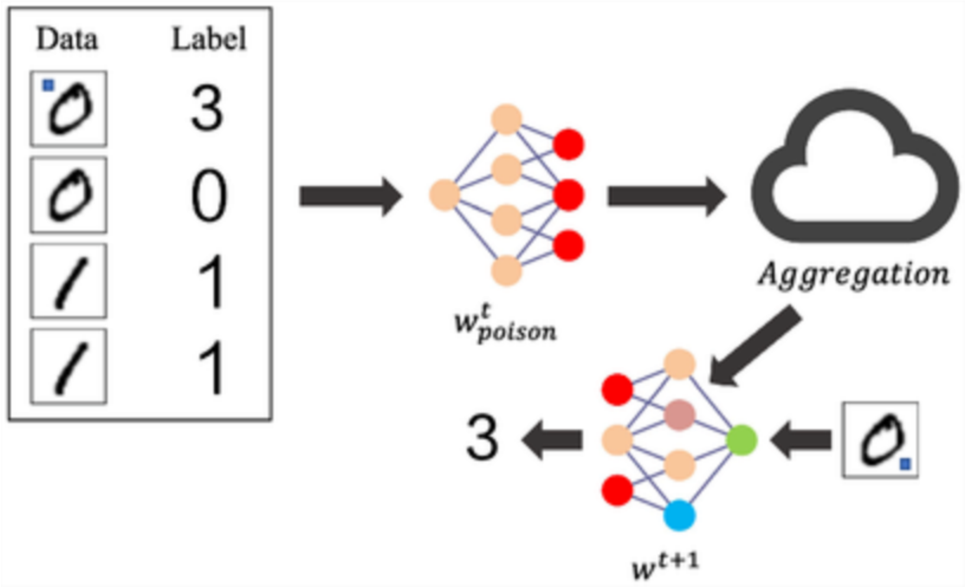


Figure 2.8: Backdoor Attacks {Source: [77]}.

The techniques discussed above exploit inherent vulnerabilities in the data-driven architecture of recommendation systems. Flip-label attacks erode the reliability of labeled data by altering the associations between items and user preferences, leading to skewed training outcomes and inaccurate recommendations. Meanwhile, clean-label attacks compromise the integrity of input features by embedding subtle manipulations that remain undetected by traditional anomaly detection systems, undermining the system's ability to distinguish malicious inputs from legitimate ones. Backdoor attacks, however, pose an even more significant threat by embedding dormant vulnerabilities directly into the model during training. These latent backdoors remain inactive until triggered under specific conditions, enabling attackers to exploit the model at any point after deployment. Such attacks are particularly insidious as they bypass existing validation protocols and undermine the sustained integrity of the recommendation framework. The effectiveness and subtlety of these attack vectors underscore the urgent need for advanced defensive mechanisms. Future defenses must prioritize robustness through adversarial training, data auditing, and dynamic validation processes to mitigate these evolving threats while preserving the performance and reliability of recommendation systems.

2.2.2 Defense Techniques

To counter these evolving threats, researchers and practitioners must implement a comprehensive and multi-faceted approach. Data cleaning and anomaly detection techniques serve as the first line of defense, enabling the identification and removal of poisoned data from training datasets [87][88]. These methods ensure that the integrity of the training data is maintained, mitigating the impact of flip-label and clean-label attacks. As illustrated in Fig. 2.5, robustness enhancement techniques such as adversarial training, regularization methods, and feature distribution analysis can significantly improve the model's resilience against sophisticated data poisoning and backdoor attacks [89][90][91]. Furthermore, during the deployment phase, empirical backdoor defenses, including independent validation, periodic model auditing, and trigger-free testing protocols, are critical to preventing the exploitation of latent vulnerabilities embedded in the model [92][93]. Combining these strategies provides a framework to safeguard recommendation systems, ensuring reliability and maintaining user trust while preserving system performance [94].

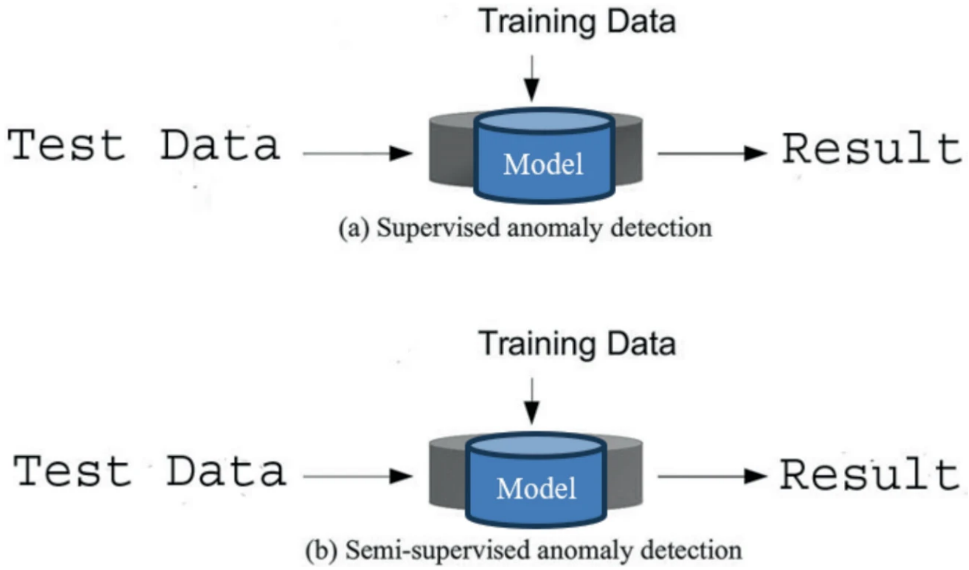

Training Data
Test Data
Model
Result
(a) Supervised anomaly detection
Training Data
Test Data
Model
Result
(b) Semi-supervised anomaly detection

Figure 2.9: Anomaly Detection {Source: [95]}.

Defensive techniques are crucial to safeguarding RS against data poisoning, presenting major risks to the robustness and trust of such recommendation frameworks. These techniques can be broadly categorized into three key areas: data cleaning, robustness enhancement, and empirical backdoor defenses. Data cleaning methods, such as anomaly detection, focus on identifying and removing suspicious or malicious data entries from the training dataset [88]. A typical anomaly detection method is shown in Fig. 2.9. By analyzing rating distributions, user activity patterns, or input-output correlations, these techniques aim to detect anomalies introduced by attackers [96]. For instance, rating distributions that deviate significantly from normal patterns or user behaviors that exhibit abnormal clustering can be flagged for further inspection. These methods not only minimize the risk of poisoned data corrupting the model but also ensure that the overall quality of the dataset remains intact. Advanced data cleaning techniques may also incorporate machine learning-based detection algorithms to dynamically adapt to evolving attack strategies, providing a proactive layer of protection.

Robustness enhancement strategies aim to make the RS less susceptible to data poisoning attacks by improving its ability to resist manipulation [97]. One widely used method is adversarial training, where models are deliberately exposed to adversarial examples during training to strengthen their resilience. As presented by Fig. 2.10. By incorporating examples of poisoned data into the training process, the model learns to identify and mitigate the impact of such inputs in real-world scenarios. Additionally, regularization techniques are employed to reduce overfitting, ensuring that the model does not disproportionately favor poisoned data during training. For example, techniques such as L_2 regularization penalize extreme weights in the model, limiting its susceptibility to data anomalies [98]. Together, these approaches improve the model's generalization capacity over both benign and adversarial data distributions, effectively reducing the influence of poisoned data on its outputs while maintaining the system's recommendation quality.

Empirical backdoor defenses specifically target backdoor attacks, which are particularly challenging due to their ability to remain dormant until triggered by specific conditions. As illustrated in Fig. 2.11, backdoor attacks introduce a trigger, such as a specific word, character, or sentence, into the training dataset, resulting in the creation of an infected deep neural network (DNN). During training, these triggers are embedded alongside carefully crafted malicious labels, causing the DNN to function normally under standard conditions while secretly embedding a vulnerability. This infected DNN remains

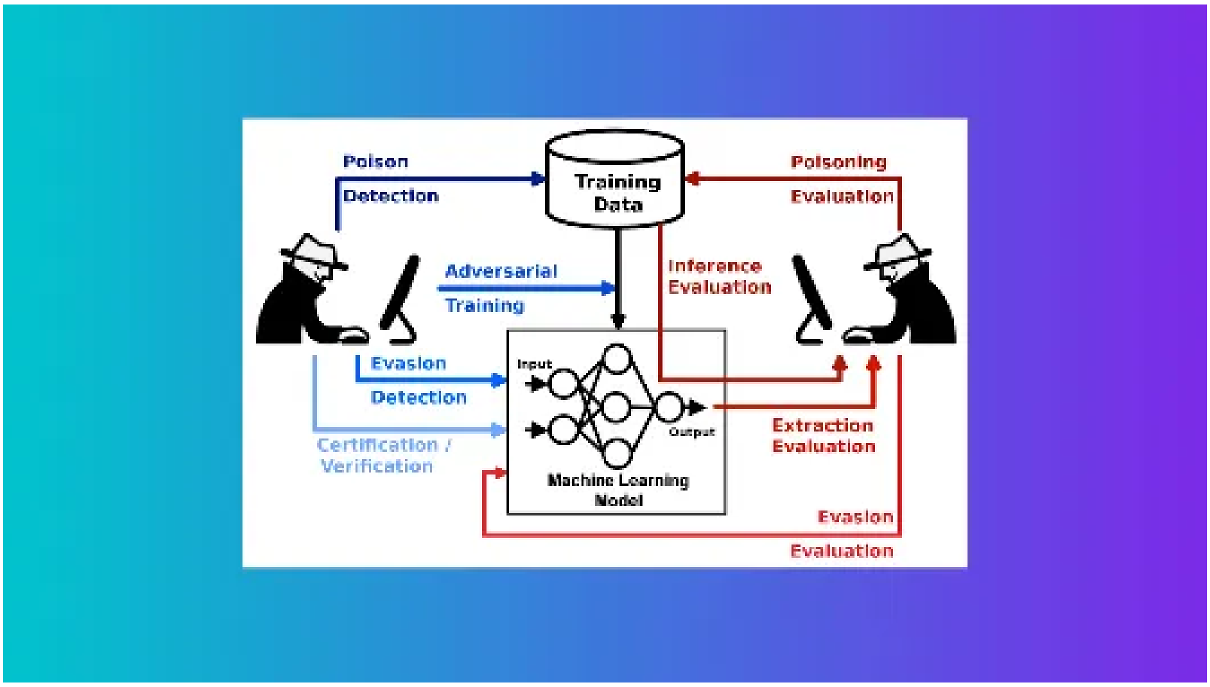


Figure 2.10: Adversarial Training in Robustness Enhancement {Source: [99]}.

undetected until the trigger is present during inference, at which point the backdoor activates. The model then misclassifies the input, producing outputs that align with the attacker's objectives. The ability of backdoor attacks to remain dormant and activate only under specific conditions makes them especially difficult to detect, posing a significant challenge to the security and reliability of RS and other machine learning models.

These defenses analyze feature distributions within the data to identify hidden triggers introduced by attackers. For example, as shown in Fig. 2.11, discrepancies in feature distribution, such as unusual clusters, outlier patterns, or data points with unique characteristics, they can signal the presence of backdoor vulnerabilities. Attackers often embed triggers that subtly alter the data's distribution, creating anomalies that can be detected through careful analysis. Techniques like feature distribution analysis, independent validation, and trigger-free testing are employed to identify and neutralize these embedded triggers. Additionally, advanced methods such as trigger isolation and reconstruction are used to pinpoint and remove malicious patterns injected during the training phase. By isolating these patterns, these defenses can mitigate backdoor risks while preserving the model's integrity. Periodic audits and real-time monitoring further enhance these approaches, ensuring the recommendation system remains robust against evolving attack

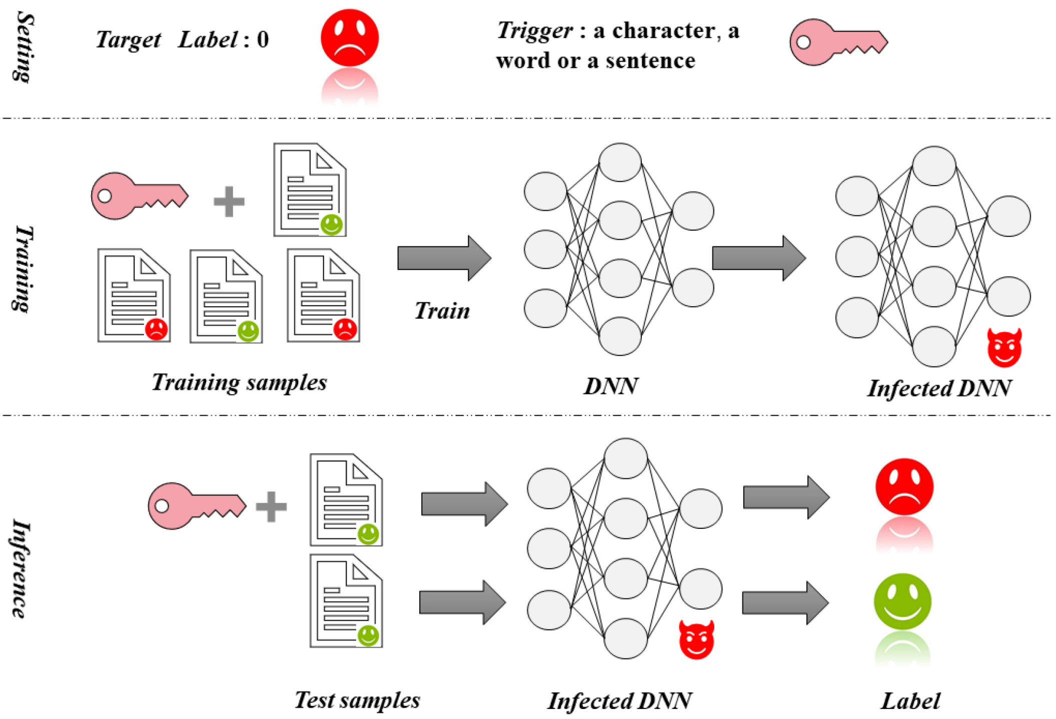


Figure 2.11: Backdoor Defense in Robustness Enhancement {Source: [100]}.

strategies. Together, these strategies provide a comprehensive framework for safeguarding systems from backdoor vulnerabilities.

Periodic audits of training data and model outputs are also crucial, as they enable the timely identification of new triggers that attackers may attempt to introduce over time. These audits involve examining both historical and newly acquired data to detect irregularities or hidden patterns that may indicate backdoor vulnerabilities [101]. As demonstrated in Fig. 2.11, integrating these audits with existing defense mechanisms provides an added layer of protection against evolving attack strategies. Together, these methods form a comprehensive framework to mitigate backdoor risks, ensuring the robustness and reliability of recommendation systems in diverse operational environments [102][103]. By proactively addressing backdoor threats, these defenses help maintain the performance, security, and integrity of RS models, safeguarding them against increasingly sophisticated methodologies while ensuring a seamless user experience.

2.3 Recommendation Unlearning Verification

Building on the challenges and countermeasures discussed in Section 2.2 regarding data poisoning attacks, it becomes evident that ensuring data integrity is only part of the solution for safeguarding recommendation systems. With the increasing emphasis on user privacy and compliance with global regulatory frameworks like GDPR and CCPA, the ability to selectively unlearn specific user data upon request has emerged as a crucial requirement. Recommendation unlearning addresses situations where a user demands their data to be deleted while ensuring that this removal is accurately reflected in the RS model [104]. However, implementing such mechanisms introduces complex challenges, particularly in verifying whether the data has been completely removed and ensuring the model remains robust and functional. Moreover, achieving this balance between compliance and performance is critical for sustaining trust and adoption of RS technologies.

The significance of recommendation unlearning lies in its dual objectives of enhancing user privacy and maintaining system usability. Modern RS face increasing scrutiny from users and regulators to comply with privacy mandates without compromising the quality of their recommendations. While privacy-preserving mechanisms address user concerns, they must also ensure that unlearning specific data does not destabilize the model or degrade recommendation performance. For instance, overly aggressive unlearning could negatively impact the RS's predictive accuracy or introduce instability in its outputs. In this context, regulatory compliance frameworks such as GDPR demand robust privacy practices while emphasizing user rights over their data [105]. Effective unlearning processes must seamlessly meet these regulatory requirements and technical challenges while ensuring that the RS retains its functionality and accuracy for other users.

As depicted in Section 2.2, there is competition among researchers and organizations to develop unlearning algorithms, yet verifying the success of these methods remains ambiguous and complex. Unlearning verification requires rigorous testing to ensure that removed data no longer influences the model while avoiding unintended consequences, such as performance degradation. Complications arise when unlearning processes overlap with other innovations, leading to uncertainties about their effectiveness. Additionally, verifying unlearning success often requires tailored testing environments, further complicating standardization. With RS becoming integral to e-commerce, social media, and personalized services, the stakes for unlearning verification are higher than ever, as both user trust and legal compliance hinge on effective solutions to these challenges.

For approaches to model unlearning verification, they include strategies like backdoor triggers, probabilistic verification, and fingerprinting. Backdoor triggers can embed identifiable patterns into the model during training, serving as markers to assess whether unlearning has occurred. This technique allows evaluators to check whether these markers persist after the unlearning process, providing concrete evidence of its success or failure. Probabilistic verification, on the other hand, uses statistical models to estimate the likelihood that specific data has been fully unlearned. Fingerprinting techniques further analyze the model's outputs to detect residual influences of removed data, ensuring the data's complete elimination. Together, these methods form a multi-faceted framework for unlearning verification, addressing both privacy concerns and technical challenges while maintaining system performance and reliability.

In this context, Recommendation Unlearning Verification (RUV) emerges as a critical area of research, aiming to bridge the gaps between privacy preservation, legal compliance, and the maintenance of reliable system performance in recommendation systems. As data privacy regulations such as GDPR and CCPA continue to evolve, the ability to selectively unlearn data upon user requests becomes not only a technical necessity but also a legal obligation. RUV ensures that data removal processes are verifiable and do not leave residual effects on the model, thereby fostering user trust. By addressing the inherent challenges, such as potential degradation of model performance or incomplete data removal, and leveraging innovative verification techniques like backdoor triggers and probabilistic verification, RUV offers a pathway to build more robust and trustworthy RS. Future developments in RUV hold the potential to enable RS to meet privacy demands while sustaining high-quality recommendations. This section lays the groundwork for exploring advanced unlearning algorithms, their challenges, and practical applications in ensuring both privacy compliance and model integrity.

2.3.1 Machine Unlearning

Machine unlearning has become a vital research area due to its necessity in addressing privacy concerns, ensuring data availability, and complying with legal frameworks like the "right to be forgotten". This concept has gained prominence as machine learning systems increasingly incorporate sensitive user data, creating potential privacy risks [106][107]. Machine unlearning focuses on effectively removing specific data samples from a trained model while preserving its functionality as if the data had never been included. This ability to ”forget” user data is essential in scenarios where individuals request the deletion of their contributions to comply with regulations such as GDPR or CCPA [108]. It is particularly significant in recommendation systems, where personal data plays a central role in generating accurate predictions.

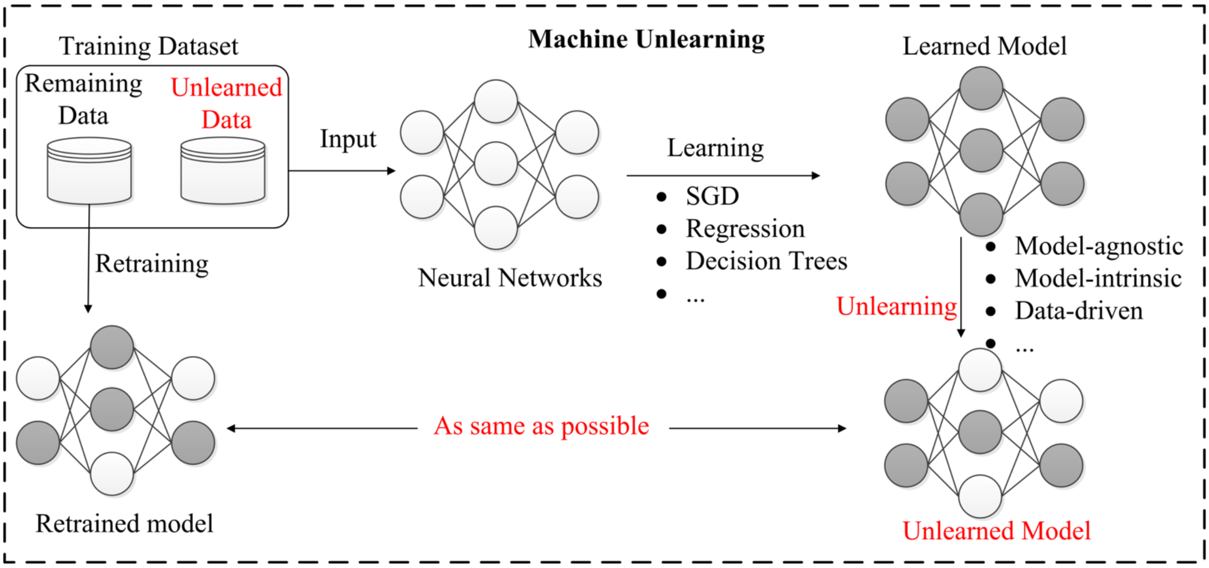


Figure 2.12: Machine Unlearning Technique.

As depicted in Fig. 2.12, machine unlearning operates through two main approaches: data reorganization and model manipulation, each addressing privacy challenges from different angles. Data reorganization involves removing unwanted data samples from the training dataset and retraining the model with the remaining data. Although effective, this approach is computationally expensive, especially for large-scale models or frequent unlearning requests. In contrast, model manipulation modifies the already-trained model by identifying and eliminating the influence of specific data samples. Techniques such as gradient reversal, weight adjustments, or targeted parameter updates allow for efficient and precise unlearning without retraining the model from scratch. Both approaches aim to ensure that the resulting model functions as if the excluded data had never been used, balancing privacy compliance with system performance.

In the data reorganization method, the training dataset is partitioned into two subsets: remaining data and unlearned data. The unlearned data, in red in the accompanying figure, represents the specific samples that must be excluded from the model. This process begins by identifying and isolating these data points to ensure they no longer contribute to the learning process. To achieve this, the model undergoes a retraining procedure using only the remaining data subset. The aim is to generate a retrained model that preserves the functionality and performance of the original model while completely eliminating the influence of the excluded data. This ensures compliance with privacy regulations.

While this approach is highly effective in achieving the desired unlearning outcome, it comes with significant computational challenges. Retraining a model, particularly for large-scale datasets or complex deep learning architectures, requires substantial computational resources and time. Moreover, if unlearning requests occur frequently or involve a large volume of data, the repeated retraining cycles can strain system resources and reduce overall efficiency. Despite these challenges, the data reorganization method remains a cornerstone in machine unlearning research, providing a robust framework for ensuring privacy while maintaining model reliability and accuracy. This trade-off between privacy and efficiency continues to drive innovations in this area.

In contrast, model manipulation offers an efficient alternative to retraining by directly modifying the learned model to erase the contributions of specific data without starting the training process from scratch. This approach is particularly advantageous for scenarios where retraining would be computationally expensive or infeasible due to the size and complexity of the dataset. Model manipulation encompasses techniques that are broadly categorized as model-agnostic, model-intrinsic, and data-driven. As illustrated in Fig. 2.12, these methods analyze key aspects of the model, such as gradients, weights, or contributions, to identify and eliminate the effects of the unlearned data.

For instance, model-intrinsic approaches leverage the internal parameters of deep neural networks, such as weight adjustments or gradient updates, to directly remove the impact of specific data points. This method ensures that the model's structure and performance are preserved while achieving the desired unlearning effect. Model-agnostic techniques, on the other hand, operate externally, treating the model as a black box. These techniques utilize external optimization or inference processes to mitigate the influence of the unlearned data without needing detailed access to the model's internal parameters. Finally, data-driven methods focus on identifying patterns or dependencies within the dataset to guide unlearning. By combining these strategies, model manipulation provides a robust and efficient framework for implementing machine unlearning in diverse applications, addressing privacy and operational challenges effectively.

Fig. 2.12 illustrates the unlearning process step-by-step, providing a visual representation of how machine unlearning operates in practice. The process begins with the training of the original model using a dataset that includes both the retained and the unlearned data samples. These unlearned samples are later identified for exclusion based on user requests or privacy regulations, such as the “right to be forgotten”. Once the unlearning request is triggered, the process diverges depending on the chosen method. In the case of data reorganization, the unlearned data is removed, and the remaining data is used to retrain the model from scratch, ensuring that the unlearned samples no longer influence the model’s predictions. Alternatively, in model manipulation, parameter adjustments are applied directly to the model without requiring retraining. This step is critical as it ensures that the resulting “unlearned model” approximates the performance of a model trained solely on the retained data. The figure emphasizes that the goal of unlearning is to mitigate the influence of specific data while maintaining the integrity, functionality, and performance of the model. These approaches are crucial in addressing both privacy concerns and computational challenges in modern machine learning systems.

Recent advances in machine unlearning have focused on refining its methodologies to address increasingly complex requirements across diverse applications. One notable example is the work by Yuan et al., who proposed a federated recommendation framework to effectively erase user contributions during federated learning [109]. This framework tackles unique challenges associated with distributed data privacy, where user data is often fragmented across multiple devices or systems. By developing mechanisms to unlearn specific user contributions without requiring centralized data storage, their method preserves the decentralized nature of federated learning while maintaining privacy compliance. Such innovations highlight the growing importance of unlearning techniques in systems that rely on large-scale, distributed datasets.

Moreover, machine unlearning has evolved to balance the competing demands of privacy, accuracy, and computational efficiency. As privacy regulations such as GDPR and CCPA continue to impose stricter requirements, unlearning has become a necessary feature for organizations managing sensitive data. Additionally, the integration of advanced techniques, such as gradient analysis, parameter isolation, and distributed retraining, ensures that unlearning can be achieved without significantly degrading system performance. These advancements underscore the potential of machine unlearning to serve as a cornerstone of modern recommendation systems, offering robust solutions for privacy-preserving, high-performance models.

2.3.2 Recommendation Unlearning

Recommendation systems personalize services by learning user preferences and generating tailored recommendations. However, in certain circumstances, it becomes necessary to remove specific training data from these models, a process referred to as recommendation unlearning [110]. This process is driven by several key motivations. First, privacy protection is a critical factor, as users may want to eliminate sensitive or personal data from the model to prevent potential privacy breaches or misuse of their information. This aspect is particularly significant in today's privacy-conscious world, where regulations such as GDPR emphasize the right to data deletion. Second, utility enhancement plays an important role. Models often encounter incorrect, outdated, or harmful data during training, which can adversely affect their performance. By removing such data, the model can recover its predictive accuracy and deliver better results. Finally, usability improvement is achieved by eliminating noisy, redundant, or inaccurate data, allowing the model to generate recommendations with greater specificity and contextual relevance.

Traditional machine unlearning methods, primarily developed for image and text data, fail to consider the collaborative information inherent in recommendation systems, making them unsuitable for direct application. To address this, researchers have proposed RecEraser in Fig. 2.13, a general and efficient machine unlearning framework designed specifically for recommendation tasks. The core idea of RecEraser is to divide the training dataset into multiple subsets and train a sub-model on each subset. Specifically, RecEraser involves three main steps: first, data partitioning, using three novel algorithms to group training data based on similarity, preserving collaborative information; second, sub-model training, where a model is trained on each subset; and third, adaptive aggregation, combining predictions from multiple sub-models to enhance the effectiveness of the final aggregated output. It conducted on three public benchmark datasets demonstrate that RecEraser achieves not only efficient unlearning but also superior utility compared to state-of-the-art unlearning methods. Moreover, the researchers have made RecEraser's

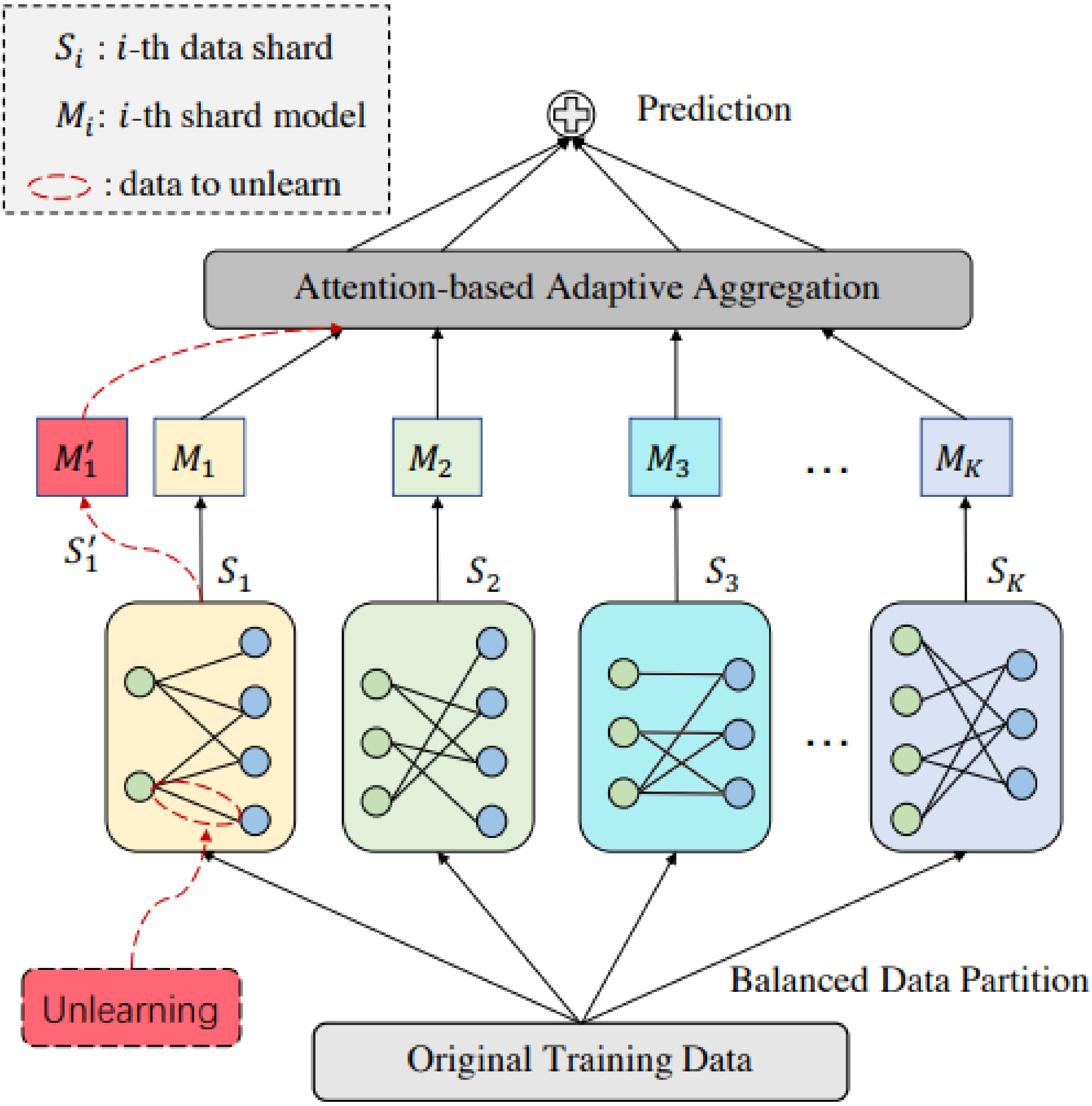


Figure 2.13: Recommendation Unlearning {Source: [104]}.

source code publicly available, enabling further research and practical applications.

In addition to RecEraser, other innovative methods for recommendation unlearning have emerged in recent years. One notable approach is the Influence Function-based Recommendation Unlearning framework (IFRU), which leverages influence functions to estimate the direct and indirect impacts of unavailable data on a trained model [111]. By doing so, IFRU enables efficient updates to the model without the need for a full retraining process, which is often time-consuming. This framework provides a practical solution for scenarios where training data must be removed due to user requests, privacy concerns, or the discovery of erroneous data. Experimental results validate the effectiveness of IFRU, demonstrating that it can maintain the performance of recommendation systems while being over 250 times faster than traditional retraining methods. This speed advantage is significant in large-scale applications where retraining from scratch is infeasible. Additionally, IFRU's design allows for flexibility, making it applicable to a wide range of recommendation algorithms. By addressing the challenge of efficient unlearning, IFRU represents a major step forward in aligning recommendation systems with data privacy requirements and improving their adaptability in dynamic environments.

Another promising approach for recommendation unlearning is Interaction and Mapping Matrices Correction (IMCorrect), which focuses on efficiently removing the effects of specific data points by directly adjusting the interaction and mapping matrices used in recommendation algorithms [112]. This method avoids the computational overhead associated with retraining while achieving a high level of accuracy in unlearning. IMCorrect identifies and modifies the components of the model most influenced by the data to be removed, ensuring that the unlearning process is both targeted and effective. This approach is particularly well-suited for real-time applications, where rapid adaptation to data removal requests is crucial. Furthermore, experiments have shown that IMCorrect maintains model performance and scalability, making it a practical solution for large-scale recommendation systems. Unlike traditional methods, which may require significant computational resources and time, IMCorrect provides a lightweight yet robust alternative. By addressing the limitations of retraining-based approaches, IMCorrect contributes to the broader goal of creating privacy-conscious, user-centric recommendation systems that can dynamically respond to changes in their data landscape.

These advancements, including IFRU and IMCorrect, provide novel technical pathways for implementing unlearning in recommendation systems. They offer efficient, scalable, and practical solutions that address challenges posed by data privacy regulations and evolving user demands. Both methods demonstrate how modern recommendation systems can balance competing priorities: protecting user privacy, enhancing system utility, and improving usability. By removing sensitive, erroneous, or outdated data without sacrificing performance or efficiency, these approaches align recommendation systems with ethical and legal standards while maintaining operational effectiveness. As data privacy and user control become increasingly important in machine learning applications, these innovative frameworks pave the way for the future development of RS, ensuring they remain adaptable, responsible, and high-performing in dynamic environments.

2.3.3 Machine Unlearning Verification

Machine unlearning has become a cornerstone for promoting ethical Artificial Intelligence (AI) usage and maintaining transparency in data-driven systems. Its role extends beyond mere compliance with regulations like GDPR, serving as a fundamental enabler of user control over personal information. By allowing users to revoke their data from machine learning models, unlearning helps mitigate risks associated with long-term data retention, such as security vulnerabilities and misuse of sensitive information. Moreover, it fosters an environment where users can trust AI systems to respect their privacy and autonomy. However, the increasing reliance on unlearning techniques has also exposed inherent risks, particularly obfuscation issues, as illustrated in Fig. 2.14. These risks highlight the gaps between theoretical unlearning and its practical implementation, emphasizing the necessity of innovative frameworks to validate and reinforce the integrity of unlearning processes, ensuring they truly serve the intended purpose.

Obfuscation risks in machine unlearning arise when service providers assert that they have successfully removed data from a model, but the model may still retain subtle, hidden traces of the deleted information. These residual traces can undermine user privacy and violate data protection requirements, creating a deceptive sense of security for users who trust the system to honor their deletion requests. Such risks are particularly problematic because they defeat the fundamental purpose of unlearning: to remove all influence of the deleted data from the model's predictions and decision-making processes. Fig. 2.14

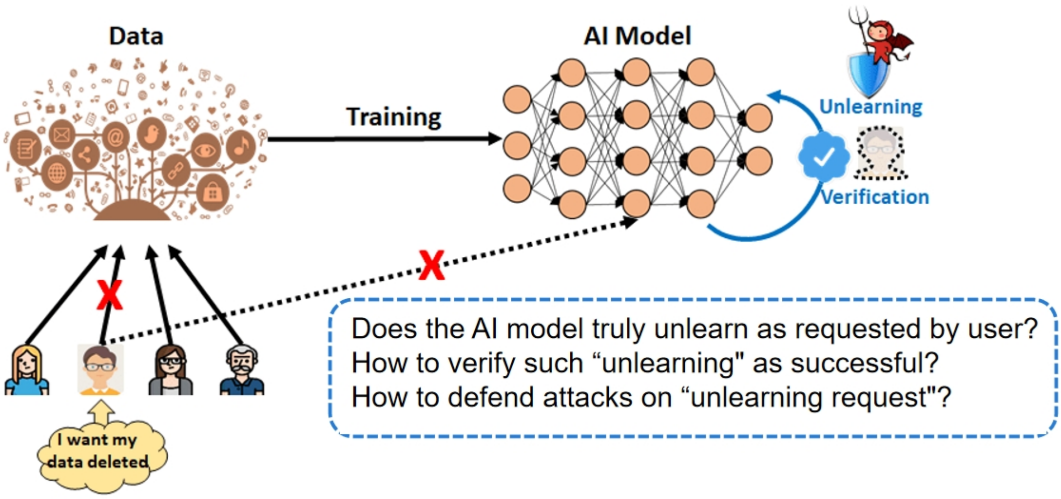
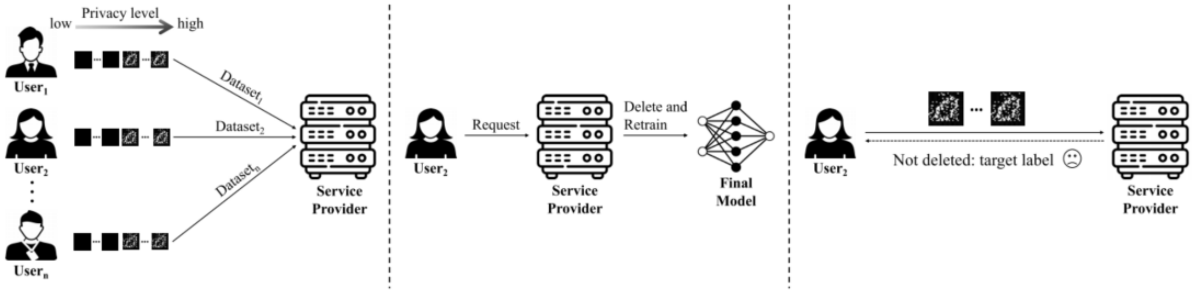

Data
Training
AI Model
Unlearning
Verification
X
X
I want my data deleted
Does the AI model truly unlearn as requested by user?
How to verify such "unlearning" as successful?
How to defend attacks on "unlearning request"?

Figure 2.14: Machine Unlearning is Facing Obfuscation Risks {Source: [113]}.

underscores three pivotal challenges in verifying machine unlearning. First, ensuring that the AI model genuinely forgets the requested data is critical. This involves assessing whether the model's behavior, predictions, or internal representations are entirely free from the influence of the deleted data. Second, verifying unlearning success is a complex task. Existing methods often fall short in providing transparent, quantifiable measures to confirm that the data's impact has been erased. Third, the system must be resilient against attacks on unlearning requests. For example, attackers might submit fraudulent deletion requests to distort the model or identify ways to exploit residual dependencies on supposedly deleted data. These challenges highlight the pressing need for reliable, efficient, and robust verification frameworks to address these risks [113].

The challenges associated with verifying machine unlearning largely arise from the intrinsic non-transparency of learning algorithms, which often operate as black boxes with limited insight into their internal workings. This lack of transparency makes it difficult to determine whether the unlearning process has genuinely removed the influence of the deleted data or if the model still retains latent traces. In many cases, traditional evaluation techniques fail to provide clear evidence of successful unlearning, leaving users and regulators uncertain about the reliability of these processes. Furthermore, the absence of standardized methods for verifying unlearning increases the risk of superficial implementations, where the data appears to be removed but its effects persist in subtle ways. These limitations underscore the urgent need for advanced and transparent verification mechanisms that not only confirm the complete removal of data but also protect systems from adversarial manipulation. By addressing these gaps, future verification frameworks can ensure that unlearning systems are both reliable and resistant to exploitation.



(a) Backdoor injection and model training (b) Data deletion and model retraining (c) Machine unlearning validation

Figure 2.15: Machine Unlearning Verification via Backdoor Triggers {Source: [114]}.

In response to the difficulties inherent in verifying machine unlearning, researchers have explored innovative solutions to guarantee that the model reliably and authentically eliminates the specified data. One particularly promising approach, illustrated in Fig. 2.15, involves the use of backdoor triggers as a mechanism for reliable verification. These backdoor triggers are unique patterns deliberately embedded into the training data during the model's initial training process. Unlike conventional methods, these triggers serve as invisible markers that remain undetectable under normal operations but can be specifically activated to test whether the model still retains traces of the removed data [114]. The strength of this approach lies in its ability to track and measure the model's residual dependency on deleted data. By analyzing the model's response to these triggers after an unlearning request, researchers can objectively determine whether the process was successful. If the model continues to respond to the backdoor triggers, it indicates incomplete unlearning. This methodology not only enhances the transparency of unlearning processes but also establishes a standardized framework for accountability, ensuring that service providers comply with privacy and unlearning requirements.

The process of verifying unlearning using backdoor triggers begins by embedding unique backdoor patterns into the training dataset, as shown in Fig. 2.15. These patterns are imperceptible to end-users and have no apparent effect on normal model operations. However, they serve as precise identifiers for tracking the data's influence within the trained model. When a user submits a request to delete specific data (Fig. 2.15(b)), the service provider is obligated to remove the corresponding data and either retrain or fine-tune the model to ensure compliance with the unlearning request. In the final step (Fig. 2.15(c)), the effectiveness of the unlearning process is validated by testing the model's response to the previously injected backdoor triggers. If the model still reacts to these triggers or exhibits behavior influenced by the deleted data, it indicates that the unlearning process was incomplete or flawed. This approach ensures a robust evaluation of unlearning efficacy by exposing subtle dependencies the model may retain. Moreover, it offers a structured framework for assessing compliance with user deletion requests, ensuring that the model no longer relies on or is influenced by data it was instructed to forget. This methodology reinforces user trust and accountability in machine learning.

This approach offers a highly effective and systematic framework for verifying the success of machine unlearning processes. By embedding backdoor triggers into the training data, researchers can evaluate whether the influence of deleted data has been completely removed. The use of these hidden markers makes it possible to detect residual traces of data that may persist in the model, even after an unlearning process has been claimed as successful. Furthermore, this method is particularly valuable for exposing potential obfuscation attempts by service providers who might assert compliance without fully removing the data's impact. By providing an objective and transparent mechanism for evaluation, the integration of backdoor triggers enhances accountability in machine learning. It also reinforces user trust by ensuring their data deletion requests are genuinely honored, thereby safeguarding privacy and promoting ethical practices in AI systems.

While the field of machine unlearning verification has seen significant advancements, the technologies and methodologies specifically tailored for recommendation unlearning verification remain underdeveloped. Recommendation systems present unique challenges due to their reliance on collaborative filtering, user-item interactions, and implicit feedback, which complicate the verification process. Ensuring that a recommendation model has truly forgotten specific data requires novel approaches that account for the interdependencies among users and items. In this context, existing verification techniques for general unlearning may not be directly applicable. Therefore, this topic will delve into the exploration of innovative methods and frameworks specifically designed to address the unique demands of recommendation unlearning verification, paving the way for more robust and reliable solutions in this domain.

2.4 Recommendation System Model Ownership Verification

Building on the challenges and countermeasures discussed in Section 2.2 regarding data poisoning attacks and the mechanisms explored in Section 2.3 for ensuring privacy through unlearning, it becomes clear that addressing data privacy and integrity represents only part of the solution for safeguarding recommendation systems. As recommendation systems increasingly operate in sensitive and competitive domains, the issue of model ownership verification has emerged as a critical priority. Ensuring that a deployed RS model is genuinely owned and maintained by the claimed entity is essential for protecting intellectual property, preventing unauthorized use, and ensuring the overall security of the system. This becomes especially important in contexts involving the distribution of RS models across multiple platforms, deployed in third-party environments, or vulnerable to theft or replication, as these situations heighten the risk of intellectual property violations.

This section delves into the technical challenges involved in verifying model ownership while ensuring that the RS maintains its functionality, efficiency, and security. Model ownership verification is a critical aspect of protecting intellectual property and preventing unauthorized usage, especially in scenarios where RS models are deployed on shared or third-party platforms. The ability to verify ownership is essential not only for safeguarding proprietary technologies but also for maintaining trust in RS systems, which are integral to industries such as e-commerce, social media, and personalized services. Achieving this requires robust mechanisms that can accurately confirm ownership without imposing significant computational or performance burdens on the system.

Striking a balance between robust ownership verification and minimal performance impact is a challenging yet vital goal. Effective ownership verification fosters trust between stakeholders, ensuring that RS solutions are used as intended and remain secure against replication or theft. Furthermore, by protecting intellectual property and promoting accountability, ownership verification mechanisms encourage the broader adoption of secure and reliable RS technologies in collaborative and competitive environments. As the landscape of recommendation system applications continues to grow, advancing ownership verification techniques will play a pivotal role in ensuring both system security and the equitable use of these powerful tools.

The significance of model ownership verification becomes even more apparent when considering the broader context of Sections 2.2 and 2.3, which address the integrity of training data and the privacy of user data, respectively. While data privacy and unlearning are integral to user trust and regulatory compliance, they do not address the risks posed by unauthorized access to or replication of RS models. By shifting the focus to model-level security, ownership verification ensures the long-term sustainability and accountability of RS deployments. Future advancements in this area must not only meet the technical demands of verification but also align with broader legal and ethical standards, ensuring that RS technologies remain both secure and trustworthy.

2.4.1 Obfuscation Risks in Model Ownership

Deep learning models have emerged as a cornerstone of modern technology, driving innovations across a wide range of applications, including recommendation systems, computer vision, natural language processing, healthcare diagnostics, and autonomous systems. Their remarkable ability to learn complex patterns from data has not only transformed industries but also created a significant demand for high-performing models capable of solving specialized problems. This widespread adoption has led to a competitive landscape where organizations invest substantial resources in developing proprietary deep-learning models, often representing their most valuable intellectual property.

However, with this rapid growth comes significant challenges, particularly in ensuring model ownership and protecting these assets. Deep learning models, being complex yet highly transferable, face increasing risks of theft and ownership obfuscation. Attackers can exploit vulnerabilities in deployment environments, such as cloud-based Machine Learning as a Service (MLaaS) platforms or end-user devices, to steal or replicate these models. Once stolen, models may be modified through techniques like fine-tuning or distillation, further masking their origins. These challenges highlight the critical need for effective security measures, such as ownership verification and watermarking, to protect the economic value of these models while ensuring their ethical and legal use in a competitive and rapidly evolving technological landscape.

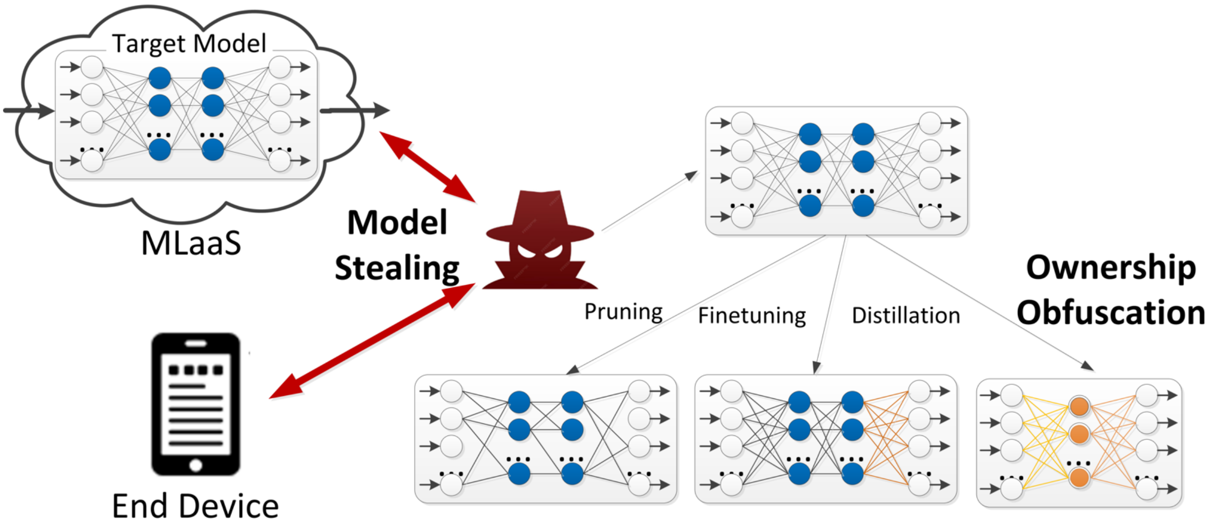


Figure 2.16: Model Ownership is Facing Obfuscation Risks {Source: [115]}.

As depicted in Fig. 2.16, attackers can exploit the deployment of models on cloud platforms, such as MLaaS, or edge devices to carry out model-stealing attacks. These attacks enable adversaries to extract confidential models using various sophisticated techniques. One common method is Application Programming Interface (API) probing, where attackers send repeated queries to the model's public-facing API to reconstruct its decision boundaries or parameters. Another approach involves side-channel analysis, which leverages indirect information, such as hardware usage or timing patterns, to infer the internal workings of the model. Once stolen, the extracted models are often subjected to further transformations, including pruning, fine-tuning, or distillation. These methods modify the original model by simplifying or adapting it, effectively creating a derivative version that is difficult to trace back to its rightful owner. For example, pruning removes unnecessary parameters, while fine-tuning adjusts the model for new tasks. Distillation transfers the knowledge of the original model to a smaller, less identifiable one [115].

These processes enable attackers to obfuscate the origins of stolen models, making it difficult for rightful owners to assert ownership. By disguising the model's lineage through techniques like fine-tuning, pruning, or distillation, adversaries can effectively undermine the economic and intellectual property value of the original model. This creates significant challenges for organizations that invest heavily in the development of proprietary deep learning models, as their competitive edge and innovation may be compromised. Moreover, the unauthorized use of stolen models can lead to ethical and legal violations, further exacerbating the risks. These challenges highlight the urgent need for robust security mechanisms, such as watermarking and ownership verification techniques. These methods can provide tamper-proof identifiers and evidence of ownership, ensuring that models remain protected against theft, misuse, and unauthorized deployment in increasingly vulnerable and competitive environments.

2.4.2 Model Ownership Verification

Model ownership verification has emerged as a crucial research area to combat the escalating risks of theft and obfuscation in deep learning models. As deep learning continues to power numerous applications, adversaries are increasingly exploiting publicly available or shared datasets to train unauthorized models. These activities pose significant challenges for dataset and model owners, making it difficult to assert their intellectual property rights and protect the economic value of their investments.

As depicted in Fig. 2.17, adversaries can use these datasets to create unauthorized models, which may then be fine-tuned or modified to mask their origins, further complicating dataset ownership claims. Similarly, several effective methods for model ownership verification have been proposed, including dataset watermarking and trigger-based testing. These techniques embed unique identifiers into datasets or model behavior, allowing owners to verify unauthorized usage. Such verification mechanisms provide robust evidence to identify theft, enforce ownership rights, and deter malicious actors, thereby safeguarding the integrity and value of deep learning systems [116].

Recent advancements in ownership verification methodologies have introduced a systematic and structured approach to safeguarding intellectual property in deep learning models. One highly effective strategy involves embedding imperceptible yet unique watermarks into training datasets before their release. These watermarks act as proprietary markers that remain undetectable during normal use but can later be leveraged to es-

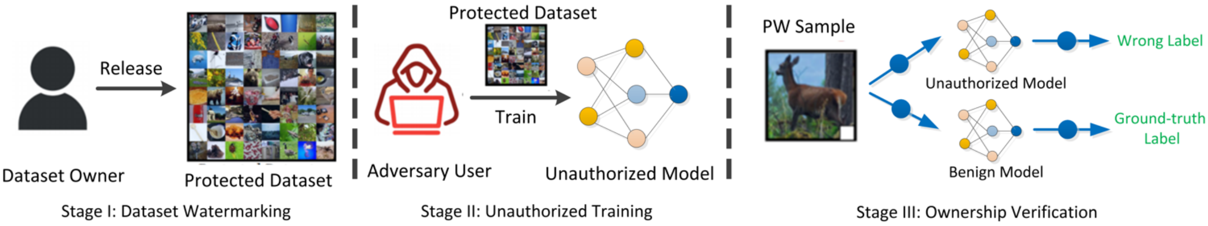


Figure 2.17: Dataset Ownership Verification via Backdoor Watermarking {Source: [116]}.

tablish ownership. The key advantage of this method lies in its ability to preserve the usability and quality of the dataset without compromising the model's training performance. For instance, researchers can integrate subtle perturbations or specific patterns into data points to create a protected dataset. This ensures that the embedded identifiers are not only robust but also seamlessly integrated into the data, providing a foundation for verifying ownership in adversarial scenarios.

When an adversary uses a protected dataset containing embedded watermarks to train an unauthorized model, the proprietary markers are inadvertently retained in the resulting model. Importantly, these watermarks remain intact because the adversary is unaware of their existence or the specific mechanism used to embed them. This unique feature allows the legitimate model owner to verify ownership through specially designed trigger samples. By testing these samples on the unauthorized model, discrepancies in output can be observed. For example, unauthorized models may produce incorrect or nonsensical labels, while the legitimate model outputs the correct ground-truth labels. This discrepancy provides a concrete and defensible method for exposing unauthorized usage. Such techniques strengthen ownership claims by offering a robust, evidence-based framework for protecting intellectual property in complex machine-learning systems.

These ownership verification techniques underscore the inherent vulnerabilities of publicly available or shared datasets, which are often subject to exploitation. By embedding watermarks into datasets, researchers provide an essential layer of protection that not only deters unauthorized use but also facilitates accountability in cases of intellectual property disputes. This process highlights the broader importance of dataset watermarking within the intellectual property protection framework, particularly as deep learning becomes integral to sensitive and competitive industries. Combining dataset watermarking with trigger-based testing creates a comprehensive pipeline for ownership verification that is both practical and effective. However, the rapid evolution of adversarial techniques poses significant challenges. As adversaries develop increasingly sophisticated methods to bypass watermarking protections, continuous innovation in ownership verification strategies is crucial for ensuring the long-term security and trustworthiness of deep learning.

2.4.3 Ambiguity in Recommendation System Intellectual Property

The intellectual property (IP) of recommendation systems based on deep learning has become an increasingly ambiguous and contentious topic in recent years. RS is foundational to industries such as e-commerce, social media, and content platforms, where they are widely deployed to deliver personalized recommendations and improve user engagement. These systems often rely on sophisticated algorithms and proprietary deep learning models, making their intellectual property highly valuable [117]. However, the rapid advancement of cutting-edge AI techniques has introduced substantial challenges in protecting and enforcing ownership of these models. As RS becomes more integral to business success, its value as a competitive asset has surged, driving the need for clear IP frameworks. Without robust mechanisms to safeguard Recommendation System Intellectual Property (RSIP), developers risk unauthorized usage, replication, or modification, further complicating the already ambiguous boundaries of ownership in this domain.

The growing competition among developers to create unique recommendation algorithms has intensified concerns surrounding RSIP. The rapid pace of innovation has resulted in significant overlaps in methods and foundational techniques, often leading to disputes over ownership. In many cases, competing entities claim rights to similar models or algorithms, complicating the legal and ethical frameworks for RSIP. This ambiguity undermines collaboration and trust between stakeholders, stifling innovation in a highly competitive industry. Additionally, the lack of standardized IP protection mechanisms has created an environment where companies may inadvertently infringe on each other's work, further fueling disputes. To address these challenges, clear guidelines and innovative verification techniques are needed to distinguish proprietary advancements from shared innovations. Doing so is essential to promote fairness and encourage continued development in recommendation system technologies.

To address the challenges of RSIP, researchers have explored strategies inspired by methods used in image-based deep learning models. Among these, techniques such as weighted watermarking, backdoor watermarking, and active watermarking have shown promise for safeguarding RSIP [118]. These techniques embed unique identifiers into RS models or their outputs, allowing developers to trace and verify their proprietary algorithms in cases of misuse or dispute. For instance, weighted watermarking applies subtle, imperceptible changes to the model's parameters, while backdoor watermarking embeds trigger patterns to verify ownership. Active watermarking further extends these approaches by enabling dynamic detection of unauthorized usage. Together, these methods provide a robust foundation for asserting ownership and clarifying disputes. As these approaches are refined, they hold the potential to standardize IP protection while fostering collaboration and innovation in recommendation system development.

In summary, the challenges surrounding recommendation system intellectual property underscore the urgent need for effective mechanisms to protect and verify ownership of these valuable technologies. Inspired by Fig. 2.18, we aim to explore ownership verification techniques specifically tailored for recommendation system models [119]. As recommendation systems become increasingly integral to driving user engagement and business success, their proprietary models and algorithms represent critical competitive assets. However, the ambiguity in RSIP, exacerbated by overlapping innovations and intense competition, poses significant risks to collaboration, trust, and progress in the industry. Advanced techniques such as watermarking and ownership verification offer promising solutions to these challenges by providing reliable methods to protect intellectual property and address disputes. As the importance of RS technologies continues to grow, exploring and developing robust IP protection frameworks will remain a crucial area of research, shaping the future of secure, reliable, and innovative recommendation systems in an ever-evolving digital landscape.

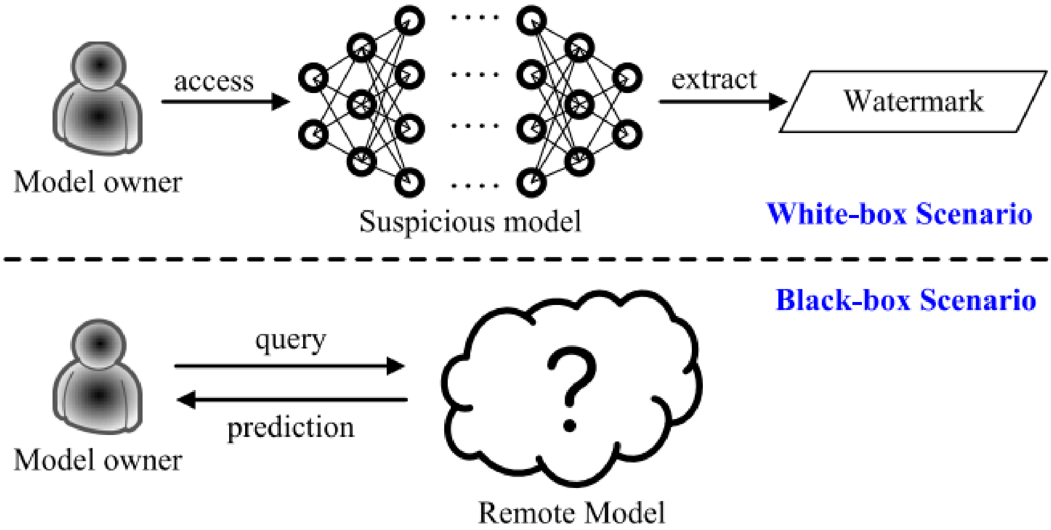


Figure 2.18: Intellectual Property Protection for Deep Learning Models {Source: [119]}.

2.5 Large Language Models as Recommendation Systems

This section explores the application of LLMs in recommendation systems and the associated privacy and security challenges. With the advancement of deep learning and natural language processing technologies, LLMs have demonstrated exceptional capabilities in understanding user behavior and generating personalized recommendations. However, the integration of LLMs into recommendation systems presents several critical challenges, including computational complexity, data privacy protection, and security vulnerabilities. This section first introduces the fundamental concepts of LLMs (Section 2.5.1), then examines their role in recommendation systems (Section 2.5.2), and finally delves into the privacy and security concerns that arise in LLM-powered recommendation systems (Section 2.5.3).

2.5.1 Large Language Models

LLMs have emerged as a breakthrough in natural language processing (NLP), demonstrating superior capabilities in text generation, semantic understanding, and information retrieval [120][121]. These models, typically built on transformer-based architectures, leverage self-attention mechanisms and deep hierarchical representations to capture complex linguistic patterns [122]. By training on massive datasets, LLMs acquire broad contextual knowledge, making them adaptable to diverse NLP applications, including machine translation, question answering, and conversational agents. The introduction of models such as GPT-4 [123] and PaLM [124] has further expanded the frontiers of NLP, enabling more sophisticated human-like interactions. However, their extensive computational requirements and large-scale training data dependencies remain significant challenges for real-world deployment, particularly in resource-constrained environments.

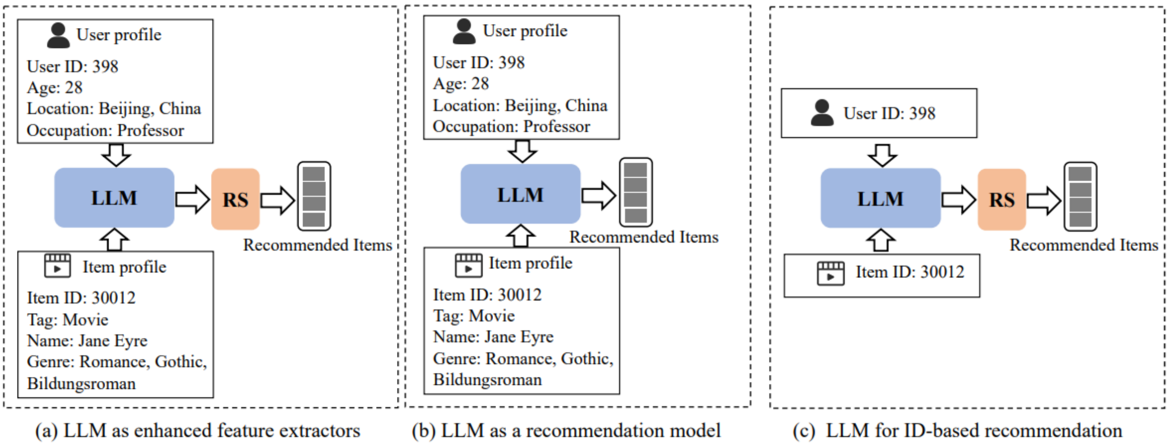
In recent years, the adoption of LLMs in recommendation systems has grown rapidly owing to their capacity to handle unstructured text information, such as user reviews, product descriptions, and social media interactions [125][126]. Unlike traditional collaborative filtering and content-based approaches, which primarily rely on structured user-item interaction matrices, LLMs offer a more nuanced understanding of implicit user preferences through contextualized representation [127]. This enables more accurate and personalized recommendations, particularly in cold-start scenarios where historical interaction data is limited. Moreover, few-shot and zero-shot learning capabilities allow LLMs to generalize to new recommendation tasks with minimal fine-tuning, reducing data dependency and enhancing model adaptability [128]. Despite these advantages, integrating LLMs into recommendation pipelines poses challenges related to inference latency, memory consumption, and scalability.

The computational cost associated with training and deploying LLMs remains a critical limitation, especially for large-scale recommendation applications [129]. High-performance hardware, such as Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), is often required to process large volumes of user-generated content in real time. Recent advances in model compression, such as quantization and distillation, have sought to mitigate these constraints by reducing model size while preserving performance [130]. Furthermore, ethical considerations, including bias mitigation and explainability, are becoming increasingly important as LLM-powered recommendation systems influence user decision-making [131]. Addressing these challenges will be essential for the widespread adoption of LLMs in recommendation systems, ensuring fairness, transparency, and efficiency in AI-driven personalization.

2.5.2 Applications of LLMs in Recommendation Systems

The integration of LLMs into recommendation systems has significantly advanced the field by enhancing content understanding, user behavior modeling, and personalized recommendations. As shown Fig. 2.19 [132]. Unlike traditional collaborative filtering or deep learning based methods, LLMs excel at processing unstructured data, such as user reviews, articles, and social media interactions, thus generating more accurate and contextually relevant recommendations. For instance, recent studies have demonstrated the potential of LLMs to interpret complex user-generated content, leading to improved recommendation diversity and user engagement [133].

Moreover, LLMs possess few-shot and zero-shot learning capabilities, enabling them to adapt to various recommendation scenarios with minimal task-specific data. This adaptability is particularly beneficial in addressing cold-start problems, where user or item interaction history is sparse. Research has shown that LLMs can infer user preferences from limited information, thereby enhancing the system's generalization and robustness



(a) LLM as enhanced feature extractors (b) LLM as a recommendation model (c) LLM for ID-based recommendation

Figure 2.19: Enhancing Recommendation with Large Language Models {Source: [132]}.

[134]. Additionally, the generative nature of LLMs facilitates the creation of novel content, which can be leveraged to enrich recommendation outputs and provide users with unique, tailored experiences [135].

However, integrating LLMs into recommendation systems presents several challenges. The substantial computational resources required for training and inference can impede real-time application, necessitating model optimization techniques. Furthermore, ensuring the interpretability of LLM-based recommendations is crucial for user trust, as the black-box nature of these models can obscure decision-making processes. Addressing potential biases inherent in LLMs is essential to prevent the propagation of unfair or discriminatory recommendations. Ongoing research aims to develop methods that mitigate these issues, such as incorporating fairness constraints and enhancing model transparency.

In summary, while LLMs offer promising advancements for recommendation systems through improved content understanding and adaptability, careful consideration of computational efficiency, interpretability, and fairness is imperative.

2.5.3 Privacy and Security of Recommendation Systems with LLMs

The integration of LLMs into recommendation systems offers significant advancements in personalization and user engagement. However, this integration introduces critical privacy and security challenges that must be addressed.

LLM-driven recommendation systems often require extensive user data to function effectively, raising concerns about potential privacy breaches. The collection and processing of sensitive information, such as browsing history and personal preferences, can lead to unauthorized data exposure if not properly managed. Recent incidents have highlighted vulnerabilities where AI models inadvertently exposed confidential user information, underscoring the need for robust data protection measures [136].

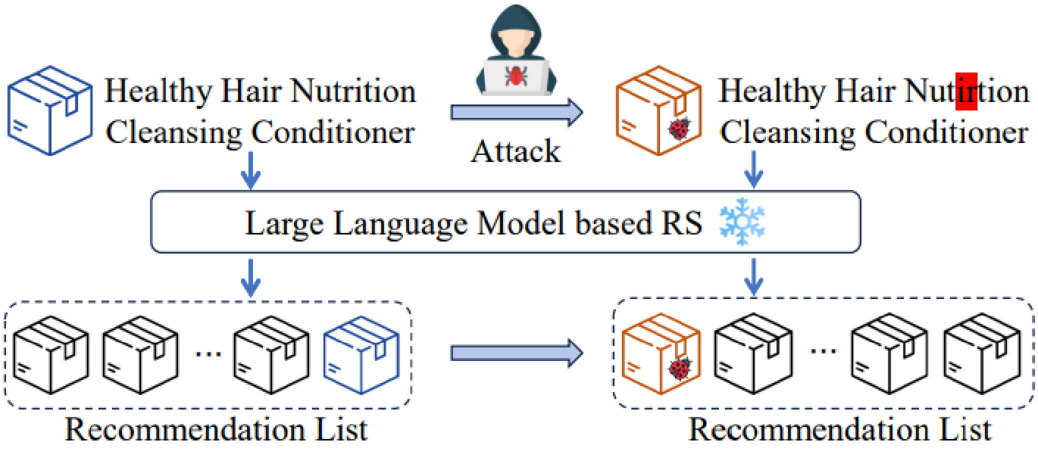


Figure 2.20: Stealthy Attack on Large Language Model Based Recommendation {Source: [137]}.

LLM-based systems are vulnerable to adversarial attacks, including data poisoning and adversarial examples. Data poisoning involves injecting malicious data into the training set, skewing the model's outputs. Adversarial examples are inputs crafted to deceive the model into making incorrect predictions. These attacks can degrade the performance of recommendation systems and lead to erroneous or harmful suggestions.

According to Fig. 2.20, a critical threat to LLM-based recommendation systems is data poisoning, in which attackers introduce harmful components into the model's input stream. These poisoned inputs can manipulate the model's internal representations, leading to biased or attacker-preferred outputs in the recommendation list. In the example, a malicious version of a product is inserted into the system, causing the LLM-based recommender to prioritize compromised items in the output. Such attacks undermine the integrity and trustworthiness of recommendation results, highlighting the urgent need for robust defense mechanisms [137][138].

The black-box nature of LLMs results in a lack of transparency in decision-making processes. Users and developers may find it challenging to understand how specific recommendations are generated, which can erode trust and hinder the identification of biases or errors within the system. Enhancing the interpretability of LLMs is crucial for building user confidence and ensuring ethical AI practices.

2.6 Summary

This chapter provides a comprehensive review of existing works on privacy-preserving mechanisms and security innovations in deep learning-based recommendation systems. It begins with an overview of recommendation systems leveraging deep learning, highlighting their strengths and vulnerabilities, followed by an examination of data poisoning attacks, including various attack techniques and corresponding defense strategies, emphasizing the importance of protecting model integrity in adversarial environments. Additionally, recommendation unlearning verification is explored as a novel approach to managing user privacy, focusing on machine unlearning and its verification processes. Furthermore, model ownership verification is discussed, addressing risks associated with obfuscation and intellectual property ambiguities, while reviewing techniques such as watermarking and ownership verification to ensure the legitimacy and security of recommendation system models. Lastly, the large language models (LLMs) as recommendation systems are briefly analyzed, identifying emerging privacy and security challenges. These insights form the basis for the subsequent chapters, which will propose innovative solutions and experimental frameworks to tackle the challenges identified in the literature, with a focus on dual defense mechanisms, recommendation unlearning verification, and model ownership verification in deep learning-based recommendation systems.

Chapter 3

3 A Dual Defense Design Against Data Poisoning Attacks in Deep Learning-Based Recommendation Systems

Thus far, this thesis has discussed various privacy and security issues in deep learning-based recommendation systems. Building upon these discussions, this chapter delves further into our proposed solutions, which offer efficient approaches to addressing these challenges, making them an ideal choice in the field of security and privacy for recommendation systems. In this chapter, we propose a dual defense strategy to address the problem of data poisoning attacks in deep learning-based recommendation systems. The proposed defense consists of two complementary layers, tackling the problem both proactively and reactively [139]. The first line, referred to as active defense, preemptively reduces the system's susceptibility to poisoning attacks by incorporating carefully crafted regularization terms into the loss function. This approach mitigates the impact of attackers while preserving the system's performance, thus significantly lowering the success rate of targeted attacks. The second line of defense, referred to as passive defense, introduces a GAN -based detection model to accurately identify and filter out poisoned data. Empirical evaluations conducted on three diverse datasets demonstrate that our dual defense strategy substantially enhances the proactive and reactive capabilities of recommendation systems, effectively countering data poisoning attacks.

The rest of the chapter is organized as follows. Section 3.1 presents the preliminary analysis of the dual defense design. Section 3.2 introduces the framework of dual defense. Section 3.3 describes the implementation of our dual defense. Section 3.4 provides the experimental results and analysis. Finally, Section 3.5 summarizes the chapter.

3.1 Preliminary Analysis of the Dual Defense Design

In this section, we conducted a preliminary analysis of the dual defense design against data poisoning attacks in deep learning-based recommendation systems. First, we explored Neural Matrix Factorization (NMF) as the core recommendation model, which serves as the foundation for understanding the vulnerabilities and strengths of the system under adversarial settings. Subsequently, we discussed various defensive strategies for recommendation systems, categorizing them into proactive and reactive approaches to highlight the need for a combined dual defense mechanism. Finally, we provided a detailed problem formulation, which defines the objectives, constraints, and attack scenarios for designing an effective defense system. This preliminary analysis establishes the theoretical groundwork for the proposed dual defense strategy, providing insights into its structure and the challenges it addresses. The findings here lay the foundation for the subsequent implementation and evaluation of the dual defense mechanism.

3.1.1 Neural Matrix Factorization

This study builds upon the NeuMF algorithm, a prominent NCF approach widely utilized in recommendation systems [140]. NeuMF effectively combines the GMF technique with multi-layer perceptron architectures to model both linear and non-linear interactions between users and items, enhancing prediction accuracy and scalability. The integration of these two architectures allows the system to capture complex patterns in user-item interactions, making it particularly effective for handling sparse data. In this work, NeuMF serves as the foundational framework for exploring vulnerabilities in recommendation systems under adversarial settings. By leveraging its robust architecture, we analyze the impact of data poisoning attacks and develop dual defense mechanisms to mitigate such threats. As illustrated in Fig. 2.2, the model structure and its extensions form the basis for implementing proactive and reactive defense strategies in this study.

The NeuMF model represents a powerful fusion of matrix factorization (MF) and multi-layer perceptron techniques, designed to capture both linear and non-linear user-item interactions in recommendation systems. The model starts by representing users and items as one-hot encoded vectors, which are then projected into dense latent spaces using separate embeddings for MF and MLP. The MF component computes an element-wise inner product of user and item latent vectors, effectively capturing the linear relationships between users and items. In parallel, the MLP component processes these latent vectors through multiple layers, applying ReLU activation functions at each layer to model complex non-linear interactions. These two components are subsequently concatenated, and the combined features are passed through the NeuMF layer, enabling the integration of both linear and non-linear outputs.

The final predictions, represented as $\hat{y_{ui}}$, combine the outputs of the MF and MLP components to estimate the likelihood of interaction between users and items. During training, NeuMF minimizes the difference between predicted ($\hat{y_{ui}}$) and actual (y_{ui}) interaction scores using a suitable loss function. Once trained, the model infers unobserved values within the user-item interaction matrix Y, resulting in a completed matrix $\hat{Y}$ that provides personalized recommendations. This dual-architecture design enables NeuMF to achieve high accuracy and flexibility, making it a robust and scalable solution for modern recommendation systems, particularly in handling sparse and complex datasets.

3.1.2 Defensive Strategies for Recommendation Systems

Existing research highlights the vulnerability of recommendation systems to a range of attacks designed to manipulate outcomes by injecting low-quality or misleading information into the system. Among these, shilling attacks are particularly prevalent and involve the creation of multiple fake profiles or duplicate items to artificially enhance an item's visibility. These attacks exploit favorable ratings or reviews to increase the likelihood of the item being recommended, thereby skewing the system's outputs [141]. On the other hand, poisoning attacks, as detailed by [70], introduce malicious data modifications, such as fake user-item interactions, to corrupt the recommendation process. These attacks are often carefully crafted to influence the system's learned patterns and degrade its performance, either by promoting a specific item or reducing the visibility of competing ones. Both types of attacks pose significant challenges to the integrity and reliability of recommendation systems, particularly as they rely on user-generated data, which is inherently susceptible to manipulation. Addressing these vulnerabilities requires the development of robust defense mechanisms to maintain system accuracy and trustworthiness while mitigating the impact of adversarial behaviors.

Various methods have been explored to counter data poisoning attacks in recommendation systems, aiming to enhance their robustness and reliability. Techniques such as data quality rules, clustering, outlier detection, and L_2 regularization have been widely adopted for mitigating the influence of poisoned data. Additionally, advanced methods like Slab defense and loss defense [142] have been proposed to improve the system's ability to withstand adversarial manipulations. While these approaches provide some level of protection, they often face limitations such as increased computational overhead, reduced scalability, or challenges in detecting highly sophisticated poisoning strategies.

To address these limitations, multi-model fusion approaches have gained attention due to their effectiveness in reducing the influence of data poisoning threats targeting individual models. For instance, the methods proposed by Vasiliki Kelli and Islam Umar leverage the diversity in predictions from multiple models to detect and filter out poisoned data more effectively [143][144]. These approaches enhance robustness by aggregating insights from multiple perspectives, reducing the likelihood of adversarial data corrupting the system. Despite their effectiveness, challenges remain in optimizing the trade-off between computational efficiency and defense capability, highlighting the need for further research into scalable and adaptive defense mechanisms for real-world applications.

GANs initially introduced by Ian Goodfellow et al., have proven to be a powerful tool in various domains, including enhancing the robustness of machine learning models against data poisoning attacks [145]. GANs consist of a generator and a discriminator that is trained in a competitive framework, enabling the generation of high-quality synthetic data that closely resembles the original dataset. By leveraging this capability, GANs can be employed to generate additional clean samples, which augment training datasets and increase the proportion of legitimate data. This augmentation process not only helps dilute the influence of poisoned or malicious data but also ensures that the model learns robust patterns from trustworthy sources.

Furthermore, the inclusion of GAN-generated clean data reduces the model's reliance on potentially corrupted samples, thereby mitigating the negative effects of poisoning attacks. This enhanced data diversity contributes to a more resilient training process, ensuring that the model maintains its performance and accuracy even in adversarial environments. As a result, GANs have emerged as a promising defense mechanism, providing an effective and scalable solution to improve model robustness against data poisoning attacks in recommendation systems and other machine learning applications.

3.1.3 Problem Formulation

We begin by conducting a comprehensive analysis of the threat model and attack methods, which encompasses the attacker's intentions, capabilities, and expertise. Understanding these factors is critical for designing effective and targeted defense mechanisms. The threat model serves as the foundation for identifying potential vulnerabilities in the system, while also defining the scope and nature of adversarial actions. Specifically, we examine the attacker's objectives, such as compromising model integrity or promoting specific items, and their capabilities, which include access to the data or system and the resources available to execute the attack. Furthermore, we analyze the strategies employed by attackers, such as data poisoning or adversarial manipulation, to better understand their impact. This detailed evaluation is essential for formulating robust defense techniques tailored to the identified threats, ensuring improved system resilience.

Attack capability: Recent research has highlighted the widespread adoption of deep learning in recommendation systems to improve prediction accuracy and user satisfaction. However, these systems are inherently vulnerable to data poisoning attacks, where adversaries inject fake data with meticulously designed ratings to compromise system performance and accuracy. Such attacks exploit the dependency of deep learning algorithms on large-scale training data, making even small-scale manipulations impactful. Utilizing frameworks like NeuMF, a widely used deep learning-based recommendation system, attackers can craft malicious data to target specific outcomes. For instance, by altering a small portion of the training data, attackers can influence the behavior of learning algorithms, skewing recommendations toward or against certain items [146]. These manipulations can lead to biased recommendations, reduced system reliability, and erosion of user trust. The complexity of deep learning models exacerbates this issue, as subtle changes in the training data are difficult to detect and can propagate through the learning process. Understanding the attacker's capabilities, such as access to training data and knowledge of the system architecture, is crucial for developing robust defense mechanisms to mitigate these threats and safeguard the integrity of recommendation systems.

Attack strategy: This attack strategy approximates the optimization problem by constructing a "poison mode" to simulate the behavior of a compromised recommendation system. The primary objective of the attacker is to manipulate the system to promote a target item while ensuring the attack remains inconspicuous. The objective function can be formalized as:

$$G[y(v)] = \|y(v)\|_2^2 + \eta \cdot \sum_{u \in S} \max\{\min_{i \in L_u} \log[\hat{y}_{ui}] - \log[\hat{y}_{ut}], -\kappa\}. \tag{3.1}$$

In this equation, $y(v)$ denotes the preference vector associated with user v, η is a coefficient controlling the impact of the attack, S represents the set of unrated users, and L_u denotes the recommendation list for user u. The terms $\hat{y}_{ui}$ and $\hat{y}_{ut}$ are the predicted ratings for item i and the target item t, respectively, while κ introduces robustness to ensure that the attack objective remains effective. Minimizing $\min_{l \in L_u} \log[\hat{y}_{ui}] - \log[\hat{y}_{ut}]$ narrows the score gap between the lowest-ranked item in L_u and the target item t, effectively pushing t into the user's top-K list. This ranking-based approach directly targets the recommendation boundary, making it more effective and stealthy than simply giving high ratings to t (details in [70]).

The attack iteratively selects filler items for fake users by leveraging predicted ratings and dynamic probabilities, injecting generated ratings to strategically promote the target item. To further enhance its effectiveness, the attacker defines the following loss function:

$$l = \mathcal{L} + \lambda \cdot G\left[\widehat{\mathbf{y}}_{(v)}\right]. \tag{3.2}$$

This loss function incorporates the original recommendation system's loss function $\mathcal{L}$, which ensures the overall model effectiveness, and $G[\hat{y}(v)]$, a term directly tied to the attack objective. Here, $\hat{y}(v)$ denotes the predicted rating vector for the fake user v, and λ is a positive coefficient balancing the trade-off between maintaining the performance and achieving the attack objective [70]. By optimizing this combined loss function, the attacker amplifies the impact of the poisoning attack while minimizing detection risks, compromising the recommendation system. This sophisticated approach underscores the critical need for robust defense mechanisms to mitigate such adversarial strategies.

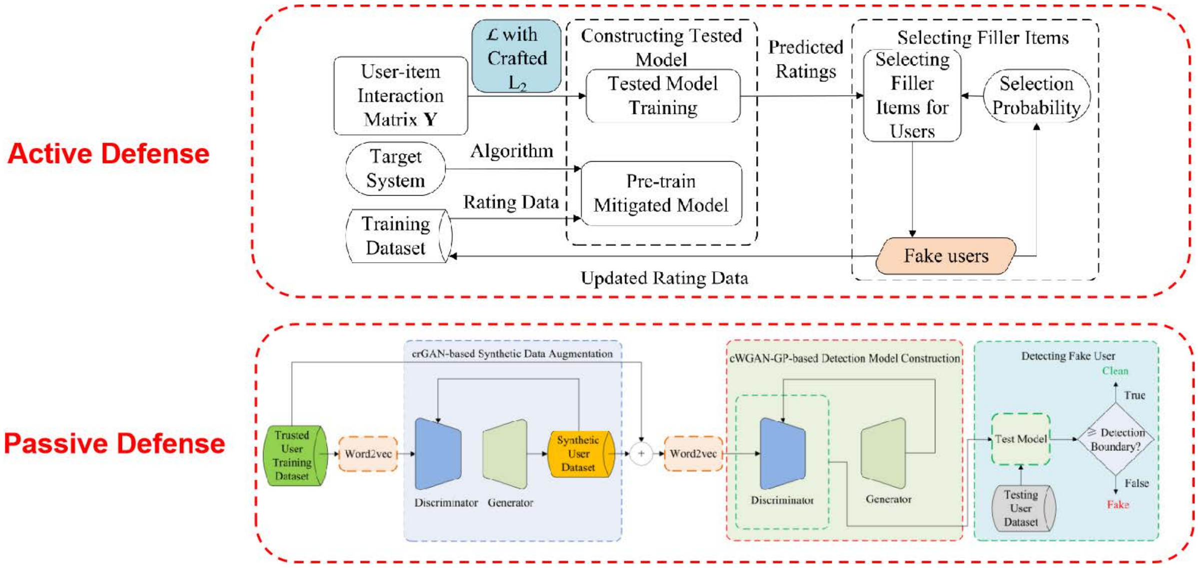


Figure 3.1: Overview of Our Dual Defense Framework.

3.2 Dual Defense Framework Overview

To address the growing vulnerability of deep learning-based recommendation systems to data poisoning attacks, we propose a comprehensive dual-defense framework that integrates two complementary approaches: active defense and passive defense, as depicted in Fig. 3.1. This framework is specifically designed to counteract malicious interference by targeting the problem from both proactive and reactive perspectives. The active defense mechanism operates preemptively, modifying the recommendation system's loss function during the training phase to strengthen its resilience against adversarial attacks. On the other hand, the passive defense mechanism functions reactively, employing a GAN-based detection model to identify and isolate fake users after the system has been exposed to poisoned data. Together, these two strategies work in tandem to mitigate the effects of data poisoning, preserving the stability, trustworthiness, and protection of recommender models under adversarial conditions.

The active defense mechanism operates before the recommendation system is fully trained. Its primary objective is to proactively enhance the system's resilience against potential attacks by incorporating a crafted L_2-regularized loss function into the model's optimization process. The active defense begins with the user-item interaction matrix Y, which serves as the training dataset. By modifying the recommendation system's loss function, denoted as $\mathcal{L}$ with a crafted L_2 regularization term, the algorithm creates a more robust learning model that minimizes the influence of adversarial inputs. This is achieved through the generation of a pre-trained mitigated model, which is then used to construct the tested model. During this process, predicted ratings are generated, and filler items are iteratively selected for fake users based on their selection probabilities. These filler items are strategically crafted to test the system's response to adversarial inputs, ensuring that the model is optimized to resist such manipulations.

While the active defense aims to prevent attacks before they occur, the passive defense mechanism is employed post-poisoning to detect and mitigate the effects of malicious activities. This component utilizes a GAN based model for enhanced detection and isolation of fake users. The process begins with trusted user training data, which is processed using Word2Vec embeddings to form a synthetic user dataset through a conditional GAN (cGAN) framework. This augmented dataset strengthens the detection model's ability to identify anomalies. To further refine detection capabilities, a Wasserstein GAN with Gradient Penalty (WGAN-GP) is employed to construct a robust detection model. The detection model distinguishes between clean and fake users by analyzing the testing dataset and comparing it to a detection boundary. Any user crossing this boundary is classified as fake, and their data is isolated to preserve the integrity of the RS.

The dual-defense framework combines these two approaches to provide a robust solution against data poisoning. The active defense reduces the system's susceptibility to adversarial manipulations during training, while the passive defense ensures post-training security by identifying and isolating malicious users. Together, they create a cohesive defense mechanism that not only mitigates immediate threats but also improves the system's long-term resilience. This integrated framework represents a significant advancement in the protection of deep learning-based recommendation systems.

In the following Section 3.3, we delve deeper into the technical implementation of both active and passive defenses, providing a detailed analysis of their components and their combined effectiveness in countering data poisoning attacks.

3.3 RS Model Dual Defense Implementation

The proposed RS model introduces a dual defense mechanism aimed at mitigating the risks posed by data poisoning attacks, combining two complementary components:

active defense design and passive defense design. These two components work in synergy to comprehensively address the vulnerabilities inherent in deep learning-based recommendation systems, tackling the problem from both proactive and reactive perspectives. The active defense focuses on preemptively strengthening the model during the training phase by modifying the loss function, making the system more resilient to adversarial data. Meanwhile, the passive defense operates post-training, utilizing a GAN-based detection framework to identify and isolate malicious users and poisoned data. Together, these strategies ensure robust protection, preserving the trust and operational soundness of recommender systems under adversarial conditions.

3.3.1 Active Defense Design

After investigating and comparing various regularization methods, we observed that many approaches exhibit limitations in effectively countering data poisoning attacks. Among these, L_2 regularization demonstrated superior performance due to its ability to enhance model robustness and mitigate the adverse effects of poisoning attacks by penalizing large weight values [147]. To further improve its effectiveness in this context, we propose a carefully crafted L_2 regularization (CLR) specifically tailored for data poisoning scenarios. This enhanced regularization technique incorporates additional terms to target adversarial patterns, reducing their influence on the model. The proposed CLR not only strengthens the system's resilience but also maintains model performance during adversarial attacks. Fig. 3.2 illustrates the comprehensive attack mitigation framework using the CLR approach, highlighting its role in minimizing the impact of poisoned data.

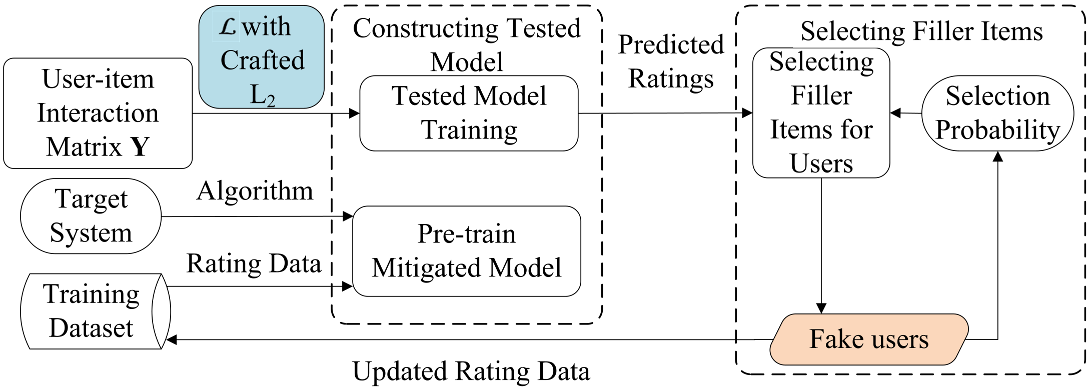


Figure 3.2: CLR against data poisoning attacks.

To mitigate the effects of unknown data poisoning attacks while maintaining the performance of the recommendation system, we integrate a carefully crafted L_2 regularization term into the original model's loss function, $\mathcal{L}$. This enhanced loss function is designed to reduce the influence of poisoned data by penalizing large model weights, thereby improving the model's robustness against adversarial manipulations. The crafted L_2 regularization term not only strengthens the system's resilience to data poisoning but also ensures that the recommendation accuracy remains uncompromised. The resulting new loss function is expressed as follows, incorporating both the original objective and the added defense mechanism.

$$L = \mathcal{L} + (N_r - 1/2) * 10 + \frac{e^{\lambda}}{2}\|\omega\|_2^2, \tag{3.3}$$

where N_r represents random noise introduced to enhance the robustness of the model against data poisoning. This noise plays a crucial role in preventing overfitting to adversarial patterns while improving the system's generalization. The exponential form of the crafted regularization term ensures positive values, aiding in the learning process of the coefficient λ, balancing the trade-off between predictive effectiveness and defensive robustness. Different from standard L_2 regularization, which primarily mitigates overfitting, this crafted variant specifically targets parameters most susceptible to adversarial manipulation. This design introduces a trade-off: stronger penalties further reduce the target item hit rate but may slightly affect recommendation quality for non-adversarial items. Through the integration of this mechanism, the poisoned model's loss function is transformed into a more robust formulation, designed to reduce the influence of adversarial inputs while maintaining the system's overall effectiveness and accuracy.

$$l' = L + \lambda \cdot G\left[\widehat{\mathbf{y}}_{(v)}\right]. \tag{3.4}$$

Here, our primary goal is to covertly protect the recommendation system from potential data poisoning attacks by implementing proactive defense mechanisms during training. To achieve this, we propose a heuristic active protection strategy, as detailed in **Algorithm** 1, which integrates crafted regularization within the training phase of the model. This approach enhances the system's robustness against adversarial manipulations while maintaining its performance. By strengthening the system preemptively, we reduce vulnerabilities and mitigate the impact of poisoned data before attacks occur.

```
Algorithm 1: Active Guard: CLR
Input: User-item interaction matrix $\boldsymbol{Y}$, Crafted $L_2$, initial loss function $\boldsymbol{\mathcal{L}}$,
       pre-train epochs $T_{pre}$, learning rate $\eta$, tested model update schedule $S$
Output: detection model $\hat{\theta}$
begin
    # STEP 1: Get Training Data $\boldsymbol{D_{trn}}$.
    $\boldsymbol{D_{trn}} \leftarrow \boldsymbol{Y}$;
    # STEP 2: Polish the initial model loss function $\boldsymbol{\mathcal{L}}$.
    Using the item Approximating Hit Ratio as $\boldsymbol{\mathcal{L}}$;
    Get polished model loss function $\boldsymbol{L}$ using Eq. (3.3);
    $\boldsymbol{L} \leftarrow$ Crafted $\boldsymbol{\mathcal{L}}(N_r, L_2)$;
    # STEP 3: Pre-train model $M_t$ on $\boldsymbol{D_{trn}}$ with $\boldsymbol{L}$.
    Start initial training to get the mitigatory poisoning model $M_t$ based $\boldsymbol{L}$.
    Get mitigatory poisoning model $\theta_t \leftarrow M_t$;
    Initialize $\boldsymbol{L} \Longleftarrow 0$, model $\theta_t$, and random optimizer
    for $t = 1 ... T_{pre}$ do
        $\theta^t \leftarrow \theta^t - \eta \bigtriangledown L(D_{trn}, \theta^t)$
    end
    return mitigatory poisoning model $\theta_t$
    # STEP 4: detection model training for data poisoning defense:
    Get tested model $\hat{\theta}$;
    for $t = T_{pre} + 1 ... T$ do
        if $t \in S$ then
            $\hat{\theta} \leftarrow$ update $\theta_t(D_{trn}, l')$ based on Eq. (3.4)
        end
    end
end
return $\hat{\theta}$
```

3.3.2 Passive Defense Design

To bolster deep learning-based recommendation systems against data poisoning attacks, we propose the incorporation of a passive defense mechanism as a complementary safeguard. This approach is designed to enhance system robustness by identifying and filtering out malicious users embedded in the training data. By leveraging statistical analysis and anomaly detection techniques, passive defense reduces the influence of adversarial data, ensuring the integrity and reliability of the recommendation system. The proposed methodology not only mitigates the impacts of data poisoning but also fortifies the system's ability to deliver accurate and secure recommendations, thus providing a significant advancement in the field of trustworthy AI.

We employ a GAN-based detection method to effectively identify fake users D_f within the dataset. This is achieved by comparing prediction results from two models: the target model, trained on real data D_T, and a simulated model. By analyzing the prediction differences between these two models, our method is able to detect fake users D_f embedded in the dataset $D_d = D_f + D_T$. The GAN framework facilitates precise modeling of adversarial behavior, leveraging its generative and discriminative capabilities to isolate anomalies. Specifically, we adopt a consistent regularization GAN (crGAN) for data augmentation and a conditional Wasserstein GAN with gradient penalty (cWGAN-GP) for simulation modeling. In both cases, the generator produces realistic synthetic samples aligned with the original distribution, while the discriminator distinguishes them from real data. To ensure reproducibility, the generator has four fully connected layers with 128, 256, and 512 hidden units, followed by an output layer whose dimension d matches the dataset-specific feature size (100 for ML-100K and 50 for Last.fm). The discriminator has three fully connected layers with 512 and 256 hidden units and a final output neuron, with LeakyReLU activations, Sigmoid output, and batch normalization with dropout (0.3). Training uses Adam ($\beta_1 = 0.5$, $\beta_2 = 0.999$), learning rate 0.0002, batch size 128, and 200 epochs, with gradient penalty 10 for both crGAN and cWGAN-GP. These architectures were validated on held-out validation sets to ensure generalization and effective fake-user detection, as illustrated in Fig. 3.3.

Data processing. The sparsity of user-item interaction data is a critical challenge for the effectiveness of recommendation algorithms, as illustrated in Fig. 2.2. Sparse interaction data often fails to capture complex relationships between users and items, reducing the accuracy and robustness of recommendation systems. Traditional one-hot encoding methods exacerbate this issue by generating high-dimensional and sparse repre-

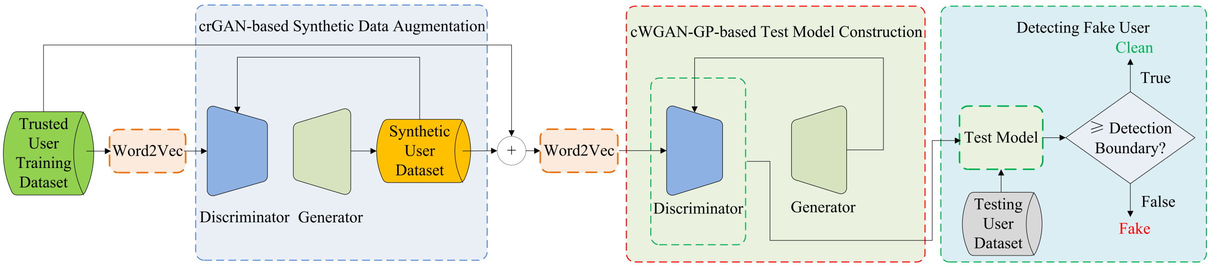


Figure 3.3: A framework for defense detection based on GAN.

sentations, which can adversely impact the performance of detection models in identifying fake users or malicious activity. To tackle this issue, we employ Word2Vec, a technique widely adopted in natural language processing tasks. Word2Vec represents words as dense, continuous vectors, capturing semantic relationships in a low-dimensional space. By leveraging Word2Vec, we transform the sparse user-item interaction data into dense vector representations, effectively mapping users and items into a shared vector space. This approach facilitates the capture of implicit relationships between users and items, improving the robustness and performance of the detection and recommendation models. Fig. 3.3 demonstrates the mapping process enabled by Word2Vec, highlighting its role in reducing sparsity and enhancing data representation.

Data enhancement. While recommendation systems typically have access to large volumes of data, maintaining data purity remains a significant challenge, especially when defending against poisoned data. To address this, it is essential to enhance data diversity, which can improve the effectiveness and robustness of the system. The second component of our defense detection process focuses on generating synthetic training data that closely matches the distribution of the original dataset D_T. We adopt a consistent regularization Generative Adversarial Network (crGAN)-based framework to achieve this goal.

In this framework, the generator (G) is designed to produce realistic synthetic samples that enhance the diversity and representativeness of the training dataset. These synthetic samples help mitigate the impact of poisoned data by providing a more robust foundation for recommendation model training and analysis. Simultaneously, the discriminator (D) evaluates the generator's output, distinguishing between real and synthetic data. This adversarial process not only improves the quality of synthetic data but also ensures that the generator continuously refines its output. By integrating crGAN, we create a more effective and resilient dataset for training robust recommendation models.

During the training process, the generator (G) and discriminator (D) engage in a collaborative adversarial learning framework to achieve consistency between the augmented data $T(x)$ and the original data x. The generator is designed to generate augmented samples that closely align with the distribution of the original data, while the discriminator works to distinguish between the real data and the generator's output. This adversarial interplay allows both components to iteratively improve their performance. The optimization process focuses on minimizing the discrepancy between $T(x)$ and x, governed by carefully designed objectives that ensure data quality and consistency.

$$\min DLcr = \min D \sum j = m^n \lambda_j \left\| D_j(x) - D_j(T(x)) \right\|^2 . \tag{3.5}$$

Adversarial training generates synthetic data, with the discriminator assessing quality. Objective functions are:

$$\begin{aligned} L_{cr}^{(i)} &= \|D(x) - D(T(x))\|^2, \\ L_D^{(i)} &= D(G(z)) - D(x). \end{aligned} \tag{3.6}$$

Constructing a simulation model for detection. To address challenges posed by data insufficiency and mitigate overfitting in deep learning-based recommendation systems, we employ a conditional Wasserstein Generative Adversarial Network with Gradient Penalty (cWGAN-GP) to construct a simulation model. This approach enhances the diversity and representativeness of the training data distribution by generating high-quality synthetic data conditioned on specific attributes. The cWGAN-GP framework ensures stability during training and preserves critical data features by incorporating a gradient penalty term, which mitigates mode collapse and improves generator performance. This simulation model effectively augments the training dataset, contributing to more robust detection and analysis capabilities.

The cWGAN-GP leverages the Wasserstein distance to evaluate discrepancies between the distributions of real and simulated samples while incorporating conditional information to guide the generation process. The Wasserstein distance, a robust metric for measuring differences between probability distributions, ensures improved stability and convergence during training. By conditioning the generation process on specific attributes, cWGAN-GP enhances the quality and relevance of the generated samples, facilitating the creation of a more representative and balanced dataset for robust model training. The mathematical definition of the Wasserstein distance is:

$$\mathrm{W}\left(p_{data}, p_{\mathrm{g}}\right) = \inf\nolimits_{\gamma\in\Pi(p_{data},p_{\mathrm{g}})} \mathrm{E}_{(x,y)\sim\gamma}[\|x-y\|]. \tag{3.7}$$

Here, p_{data}, p_{g} denote the true data distribution and the generated data distribution, respectively. $\prod\left(p_{\mathrm{data}}, p_{\mathrm{g}}\right)$ represents the joint probability that all edge distributions conform to p_{data} and p_{g}.

The cWGAN-GP generator is constructed to generate synthetic samples that approximate the distribution of real data while incorporating conditional information y to enable personalized and context-aware simulations. By linking the condition y to the generator's output distribution p_{g}, the model ensures that the generated data aligns with specific attributes or requirements. Simultaneously, the discriminator integrates the real data distribution p_{data}, the generator's distribution p_{g}, and the conditional information y into a joint hidden representation. This joint modeling enables the network to effectively evaluate the quality and consistency of the generated data, facilitating the generation of conditional data tailored to the underlying context.

$$\begin{gathered}\min\nolimits_G \max\nolimits_D V(D,G) = \mathrm{E}_{x\sim p_{data}(x)}[D(x \mid y)] - \\ \mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{g} \mid y)] - \lambda \mathrm{E}_{\hat{x}\sim \mathrm{P}_{\hat{X}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x} \mid y)\|_2 - 1\right)^2\right].\end{gathered} \tag{3.8}$$

The loss objectives defined for cWGAN-GP are as follows:

$$\begin{gathered}L(D) = -\mathrm{E}_{x\sim p_{data}(x)}[D(x \mid \mathrm{y})] + \mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{g} \mid \mathrm{y})] + \\ \lambda \mathrm{E}_{\hat{X}\sim \mathrm{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{\mathrm{x}} \mid \mathrm{y})\|_2 - 1\right)^2\right], \\ L(G) = -\mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{\mathrm{g}} \mid \mathrm{y})].\end{gathered} \tag{3.9}$$

The cWGAN-GP framework aims to minimize the objective function L, effectively reducing the distribution gap between the generated data and the real data. By optimizing the Wasserstein distance during training, the generator learns to produce realistic samples, while the discriminator becomes proficient at distinguishing between real and synthetic data. After training, the discriminator network is repurposed as a simulation model. Leveraging its ability to differentiate real from synthetic data, the discriminator provides valuable insights into the underlying data structure and assists in detecting inconsistencies, making it a powerful tool for robust data analysis and system evaluation.

Fake user detection. The primary objective of our approach is to identify fake users and prevent their inclusion in the recommendation system's training data, thereby enhancing the system's robustness and reliability. The simulation model, trained to distinguish between real and synthetic data, employs a detection threshold to classify users. Users whose outputs fall below the predefined threshold are classified as fake, indicating potential malicious activity, while those exceeding the threshold are deemed authentic. This threshold-based classification ensures precise detection of anomalies within the dataset, effectively mitigating the impact of fake users on the training process and improving the recommendation system's performance and security.

Algorithm 2 provides a detailed summary of the four-part detection mechanism designed to defend against data poisoning attacks in RS models. This framework integrates the strengths of crGAN and cWGAN-GP. The crGAN component enhances data augmentation by generating diverse, high-quality synthetic samples, while the cWGAN-GP module focuses on detecting fake users through conditional data generation and distribution alignment. Together, these components form a robust defense mechanism capable of addressing both data sparsity and adversarial threats.

A critical element of the framework is the detection threshold, which serves as the classifier for identifying fake users. Its rigorous evaluation is essential to ensure optimal accuracy, reliability, and practicality in real-world applications, strengthening the system's security against sophisticated data poisoning attacks.

Algorithm 2: Passive Guard: detection mechanism via GAN

```
Input: Trusted user data $\boldsymbol{D_T}$, fake user data $\boldsymbol{D_f}$, perturbation vector $\delta$
Output: Detection decision of each user using detection model $M_d$
begin
    # STEP 1: Get trusted data of dense features $D_T^{'}$.
    $D_T^{'} \leftarrow D_T$ ( using Word2Vec);
    # STEP 2: Get augmented training data $D_{aug}$ on crGAN.
    if Synthetic user $D_{sy}$ exist then
        load $D_{Sy}$
    else
        Generate $D_{Sy}$ using crGAN
        load Generate $D_{Sy}$
    end
    $D_{aug} \leftarrow D_T^{'} + D_{Sy}$.
    # STEP 3: Constructing detection simulation model $M_d$.
    clean data $D_c \leftarrow D_{aug}$;
    $D_G \overset{\mathbf{G}}{\leftarrow} \delta$.
    for each training iteration do
        Update D ( D_loss ($D_c$, $D_{Sy}$))
        Update G ( G_loss)
    end
    Get detection simulation model $M_d \leftarrow \mathbf{D}$.
    # STEP 4: Detecting fake user using $M_d$.
    for user $u$ in tested dataset do
        if $M_d(u) \geq boundary$ then
            $u \Longrightarrow clean$
        else
            $u \Longrightarrow fake$
        end
    end
    return 0
end
```

Table 3.1: The Summary of Three Datasets.

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

3.4 Experimental Results and Analysis

This section provides a comprehensive evaluation of the proposed framework, including experimental setup, key influencing factors, and the performance of the detection mechanism. The analysis demonstrates the effectiveness of our approach in achieving proactive defense, maintaining model integrity, and ensuring accurate fake user detection. Furthermore, ablation studies are conducted to highlight the contributions of individual components within the framework.

3.4.1 Experimental Set Up

Selecting datasets and models. To evaluate the effectiveness of our dual-defense framework, we conducted experiments on three widely used datasets: ML-100K, ml-1m, and Last.fm [70]. Details of these datasets are provided in Table 3.1. The Last.fm underwent preprocessing, including binarizing user-item interactions, removing duplicate tags, and filtering data to mitigate cold-start issues. These steps ensured data quality and consistency across experiments. For the defense mechanism, we selected NeuMF as the target model due to its ability to effectively model implicit feedback. By capturing both linear and nonlinear user-item relationships, NeuMF enhances recommendation quality, making it a suitable benchmark for assessing the robustness of proposed defense approach.

Evaluation metrics. In the active defense line, the hit rate ($HR(t)$) of the target item is employed as the primary evaluation metric to assess the effectiveness of the proposed framework in defending against data poisoning attacks. The hit rate measures the proportion of instances in which the target item is successfully recommended within the top-t ranked items for a user. This metric provides a quantitative evaluation of the recommendation system's robustness against adversarial manipulations. By monitoring changes in $HR(t)$, we can evaluate the system's ability to resist attacks and maintain recommendation accuracy. The formula for calculating $HR(t)$ is as follows:

$$HR(t) = \frac{\sum_{i=1}^{n} I\left\{t_i \in Top\ K_i\right\}}{n}. \tag{3.10}$$

The indicator function I is defined as 1 when the condition $t_i \in Top\ K_i$ is satisfied, and 0 otherwise. Here, n represents the total number of items. In the passive defense line, we evaluate the performance of the GAN-based detector against data poisoning attacks using three key metrics: accuracy, recall (TP/(TP + FN)), and $F1$ score [148]. Accuracy measures the overall correctness of the detector, while recall focuses on the detector's ability to identify fake users effectively. The $F1$ score provides a balanced measure by combining precision and recall, offering a comprehensive assessment of the detector's performance in identifying malicious data within the training set. These metrics ensure a robust evaluation of the proposed detection framework.

$$F1 = \frac{Precision \times Recall}{2 \times (Precision + Recall)}. \tag{3.11}$$

Precision, defined as $TP/(TP+FP)$, evaluates the proportion of correctly identified fake users among all users classified as fake. Here, TP represents the number of correctly detected fake users, FN is the number of fake users misclassified as real, and FP denotes real users misclassified as fake.

Implementation specifics. In the active defense approach, simulation experiments were conducted on the NeuMF model, utilizing CLR as the primary defense mechanism. Various regularization parameters ($\lambda = 0.01, 0.1, 1.0, 3.0$) were tested to analyze their impact on defense performance. The hit rates ($HR(t)$) of target items under different conditions were compared, including raw data poisoning without defense [70], local differential privacy (LDP) [149], and HINT defense methods [150], to evaluate the effectiveness of the CLR approach.

For passive defense, Word2Vec was employed to preprocess data, addressing the sparsity of the user-item interaction matrix. The output dimensions were set to 10x10 for ML-100K, 16x16 for ml-1m, and 20x20 for Last.fm, ensuring enhanced data representation. This approach improved the quality and robustness of the detection model by

Table 3.2: Defensive Results for the Active Defense Method.

Dataset	Methods	Attack size							
		Random target items				Unpopular target items			
		0.5%	1%	3%	5%	0.5%	1%	3%	5%
ML-100K	Data poisoning attack	0.0034	0.0046	0.0100	0.0151	0.0007	0.0019	0.0111	0.0206
	LDP	0.0030	0.0035	0.0065	0.0087	0.0001	0.0002	0.0012	0.0022
	HINT	0.0032	0.0035	0.0069	0.0080	0.0001	0.0004	0.0014	0.0033
	CLR	**0.0026**	**0.0031**	**0.0044**	**0.0049**	**0.0001**	**0.0002**	**0.0010**	**0.0021**
Last.fm	Data poisoning attack	0.0047	0.0068	0.0144	0.0243	0.0012	0.0026	0.0086	0.0161
	LDP	0.0034	0.0050	0.0120	0.0210	0.0005	0.0017	0.0058	0.0118
	HINT	0.0032	0.0055	0.0069	0.0163	0.0006	0.0014	0.0047	0.0117
	CLR	**0.0031**	**0.0040**	**0.0121**	**0.0183**	**0.0005**	**0.0011**	**0.0061**	**0.0108**

creating dense vector representations for more effective analysis.

3.4.2 Achieving Proactive Defense

Proactive defensive guarantee. Experimental results validate the effectiveness of incorporating a carefully designed L_2 regularization into the original recommendation system model to counter data poisoning attacks. This enhancement significantly mitigates the hit rate ($HR(t)$) of target items under adversarial conditions. As shown in Table 3.2, with the introduction of only 0.5% fake users into the ML-100K dataset, our proposed defensive mechanism achieved a notable reduction in $HR(t)$ for random target items by 0.08%, far exceeding the 0.02% reduction achieved by HINT under comparable conditions. Moreover, as shown in Table 3.2, the proposed method consistently outperforms baselines across higher attack sizes (1%, 3%, and 5%), demonstrating robustness even under more aggressive attack scenarios. This substantial improvement in reducing target item vulnerability highlights the robustness of our defense approach, making it a more reliable solution against data poisoning attacks. These findings underscore the potential of our first line of defense to set new benchmarks in securing recommendation systems against adversarial threats.

Our proposed defense mechanism demonstrates exceptional performance compared to existing methodologies, even in challenging scenarios where attackers have partial knowledge of common user ratings. Comprehensive experiments conducted on two widely

Table 3.3: Defense Results of Partial Knowledge for Attacker.

Knowledge level	Methods	Random target items
30%	Data poisoning attack	0.0092
	LDP	0.0086
	HINT	0.0090
	CLR	**0.0072**

used datasets validate the robustness of our approach. The results reveal that when only 30% of the original user-item interaction matrix is scored, the attack hit rate ($HR(t)$) decreases significantly to 0.0092, indicating a notable reduction in the impact of data poisoning attacks. Furthermore, our novel defense strategy enhances this reduction even further, lowering the hit rate to an impressive 0.0020 for randomly selected target items.

These findings, detailed in Table 3.3, highlight the efficacy of our method in safeguarding recommendation systems. By effectively countering attacks under conditions of partial data visibility, our defense mechanism sets a new standard for protecting recommendation models from adversarial threats, ensuring higher security and reliability in real-world applications.

3.4.3 Maintaining Model Integrity

The integration of CLR into the recommendation system's original loss function serves as a robust active defense mechanism while maintaining the system's performance integrity. To validate this, we compared the recommendation system's performance using the widely accepted evaluation metric, Normalized Discounted Cumulative Gain (NDCG). Experimental results indicate that incorporating CLR has a negligible impact on the recommendation system's performance. As shown in Table 3.4, under the condition that default epochs and other parameters remain constant, the NDCG values across the three datasets (ML-100K, ml-1m, and Last.fm(Music)) exhibit minimal variation.

Table 3.4: Impact of the First Defense Line on Recommendation System Performance.

	ML-100K	ml-1m	Last.fm
Non-CLR	0.31287	0.35960	0.38990
CLR-ed	0.31371	0.35779	0.38912
NDCG Change	**+0.00084**	**-0.00181**	**-0.00078**

The observed changes in NDCG are within 0.002, demonstrating that the proposed defense mechanism effectively enhances security against data poisoning attacks without degrading the quality of recommendations. These results highlight the compatibility of CLR with existing recommendation systems, ensuring robust defense while preserving model integrity and overall user experience in real-world applications.

3.4.4 Effective Fake User Detection

Effective detection guarantee. To establish a comprehensive defense against data poisoning attacks in deep learning-based recommendation systems, we implement a second line of defense: passive defense. This approach leverages a GAN-based detection model to identify and mitigate the influence of fake users embedded within the training data. By employing this model, we aim to detect adversarially generated fake users at various scales, ensuring robust protection against poisoning attacks.

To evaluate the effectiveness of our detection mechanism, we conducted extensive experiments using datasets such as ML-100K. These experiments tested the model's ability to identify fake users across different attack intensities and scales, ensuring adaptability to diverse adversarial scenarios. Results demonstrate that the GAN-based detection model efficiently detects and filters fake users, preserving the integrity of the training data and minimizing the risk of degraded recommendation quality. This passive defense mechanism complements the active defense line, forming a dual-layered approach that strengthens the resilience of recommendation systems against data poisoning attacks, ensuring reliable and secure recommendations in practical applications.

Accuracy of GAN detection. The accuracy of our second line of the GAN detection defense method, was evaluated on the three classical datasets. Traditional rating-based detection methods have demonstrated limited effectiveness, with an accuracy of only around 70% [70]. In contrast, our GAN-based detection method achieved a significant improvement, with an accuracy of approximately 90% across all tested datasets.

Figs. 3.4, 3.5, and 3.6 illustrate significant improvements in terms of performance and robustness in identifying fake users. This marked improvement underscores the enhanced defense efficiency of our method, making it a powerful tool for combating data poisoning attacks. By reliably detecting adversarial activities, the GAN detection method contributes to the security and stability of RS models, ensuring robust recommendations even in the presence of malicious attempts to manipulate the training data.

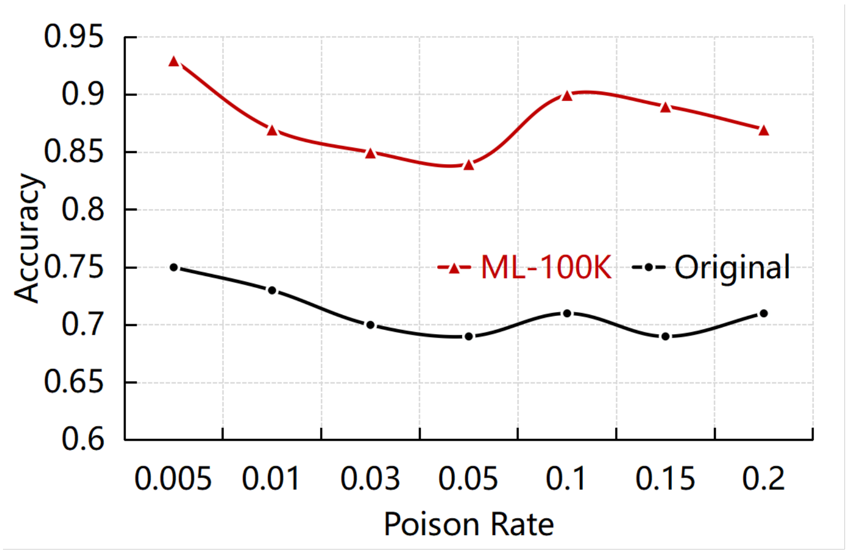


Figure 3.4: Accuracy of Detection on ML-100K.

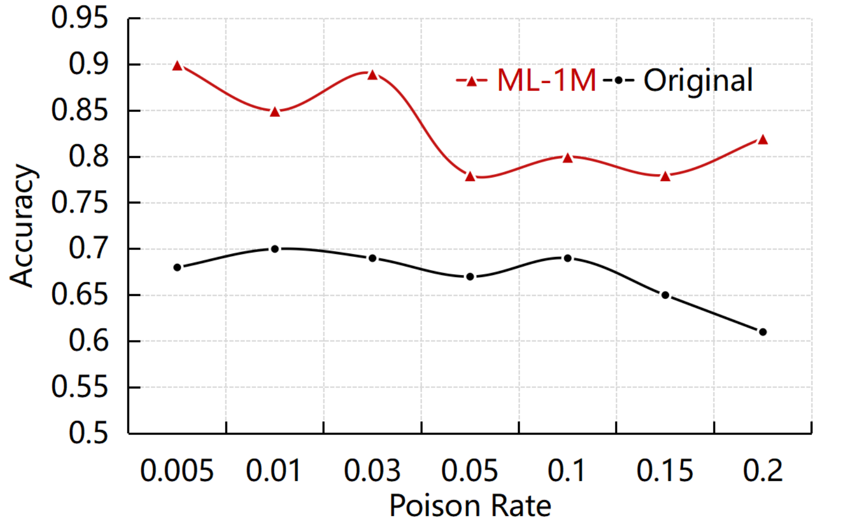


Figure 3.5: Accuracy of Detection on ml-1m.

F1 score of GAN detection. To comprehensively validate the effectiveness of our detection mechanism, we evaluated the $F1$ score, a critical metric that balances precision and recall, offering a robust measure of detection performance. As shown in Figs. 3.7, 3.8, and 3.9, our GAN-based detection method consistently achieved an average $F1$ score of approximately 85%, significantly outperforming previous detection methods, which achieved an average $F1$ score of around 65%. This marked improvement underscores

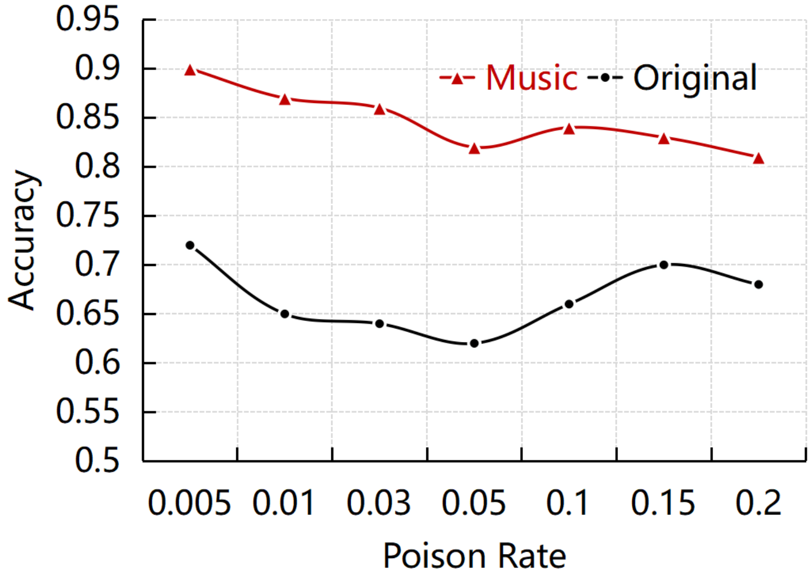


Figure 3.6: Accuracy of Detection on Music.

the capability of our approach to accurately identify fake users across various datasets. By integrating precision and recall, our method ensures a reliable and robust detection mechanism that strengthens the defense framework against data poisoning attacks, highlighting its superiority in practical applications. These findings demonstrate the enhanced resilience and applicability of our detection strategy in real-world scenarios.

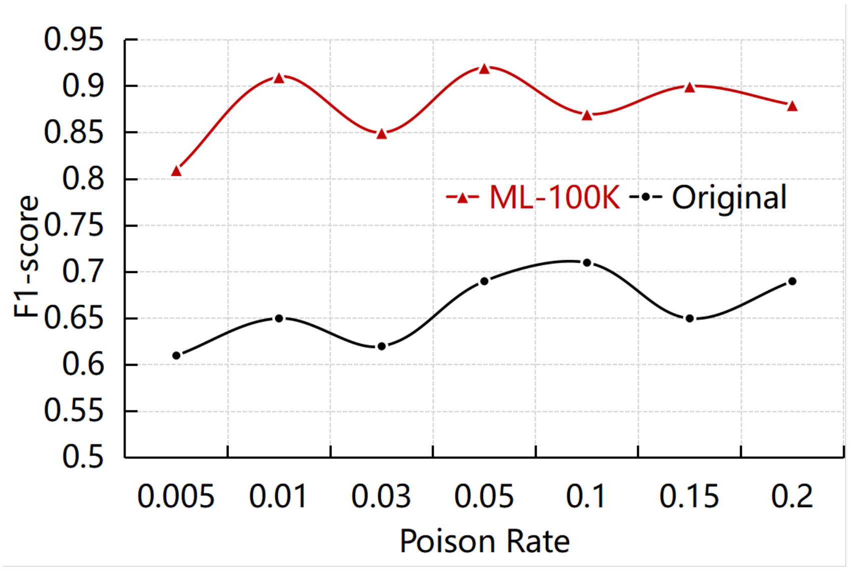


Figure 3.7: F1 score of Detection on ML-100K.

Evaluation of Recall Metric in GAN-based Detection. Recall is a critical metric for evaluating the performance of detection models, particularly in the context of

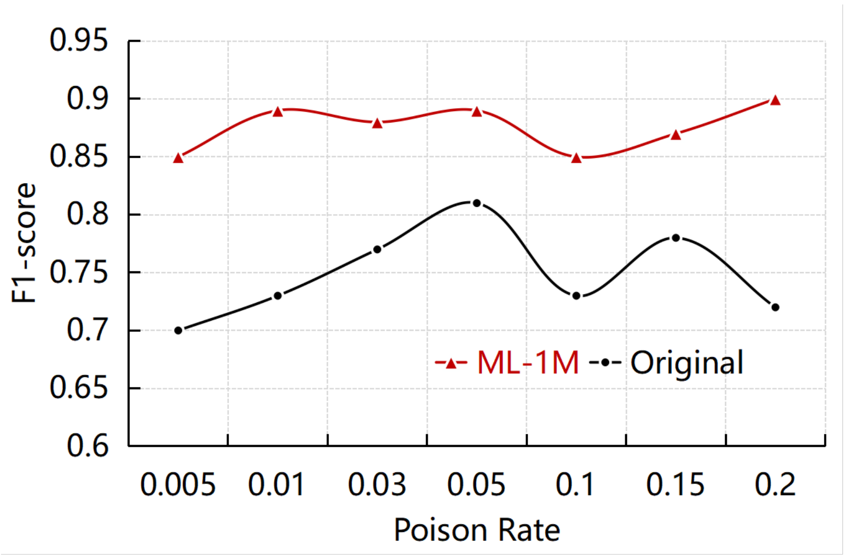


Figure 3.8: F1 score of Detection on ml-1m.

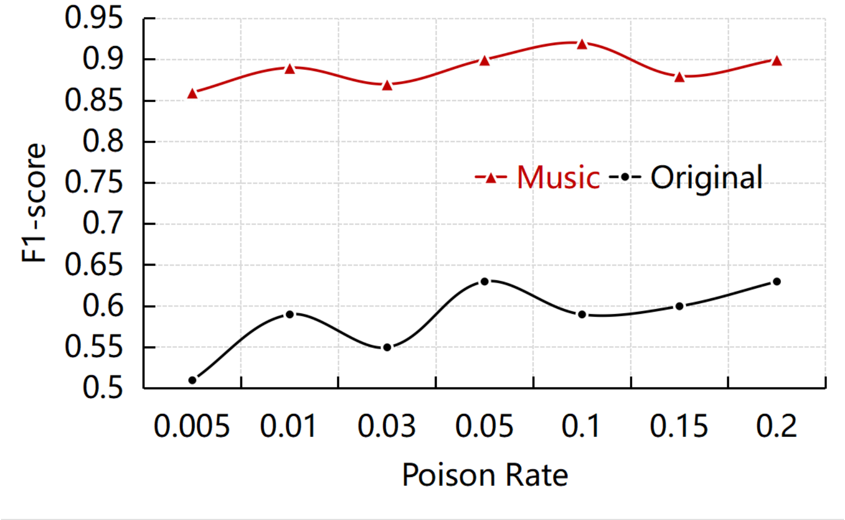


Figure 3.9: F1 score of Detection on Music.

fake user detection. It measures the ability of our defense model to correctly identify fake users within the dataset, minimizing missed detections. As illustrated in Figs. 3.10, 3.11, and 3.12, our GAN-based detection model consistently achieved a recall of approximately 90%. This high recall value underscores the model's effectiveness in detecting fake users across diverse datasets, such as ML-100K, ml-1m, and Last.fm. The results highlight the robustness and reliability of our approach in safeguarding recommendation systems and ensuring their integrity against malicious data poisoning attacks.

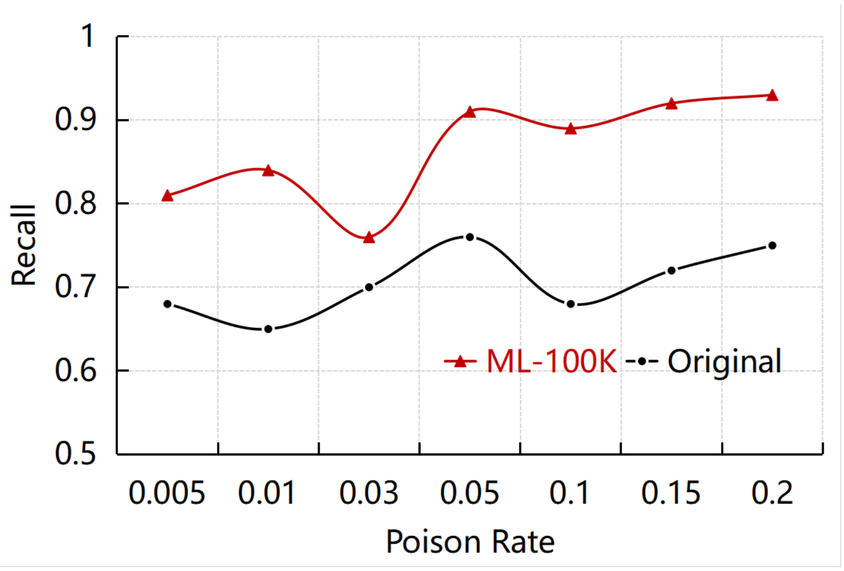


Figure 3.10: Recall of Detection on ML-100K.

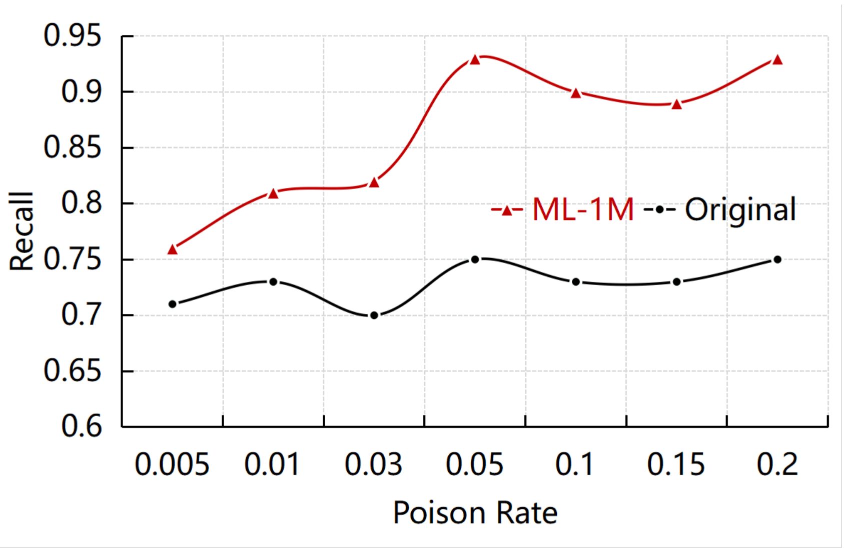


Figure 3.11: Recall of Detection on ml-1m.

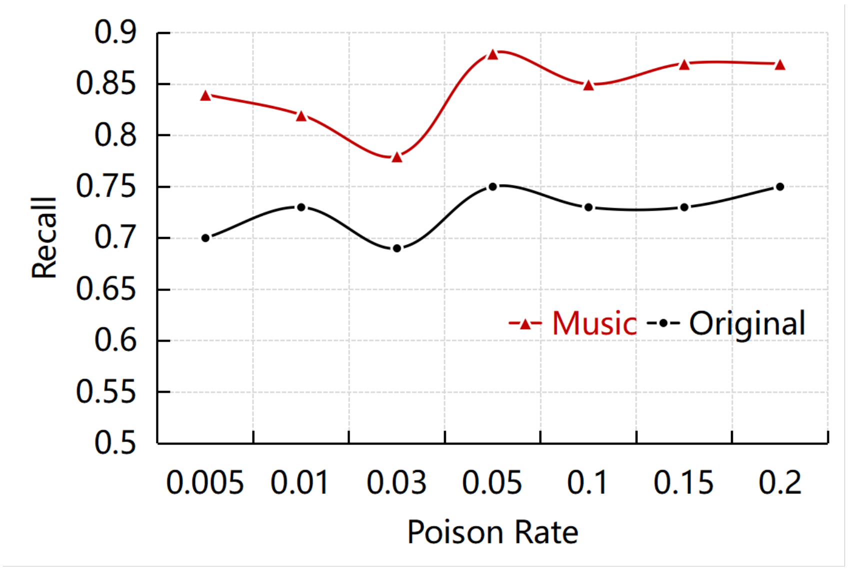


Figure 3.12: Recall of Detection on Music.

3.4.5 Ablation Studies

1)First Line of Defense (Active Defense)

The impact of different numbers of fake users. To evaluate the effectiveness of our active defense mechanism, we analyzed its performance under varying numbers of injected fake users. With a greater proportion of injected fake users, the hit rate ($HR(t)$) of the targeted item also rises across all datasets, indicating heightened adversarial impact. For instance, in the ML-100K dataset (Table 3.2), injecting 0.5% random fake users resulted in an $HR(t)$ of 0.0026, while increasing the injection to 5% raised $HR(t)$ to 0.00049. Despite this increase, our active defense method consistently demonstrates significant effectiveness in mitigating the impact of fake users, as evidenced by the results in Table 3.2, underscoring its robustness in reducing losses.

The impact of different numbers of the recommended list. Table 3.5 presents the defense results under varying sizes of the recommendation list (K). The results reveal that as K increases, the evaluation metric $HR(t)$ also rises, indicating an enhanced impact of data poisoning attacks. However, our defense method proves to be highly effective in mitigating these effects. For example, when $K = 15$, the $HR(t)$ on the ML-100K dataset is reduced to approximately 30% of the original attack's impact, demonstrating significant resilience. Moreover, even at $K = 5$, our defense strategy successfully neutralizes the poisoning attack on the Last.fm dataset, reducing $HR(t)$ to a mere 0.0003.

Table 3.5: The defense results for different recommended list size K.

Dataset	Methods	K			
		5	10	15	20
ML-100K	Data poisoning attack	0.0012	0.0019	0.0033	0.0042
	LDP	0.0006	0.0012	0.0024	0.0026
	HINT	0.0006	0.0010	0.0022	0.0028
	CLR	**0.0004**	**0.0006**	**0.0010**	**0.0019**
Last.fm	Data poisoning attack	0.0007	0.0026	0.0042	0.0061
	LDP	0.0006	0.0017	0.0029	0.0040
	HINT	0.0004	0.0021	0.0034	0.0046
	CLR	**0.0003**	**0.0021**	**0.0023**	**0.0037**

These results underscore the robustness of our approach in safeguarding recommendation systems against adversarial manipulations across varying recommendation list sizes.

2)Second Line of Defense (Passive Defense)

The impact of poison rates on detection accuracy. The effect of varying poison rates on detection accuracy is analyzed, as shown in Figs. 3.4, 3.5, and 3.6. Results indicate that lower poisoning rates for target items correspond to higher detection accuracy. For instance, using the ML-100K dataset, when the hit rate ($HR(t)$) is 0.005, the detection accuracy reaches a peak of approximately 93%. This improvement occurs because fewer fake users reduce the complexity of identifying anomalies, allowing the model to more effectively distinguish fake users from authentic ones. These findings underscore the model's robustness in maintaining high detection accuracy under low poisoning rates, ensuring the integrity of recommendation systems against adversarial attacks.

The impact of poison rates on $F1$ score. Figs. 3.7, 3.8, and 3.9 illustrate the variation in $F1$ score with different poisoning rates. We observed interesting trends. First, an optimal $F1$ score exists for each dataset and poisoning rate. Secondly, increasing the poisoning rate does not consistently lead to a higher or lower $F1$ score. The $F1$ score peaks at a poisoning rate of 0.05 for the ML-100K, while for the ml-1m and Last.fm, the maximum $F1$ score occurs at poisoning rates of 0.2 and 0.1, respectively. This non-linear impact indicates varying sensitivity to poisoning attacks across datasets, likely due to differing characteristics and user behavior patterns.

The impact of poison rates on recall. Figs. 3.10, 3.11, and 3.12 depict the variations in recall values under different data poisoning rates. The results reveal a clear trend: recall generally increases as the poisoning rate rises. Notably, across all three datasets, the highest recall values are consistently achieved when the poisoning rate reaches 0.2. This trend indicates that in scenarios with higher poisoning rates, the model becomes more effective at identifying fake users, likely due to the increased presence of distinct patterns associated with adversarial behaviors. These findings highlight the robustness and adaptability of the detection model, which maintains a strong capability to accurately identify fake users even in environments with elevated data poisoning rates.

3.5 Summary

In this chapter, we propose a dual defense mechanism to counter data poisoning attacks in deep learning-based recommendation systems. The first line of defense, active defense, is implemented using Contrastive Learning Regularization (CLR). Our experiments demonstrate that CLR effectively reduces the hit rate of data poisoning attacks across various attack intensities, with minimal impact on the recommendation system's performance. The second line of defense, passive defense, leverages a GAN-based detection model to identify fake users with high accuracy. The incorporation of real synthetic data further enhances the defense capability by expanding the training dataset, enabling the simulation model to better identify predictive differences between real and fake users. Comprehensive evaluations of three datasets validate the effectiveness of this dual-defense approach. Future research could focus on developing advanced fake user detection methods and designing more robust recommendation system models to further strengthen defenses against data poisoning attacks.

Building upon the dual defense framework proposed in this chapter to address data privacy concerns in RS, the next chapter explores a new direction that enhances model reliability while ensuring data privacy through recommendation unlearning verification. Specifically, it introduces a scheme that uses non-influential trigger data to verify whether deleted user information has been fully removed from the model's influence.

Chapter 4

4 A Novel Scheme For Recommendation Unlearning Verification (RUV) Using Non-influential Trigger Data

In the previous chapter, we explored defense mechanisms against data poisoning attacks in deep learning-based recommendation systems, demonstrating effective strategies to safeguard models against adversarial manipulations. Building on this proposition, this chapter shifts focus to recommendation unlearning verification, addressing the emerging need to ensure compliance with unlearning requests in recommendation systems.

Machine unlearning has garnered widespread attention due to its significance in privacy preservation, model usability, and adherence to legal regulations. It requires model providers to effectively remove users' data from models upon receiving unlearning requests. Recommendation systems, widely applied in big data environments, have been extensively researched in the field of deep learning. However, little attention has been paid to evaluating the effectiveness of unlearning approaches within pure tabular data-based recommendation scenarios. To bridge this gap, we propose a recommendation unlearning verification (RUV) scheme based on non-influential trigger data [151]. This approach allows users to assess whether a recommendation system adheres to unlearning requests by monitoring the recommendation rate of selected target items. Evaluation results on real-world datasets confirm the efficiency and accuracy of our RUV scheme, offering a practical solution for verifying unlearning compliance in recommendation systems while ensuring user privacy and system reliability.

The rest of the chapter is organized as follows. Section 4.1 introduces the foundations of RUV. Section 4.2 outlines the RUV framework. Section 4.3 describes the operational mechanism of RUV. Section 4.4 presents the experimental results and analysis. Finally, Section 4.5 provides a summary of the chapter.

4.1 Foundations of RUV

The primary objective of our work is to verify whether an RS model can effectively perform an unlearning action upon receiving an unlearning request from a curious user. This verification is crucial to guarantee adherence to privacy-centric regulatory frameworks while upholding user confidence. In this section, we outline the goals and responsibilities of the three key roles involved in the RUV process: the curious users, the recommendation system, and the Certification Authority (CA). Curious users aim to verify that their unlearning requests have been correctly executed by observing changes in the recommendation results. The RS is responsible for processing unlearning requests and ensuring that relevant data is effectively removed from its models. The Certification Authority serves as an independent entity to oversee the process, verify compliance, and provide assurances to users regarding the integrity of the unlearning process. This collaborative effort ensures that unlearning requests are handled securely and transparently.

- **Curious users' goals.** In a recommendation system, numerous users are interacting with the platform. Among these users, we define a specific subset known as curious users, who are particularly concerned with verifying whether the system genuinely acts upon their unlearning requests. These users are motivated by privacy concerns and aim to ensure that their data is effectively removed from the system's influence after submitting an unlearning request. Curious users actively monitor and evaluate the system's behavior to detect any potential discrepancies, holding the system accountable for compliance with privacy standards. Their role is crucial in fostering trust and transparency in the recommendation unlearning process.

- **Curious users' background knowledge.** In this scenario, a curious user plays an important role in verifying the unlearning capabilities of an RS. A curious user can leverage their own data as training data for the recommendation model, providing a baseline to monitor the model's behavior. They possess the ability to submit both recommendation requests and unlearning requests to the RS, ensuring their data is removed from the system as per their request. Additionally, the user can request the RS to demonstrate compliance with unlearning protocols through the RUV framework. However, it is essential to note that the power of a curious user is limited. These users are primarily concerned with privacy-related issues and are not equipped to perform large-scale testing or submit a substantial number of requests under multiple identities. This constraint ensures that the verification process remains practical and fair while preventing potential misuse of the system.

- **The objectives of RS.** In this verification setting, the primary goal of the RS is to acquire sufficient training data to build a high-quality model capable of providing accurate and reliable recommendations to users. At the same time, the RS must efficiently handle user requests for data unlearning and perform model retraining to comply with privacy regulations and user expectations [152]. For instance, when a user submits an unlearning request, the system is expected to process the request promptly, ensuring that the corresponding data is effectively removed and the model is updated accordingly. However, some recommendation systems may neglect or delay handling such requests to save computational resources and retraining costs. This potential non-compliance underscores the need for Recommendation Unlearning Verification, which ensures that the RS adheres to user requests and maintains its integrity in meeting privacy-preserving obligations.

- **The abilities of RS.** The RS can receive, store, and utilize user data to train high-quality models designed to provide accurate and reliable recommendation services. By leveraging the full scope of available user data, the RS can optimize its algorithms to deliver personalized recommendations, enhancing user experience and system efficiency. Furthermore, the RS is equipped to handle unlearning requests from curious users, enabling it to remove specific user data from its training set and perform model retraining to ensure compliance with privacy-preserving regulations and user demands [153].

- **CA's goals.** As an independent third party, the primary goal is to fairly and impartially evaluate the output of the recommendation system and relay the findings back to the curious user. This third party plays a critical role in ensuring transparency and accountability in the verification process. It aims to minimize the potential risks of bias or manipulation that could arise if curious users directly evaluate the RS output. By overseeing the interactions between curious users and the RS, the third party controls user privileges and ensures that verification procedures are conducted objectively and consistently. This approach strengthens the fairness of the verification process and enhances trust in the RS's compliance with unlearning and privacy-preserving requirements.

- **CA's authority.** In existing architectures for unlearning validation, curious users and the validated recommendation models are typically considered independent entities, and both are treated as untrustworthy [154][155]. To address this trust gap and simulate realistic scenarios for RUV, we introduce a CA. The CA is a trusted third party recognized by both users and the RS. It acts as a mediator to ensure fairness and transparency in the verification process. The CA has access to critical information, such as information on user participation within the RS's training data. During the RUV process, the CA can issue multiple query requests to the RS, evaluate the system's compliance with unlearning requests, and leverage its authority to oversee the process. By bridging the trust gap, the CA ensures that the verification procedure is objective and reliable for all stakeholders.

4.2 RUV Framework

Our design focuses on verifying recommendation unlearning within the framework of a Certification Authority. As illustrated in Fig. 4.1, the RUV mechanism comprises three main steps to ensure fairness, transparency, and effectiveness in verifying unlearning requests. Below, we provide a detailed analysis of the workflow:

Fig. 4.1 (step 1) illustrates the preparation phase of our proposed verification framework. In this stage, we assume that the recommendation system model is constructed using a known loss function $\mathcal{L}$, which guides the generation of fake data D_F (comprising D_{Fu} and D_{Fi}). For each D_{Fu}, the RS model predicts ratings for all items and selects certain items as fake items, forming D_{Fi}. These generated D_{Fu} are incrementally added to the training set, and the RS model is continuously updated until a sufficient number of fake items is generated. Importantly, the inclusion of fake items D_{Fi} is designed to be non-influential to the model's training process, ensuring the integrity of the personalized recommendation model. Once the RS receives data from all users, it performs incremental training to construct a robust and accurate recommendation model while embedding the generated triggered data for subsequent unlearning verification.

After injecting some trigger data into the RS model via the curious user, and upon completion of model training, the curious user requests the RS to pre-check the recommendation rate of the target item t through the CA. This process is illustrated in Fig. 4.1 (step 2). In real-world scenarios, curious users often have limited capacity to submit unlearning requests. An untrusted RS may choose to ignore these requests to save computational and retraining costs. To address this, the CA acts as an independent third party to ensure compliance by verifying the RS's actions. This pre-checking phase establishes a baseline for the recommendation rate of the target item and prepares the system for subsequent verification steps. By providing oversight, the CA enhances transparency and accountability, ensuring the integrity of the unlearning verification process.

In Step 1, a curious user injects fake trigger data into the training dataset and later submits an unlearning request to the recommendation system. The RS may claim compliance with the request, regardless of whether the unlearning action was genuinely executed. To verify this, the user engages a trusted third-party Certification Authority to analyze the recommendation rate of the target item. As depicted in Fig. 4.1 (Step 3), if the unlearning process was truly performed, the recommendation rate for the target item should show a noticeable decrease. This verification step ensures that the RS is held accountable for processing unlearning requests. By detecting inconsistencies in the recommendation rate, the CA helps identify whether the RS has complied with unlearning requirements, providing a transparent and robust mechanism for ensuring user data privacy.

Together, these steps establish a robust and comprehensive RUV mechanism. The combination of triggered data injection, independent oversight by the CA, and precise recommendation rate analysis ensures that recommendation systems remain accountable and transparent in addressing unlearning requests. Subsequent sections will explore the implementation details and evaluation results of this mechanism.

4.3 RUV Operational Mechanism

In this section, we detail the implementation of recommendation unlearning verification. Our proposed design workflow can be summarized into three main steps: (1) inject-

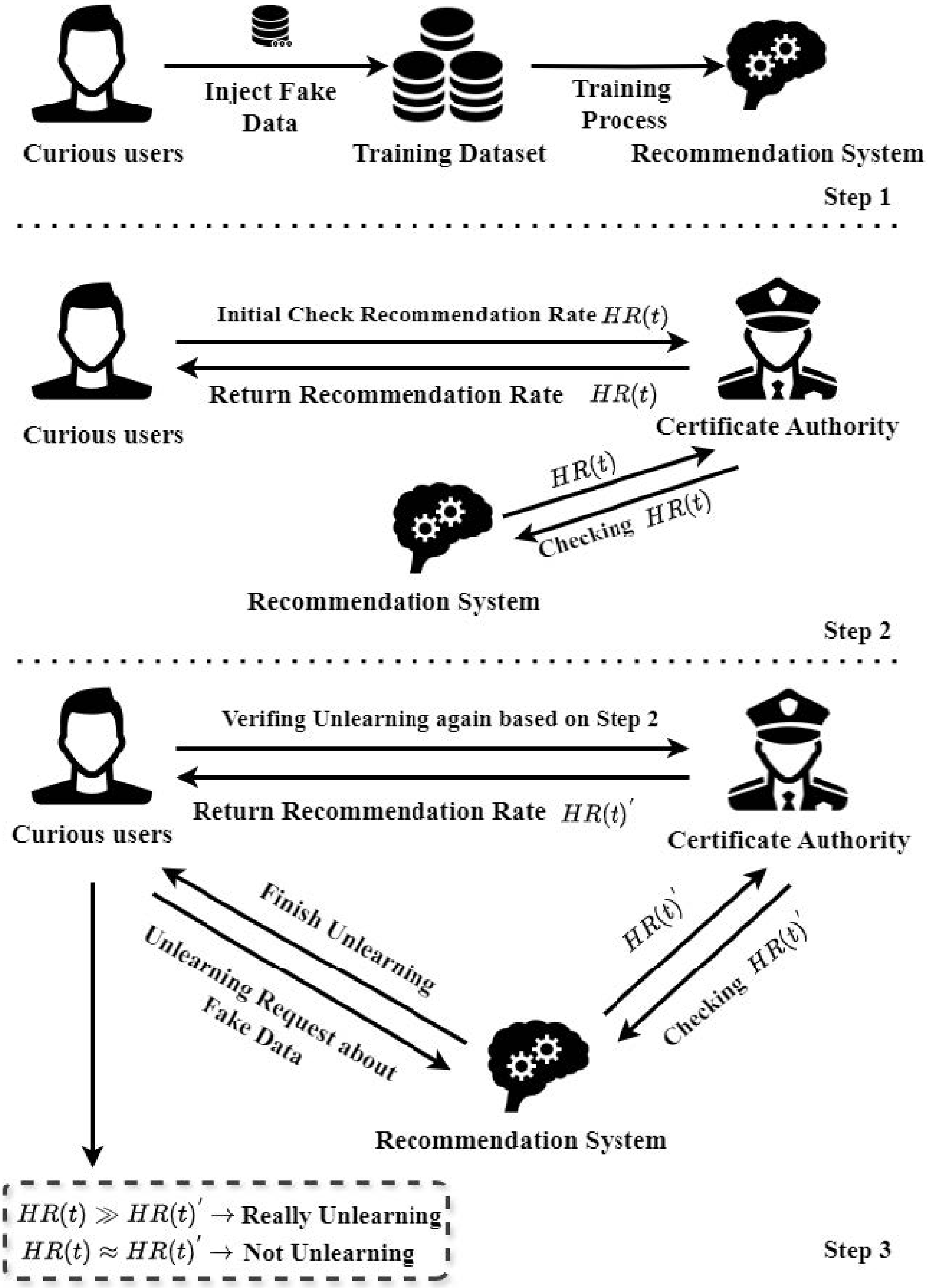

Curious users
Inject Fake Data
Training Dataset
Training Process
Recommendation System
Step 1
Initial Check Recommendation Rate $HR(t)$
Return Recommendation Rate $HR(t)$
Curious users
Certificate Authority
$HR(t)$
Checking $HR(t)$
Recommendation System
Step 2
Verifing Unlearning again based on Step 2
Return Recommendation Rate $HR(t)'$
Curious users
Certificate Authority
Finish Unlearning
Unlearning Request about Fake Data
$HR(t)'$
Checking $HR(t)'$
Recommendation System
$HR(t) \gg HR(t)' \rightarrow$ Really Unlearning
$HR(t) \approx HR(t)' \rightarrow$ Not Unlearning
Step 3

Figure 4.1: Framework of Recommendation Unlearning Verification (RUV).

ing non-influential triggered data into the training dataset, (2) introducing a Certification Authority for fair and comprehensive verification, and (3) analyzing the recommendation rate of target items to validate unlearning compliance.

Algorithm 3 outlines a heuristic approach for addressing the Recommendation Unlearning Verification problem. The proposed verification framework is structured into three sequential steps: injecting non-influential trigger data, leveraging a Certification Authority for compliance checking, and analyzing the recommendation rate to determine unlearning effectiveness.

4.3.1 Injecting Non-influential Trigger Data and Model Training

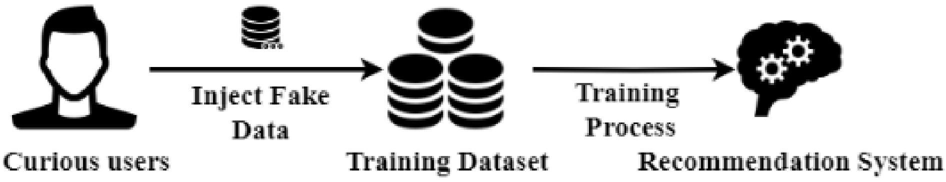


Figure 4.2: Injecting Non-influential Trigger Data and Model Training.

The first step of our method is Fig. 4.2. Our work is grounded in the Neural Collaborative Filtering architecture, a representative deep learning paradigm for recommendation systems. The interactions, represented as $\{D_u, D_i, r_{max}\}$, are structured in a matrix Y, where preference ratings range from 0 to 5. To facilitate recommendation unlearning verification, we inject m fake user-item pairs into the original dataset, progressively adding them one by one based on varying privacy levels. This process generates the final training dataset D, incorporating both real and fake data. To ensure efficient and verifiable unlearning, the model M is iteratively updated using an optimized loss function designed for this purpose. This loss function balances accuracy and adaptability, enabling the RS to accommodate unlearning requests while maintaining high-quality recommendations. The proposed method enhances the transparency and reliability of unlearning mechanisms in recommendation systems.

$$\mathcal{L} = l + \alpha \cdot V\left[y(D_{Fu})\right], \tag{4.1}$$

l is the loss function of the original recommendation system, which is not crafted, and α is a weighting factor to balance the relative importance of L and the verification-related loss function $V\left[y(D_{Fu})\right]$. The $V\left[y(D_{Fu})\right]$ is a loss function related to the verification

Algorithm 3: Recommendation Unlearning Verification (RUV)

```
Input: Curious user $U$, fake data $D_F$, fake user $D_{Fu}$, fake item $D_{Fi}$, original
       training dataset $D_o$, target item $t$, parameters $m$, $N$
Output: The verification result $q$
begin
    # Injecting $D_F$ and getting recommendation system model $M$:
    for $D_{F1}$, $D_{F2}$, ..., $D_{Fm}$ do
        Initialize the recommendation model $M$ considering progressively augmented
          input data.
        Add the $D_F(D_{Fu}, D_{Fi}, r_{Fmax})$ to $D_o$;
        Get User-item interaction matrix $Y \leftarrow D = D_o \cup D_F$;
        Training $M$ on $Y$ with $\mathcal{L}$;
    return Recommendation system $M$.
    # Pre-check Recommendation Rate $HR(t)$ of $t$:
    for each $R_U$ from curious users $U$, $U \in \{1, N\}$ do
        $U$ check the $HR(t)$ through $CA$ on $M$;
        $M \xrightarrow{generate} HR(t)$ using Eq.(1);
        $CA \xleftarrow{receive} HR(t)$;
    return $HR(t)$ to curious users $u$ from $CA$.
    # Recommendation unlearning verification:
    for each $R_u$ from curious users $U$, $U \in \{1, N\}$ do
        $U \xrightarrow{R_u} M$;
        $U \xleftarrow{finish\ R_u} M$;
        $U$ check the $HR(t)'$ through $CA$ on $M$;
        $M \xrightarrow{generate} HR(t)'$ using Eq.(1);
        $U \xleftarrow{CA} HR(t)'$;
        if $HR(t) >> HR(t)'$ then
            set $q = 1$
        else if $HR(t) \approx HR(t)'$ then
            set $q = 0$
        else
            set $q = -1$
    return verification result $q$
```

target item, which is calculated as shown in equation (4.2).

$$
\begin{aligned}
V\left[y(D_{Fu})\right] &= \|y(D_{Fu})\|_2^2 + \beta \cdot l', \\
y(D_{Fu}) &\in [0, r_{max}].
\end{aligned}
\tag{4.2}
$$

In the proposed loss function, β serves as a weighting factor to balance the relative importance between the regularization term and the loss function. Additionally, $y(D_{Fu})$

represents a vector of predicted ratings for all items associated with the fake user D_{Fu}, ensuring targeted unlearning efficiency. Here, $l^{'}$ (Eq. 4.3) promotes the target item's ranking by reducing its score gap with other items, while $\|y(D_{Fu})\|_2^2$ preserves stealthiness of fake-user ratings, and $\beta \cdot l^{'}$ enhances attack effectiveness.

To optimize the recommendation system, we minimize $l^{'}$, aiming to maximize the frequency at which the target item t occurs in the *Top K* ranked list r_t. This approach effectively enhances the recommendation rate $HR(t)$ for t. The loss function is applied to all regular users u who did not rate the target item t, ensuring that the system prioritizes the inclusion of t in the recommendations without disrupting the preferences of other users, thereby improving overall recommendation performance.

$$l^{'} = \sum_{u \in A} \max \left\{ \min_{i \in r_t} \log \left[y(D_{ui}) \right] - \log \left[y(D_{ut}) \right], -s \right\}, \tag{4.3}$$

where A is the set of regular users. The target item t ranks higher in more users' lists as $l^{'}$ becomes smaller, and the higher the recommendation rate $HR(t)$ is. Furthermore, a configurable parameter $s > 0$ makes sure that the target item t is positioned significantly higher than the lowest-ranked entry in the recommendation list. The recommended list will be empty when the target item t appears on the recommended list. Otherwise, we should consider the user's needs and improve the ranking of the target item.

4.3.2 Pre-check Recommendation Rate HR(t)

To evaluate the effectiveness of recommendation unlearning, we use the recommendation rate $HR(t)$ of the target item t as the key metric for verification, as shown in Fig. 4.3. At the initial stage of performing recommendation unlearning verification, suppose there are N curious users U who wish to submit recommendation unlearning requests R_U to the recommendation system M. These users must first determine the recommendation rate $HR(t)$ of the target item t prior to unlearning. This initial measurement serves as essential preparation for subsequent RUV comparison tasks, particularly in the last step.

In real-world recommendation unlearning verification scenarios, the curious users rely on the Certification Authority to obtain the baseline recommendation rate $HR(t)$ of the target item t before unlearning occurs. The CA ensures the accuracy and transparency of this initial measurement, which provides a reference point for determining whether

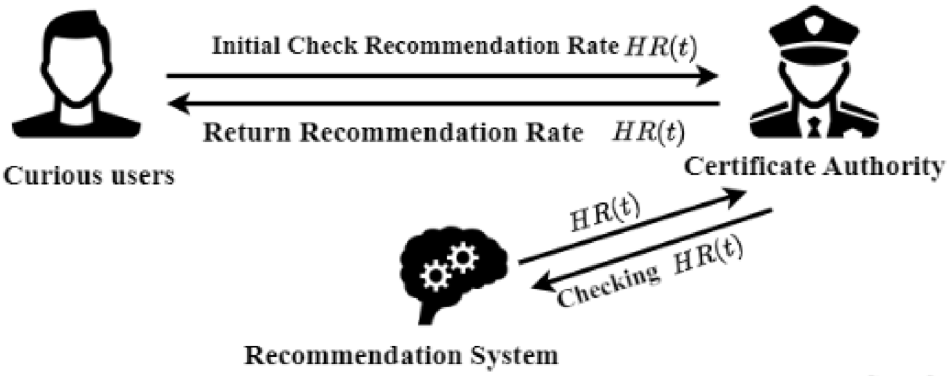


Figure 4.3: Pre-check recommendation rate HR(t).

the RS has effectively removed the influence of t after processing the unlearning request. This step is crucial for a reliable and fair unlearning verification process.

4.3.3 Recommendation Unlearning Verification Process

Fig. 4.4 illustrates the most critical step in the Recommendation Unlearning Verification process, which ensures that the recommendation system adheres to unlearning requests submitted by curious users. When a curious user U submits an unlearning request R_U to the recommendation system M, the RS returns an unlearning signal to the user, regardless of whether it genuinely executes the unlearning operation. To verify M's compliance, the curious user collaborates with a trusted third-party CA. After the unlearning process is claimed to be completed, the CA assists U in obtaining the updated recommendation rate $HR(t)'$ for the target item t. The verification is conducted by comparing the recommendation rates $HR(t)$ (before unlearning) and $HR(t)'$ (after unlearning). If $HR(t) > HR(t)'$, the RS has successfully executed the unlearning operation, and the verification result is set to $q = 1$. Conversely, if $HR(t) \approx HR(t)'$, the RS has failed to comply with the unlearning request, and $q = 0$. In cases where the process encounters inconsistencies, the verification is deemed invalid, and $q = -1$. For example, this may occur if adversaries exploit the unlearning process to promote the target item, or if system noise and model retraining dynamics inadvertently increase its ranking.

Several factors are critical in this verification step. First, key-based trigger data, such as the target item t, serves as the foundation for monitoring whether the RS complies with unlearning requests. Second, the CA is essential to ensuring fairness and transparency by independently verifying the updated recommendation rate $HR(t)'$. Lastly, the com-

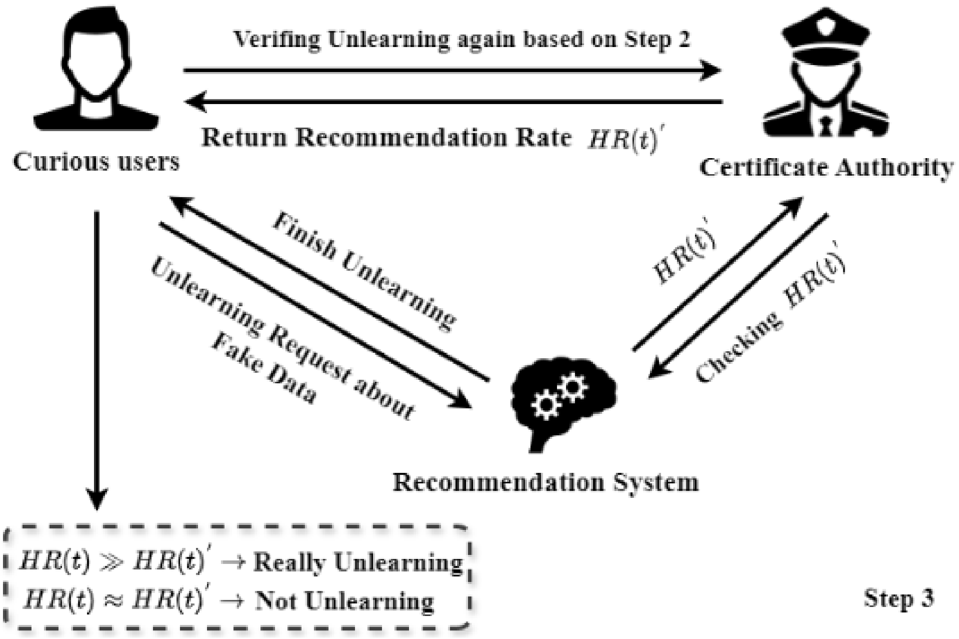

Verifing Unlearning again based on Step 2
Return Recommendation Rate $HR(t)'$
Curious users
Certificate Authority
Finish Unlearning
Unlearning Request about Fake Data
$HR(t)'$
Checking $HR(t)'$
Recommendation System
$HR(t) \gg HR(t)' \rightarrow$ Really Unlearning
$HR(t) \approx HR(t)' \rightarrow$ Not Unlearning
Step 3

Figure 4.4: Recommendation Unlearning Verification Process.

parison between $HR(t)$ and $HR(t)'$ provides a clear indication of whether the unlearning operation has been successfully performed. By verifying that the RS has adhered to unlearning requests, the curious user can confidently continue utilizing the recommendation service, assured that their data privacy has been respected and handled appropriately. This process strengthens trust and accountability in RS.

4.4 Experimental Results and Analysis

This section presents a comprehensive evaluation of the proposed framework, covering the experimental setup, metrics, and key performance factors. The analysis is structured to validate the framework's ability to guarantee recommendation unlearning, ensure trigger data invisibility, and maintain model integrity. The results confirm the effectiveness of the approach in providing robust unlearning verification while maintaining the effectiveness of the recommendation model. Furthermore, ablation studies are conducted to analyze the contributions of each component in the framework, highlighting their respective roles in achieving the desired outcomes. These evaluations offer in-depth insights into the system's holistic effectiveness and reliability in real-world scenarios.

4.4.1 Set Up and Metrics

Datasets and models. To assess the efficacy of the proposed Recommendation Unlearning Verification framework, we use three real-world datasets: MovieLens-100K (ML-100K), MovieLens-1M (ml-1m), and Last.fm. The ML-100K dataset consists of 100,000 user ratings provided by 943 users for 1,682 movies. The ml-1m dataset is larger, containing 1,000,209 ratings for 3,706 movies by 6,040 users. The Last.fm dataset includes 186,479 tag assignments involving 1,892 users and 17,632 artists.

To ensure consistent evaluation, we preprocess the Last.fm dataset by binarizing user-artist interactions, removing duplicate entries, and applying iterative filtering to address the "cold start" problem, which can negatively affect recommendation quality. These preprocessing steps transform the data into implicit feedback, suitable for use in recommendation systems. We test our RUV framework using NCF, a widely adopted deep learning-based recommendation model known for effectively capturing user-item interactions. These datasets and models provide a robust foundation for evaluating the unlearning verification process across diverse scenarios.

Evaluation metrics. The recommendation rate $HR(t)$ of the target item t is employed as the primary evaluation metric to assess the effectiveness of the recommendation unlearning operation. $HR(t)$ quantifies the frequency at which t appears in the top-K recommendations and is computed using the following equation:

$$HR(t) = \frac{\sum_{i=1}^{n} I\left\{t_i \in Top\ K_i\right\}}{n}. \tag{4.4}$$

In the calculation of the recommendation rate $HR(t)$, I represents an indicator function that equals 1 when the condition $t_i \in Top\ K_i$ is satisfied, indicating that the target item t appears within the top-K recommendations for user i, and 0 otherwise. Here, n denotes the total number of users.

To evaluate the effectiveness of the recommendation unlearning operation, we perform the following steps: (1) Before the unlearning process, we measure and record the initial recommendation rate $HR(t)$ of the target item t, providing a baseline for comparison. (2) After the unlearning operation, the updated recommendation rate $HR(t)'$ for t is obtained through the CA. Finally, we compare the results from here's steps (1) and (2)

Table 4.1: The results of the $HR(t)$ for recommendation unlearning verification.

Dataset	Recommendation Rate		Comparation		unlearning
	$HR(t)$ *before unlearning*	$HR(t)'$ *after unlearning*	$HR(t) > HR(t)'$	$HR(t) \approx HR(t)'$	✓/ ×
ML-100K	0.0034	0.0025	✓	×	✓
ml-1m	0.00022	0.00021	×	✓	×
last.fm	0.0047	0.0024	✓	×	✓

to determine the impact of the unlearning operation. For accuracy and reliability, all results are averaged over three independent trials by default, ensuring consistency and robustness in the evaluation.

Implementation details. We evaluate the proposed unlearning verification method for recommendation systems by monitoring the change in the recommendation rate $HR(t)$ of the target item t from the $Top\ K$ recommendation list. The process begins by injecting m fake data into the training dataset. To improve recommendation accuracy and model robustness, negative samples (items disliked by users) are paired with positive samples (items interacted with by users) for each user. The model is trained using NCF for 30 epochs, employing the Adam optimizer and an $HR(t)$-based early stopping strategy to avoid overfitting. After training, the recommendation rate $HR(t)$ of the target item is recorded for both the initial training model and the model after performing unlearning. This comparison provides an accurate evaluation of the effectiveness of the unlearning operation within the RS framework.

4.4.2 Recommendation Unlearning Guarantee

Table 4.1 presents the verification results for recommendation unlearning, demonstrating the effectiveness of the proposed approach. In this scenario, a curious user selects a specific piece of data for an unlearning request. The verification process evaluates the success of the unlearning operation by analyzing the recommendation rate $HR(t)$ of the target item t. In real-world applications, once an unlearning request is submitted, the RS retrains the model, which is not directly exposed to the user, ensuring data privacy.

For the experiments, we inject 500 fake user data points into the training dataset. The default parameters are set as $Top\ K = 10$, $n = 30$, and $nn = 4$. On the ml-1m dataset, the results show $HR(t) \approx HR(t)'$, indicating that the unlearning operation was

Table 4.2: Invisibility of Trigger Data in RUV.

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
	KIA	0.1267	0.1273	0.1283	0.1290	0.3800	0.3443	0.2353	0.2360

insufficient. Conversely, on the ML-100K dataset, $HR(t) > HR(t)'$, confirming that the unlearning operation was effectively performed. This weaker effect may be attributed to the larger scale and higher sparsity of ml-1m, which dilutes the impact of 500 injected points, making their removal less influential on overall recommendations. These results, as outlined in Table 4.1, validate the proposed recommendation unlearning verification method. The significant drop in $HR(t)'$ compared to $HR(t)$ provides evidence that the RS successfully removed the requested data from its model, ensuring compliance with the unlearning request and maintaining user privacy.

However, the novelty of our approach lies in leveraging a pure tabular data-based recommendation system model, which fundamentally differs from existing methods that often rely on complex embeddings or hybrid data representations [156][157]. This distinction makes a direct comparison with these methods infeasible, as they operate in fundamentally different settings. Instead, we have conducted an extensive evaluation of the proposed approach, focusing on the influence of various factors such as data quality, model parameters, and unlearning efficiency within the context of our current framework. These evaluations offer meaningful perspectives on the robustness, effectiveness, and adaptability of our method in similar scenarios. The results highlight its potential as a reliable solution for addressing privacy-preserving unlearning in pure tabular data-based RS models, paving the way for its application in broader contexts.

4.4.3 Trigger Data Invisibility Guarantee

To evaluate the invisibility of trigger data in the RUV framework, we conducted experiments on the ML-100K dataset under two potential attack scenarios: Support Vector Machine (SVM) attacks and Key Item Analysis (KIA). As shown in Table 4.2, the false positive rate (FPR) consistently remains low, averaging around 12% across varying trigger data sizes ($m = 50, 1000, 3000, 5000$). This indicates that the injected trigger data

Table 4.3: Model Integrity in RUV.

	ML-100K			ml-1m			Last.fm		
Non-trigger data	0.31075			0.33608			0.39138		
Triggered data	m=5	m=50	m=100	m=5	m=50	m=100	m=5	m=50	m=100
	0.30842	0.31362	0.30588	0.33422	0.33873	0.33954	0.37451	0.37962	0.37206
NDCG change	**- 0.00233**	**+ 0.00287**	**- 0.00487**	**+ 0.00186**	**+0.00265**	**+0.00346**	**-0.01687**	**-0.01176**	**-0.01932**

remains largely undetectable under adversarial analysis, maintaining its stealthiness.

Similarly, the false negative rate (FNR) is observed to stay within acceptable limits under both SVM attacks and KIA scenarios. For instance, at $m = 3000$, the FNR is 23.5%, demonstrating that the trigger data, while difficult to detect, allows sufficient reliability for verification purposes. These results confirm that the injected trigger data is effectively invisible to adversarial models, ensuring the integrity and security of the RUV framework while facilitating accurate and reliable recommendation unlearning verification. This balance between invisibility and verifiability underscores the robustness of our approach to preserving the privacy of the verification process. Specifically, the pure tabular data-based design makes trigger data closely resemble normal user data, improving stealthiness but inevitably increasing the FNR. A moderate FNR (e.g., 23.5%) is thus considered acceptable, as it maintains sufficient detectability for verification while ensuring that the trigger data remains inconspicuous to potential adversaries.

4.4.4 Ensuring the Model Integrity

To evaluate the impact of trigger data injection on the integrity of RS models, we carried out experiments on three datasets. As shown in Table 4.3, model performance was assessed using the Normalized Discounted Cumulative Gain (NDCG), a key metric for recommendation quality. The results indicate that the performance impact of injecting trigger data is minimal, with changes in NDCG values remaining within 0.02 across all datasets, regardless of the size of the injected trigger data (m).

For example, in the ML-100K dataset, NDCG changes from -0.00487 to $+0.00287$, showing negligible performance degradation. Similarly, in the ml-1m dataset, changes vary between $+0.00186$ and $+0.00346$, while in the Last.fm dataset, the largest change is -0.01932, still well within acceptable limits. These results confirm that the injected trigger data does not compromise the overall recommendation quality or disrupt the recommendation system's functionality.

This demonstrates that our approach effectively integrates trigger data for verification purposes without negatively impacting the RS's recommendation performance, ensuring model integrity while maintaining the reliability of the recommendation system.

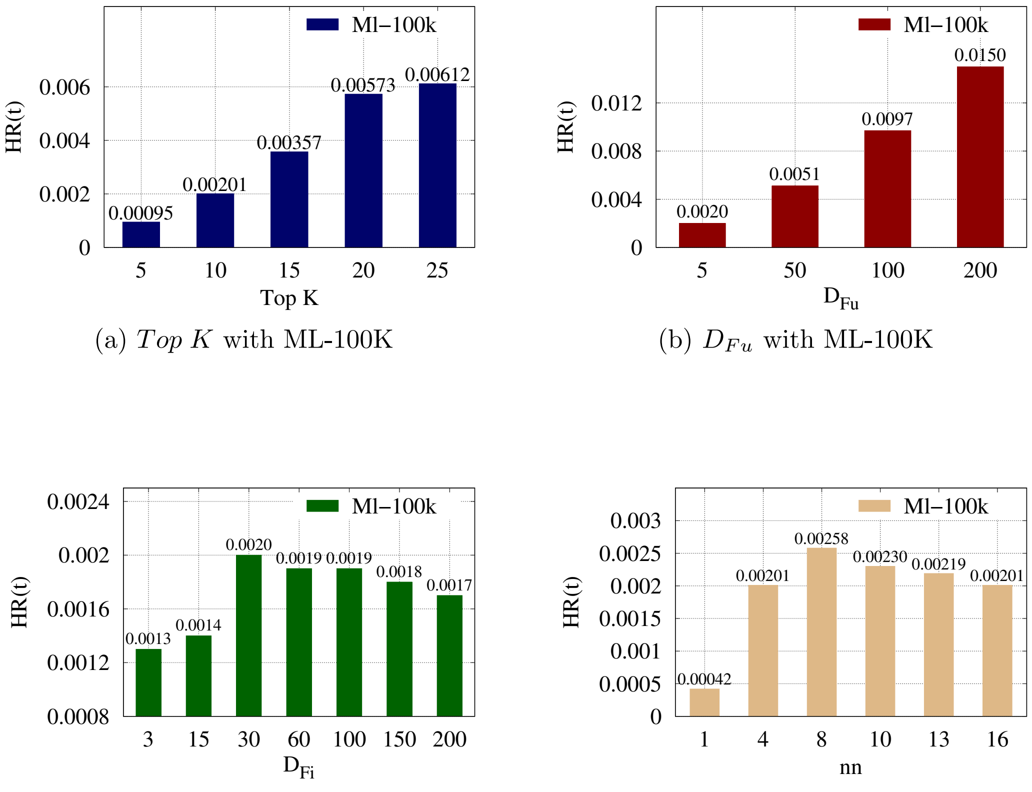
4.4.5 Ablation Studies

Impact of the Top K. The impact of $Top\ K$ on the recommendation rate $HR(t)$ is evaluated across three datasets, with results illustrated in Fig. 4.5(a), Fig. 4.6(a), and Fig. 4.7(a). The analysis reveals a consistent trend: as $Top\ K$ increases, the recommendation rate $HR(t)$ for the target item t also increases across all datasets. For example, in the ML-100K dataset, the recommendation rate $HR(t)$ for the target item rises approximately 5.4 times when $Top\ K = 5$ is expanded to $Top\ K = 25$.

This increase is attributed to the fact that larger recommendation lists enhance the probability of the target item t appearing within the top-ranked recommendations. The trend is consistent across the ml-1m and Last.fm datasets, further validating this observation. These findings highlight the influence of $Top\ K$ on recommendation effectiveness, emphasizing the need to consider list size as a factor when evaluating recommendation unlearning and system performance in practical applications. This ensures a comprehensive understanding of system behavior under different operational settings.

Impact of the fake users. In our experiments, we observe that the recommendation hit rate $HR(t)$ for target items increases as the number of injected fake users D_{Fu} grows across all datasets. This trend is illustrated in Fig. 4.5(b). For instance, on the ML-100K dataset, $HR(t)$ initially reaches 0.0020 with a small number of fake users. However, when 5% of fake users D_{Fu} are injected, $HR(t)$ increases significantly to 0.0151.

This behavior is expected because, as the proportion of fake users incorporated into the training dataset increases, the target items associated with these fake users (non-influential trigger data) appear more frequently in the overall training data. This increased frequency enhances the model's likelihood of recommending the target items, thereby boosting $HR(t)$. While this observation highlights the potential for manipulating recommendation outcomes by injecting fake users, it also underscores the importance



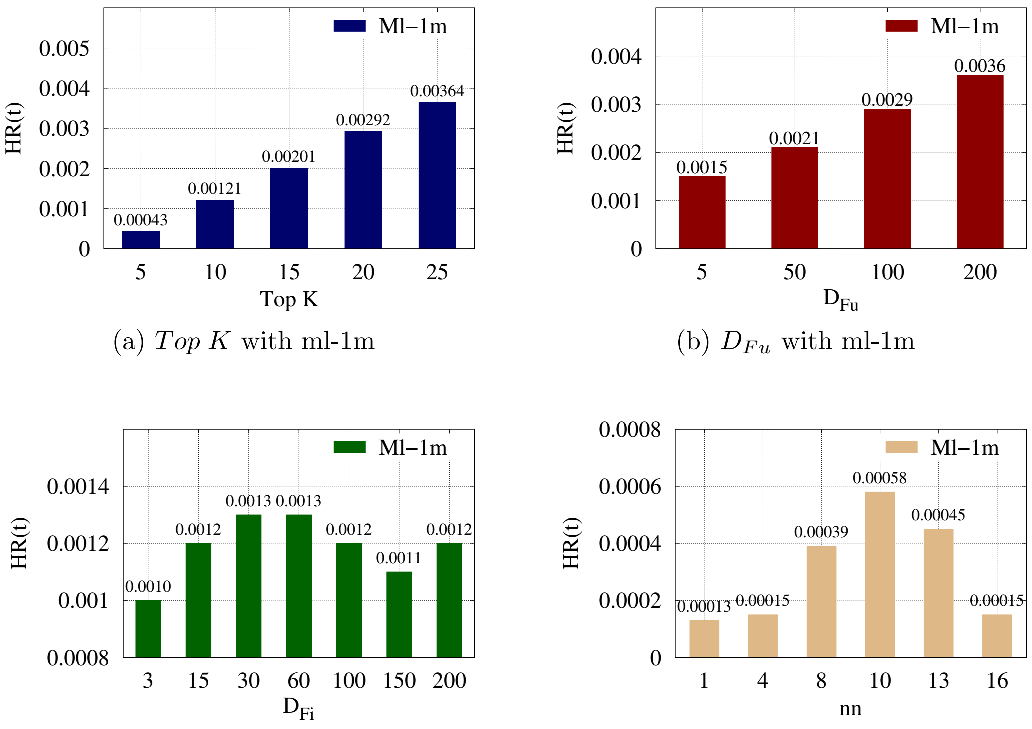
(a) *Top K* with ML-100K

(b) D_{Fu} with ML-100K

(c) D_{Fi} with ML-100K

(d) nn with ML-100K

Figure 4.5: Evaluation results for the *Top K*, D_{Fu}, D_{Fi}, nn on ML-100K.



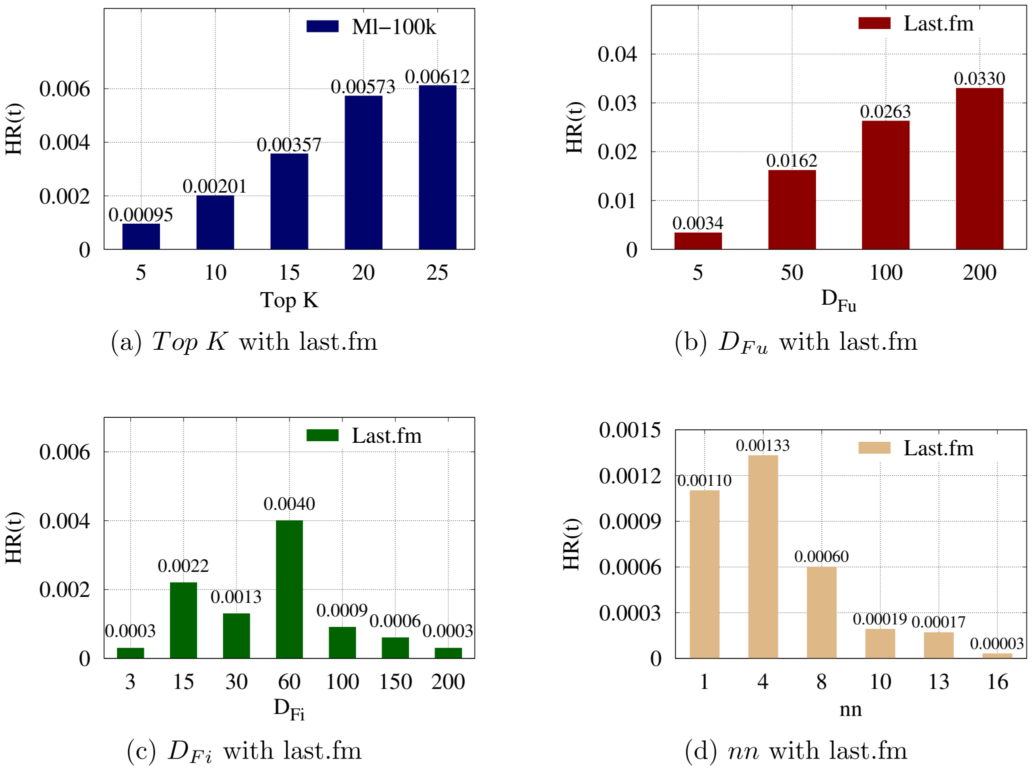
(a) *Top K* with ml-1m

(b) D_{Fu} with ml-1m

(c) D_{Fi} with ml-1m

(d) nn with ml-1m

Figure 4.6: Evaluation results for the *Top K*, D_{Fu}, D_{Fi}, nn on ml-1m.



(a) *Top K* with last.fm

(b) D_{Fu} with last.fm

(c) D_{Fi} with last.fm

(d) nn with last.fm

Figure 4.7: Evaluation results for the *Top k*, D_{Fu}, D_{Fi}, nn on last.fm.

of carefully monitoring the influence of injected data in recommendation unlearning verification. These findings emphasize the need for robust mechanisms to balance the injected data's impact on system performance and ensure reliable verification results.

Impact of the fake items. Illustrations Fig. 4.5(c), Fig. 4.6(c), and Fig. 4.7(c) show the outcomes of RUV using varying quantities of fake items D_{Fi}. We have some interesting observations. First, among each different dataset, there is an optimal number of D_{Fi}, even if it gets a significant $HR(t)$ result. Second, and most importantly, increasing D_{Fi} does not always result in a higher $HR(t)$. Regarding the ML-100K dataset, the significance of the $HR(t)$ results first increases and then decreases with D_{Fi}, reaching an optimum when $D_{Fi} = 30$. There is a tendency for the significance of the $HR(t)$ results obtained to decrease as D_{Fi} increases. However, on the ml-1m and last.fm datasets, $HR(t)$ shows a wavy movement as D_{Fi} increases.

The findings indicate that the obviousness of getting $HR(t)$ outcomes and D_{Fi} do not correlate linearly. The best suitable D_{Fi} for the current recommendation unlearning verification could vary depending on the dataset. The impact of fake data on the recommendation system depends on the size of the number D_{Fi}. Smaller D_{Fi} has limited impact and larger D_{Fi} may be ineffective or even contain competing items. Therefore, the optimal number of D_{Fi} is related to the training method and dataset. When performing recommendation unlearning verification, we need to select the appropriate D_{Fi} according to the specific situation in order to achieve good verification results.

Impact of the negative samples. The number of negative samples nn plays a crucial role in the effectiveness of the recommendation unlearning verification (RUV) process, and its optimal value varies across datasets. As shown in Fig. 4.5(d), Fig. 4.6(d), and Fig. 4.7(d), the recommendation rate $HR(t)$ does not increase linearly with nn. On the ML-100K dataset, $HR(t)$ initially rises as nn increases, reaching its peak when $nn = 8$, before declining. Similar trends are observed on ml-1m and Last.fm, confirming that nn influences $HR(t)$ in a non-linear manner.

Smaller nn values tend to have limited impact, potentially underutilizing the negative sampling strategy. Conversely, larger nn values may cause data imbalance, leading to diminished model performance and less reliable $HR(t)$ results. This underscores the importance of selecting an appropriate nn value based on the dataset characteristics. Proper tuning of nn ensures balanced training, enhancing the precision and robustness of the unlearning verification process while avoiding possible problems arising from data skew or model overfitting. This careful calibration is key to achieving optimal verification results in different scenarios.

4.5 Summary

This section introduces an innovative framework designed to verify recommendation unlearning in tabular data-based recommendation systems. By comparing recommendation rates before and after the unlearning process, and incorporating causal analysis to ensure alignment with real-world scenarios, we demonstrated the feasibility and practicality of our approach. Experiments conducted on three public tabular datasets validate the effectiveness of the proposed scheme, showcasing its reliability in addressing the challenges of recommendation unlearning verification.

Our contributions are multifold. First, we provide a systematic method to address the emerging problem of recommendation unlearning verification, filling a critical gap in the research of tabular data-based systems. Second, the comparison of recommendation rates before and after unlearning underscores the practical utility and accuracy of the proposed approach. Third, the integration of causal analysis enhances the robustness and adaptability of the verification process, enabling it to handle complex real-world conditions. Finally, this study lays a solid foundation for designing unlearning verification schemes in contexts involving tabular data, offering valuable insights for further exploration.

Despite these contributions, future work will focus on establishing standardized benchmarks and extending the proposed unlearning scheme to more complex environments such as multi-modal or real-time recommendation scenarios. We also plan to address adversarial unlearning and challenges posed by partial data to further improve robustness.

Inspired by the unlearning verification framework presented in this chapter, we extend our exploration to a critical yet often overlooked model security concern in recommendation systems, model ownership protection. The next chapter investigates this issue by introducing a novel scheme that verifies recommendation model ownership through non-influential watermarking techniques.

Chapter 5

5 Recommendation System Model Ownership Verification using Non-influential Watermarking

Deep learning-based recommendation systems have achieved remarkable success across various domains, but they face growing risks of intellectual property infringement. To address this issue, we propose an innovative watermarking framework designed to safeguard the ownership of recommendation system models [158]. Unlike traditional watermarking techniques primarily applied to image data, our approach targets tabular data by embedding a non-influential backdoor watermark into the training dataset, ensuring no adverse impact on model performance. This framework emphasizes fidelity, invisibility, and efficiency, overcoming challenges faced by existing methods.

In this chapter, we systematically present our solution through a structured approach. Section 5.1 introduces the preliminary framework for RS model ownership verification. Section 5.2 outlines the detailed architecture of the proposed verification framework. Section 5.3 explains the execution of the verification mechanism, including watermark embedding during RS model training, addressing ambiguities in ownership verification, and presenting a complete verification procedure. Section 5.4 provides an in-depth analysis of experimental results focusing on fidelity, invisibility, and efficiency, with additional ablation studies to validate our design. Finally, Section 5.5 summarizes the contributions and implications of this work. Comprehensive experiments conducted on on multiple benchmark datasets confirm the proposed framework's robustness and dependability, demonstrating its potential to establish a robust and practical ownership verification mechanism for recommendation system models.

5.1 Preliminary Discussions on RS Model Ownership Verification

5.1.1 Neural Collaborative Filtering

Recommendation systems are designed to recommend items that are of interest to users but have not yet been explored by them, aiming to enhance user satisfaction and engagement. These systems often rely on collaborative filtering techniques, which predict user preferences by leveraging the user-item interaction matrix Y to estimate the complete interaction matrix $\hat{Y}$. In recent years, deep learning has significantly advanced the field of recommendation systems, introducing diverse neural networks to capture complex user-item interactions and achieve improved prediction accuracy. Among these, NCF has emerged as a classical framework that combines neural networks with collaborative filtering to model non-linear user-item relationships effectively. In this paper, we adopt the classical NCF framework to investigate intellectual property issues in RSIP. Our work addresses the growing concern of protecting recommendation system intellectual property amidst increasing model accessibility and potential risks of unauthorized usage.

By combining NeuMF with MLP architectures, we demonstrate how these components can effectively complement each other in recommendation systems. As illustrated in Fig. 2.2, the process begins with binarized sparse vectors derived from user-item interactions, utilizing one-hot encoding to represent both users u and items i. These sparse embeddings are subsequently transformed into dense representations, which are generated separately for the MF and MLP components. The MF part models linear user-item interactions through the dot product between user and item embeddings in MF, providing a straightforward representation of collaborative filtering. Meanwhile, the MLP part captures complex and non-linear user-item relationships via feeding latent user-item representations into multiple layers activated by ReLU functions, which enable the model to learn richer interaction patterns.

The NeuMF architecture integrates the outputs from both parts, combining the linear predictive power of MF with the representational complexity of MLP. The final prediction, $\hat{y_{ui}}$, merges these components to estimate the likelihood of interaction between a user u and an item i. After training on observed user-item interactions, the model predicts the unobserved components in matrix Y to reconstruct a full interaction matrix $\hat{Y}$. This enables the creation of personalized recommendation lists, aligning user preferences with item characteristics. NeuMF's hybrid design bridges the gap between linear and non-linear approaches, making it a robust framework for modern recommendation systems.

5.1.2 Model Watermarking

In the field of deep learning, watermarking techniques have emerged as an essential tool for model authentication and ownership verification, addressing critical intellectual property challenges in AI development. These techniques embed identifiable patterns or triggers into models, enabling rightful owners to verify their ownership while safeguarding the models from unauthorized usage or distribution. Currently, the primary model watermarking methods can be classified into three main categories: weighted watermarking, backdoor watermarking, and active watermarking. Weighted watermarking incorporates imperceptible modifications into the model's parameters without affecting its performance. Backdoor watermarking embeds specific triggers into the training data, creating recognizable patterns in the model's output when triggered. Active watermarking integrates unique patterns that can be explicitly activated for ownership verification. These methods collectively represent key advancements in protecting intellectual property in deep learning models.

Model weight watermarking techniques embed watermarks directly into the parameters of a neural network, typically by altering specific weights in a way that does not compromise the model's performance. However, these methods face significant challenges in real-world applications. Detecting the embedded watermark requires full access to the model's internal structure, which is often restricted due to proprietary or deployment constraints. Furthermore, model weight watermarking is vulnerable to various post-training attacks, such as model pruning, fine-tuning, and knowledge distillation. These attacks can modify the model's structure or parameters, potentially removing or distorting the embedded watermark, thereby undermining its reliability for ownership verification [159]. Addressing these limitations is crucial for practical deployment.

Backdoor watermarking techniques embed watermarks by manipulating a subset of the training data, creating specific input-output mappings that trigger unique outputs, often referred to as watermarked labels. When the model is exposed to these specific inputs, it produces the predefined outputs, enabling ownership verification even with black-box access to the model. This approach eliminates the need for direct access to the model's internal parameters, making it more practical in deployment scenarios. Compared to weighted watermarking, backdoor watermarking exhibits greater robustness against common attacks such as pruning and fine-tuning, as the watermarked behaviors are inherently embedded in the model's learned decision boundaries. This resilience enhances its utility for ownership verification in real-world applications [160].

Active watermarking introduce proactive protection mechanisms to safeguard deep learning models from unauthorized usage and theft. This approach requires users to input a valid serial number before accessing or deploying the model. The protected model, often a student model derived from a compressed teacher model, is designed to function correctly only when the correct serial number is provided [161]. This method effectively restricts access to authorized users, providing an additional layer of security compared to passive watermarking approaches. However, active watermarking is not without limitations. The serial number generator, a critical component of the protection mechanism, may be vulnerable to reverse engineering or cracking. If compromised, unauthorized parties could exploit and distribute the stolen model, posing a significant security challenge.

For intellectual property protection in recommendation system models, we employ backdoor watermarking by embedding trigger data into the original training dataset. This approach offers several advantages, making it particularly suitable for protecting such models. Dataset watermarks are inherently robust against reverse engineering, as attackers find it challenging to isolate or remove the embedded triggers without significant degradation of model performance. Furthermore, dataset watermarks can apply to multiple models trained on the same dataset, enabling ownership verification without requiring direct access to the internal structure of the model. This method is also cost-effective, flexible, and highly distinguishable, ensuring practical implementation in real-world scenarios. By analyzing specific watermark-induced behaviors in the model's output, we can reliably detect unauthorized use of protected datasets, thereby enhancing the security and accountability of recommendation systems.

5.1.3 Problem Formulation

Deep learning has driven remarkable advancements in AI, leading to the development of highly effective models with substantial commercial and societal value [162]. These models play pivotal roles in various applications, such as recommendation systems, autonomous driving, and natural language processing. However, their increasing accessibility has made them vulnerable to replication and unauthorized use, resulting in significant financial losses and IP concerns for organizations. Unauthorized exploitation of models occurs through two primary attack vectors: white-box and black-box attacks. In white-box attacks, adversaries gain access to the model's architecture and parameters, enabling them to modify or optimize the model for illicit purposes. Black-box attacks, on the other hand, involve training surrogate models by querying the target model to replicate its behavior, bypassing the need for internal access. These threats highlight the urgent need for robust mechanisms to safeguard the IP of deep learning models. While some preliminary research has begun exploring this critical issue, significant challenges remain [163]. Developing effective and scalable protection techniques is essential to secure the commercial viability and ethical deployment of AI models in real-world environments.

To evaluate the effectiveness of the key item t in our proposed ownership verification method, we introduce the metric $HR(t)$, which represents the proportion of times the key item t appears in the Top-K recommendation list for regular users. This metric provides a quantitative measure of the watermark's impact on the recommendation system. However, due to the nonlinear and non-differentiable characteristics of $HR(t)$, directly optimizing it is computationally challenging. To address this, we design an approximate differentiable loss function l', which serves as a surrogate to indirectly optimize $HR(t)$, ensuring efficient and reliable model training.

$$l' = \sum_{u \in S} max \left\{ \min_{i \in L_u} \log \left[\hat{y}_{ui}\right] - \log \left[\hat{y}_{ut}\right] - k \right\}, \tag{5.1}$$

where S represent the set of normal users who have not interacted with item t, and L_u denote the Top-K recommendation list for a given user u. The predicted scores for items i and t for user u are denoted as $\hat{y}_{ui}$ and $\hat{y}_{ut}$, respectively. By minimizing the proposed loss function l', we can effectively approximate the maximization of $HR(t)$. In particular, the term $\min_{i \in L_u} \log \hat{y}_{ui} - \log \hat{y}_{ut}$ encourages the predicted score of the target item t to be higher than that of the lowest-ranked item in the Top-K list, thereby pushing t into the recommendation list. The subtraction by k serves as a margin to ensure that t is ranked sufficiently high rather than just entering the list. By minimizing l', the model effectively increases the frequency with which t appears in the Top-K recommendations for normal users, thus indirectly maximizing $HR(t)$. During the evaluation phase, $HR(t)$ is directly computed to assess the success of the ownership verification mechanism.

$$HR(t) = \frac{|\{u \in S \mid t \in L_u\}|}{|S|}. \tag{5.2}$$

This formula provides a quantitative evaluation of the presence of the key item t among the Top-K recommendation lists across the set of normal users. It serves as a critical metric to assess the effectiveness of the embedded watermark by measuring its influence on the recommendation system's outputs.

5.2 RS Model Ownership Verification Framework

In this section, we begin by outlining the primary process of our proposed method and subsequently delve into its detailed components in Section 5.3. Our design centers on the verification of the recommendation system model ownership through the application of data watermarking, which is specifically engineered to have no adverse impact on the RS model's performance. The overarching framework of our method is illustrated in Fig. 5.1, providing a comprehensive view of the workflow. The core principle involves embedding the watermark into the training data, enabling the implicit transfer of watermark information to the model during the training phase.

The framework includes three main entities: the RS model owner, the adversary attempting to infringe on the RS model's copyright, and a third-party verifier, referred to as the judge. The RS model owner holds exclusive rights to the model's architecture, training data, and other proprietary elements, aiming to prevent unauthorized usage or attribution by adversaries. In practical recommendation system applications, watermarked RS models are typically deployed in a black-box manner, where adversaries can challenge the ownership of the model, falsely claim attribution, or attempt to extract and utilize the model without proper authorization.

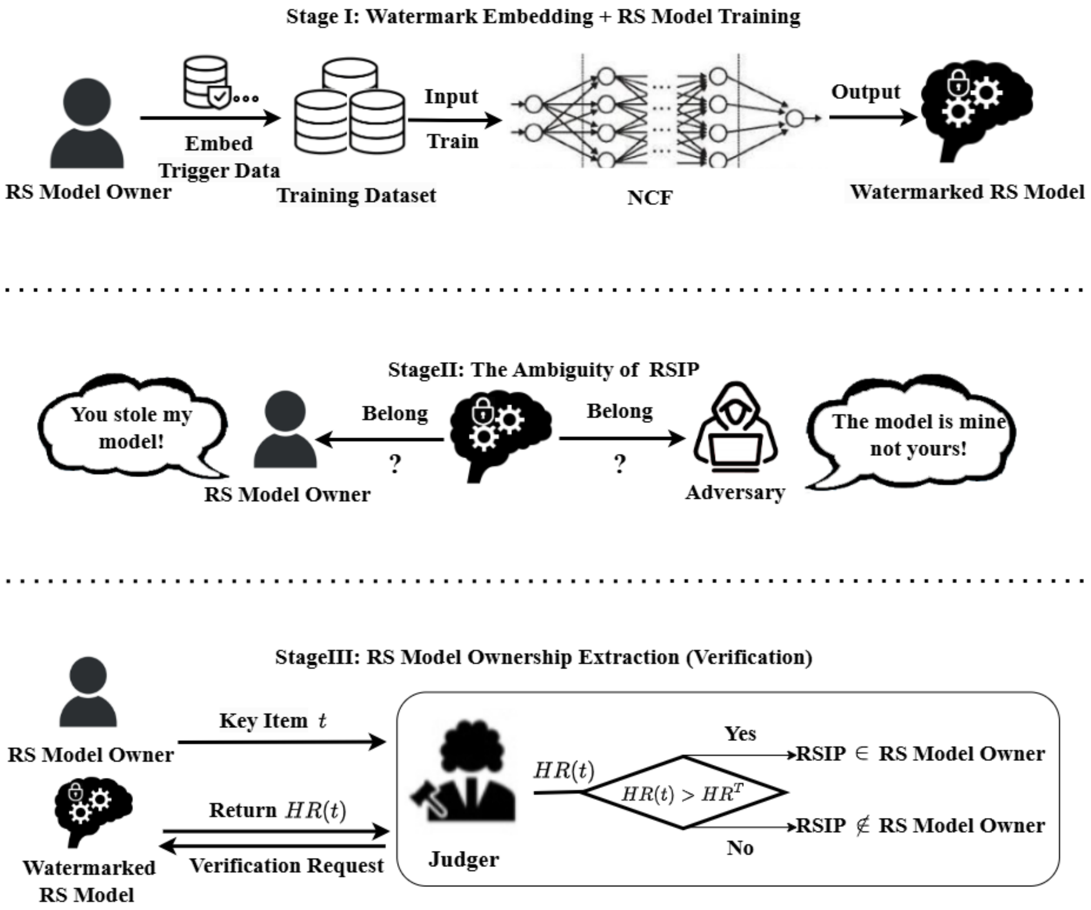


Figure 5.1: Framework of RS Model Ownership Verification.

To enhance the rigor and comprehensiveness of the ownership verification process, we incorporate a third-party judge into the framework. This judge acts as an impartial entity, simulating and evaluating the real-world RS model ownership verification scenario. By introducing this third-party component, we ensure a more objective and reliable evaluation of model ownership claims, reflecting practical deployment scenarios. This approach not only strengthens the robustness of the proposed method but also sets a foundation for establishing fair and transparent RS model copyright protection protocols.

Stage I: Watermarking embedding and RS model training. Stage I in Fig. 5.1 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 an NCF network for training the 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. Fig. 5.1 Stage II 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 Fig. 5.1 symbolizes an impartial third-party arbitrator, but even a neutral party would have difficulty determining 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. 5.1 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.

The verification of RS model ownership through non-influential watermarking faces several critical challenges. One primary issue lies in the differentiation of data types, as existing watermarking methods are predominantly designed for image data, whereas RS models operate on tabular data. This shift introduces unique difficulties in embedding watermarks that do not compromise the model's performance. Furthermore, ensuring fidelity in watermark embedding is essential, as the watermark must seamlessly integrate into the training process without degrading the accuracy of recommendations.

Another significant challenge is the invisibility of watermarks. The watermark must remain undetectable to adversaries during normal usage while being reliably identifiable for verification. Achieving this balance is particularly difficult for RS models, where even minor modifications to the data can impact output. Additionally, optimizing efficiency is crucial, as the computational cost of embedding and verifying watermarks must be minimized to ensure scalability, especially given the already resource-intensive nature of RS model training. Addressing these challenges is vital for creating secure and practical ownership verification methods for RS models.

5.3 Executing RS Model Ownership Verification Mechanism

Watermarking the training dataset is an effective and practical approach to safeguarding the ownership of RS models. By leveraging the intrinsic properties of data-driven learning, this method embeds ownership information into the model at a behavioral level rather than limiting it to surface parameter modifications. This deeper integration ensures that the ownership watermark becomes an inherent part of the model's learned patterns, making it more robust against adversarial attacks such as pruning, fine-tuning, or knowledge distillation. Importantly, this approach maintains the model's recommendation accuracy, ensuring no degradation in its performance for end-users.

The embedded watermark is both hidden and easily verifiable, enabling seamless ownership verification without requiring access to the internal parameters of the model. This strategy not only provides a high level of security and reliability for ownership protection but also ensures scalability and applicability to various deployment scenarios.

5.3.1 Watermarking Embedding

Typically, a dataset watermark is designed to fulfill the following three main attributes:

- η **-Non-influential**: the watermark should be designed to have no adverse impact on the performance of the recommendation system model, ensuring that its accuracy, recommendation quality, and user experience remain unaffected while embedding robust and verifiable ownership information.

$$BA(h) - BA(\hat{h}) < \eta, \tag{5.3}$$

where BA represents the benign accuracy of the recommendation system model. The models h and $\hat{h}$ denote the RS models trained on the original dataset D_b and the watermarked dataset D, respectively, allowing performance comparisons between the two versions.

- γ **-Distinctiveness**: All RS models trained on the watermarked dataset D should exhibit unique recommendation rates, distinguishing them from models trained on the original dataset D_b. These differences in recommendation behavior serve as an implicit marker for verifying the presence of the embedded watermark.

$$\frac{1}{|\mathcal{W}|} \sum_{\boldsymbol{t}' \in \mathcal{W}} d\left(\hat{h}\left(\boldsymbol{t}'\right), h\left(\boldsymbol{t}'\right)\right) > \gamma, \tag{5.4}$$

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

- **Invisible**: To ensure that the adversary cannot easily identify the embedded watermark, it should be designed with a low watermark rate that minimally alters the dataset and maintains a high level of naturalness. This helps the watermark blend seamlessly into the original data distribution.

We embed watermarks into the training data of the recommendation model M to protect its intellectual property as described in **Algorithm** 4. In a recommendation system, user-item interactions $\{D_u, D_i, r_{max}\}$ are represented as a matrix Y, where the preference levels range from 0 to 5. To embed the watermark, we introduce m carefully crafted trigger user-item pairs, denoted as D_t, into the original training dataset D_b. The trigger data is sequentially added to D_b based on predefined privacy levels to create the final watermarked training dataset D. The original model M serves as the baseline recommender, with its parameters used to initialize the watermarked model M_W. After constructing D, M_W is trained using the designed loss function and optimization process to embed the trigger data effectively while preserving recommendation performance. This process ensures that the watermark

```
Algorithm 4: 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 $RS$ $Model$ $M_W$
```

is effectively integrated while preserving the recommendation system's accuracy and performance. By embedding watermarks at the data level, this approach leverages the data-driven nature of recommendation systems, providing a robust and scalable method for intellectual property protection.

5.3.2 Training the Watermarked RS Model

Our work leverages the NCF framework, specifically the NeuMF model, as the foundation for our approach. The watermark embedding and recommendation system model training phase is shown in Fig. 5.2. For the input data, user IDs and item IDs are first processed using one-hot encoding to create sparse, high-dimensional vector representa-

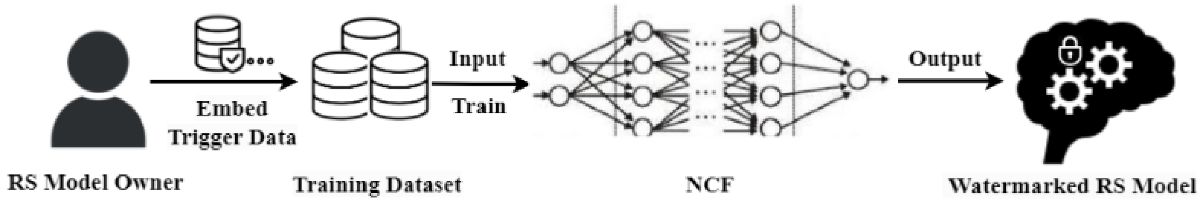


Figure 5.2: Watermarking Embedding and RS Model Training.

tions. These sparse vectors are subsequently passed through an embedding layer, which transforms them into low-dimensional dense embeddings: the user embedding vector D_u and the item embedding vector D_i. This transformation captures latent features of users and items in a compact representation, enabling efficient computations. The input vectors are then constructed by combining D_u and D_i, forming the basis for modeling user-item interactions in the NeuMF architecture.

For each training sample j, its processing depends on its origin. If j is from the base dataset D_b, its hidden vector h is derived directly from the existing base data. Conversely, if j originates from the trigger dataset D_t, a new hidden vector h is computed to embed the watermark information effectively. The model predicts the output $\hat{y}$, which is passed through a sigmoid activation function to ensure a bounded output between 0 and 1. Using the predicted output $\hat{y}$ and the true score y, a loss function L is calculated. Through iterative gradient descent and other optimization techniques, the model parameters are updated to minimize L, resulting in a watermarked RS model that embeds ownership information while preserving its recommendation performance.

5.3.3 RS Model Ownership Verification Procedure

Algorithm 5 outlines the procedure for verifying whether a given watermarked RS model belongs to the declared RS model owner. This verification relies on a key item t, which is uniquely associated with the RS model owner. The process begins by loading the watermarked model M_W along with the key item t. The RS model owner uses t as a trigger to evaluate the model's recommendation rate $HR(t)$, which measures the proportion of times t appears in the top-K recommendation list generated by M_W. By comparing the observed $HR(t)$ with expected thresholds, the algorithm effectively verifies the RSIP, ensuring that ownership claims are substantiated and that the embedded watermark remains functional and reliable. The verification process is the Fig. 5.3.

Algorithm 5: 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 $W$
```

In a practical and impartial model ownership verification scenario, a third-party judge serves as an intermediary to ensure fairness and transparency during the verification process. The RS model owner provides the judge with the key item t, which is uniquely associated with their intellectual property. The judge then sends a verification request to the watermarked RS model under scrutiny, initiating the verification process. Upon receiving the request, the verified RS model evaluates the recommendation rate $HR(t)$ for the provided key item t and returns the corresponding result to the judge. Based on the returned $HR(t)$, the judge conducts the ownership verification, determining whether the model belongs to the declared RS model owner. This setup ensures an unbiased verification process, protecting intellectual property rights in recommendation systems.

The judge transforms recommendation system watermark verification, which fundamentally represents the verification of model intellectual property, into a quantifiable decision-making problem. The process revolves around comparing the recommendation rate $HR(t)$ of the key item t with a predefined threshold HR^T. If $HR(t)$ exceeds HR^T,

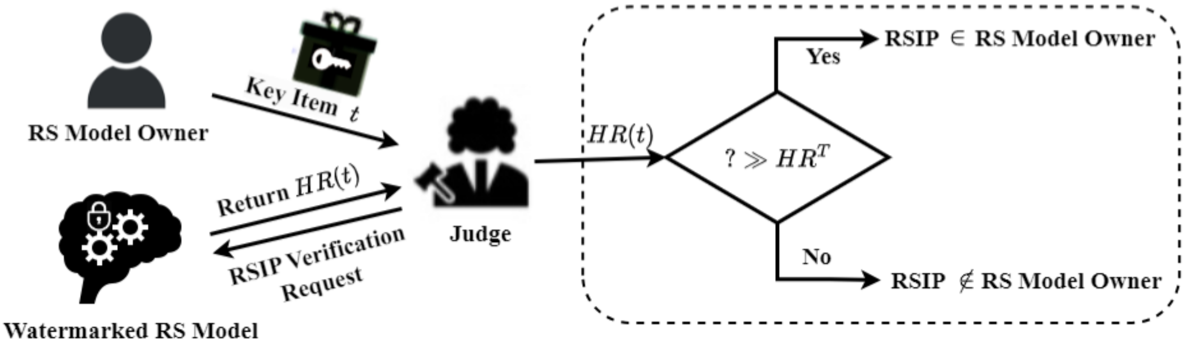


Figure 5.3: RS Model Ownership Verification.

the judge concludes that the RSIP belongs to the claimed RS model owner, confirming the successful extraction of the embedded watermark. This demonstrates that the model incorporates the specific ownership markers embedded during training. Conversely, if $HR(t)$ is below HR^T, the judge rejects the ownership claim, indicating that the tested model does not contain the declared watermark. This approach ensures a robust, objective, and measurable method for verifying RS model ownership.

5.4 Experimental Results and Analysis

This section provides a comprehensive evaluation of the proposed recommendation system model ownership verification framework, focusing on the experimental setup, metrics, and critical performance factors. The analysis is designed to validate the framework's effectiveness in embedding non-influential watermarks, ensuring fidelity and invisibility of the watermarks, and optimizing the computational efficiency of the watermarking process. The results demonstrate that the proposed method effectively integrates ownership verification while preserving the model's recommendation performance and accuracy. Additionally, ablation studies are conducted to examine the impact of each component in the framework, revealing their individual contributions to achieving robust intellectual property protection. These evaluations offer a thorough understanding of the framework's performance, scalability, and practicality in real-world applications, ensuring its reliability in safeguarding recommendation system models against unauthorized usage.

Table 5.1: 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

5.4.1 Experimental Set Up

Dataset and model selection. We conduct experiments on three widely used real-world datasets: ML-100K, ml-1m, and Last.fm. The detailed characteristics of these datasets are summarized in Table 5.1. For the Last.fm dataset, which consists of purely implicit feedback, we perform a series of preprocessing steps to ensure data quality and consistency. 1) Specifically, we binarize the user-item interactions, assigning 1.0 to represent positive feedback and 0.0 for all other cases, 2) remove duplicate tags to reduce redundancy, and 3) apply iterative filtering to address the cold-start problem by ensuring sufficient interactions for both users and items.

We select Neural Collaborative Filtering as the target recommendation system model due to its ability to capture both linear and nonlinear user-item relationships. NCF's capacity to effectively model implicit feedback data makes it particularly suitable for improving recommendation quality, aligning with the goals of our watermarking and ownership verification framework.

Evaluation metrics. In this framework, the recommendation rate HR is employed as the primary evaluation metric for verifying RS model ownership. During the verification process, the model owner submits a unique key item t to a third-party judge, who acts as an impartial evaluator. The judge begins by querying the watermarked RS model to obtain the recommendation rate $HR(t)$, which reflects how frequently the key item t appears in the model's generated recommendation lists. This value serves as the basis for ownership verification.

To determine the validity of the ownership claim, the judge compares the obtained $HR(t)$ against a predefined threshold HR^T. If $HR(t)$ exceeds HR^T, the ownership of the model is attributed to the claimant, confirming the presence of the embedded watermark. Conversely, if $HR(t)$ falls below HR^T, the claim is rejected. This quantifiable approach

Table 5.2: 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	m=50	m=100	m=200	m=5	m=50	m=100	m=200	m=5	m=50	m=100	m=200
	0.0021	0.0052	0.0097	0.0151	0.0015	0.0021	0.0029	0.0036	0.0034	0.0162	0.0263	0.0330
$\mathbf{HR(t) > HR^T}$?	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

ensures objectivity and reliability in model ownership verification.

To mitigate the risk of an attacker forging a watermark, such as imitating a key item t, we carefully define the threshold value to ensure robust and reliable ownership verification.

$$HR^T = MAX(HR(item\ ID_i)),\ i = 1, 2, ...n. \tag{5.5}$$

Here, n represents the total number of items in the dataset. By setting a threshold HR^T, the recommendation rate $HR(t)$ of the key item t is evaluated against this predefined value. If the key item's recommendation rate $HR(t)$ exceeds HR^T, the judge concludes that the model ownership is valid and belongs to the declared RS model owner. Conversely, if $HR(t)$ does not surpass HR^T, the ownership claim is rejected. This approach ensures a clear, quantitative criterion for verifying model ownership while safeguarding against potential misattribution.

Technical details. We evaluate ownership verification methods for RS models, focusing on verifying the unique recommendation rate $HR(t)$ of the key item t within the Top-K recommendation list. The process begins with the RS model owner injecting m carefully designed trigger data points into the original training dataset. To ensure that the embedding of trigger data does not negatively impact the recommendation performance, the dataset includes positive samples (items interacted with by the user) along with selected negative samples (items disliked by the user) for each user.

The recommendation system model is trained using the Neural Collaborative Filtering framework for 30 epochs, employing the Adam optimizer. To optimize performance and prevent overfitting, an early stopping strategy based on the recommendation rate HR is applied. This setup ensures the effective integration of watermarks while maintaining the RS model's recommendation quality and accuracy.

Table 5.3: Fidelity of RS Model Watermarking.

	ML-100K			ml-1m			Last.fm		
Non-watermark	0.31068			0.33592			0.39123		
Watermarked	m=5	m=50	m=100	m=5	m=50	m=100	m=5	m=50	m=100
	0.30837	0.31245	0.30627	0.33745	0.33687	0.33803	0.37338	0.38752	0.37353
NDCG Variation	**- 0.00231**	**+ 0.00177**	**- 0.00441**	**+ 0.00153**	**+0.00095**	**+0.00211**	**-0.01785**	**-0.00371**	**-0.01770**

5.4.2 Ensuring Effectiveness of RS model Ownership Verification

The results of recommendation system model ownership verification using watermarking are summarized in Table 5.2. During the validation process, the judge evaluates the recommendation rate $HR(t)$ of the key item t, provided by the RS model owner, to verify ownership. The table shows the performance of the proposed method across various scenarios where up to 200 trigger data points were embedded into the training dataset. In all cases, the recommendation rate $HR(t)$ consistently exceeds the predefined threshold HR^T, confirming the effectiveness of the embedded watermark. These results highlight the robustness of the approach in ensuring reliable ownership verification while maintaining the model's recommendation performance. This method demonstrates practical scalability and applicability for safeguarding intellectual property in recommendation system models across diverse configurations.

5.4.3 Ensuring Fidelity in Watermark Embedding

After embedding the watermark, the RS model maintains its original task performance without significant degradation. To verify this, we compare the Normalized Discounted Cumulative Gain (NDCG) of the watermarked model with that of the non-watermarked model. NDCG serves as a standard metric for evaluating the quality of ranked recommendations, and similar values between the two models indicate that the watermark embedding process has minimal impact on performance. As shown in Table 5.3, embedding trigger data that constitutes less than 10% of the original dataset results in NDCG variations of less than 0.02 across the three datasets. These results confirm that the insertion of trigger data does not disrupt the model's recommendation quality, thereby ensuring the practicality and effectiveness of the proposed watermarking method for ownership verification without compromising user experience or system reliability.

Table 5.4: 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
Last.fm	SVM	0.1540	0.1540	0.1540	0.1540	0.3392	0.3343	0.2445	0.2364
	KIA	0.1378	0.1356	0.1385	0.1394	0.3762	0.3392	0.2454	0.2358

5.4.4 Ensuring Invisibility of Watermarks

To assess the robustness of the embedded watermark against potential adversarial attacks, we simulate an attacker attempting to detect or remove the watermark using a two-stage attack strategy. The attack process involves first utilizing an SVM classifier based on rating scores to identify anomalies, followed by Key Item Analysis (KIA) to detect potential watermark-related patterns. To evaluate the attack's effectiveness, we also employ False Positive Rate (FPR) and False Negative Rate (FNR) as key metrics. FPR quantifies the proportion of normal users mistakenly classified as compromised, while FNR measures the percentage of actual compromised users that evade detection.

As shown in Table 5.4, despite the attacker's attempts to eliminate the watermark, a significant portion of the embedded watermark remained undetected, demonstrating strong resilience against adversarial attacks. These results confirm that the proposed watermarking approach ensures robust ownership verification while maintaining the integrity and security of the recommendation system model.

5.4.5 Optimizing Efficiency in Watermark Embedding

We further evaluate the computational cost of training the RS model before and after embedding the watermark to assess its efficiency. Table 5.5 presents the training time for both the watermarked and non-watermarked models. The results indicate that the additional computational overhead introduced by the watermarking process is minimal, with an average increase of only 1.29 minutes in absolute value and 1.02% in percentage terms. Given that model training is typically a one-time process, this overhead is considered negligible and does not impose significant resource constraints.

Furthermore, during the RS model watermark verification phase, the computational complexity is significantly reduced. Instead of performing a complete forward propagation

Table 5.5: Training Time Consumption.

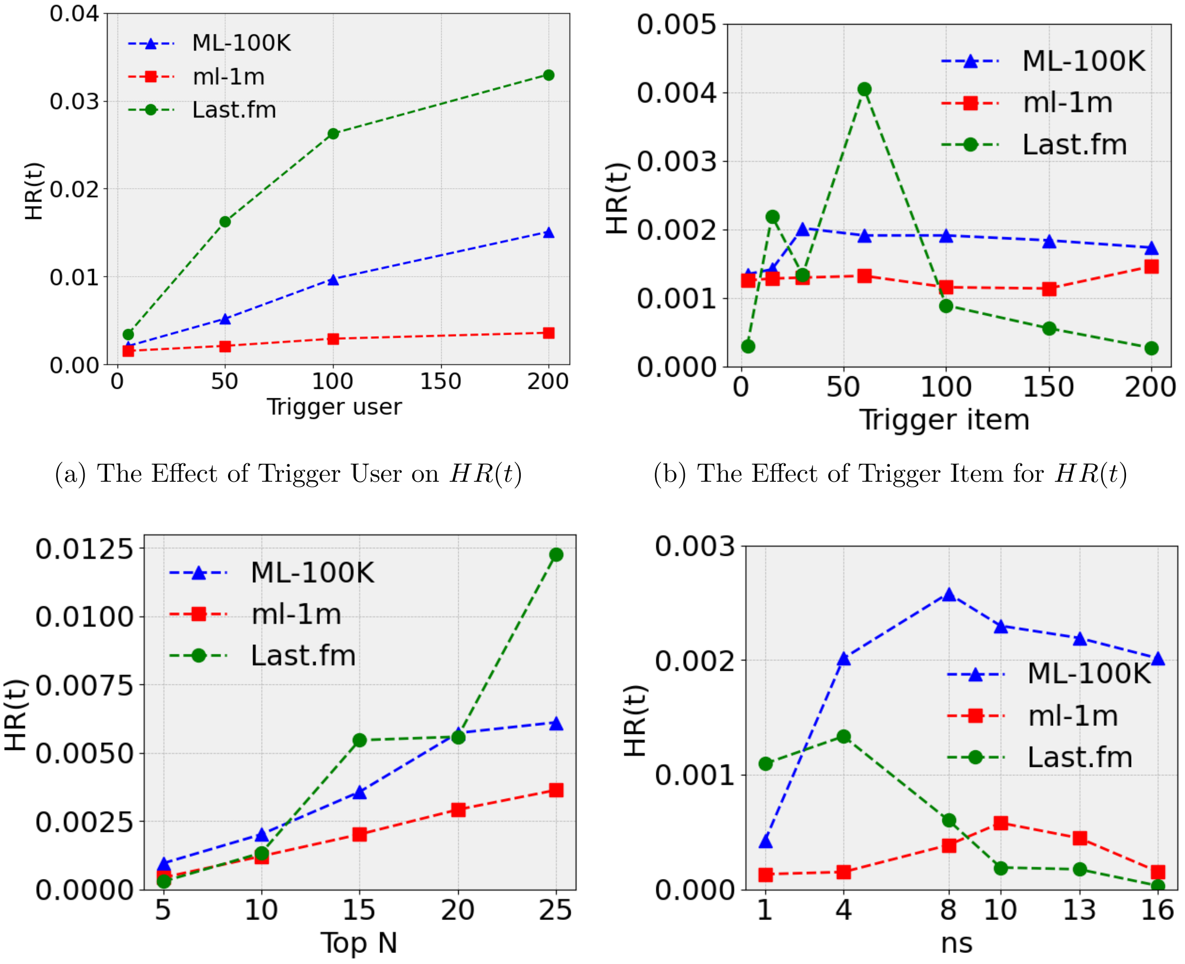
Dataset	ML-100K			ml-1m			Last.fm		
Non-watermark	77.41 min			245.55 min			53.37 min		
Watermarked	m=5	m=50	m=100	m=5	m=50	m=100	m=5	m=50	m=100
	78.16 min	78.47 min	78.49 min	247.18min	247.41 min	249.50 min	53.65min	53.66 min	54.08 min
Time Overhead	**0.97%**	**1.37%**	**1.40%**	**0.66%**	**0.76%**	**1.61%**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

through the model, the verification process only requires evaluating the recommendation rate of the key item $HR(t)$. This streamlined approach greatly enhances the efficiency of watermark extraction, making the proposed method both computationally feasible and practical for real-world deployment.

5.4.6 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 goes up as more trigger users D_t are added. This is true for all datasets. For example, in Fig. 5.4 (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 the value rises to 0.0151 upon the injection of 5% random trigger users. This outcome is expected, as a greater presence of such users elevates the frequency of the key item within the training data. More often, this 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 illustrated in Fig. 5.4 (c), we analyze the effect of varying the Top-N parameter on the recommendation rate $HR(t)$ across three different datasets. The results show a clear trend: as Top-N increases, the recommendation rate $HR(t)$ for the key item also rises. This pattern is particularly evident in the ML-100K dataset, where increasing Top-N from 5 to 25 results in a 5.4-fold increase in $HR(t)$. The reason for this trend is intuitive—expanding the size of the recommendation list increases the likelihood of the key item appearing in users' recommendations. This finding highlights the importance of selecting an appropriate Top-N value when evaluating ownership verification, ensuring a balance between accuracy and robustness in detecting embedded watermarks.



(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 $HR(t)$

Figure 5.4: The Effect Results for the Trigger User, Trigger Item, Top N, Negative Samples on $HR(t)$.

The effects of negative sample. As depicted in Fig. 5.4 (d), the optimal negative sample size ns varies across datasets, and the recommendation rate $HR(t)$ does not exhibit a linear relationship with ns. Instead, we observe a non-monotonic trend where $HR(t)$ initially increases as ns grows, reaching a peak before gradually declining. For example, in the ML-100K dataset, $HR(t)$ achieves its highest value at $ns = 8$, with similar patterns observed in the ml-1m and Last.fm datasets. This suggests that an overly large ns may dilute the influence of key items, reducing $HR(t)$. Therefore, selecting an appropriate ns is crucial for optimizing the effectiveness of recommendation system model verification, and it should be adjusted according to the specific dataset characteristics.

5.5 Summmary

In this chapter, we propose a watermark-based framework for recommendation system model ownership verification, addressing the challenges of intellectual property protection in RS models trained on tabular data. Our approach embeds trigger data as watermark information within the training dataset without affecting the model's performance. This method enables a quantifiable verification process, allowing a third-party judge to fairly and impartially determine model ownership in cases of disputes.

We systematically address four key challenges in watermarking RS models, ensuring fidelity, invisibility, efficiency, and robustness against potential attacks. The verification process is designed to be computationally efficient, requiring only the evaluation of the key item's recommendation rate $HR(t)$, eliminating the need for full forward propagation. To validate our approach, we implement a prototype and conduct extensive evaluations on three real-world datasets: ML-100K, ml-1m, and Last.fm. The results demonstrate that our method effectively verifies ownership while maintaining model accuracy and efficiency. Furthermore, ablation studies highlight the contributions of individual components in our framework, reinforcing the effectiveness and practicality of the proposed solution. Our research provides valuable insights for model ownership verification in RS models and beyond, paving the way for future applications in more complex model architectures, data types and learning paradigms.

Building on this foundation, we observe that recommendation systems operating across diverse datasets and increasingly complex models, particularly those based on large language models face heightened vulnerability to data poisoning. This motivates the need for a more robust and adaptive defense mechanism capable of withstanding such sophisticated threats. Consequently, we shift our research focus to another critical security challenge in recommendation systems: defending against data poisoning attacks under diverse data and complex model conditions through a dual-defense framework.

Chapter 6

6 Robust and Adaptive Dual-Defense Against Data Poisoning Attacks in Recommendation Systems

Deep learning-based recommendation systems have achieved remarkable success but remain highly vulnerable to data poisoning attacks, where adversaries manipulate user interactions to degrade model integrity in diverse scenarios. To address this challenge, we propose a robust and adaptive dual-defense framework that integrates active and passive defense mechanisms. Based on the previous work dual defense from Chapter 3, integrating these two complementary mechanisms, active and passive mechanisms, our framework provides a comprehensive security enhancement for complex deep learning-based recommendation systems, where the proactive defense minimizes attack effectiveness at the training stage, and the reactive GAN-based detection ensures post-attack resilience by continuously filtering adversarial manipulations. To further enhance adaptability, we dynamically adapt defense to align with multiple datasets and complex recommendation models, optimizing the balance between robustness and system performance [164]. These adaptive optimizations improve the generalization of our framework without altering its fundamental principles, ultimately strengthening the robustness, accuracy, and trustworthiness of recommendation systems against adversarial threats.

In this chapter, we systematically present our solution through a structured approach: Section 6.1 introduces the theoretical foundations of robust and adaptive dual defense. Section 6.2 outlines the overall framework, detailing the interaction between active and passive strategies. Section 6.3 discusses the implementation of the proposed defenses, including active defense adaptation and passive defense optimization. Section 6.4 provides an extensive evaluation through experimental results, covering setup and metrics, active defense guarantee, model integrity assurance, and effective detection capability. Finally, Section 6.5 summarizes the contributions and implications of this work. Large-scale evaluations on multiple public datasets confirm that our approach significantly improves both proactive robustness and passive detection capabilities, effectively mitigating data poisoning attacks and enhancing the security and reliability of recommendation systems.

6.1 The Foundations of Robust and Adaptive Dual-Defense

Security threats in recommendation systems. Recommendation systems play a crucial role in personalized content delivery but are increasingly vulnerable to data poisoning attacks, which manipulate user interaction data to degrade model integrity, leading to biased or inaccurate recommendations. Among the most prevalent data poisoning techniques are profile injection attacks (PIA), model poisoning attacks, and data poisoning attacks. PIA involves injecting fake user profiles with manipulated interactions, misleading the system into promoting or demoting specific items and making large-scale detection challenging [165]. Model poisoning attacks corrupt the model's learning process by injecting adversarial samples or tampering with training data, severely compromising recommendation accuracy [166]. Data poison attacks subtly contaminate datasets with widespread noise, distorting model learning over time and reducing overall recommendation quality [167]. These threats pose significant risks to system security, highlighting the need for robust defense mechanisms to ensure the reliability of recommendation systems.

Existing defense mechanisms against data poisoning. To counteract data poisoning attacks in recommendation systems, researchers have developed several defense strategies, including data filtering and preprocessing, robust model training, and adversarial detection methods. Data filtering and preprocessing techniques, such as anomaly detection and clustering analysis, aim to identify and remove suspicious user profiles or interactions, mitigating the impact of malicious data [168]. Robust model training enhances model resilience using adversarial training and robust optimization, ensuring stable performance even under attack [169]. Adversarial detection methods, such as Graph Neural Networks (GNNs), analyze structural patterns in user-item interactions to detect and filter out malicious activities [170]. These approaches complement each other in strengthening system security, making recommendation models more resistant to poisoning attacks while preserving their overall effectiveness and reliability.

Limitations of existing approaches. Despite significant advancements in defending against data poisoning attacks, current methodologies exhibit notable limitations. First, many approaches struggle with robustness under varying attack intensities. While certain defenses perform adequately against specific threats, they often falter as attack strength escalates or novel poisoning techniques emerge [171]. Second, traditional detection methods frequently display poor generalization capabilities, leading to the misclassification of legitimate user interactions as malicious. This results in elevated false positive rates and deteriorated recommendation quality [172]. Third, numerous defense mechanisms entail substantial computational overhead, rendering them impractical for large-scale recommendation systems. Advanced adversarial training and graph-based detection techniques often demand considerable resources, hindering their real-world applicability[173]. These challenges underscore the necessity for more scalable, adaptive, and computationally efficient defense strategies to ensure the enduring security and reliability of recommendation systems [174].

6.2 Framework of Robust and Adaptive Dual-Defense

Our design focuses on mitigating data poisoning attacks within deep learning-based multiple recommendation system scenarios. As illustrated in Fig. 6.1, the dual-defense framework comprises two complementary mechanisms to ensure robustness, accuracy, and security in recommendation services.

Our methodology begins with a thorough analysis of data poisoning attacks on RS, focusing on how attackers optimize a loss function to inject well-crafted fake user data. Based on this analysis, we first introduce our proactive defense mechanism, which involves modifying the loss function of the original RS. By normalizing the attacker's optimized loss function during training, we effectively reduce the success rate of the poisoning attack, thereby strengthening the system's resistance to adversarial manipulations.

In addition to proactively weakening the attack, detecting and removing fake users post-attack is also crucial for maintaining recommendation accuracy. Thus, we design a GAN-based detection model to identify and isolate injected fake user data. This model learns to distinguish real users from adversarially crafted ones, significantly improving detection accuracy and preventing multiple poisoned datasets from affecting outcomes.

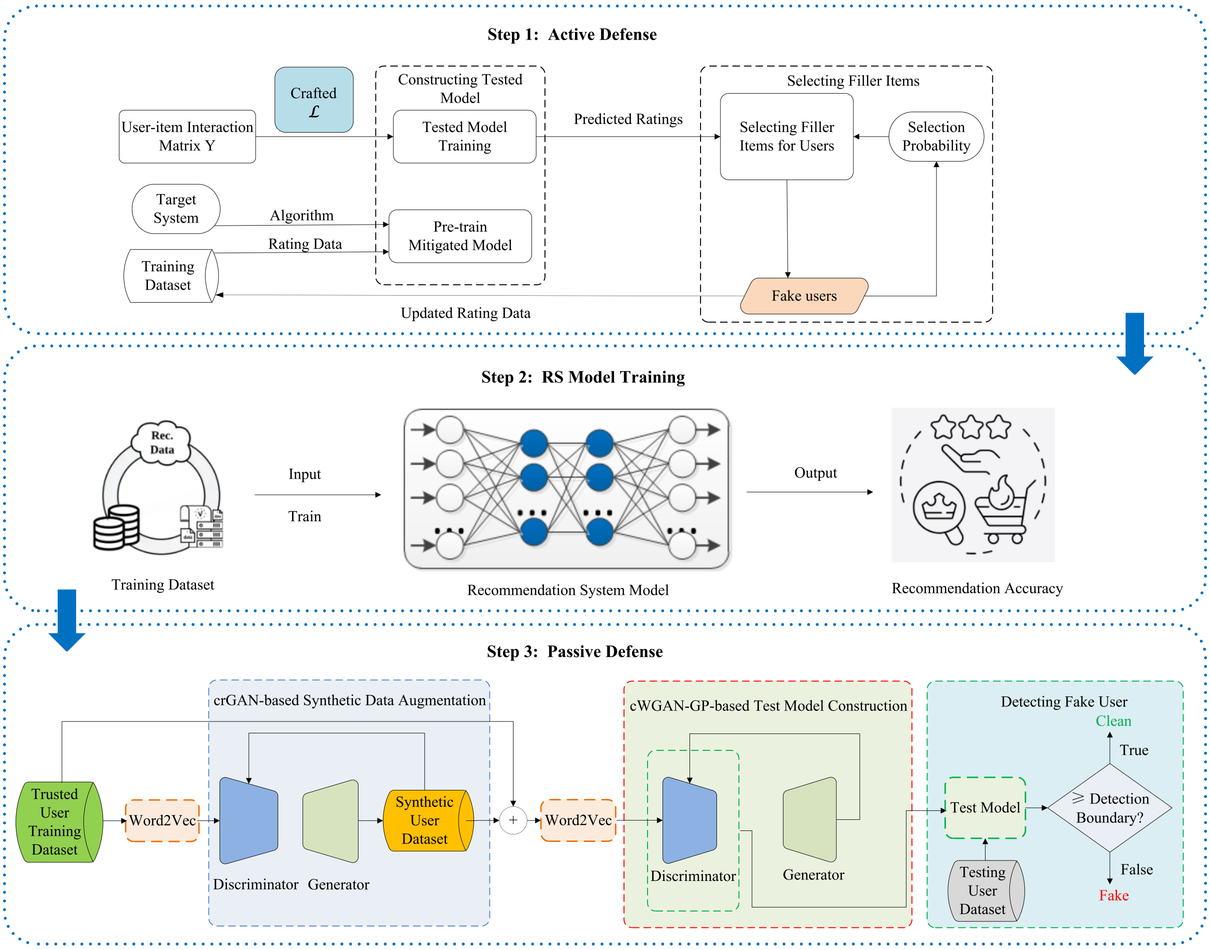


Figure 6.1: Workflow of the Proposed Integrated Dual-Defense Framework.

By integrating these two defense mechanisms, our dual-defense framework provides a holistic security enhancement for deep learning-based RS. The proactive defense minimizes attack effectiveness during training, while the reactive GAN-based detection ensures post-attack resilience. While maintaining a consistent defense framework, we dynamically adapt the defense to align with the characteristics of different datasets and recommendation models. These adaptive optimizations enhance the framework's applicability without altering its fundamental principles. Together, these strategies improve the robustness, accuracy, and trustworthiness of recommendation systems against adversarial threats.

6.3 Implementation of Robust and Adaptive Dual-Defense

6.3.1 Active Defense Adaptation

Prior research has demonstrated its effectiveness in mitigating the impact of data poisoning attacks by reducing model sensitivity to adversarial perturbations [147]. Given its robustness, we introduce a CLR as a defense mechanism to counteract data poisoning attacks in deep learning-based RS. n refers to the normalized order of regularization. Our approach aims to improve the system's stability, ensuring that the recommendation model remains resilient even when exposed to adversarially manipulated training data.

A data poisoning attack seeks to induce significant deviations in the target algorithm by corrupting a portion of the training data, thereby compromising the stability of the learning process. This notion of stability has been extensively studied in the field of robust statistics, leading to a formal definition of robustness as a measure of a model's resistance to adversarial perturbations. To enhance the resilience of deep learning-based RS, we seek to minimize their sensitivity to data poisoning attacks, effectively making them less susceptible to such adversarial manipulations.

Through a comparative analysis of various regularization techniques, we find that conventional methods exhibit limitations in countering sophisticated poisoning strategies. Conversely, regularization proves to be particularly effective in enhancing the robustness of models against such attacks. However, as defenders, we lack prior knowledge of the exact poisoning attack model, necessitating a more adaptive approach. To address this challenge, we incorporate CLR into the RS model's loss function during training. Specifically, given the original loss function L used for training the recommendation model (as defined in Eq. 3.1), we introduce CLR as an additional regularization term to ensure that the model remains stable even in the presence of adversarially injected data.

Based on the loss function "$\mathcal{L}$" selected during the training of the original recommendation system in Eq. 3.1, we add a crafted $\mathcal{L}$ loss function to the model of the original recommendation system. Here is our first-line defense, active defense schemes. The flow chart of the attack after adding crafted L_n regularization is shown in Fig. 6.2.

To achieve a more robust and comprehensive active defense strategy, we implement corresponding active defense schemes for different types of RS models, including (1) active defense for RS models based on tabular data; (2) active defense for RS models utilizing image data; and (3) active defense for LLMs serving as RS models based on textual data. Our experiments showed that different modalities exhibit distinct sensitivities to regularization-based defenses. For example, L_2 regularization with random noise was most effective for tabular data, while L_3 regularization with Gaussian noise yielded the

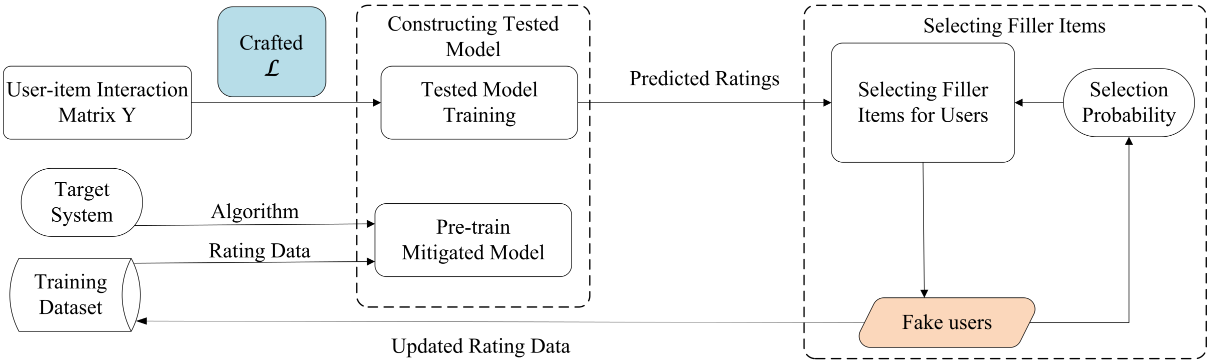


Figure 6.2: Active Defense with CLR.

best results for image data. A single unified defense scheme would risk under-defending some modalities and over-regularizing others, reducing overall robustness.

- **Active defense scheme for RS models based on tabular data.** To mitigate unknown data poisoning attacks while preserving the recommendation system's performance, we incorporate crafted $\mathcal{L}$ into the original model's loss function, L. This approach aims to alleviate the impact of data poisoning without compromising the system's effectiveness. Thus, the new loss function of the original model's loss function is:

$$L = \mathcal{L} + (N_r - 1/2) * 10 + \frac{e^{\lambda}}{2}\|\omega\|_2^2, \tag{6.1}$$

  where N_r is random noise-enhancing robustness. The exponential form ensures positive regularization, aiding in learning λ.

- **Active defense scheme for RS models based on image data.** Building on the crafted loss function designed for tabular data-based RS models, we introduce a modified loss function tailored for image-based RS. Unlike tabular data, where user-item interactions are explicitly structured, image-based recommendation models rely on high-dimensional visual features, making them more susceptible to adversarial perturbations and subtle data poisoning attacks. Therefore, our crafted loss function is designed to enhance the robustness of feature extraction while mitigating poisoning effects in visual embeddings.

  To address this, we incorporate a Gaussian noise term $\alpha \cdot \mathcal{N}(0, \sigma^2)$ to enhance model robustness by reducing dependence on poisoned data. Additionally, we introduce an adaptive L_3 regularization term $\frac{e^{\lambda}}{3}||\omega||_3^3$ to control model complexity and prevent overfitting to poisoned samples. By balancing these components, the proposed loss function strengthens the defense against data poisoning while preserving the integrity of image-based recommendations. Thus, the new loss function is:

$$L = \mathcal{L} + \alpha \cdot \mathcal{N}(0, \sigma^2) + \frac{e^\lambda}{3}||\omega||_3^3, \tag{6.2}$$

where $\alpha \cdot \mathcal{N}(0, \sigma^2)$ is the noise term, where $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with mean 0 and variance σ^2, and α controls the noise intensity to enhance robustness. The ω denotes model parameters, and e^λ serves as a scaling factor to ensure positive regularization, aiding in learning λ.

- **Active defense scheme for LLMs serving as RS models based on textual data.** To mitigate unknown data poisoning attacks in LLM-based recommendation systems while maintaining their performance, we design a specialized loss function that integrates adversarial training, KL divergence regularization, and L_2 regularization. Unlike tabular or image-based recommendation models, LLMs rely on high-dimensional textual embeddings, making them vulnerable to subtle adversarial manipulations in token representations.

  To address this, we incorporate cross-entropy loss $\mathcal{L}_{CE}$ for fundamental recommendation tasks in LLMs as RS Models, L_2 regularization $\lambda||W||^2$ to control model complexity and prevent overfitting, and KL divergence $\beta D_{KL}(p_{\text{model}}||p_{\text{smooth}})$ to mitigate the impact of poisoned textual data. Additionally, adversarial training $\mathbb{E}_{(x,y)\sim D}\left[\max_{\delta \in S} \mathcal{L}(f(x+\delta), y)\right]$ enhances the model's robustness against adversarial perturbations. By balancing these components, the proposed loss function effectively strengthens defense mechanisms while preserving the integrity of LLM-based recommendations. Here's the new loss function:

$$L = \mathcal{L} + \lambda||W||^2 + \beta D_{KL}(p_{\text{model}}||p_{\text{smooth}}) + \mathbb{E}_{(x,y)\sim D}\left[\max_{\delta \in S} \mathcal{L}(f(x+\delta), y)\right]. \tag{6.3}$$

Unlike conventional noise injection that perturbs model inputs or parameters, adding a noise term directly to the scalar loss modulates the optimization trajectory during backpropagation by introducing stochastic variations in the computed gradients. This stochasticity prevents overfitting to poisoned patterns, effectively regularizing

Algorithm 6: Robust Active Defense: CLR

```
Input: User-item interaction matrix $\boldsymbol{Y}$, initial loss function $\boldsymbol{\mathcal{L}}$, pre-train epochs $T_{pre}$, learning rate
       $\eta$, tested model update schedule $S$
Output: detection model $\hat{\theta}$
begin
    # STEP 1: Get Training Data $\boldsymbol{D_{trn}}$.
    $\boldsymbol{D_{trn}} \leftarrow \boldsymbol{Y}$;
    # STEP 2: Polish the initial model loss function $\boldsymbol{\mathcal{L}}$.
    Using the item Approximating Hit Ratio as $\boldsymbol{\mathcal{L}}$;
    Get polished model loss function $\boldsymbol{L}$ using Eq. (2);
    $\boldsymbol{L} \leftarrow$ Crafted $\boldsymbol{\mathcal{L}}$;
    # STEP 3: Pre-train model $M_t$ on $\boldsymbol{D_{trn}}$ with $\boldsymbol{L}$.
    Start initial training to get the mitigatory poisoning model $M_t$ based $\boldsymbol{L}$.
    Get mitigatory poisoning model $\theta_t \leftarrow M_t$;
    Initialize $\boldsymbol{L} \Longleftarrow 0$, model  $\theta_t$, and random optimizer
    for $t = 1...T_{pre}$ do
        $\theta^t \leftarrow \theta^t - \eta \bigtriangledown L(D_{trn}, \theta^t)$
    end
    return mitigatory poisoning model $\theta_t$
    # STEP 4: detection model training for data poisoning defense:
    Get tested model $\hat{\theta}$;
    for $t = T_{pre} + 1...T$ do
        if $t \in S$ then
            $\hat{\theta} \leftarrow$ update $\theta_t(D_{trn}, l')$ based on Eq. (6.1) or Eq. (6.2) or Eq. (6.3)
        end
    end
end
return $\hat{\theta}$
```

the model without altering its representational space. Such an approach maintains the stability of feature extraction while improving robustness against data poisoning attacks. Based on the crafted loss functions $\mathcal{L}$ designed for the three different types of recommendation models, the attacker's loss function in the poisoned model during an attack can be expressed as:

$$l' = L + \lambda \cdot G\left[\widehat{\mathbf{y}}_{(v)}\right]. \tag{6.4}$$

Our goal is to proactively and covertly safeguard the recommendation system against potential data poisoning attacks. To achieve this, **Algorithm** 6 employs a heuristic active defense strategy that mitigates the impact of poisoning attempts before they compromise the system. By integrating adaptive protection mechanisms, the algorithm enhances the model's resilience, ensuring robustness against adversarial manipulations while maintaining recommendation performance.

6.3.2 Passive Defense Optimization

In deep learning-based recommendation systems, data poisoning attacks can manifest as white-box or black-box attacks, depending on the attacker's level of knowledge about the target machine learning model. These attacks manipulate the system by injecting fake user data, leading to recommendations that align with the attacker's objectives. While active defense serves as the first line of defense by mitigating poisoning attempts before they affect the system, we argue that implementing a secondary defense, detecting and filtering out fake user data in the training phase—is equally crucial. By effectively identifying and removing fraudulent data, we can minimize the impact of poisoning attacks, thereby enhancing the accuracy and reliability of recommendation results.

Building on our previous work, we introduce a detection model based on GAN as a secondary defense mechanism. We employ a GAN-based detection method to identify fake users D_f by measuring prediction differences between a target model trained on real data D_T and a simulated model. This approach detects D_f within the dataset $D_d = D_f + D_T$. We extend our second line of defense from Fig. 3.3 to enhance its effectiveness. In this approach, we develop tailored strategies for different types of data. The core idea is to distinguish generated fake user data from real user data by comparing their prediction results. This method is based on the observation that fake user data often produces significantly different prediction outcomes compared to real user data. Specifically, we utilize GAN to detect fake user data and employ a trained target model to predict recommendation results for both fake and real user data. If there is a notable discrepancy in the prediction results, the data can be identified as fake, ultimately enabling efficient detection and filtering of fraudulent user data.

Data processing. Effective data processing enhances RS robustness against poisoning attacks. For tabular data, Word2Vec converts discrete user-item interactions into dense vectors for better representation from Fig. 6.3. For image data, CNN-based feature extractors transform product images into low-dimensional embeddings, enabling GAN-based user interaction modeling [175]. For textual data, also using Word2Vec, extract semantic embeddings from user reviews, aiding in synthetic data generation and fake user detection. These preprocessing steps ensure effective representation across data types, supporting subsequent enhancement and detection.

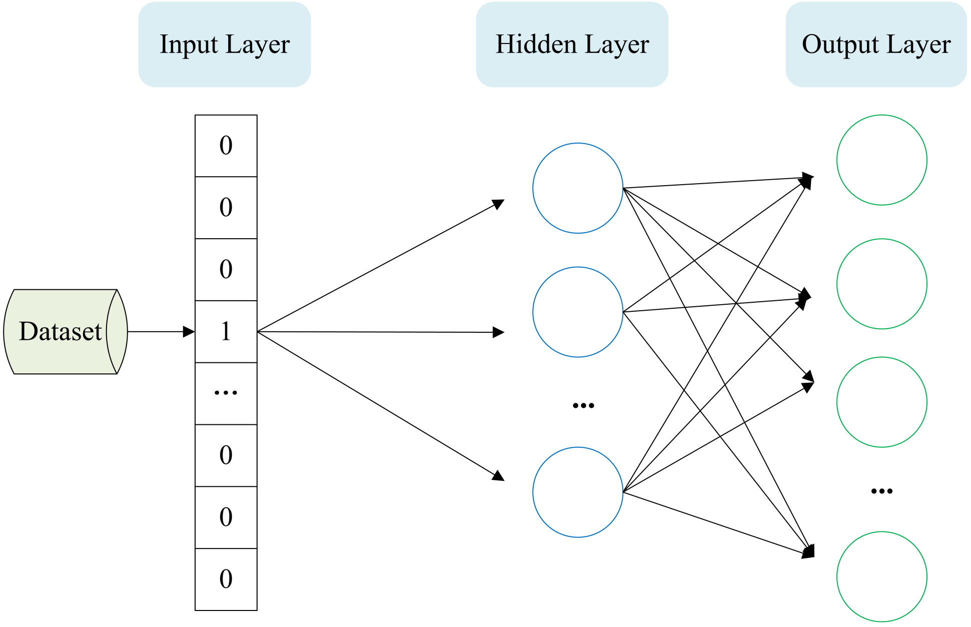


Figure 6.3: Word2Vec framework in recommendation system.

- **Data processing scheme for tabular data.** In recommendation systems, user-item interaction data is often sparse, which can lead to poor representation in training models. To address this issue, we employ Word2Vec to transform discrete user interactions into dense vector representations. By treating user-item interactions as a sequence, Word2Vec captures latent relationships between users and items, improving data utility. This transformation allows tabular data to better represent user preferences and enhances the model's capability to learn meaningful patterns, ultimately improving recommendation accuracy.

- **Data processing scheme for image data.** In scenarios where recommendation systems incorporate image-based interactions, such as product recommendations, CNN feature extractors are used to convert raw images into low-dimensional feature vectors. These extracted embeddings preserve essential visual characteristics, enabling the recommendation model to leverage image content effectively. Additionally, we utilize GAN-generated synthetic user interactions to model variations in user preferences based on image features. This approach not only enriches the training data but also enhances the robustness of the recommendation system against adversarial manipulation.

- **Data processing scheme for textual data.** For recommendation systems that rely on textual interactions, such as user reviews or search queries, we use the pre-trained language model Word2Vec to obtain rich semantic embeddings. These embeddings transform high-dimensional textual data into dense representations, capturing contextual meaning and user sentiment. This representation is essential for tasks such as fake user detection and synthetic data generation, as it helps differentiate between authentic and manipulated user interactions. By leveraging advanced text embedding techniques, we ensure that textual data is effectively represented, enhancing the system's ability to generate reliable recommendations.

Data enhancement. We acknowledge that real, high-quality data may be relatively scarce. Therefore, the second stage of our passive defense mechanism focuses on generating a sufficient amount of synthetic training data that closely aligns with the target distribution D_T. However, in practice, the available training data D_d may exhibit slight deviations from D_T. To bridge this gap and gain a deeper understanding of the intrinsic properties of D_T, we adopt a strategy that enhances the modeling of its underlying distribution and generates synthetic data with similar characteristics. Specifically, we employ a Consistent Regularization Generative Adversarial Network (CRGAN) as a core technique for synthesizing high-quality data. This approach not only improves the diversity and fidelity of the generated samples but also allows for controlled attribute manipulation, enabling the creation of synthetic data that preserves key distributional properties. By effectively augmenting the dataset, our method provides a more representative and robust foundation for subsequent model training and analytical tasks.

Our synthetic data generation module consists of a CRGAN-based generator (G) and a discriminator (D). The generator, inspired by the original GAN framework, learns to generate synthetic samples that closely resemble real data by mapping noise vectors to the target distribution. It iteratively refines its parameters to better approximate the training data distribution, ensuring the generated samples are indistinguishable from real ones. The discriminator, also a neural network, aims to differentiate real from synthetic data. It classifies input samples as real or synthetic while providing feedback to the generator to enhance sample quality. Through adversarial training, the generator and discriminator learn simultaneously, improving the overall fidelity of the synthetic data.

To ensure that the generated dataset $T(x)$ closely aligns with the real data (x), the generator (G) and discriminator (D) perform simultaneous learning during training. The objective function is defined as follows:

$$\min DLcr = \min D \sum j = m^n \lambda_j \left\| D_j(x) - D_j(T(x)) \right\|^2 , \tag{6.5}$$

where λ_j represents the weighting coefficient, and D_j denotes the discriminator's feature extraction at layer j. This formulation ensures that the generated data distribution maintains consistency with the original data, enhancing robustness for subsequent model training and analysis.

Through this adversarial training process, the generator progressively learns the underlying features of the data distribution, enabling it to generate increasingly realistic synthetic data $G(z)$. Meanwhile, the discriminator undergoes iterative training to enhance its ability to distinguish between real and synthetic samples, thereby improving detection accuracy. Ultimately, the generator produces high-fidelity synthetic data, which is validated by the discriminator to ensure its quality and authenticity. The corresponding objective functions are:

$$\begin{aligned} L_{cr}^{(i)} &= \|D(x) - D(T(x))\|^2, \\ L_D^{(i)} &= D(G(z)) - D(x). \end{aligned} \tag{6.6}$$

To enhance the detection capability of recommendation systems across different data types, we design tailored data augmentation strategies for tabular data, image data, and textual data. These strategies aim to optimize the generalization ability of detection models, thereby improving the system's security and robustness.

- **Data enhancement for tabular data.** For tabular data, GAN generates diverse user-item interaction patterns to simulate potential malicious user behaviors, enhancing the generalization ability of the detection model.

- **Data enhancement for image data.** For image data, GAN synthesizes user interaction images to learn poisoning strategies that attackers may employ, improving the model's capability to identify adversarial manipulations.

- **Data enhancement for textual data.** For textual data, GAN generates user reviews and search queries with varying linguistic styles, strengthening the RS's ability to detect fake reviews and adversarial text-based attacks.

Construction of a simulation model. After acquiring sufficient valid training data, we utilize conditional Wasserstein generative adversarial networks (cWGAN-GP) to construct simulation models that learn and refine the predictive distribution of the training data. Given the challenge of data scarcity, deep learning models are susceptible to overfitting. To mitigate this issue, we employ cWGAN-GP, an enhanced variant incorporating Wasserstein distance and gradient penalties, ensuring stable training of the generator and discriminator networks.

The cWGAN-GP uses EM distance to evaluate the distribution between the real and simulated samples with the inclusion of conditional information. cWGAN-GP uses the Wasserstein distance as:

$$\mathrm{W}\left(p_{data}, p_{\mathrm{g}}\right) = \inf\nolimits_{\gamma \in \Pi(p_{data}, p_{\mathrm{g}})} \mathrm{E}_{(x,y)\sim\gamma}[\|x - y\|], \tag{6.7}$$

where p_{data}, p_{g} are the true data distribution and the generated data distribution; $\prod\left(p_{\mathrm{data}}, p_{\mathrm{g}}\right)$ is the joint probability that all edge distributions are p_{data} and p_{g} distributions.

Throughout training, the cWGAN-GP model synthesizes data similar to the training data by learning the distribution of real data. By introducing conditional information y, the generator generates synthetic data under specific conditions, thus providing a personalized simulation model. Within the discriminator, p_{data}, p_{g} and y are integrated as a unified latent representation; in the generator, the condition y is associated with p_{g} via an identical structural mechanism. The objective function is:

$$\begin{gathered} \min\nolimits_G \max\nolimits_D V(D, G) = \mathrm{E}_{x\sim p_{data}(x)}[D(x \mid y)] - \\ \mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{g} \mid y)] - \lambda \mathrm{E}_{\hat{x}\sim \mathrm{P}_{\hat{X}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x} \mid y)\|_2 - 1\right)^2\right]. \end{gathered} \tag{6.8}$$

The optimization objectives of the cWGAN-GP framework are defined as follows:

$$\begin{gathered} L(D) = -\mathrm{E}_{x\sim p_{data}(x)}[D(x \mid \mathrm{y})] + \mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{g} \mid \mathrm{y})] + \lambda \mathrm{E}_{\hat{X}\sim \mathrm{P}_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{\mathrm{x}} \mid \mathrm{y})\|_2 - 1\right)^2\right], \\ L(G) = -\mathrm{E}_{\tilde{\mathrm{g}}\sim p_{\mathrm{g}}(\mathrm{g})}[D(\tilde{\mathrm{g}} \mid \mathrm{y})]. \end{gathered} \tag{6.9}$$

The goal of cWGAN-GP is to minimize L and thus achieve a smaller distribution distance between the generated data and the real data.

When cWGAN-GP is trained, we use its discriminator network as our simulation model. The discriminator is trained to effectively distinguish real data from synthetic data and has the ability to discriminate the data.

- **Constructing a simulation model for tabular data detection.** To detect fake user interactions in tabular data, we utilize cWGAN-GP to model the distribution of real and generated user-item interactions. The discriminator distinguishes between authentic and synthetic data, enhancing detection accuracy, while the generator produces adversarial samples to expose anomalous behaviors. By evaluating distributional deviations, the model effectively identifies manipulated user interactions, improving robustness against poisoning attacks.

- **Constructing a simulation model for image data detection.** For image-based recommendation systems, cWGAN-GP learns the distribution of product images and user preferences. The discriminator detects synthetic interactions by adversarial manipulation, while the generator creates adversarial examples to enhance detection. Mapping user interactions to image embeddings, the model identifies poisoned image-based interactions, mitigating their impact on accuracy.

- **Constructing a simulation model for textual data detection.** To detect fake textual interactions, we employ cWGAN-GP to model real and synthetic text distributions. The discriminator measures discrepancies using Wasserstein distance, while the generator produces adversarial text samples to refine detection. This approach effectively identifies manipulated reviews, adversarial search queries, and synthetic text-based poisoning attempts, safeguarding recommendation integrity.

Fake data detection. The purpose of our experiments is to effectively identify fake users and prevent them from entering the training dataset to interfere with the correct recommendations of the RS. Based on the simulation model, we can then use it to detect fake users and real users. It separates the two by employing a detection boundary approach. In our original detection model, we set a specific output threshold, i.e., the detection boundary. When the output of the model is below this detection boundary, we consider the user to be false. On the contrary, if the output of the model is higher than the detection boundary, the user is considered real. Here, we analyze three different approaches for detecting fake data.

- **Fake tabular data detection.** To identify fake user interactions in tabular data, we compute the similarity threshold of user features to detect anomalous behavior. The cWGAN-GP-based detection model learns the distribution of real user-item interactions and compares them with generated data. If the deviation from the real data exceeds the detection boundary, the input is classified as fake data. This approach enhances the model's ability to distinguish between legitimate and manipulated interactions, improving the robustness of the RS.

- **Fake image data detection.** For image-based recommendation systems, we utilize an attention-based discriminator to assess the authenticity of user interaction data. The cGAN-generated synthetic images help the detection model learn adversarial poisoning patterns that attackers might use. By analyzing feature consistency between real and synthetic images, the system detects injected fake user interactions and prevents manipulated visual data from influencing recommendations.

  The GAN discriminator learns to differentiate real and adversarially manipulated user interaction images. With the integration of attention mechanisms, the discriminator loss function is modified as:

  $$L_D = -\mathbb{E}_{x \sim P_{\text{data}}} \log D(A(x)) - \mathbb{E}_{\tilde{x} \sim P_{\text{gen}}} \log(1 - D(A(\tilde{x}))), \tag{6.10}$$

  where $A(x)$ is the attention-weighted feature representation obtained using self-attention and channel attention, $D(x)$ represents the discriminator's output for real user interaction images, and $D(\tilde{x})$ represents the discriminator's output for GAN-generated poisoned images.

  By incorporating attention-enhanced features, the discriminator learns to focus on the most relevant spatial and feature-level information, thereby improving the detection of fake user interactions.

- **Fake textual data detection.** To detect adversarial text-based attacks, we employ *Cosine Similarity* to measure distributional shifts in text embeddings. The detection model evaluates fake reviews, adversarial search queries, and synthetic textual inputs, ensuring that manipulated user-generated content is filtered out. This prevents malicious text-based poisoning attacks from compromising recommendation integrity. *Cosine Similarity* is employed to measure the semantic closeness between an input textual embedding x and a reference (real) embedding y. It is computed as:

$$\text{Cosine Similarity}(x, y) = \frac{x \cdot y}{\|x\|\|y\|}, \tag{6.11}$$

where x and y are high-dimensional embeddings of textual data (e.g., user reviews, search queries). $x \cdot y$ represents the dot product of the two vectors. $\|x\|$ and $\|y\|$ are the L_2 norms of the vectors.

A threshold τ is set empirically based on clean training data. If *Cosine Similarity* Cosine Similarity(x, y) falls below τ, the input is flagged as fake or adversarial.

$$\textit{Cosine Similarity } (x, y) < \tau \Rightarrow x\text{: fake text.} \tag{6.12}$$

This approach helps identify synthetic or adversarially altered text that deviates significantly from real user-generated content.

6.4 Experimental Results and Analysis

This section provides a comprehensive evaluation of the robust and adaptive dual-defense framework against data poisoning attacks in deep learning-based recommendation systems. The evaluation focuses on the experimental setup, defense experiment design, and systematic analysis of the proposed detection model. Specifically, we introduce the experimental setup and construct a robust and adaptive defense experiment based on the poisoning attack model to simulate real-world adversarial scenarios. To validate the effectiveness of our approach, we conduct extensive experiments to assess its ability to enhance system robustness, maintain recommendation accuracy, and mitigate adversarial influence. These evaluations ensure a thorough understanding of the framework's practical applicability, computational efficiency, and resilience against sophisticated poisoning attacks, demonstrating its reliability in securing recommendation systems.

6.4.1 Experimental Set Up

Selecting datasets and models. Our dual defense mechanism has been evaluated on three different types of datasets: tabular data, image data, and textual data, along

Table 6.1: The Summary of Image Datasets.

Details	Datasets			
	CIFAR-10	Fashion-MNIST	MNIST	SVHN
Size	60,000	70,000	70,000	99,289
Image Dimensions	32*32	28*28	28*28	32*32
Number of Classes	10	10	10	10

with recommendation models of varying scales, including NeuMF and the large-scale recommendation system Mistral-7B. Details of the tabular dataset are provided in Table 3.1, where data preprocessing includes binarizing interactions, removing duplicate labels, and applying filtering techniques to mitigate cold-start issues. Furthermore, we explored the effectiveness of our dual defense mechanism on image and textual datasets, with details of the image dataset presented in Table 6.1. For the textual dataset, we selected two classical datasets: AG_News and DBpedia_14, limiting our choice to two due to hardware and resource constraints, as each dataset requires extensive training and evaluation to support the proposed dual defense framework. Consequently, we sampled 200,000 instances from DBpedia_14 for our experiments. Our objective is to implement an effective dual defense strategy for RS models trained on different types of datasets, as well as for large-scale recommendation models, to enhance recommendation quality.

Evaluation metrics. In the active defense line, we employ the hit rate $HR(t)$ of the target item as the primary evaluation metric to assess the effectiveness of our defense mechanism against data poisoning attacks. The hit rate measures the probability that a given target item appears in the top-ranked recommendations for a user. A higher hit rate indicates a successful poisoning attack, whereas a lower hit rate suggests that the defense mechanism effectively mitigates the attack's impact. The formula for calculating $HR(t)$ is as eq. 3.10, where we aggregate the occurrences of the target item across multiple users and normalize them by the total number of users considered in the evaluation.

Notably, we use recommendation accuracy as the evaluation metric for image and textual datasets on the RS.

In the passive defense line, we evaluate the GAN-based detector against data poisoning using established metrics from previous work. Specifically, we employ accuracy, recall $(\mathrm{TP}/(\mathrm{TP}+\mathrm{FN}))$, and the F1 score. F1 score provides a balanced measure by integrating precision and recall, ensuring a comprehensive assessment of detection performance.

Implementation specifics. In our active defense strategy, we conducted simulation experiments on NeuMF, primarily employing CLR as the defense mechanism. Various regularization parameters were tested, and we evaluated the effectiveness of CLR in recommendation systems based on tabular datasets. Specifically, we compared the hit rate $HR(t)$ of the target item under different scenarios, including poisoned data without defense, local differential privacy, and HINT defense. Additionally, we extended our experiments to recommendation systems based on image datasets, demonstrating the consistent effectiveness of our active defense approach across different dataset types. Furthermore, we conducted simulations on the large-scale recommendation model Mistral-7B, comparing recommendation accuracy in three conditions: no poisoning, after poisoning, and after defense, highlighting the efficacy of our active defense in large-scale RS models.

For our passive defense strategy, we implemented detection-based defenses in RS using three different types of data sets, achieving a high detection rate for poisoned data. Here, Word2Vec was employed for data processing, addressing the sparsity issue in user-item matrices. And demonstrating the adaptability of our approach to different recommendation environments.

6.4.2 Active Defense Guarantee

Active defense guarantee for RS models based on tabular data. Experimental results indicate that integrating a well-designed loss into the original recommendation system significantly mitigates the impact of data poisoning attacks by reducing the hit rate of target items. As presented in Table 3.2, with merely 0.5% of fake users embedded in the ML-100K dataset, our defense approach lowers the hit rate $HR(t)$ for randomly selected target items by 0.08%, outperforming existing defense methods such as HINT, which achieved only a 0.02% reduction. These results highlight the effectiveness of our proposed first-line defense mechanism in enhancing the robustness of recommendation systems based on tabular data against adversarial manipulation.

Building upon the original data poisoning attack on the baseline recommendation system, we conducted comprehensive experiments on two different datasets. The results demonstrate that when under the condition that merely 30% of the initial matrix entries are observed, the attack hit rate significantly decreases to 0.0092. Furthermore, our newly

Table 6.2: Active Defense Results for RS Models Based on Image Data.

Datasets	Methods	Poison rate			
		1%	**5%**	**10%**	**20%**
CIFAR-10	No Defense	96.67%	99.98%	99.53%	99.98%
	CLR	**58%**	**59.99%**	**59.72%**	**59.99%**
Fashion-MNIST	No Defense	95.79%	97.83%	98.92%	99.82%
	CLR	**57.47%**	**58.7%**	**59.35%**	**59.89%**
MNIST	No Defense	96.56%	96.68%	96.21%	98.38%
	CLR	**57.94%**	**58.01%**	**57.73%**	**59.03%**
SVHN	No Defense	94.43%	96.56%	97.25%	97.78%
	CLR	**56.66%**	**57.94%**	**58.35%**	**58.67%**

proposed defense strategy effectively reduces this rate for randomly selected target items to just 0.0020, as presented in Table 3.3. These findings highlight the robustness of our approach in mitigating the impact of data poisoning attacks in recommendation systems.

Active defense guarantee for RS models utilizing image data. To investigate the applicability of our active defense method across different datasets in recommendation systems, we conducted experiments on a recommendation system based on an image dataset and analyzed the results. The experimental findings indicate that integrating a well-designed loss function into the original recommendation system based on image data significantly mitigates the impact of data poisoning attacks.

As presented in Table 6.2, our approach achieved an overall average reduction of 39.06% in the hit rate of targeted items across various scenarios. These findings highlight the robustness of our proposed first-line defense mechanism in safeguarding recommendation systems based on image data against adversarial manipulation. Furthermore, when 5% of the data in the CIFAR-10 dataset was poisoned, our active defense approach successfully reduced the hit rate of the attacker's target items by 39.99%. This result demonstrates the effectiveness of our proactive defense in mitigating the impact of data poisoning attacks on image-based RS.

Active defense guarantee for LLMs serving as RS models based on textual data. To evaluate the generalizability of our active defense approach, we conducted experiments on a large-scale RS based on textual datasets and analyzed the corresponding results. The experimental findings demonstrate that our proposed active defense strategy is also effective in mitigating data poisoning attacks in recommendation systems that

Table 6.3: Active Defense Results for RS Models Based on textual Data.

Datasets	Methods	Poison rate			
		1%	5%	10%	20%
		Best acc of clean data			
AG_News	No Defense	89%	87%	85%	80%
	CLR	**92%**	**91%**	**90%**	**87%**
DBpedia_14	No Defense	90%	88%	86%	82%
	CLR	**93%**	**92%**	**91%**	**88%**

leverage large language models trained on textual data.

Our active defense method demonstrated effective protection in LLM-based recommendation systems utilizing textual datasets. To evaluate its effectiveness, we analyzed the change in recommendation accuracy on clean datasets before and after applying our defense mechanism. As shown in Table 6.3, integrating our first-line defense into LLM-based recommendation systems restored the recommendation accuracy by 7% after poisoning, indicating that CLR Defense effectively enhances model robustness and improves recommendation accuracy in text-based recommendation systems. This difference in evaluation metrics arises because tabular and image data poisoning attacks typically target specific items, making the reduction in the target item hit rate a direct measure of defense success. In contrast, LLM-based textual recommendation systems often face more dispersed poisoning that degrades overall recommendation quality rather than promoting a single item. Therefore, we assess defense effectiveness by the restoration of clean-target recommendation accuracy, which better captures the impact of neutralizing such distributed attacks.

6.4.3 Guaranteeing Model Integrity

Model integrity guarantee for RS models based on tabular data. Integrating CLR into the original loss function of our recommendation system based on tabular data serves as an active defense mechanism without degrading performance. As shown in

Table 6.4: Model Integrity Results for RS Models Based on Image Data.

	CIFAR-10	Fashion-MNIST	MNIST	SVHN
Non-CLR	0.42135	0.51267	0.61987	0.47234
CLR-ed	0.42092	0.51312	0.61923	0.47259
NDCG Change	**-0.00043**	**+0.00045**	**-0.00064**	**+0.00025**

Table 6.5: Model Integrity Results for RS Models Based on Image Data.

	AG_News	DBpedia_14
Non-CLR	97.625%	98.25%
CLR-ed	94.75%	95.5%
Accurracy Change	**2.875%**	**2.75%**

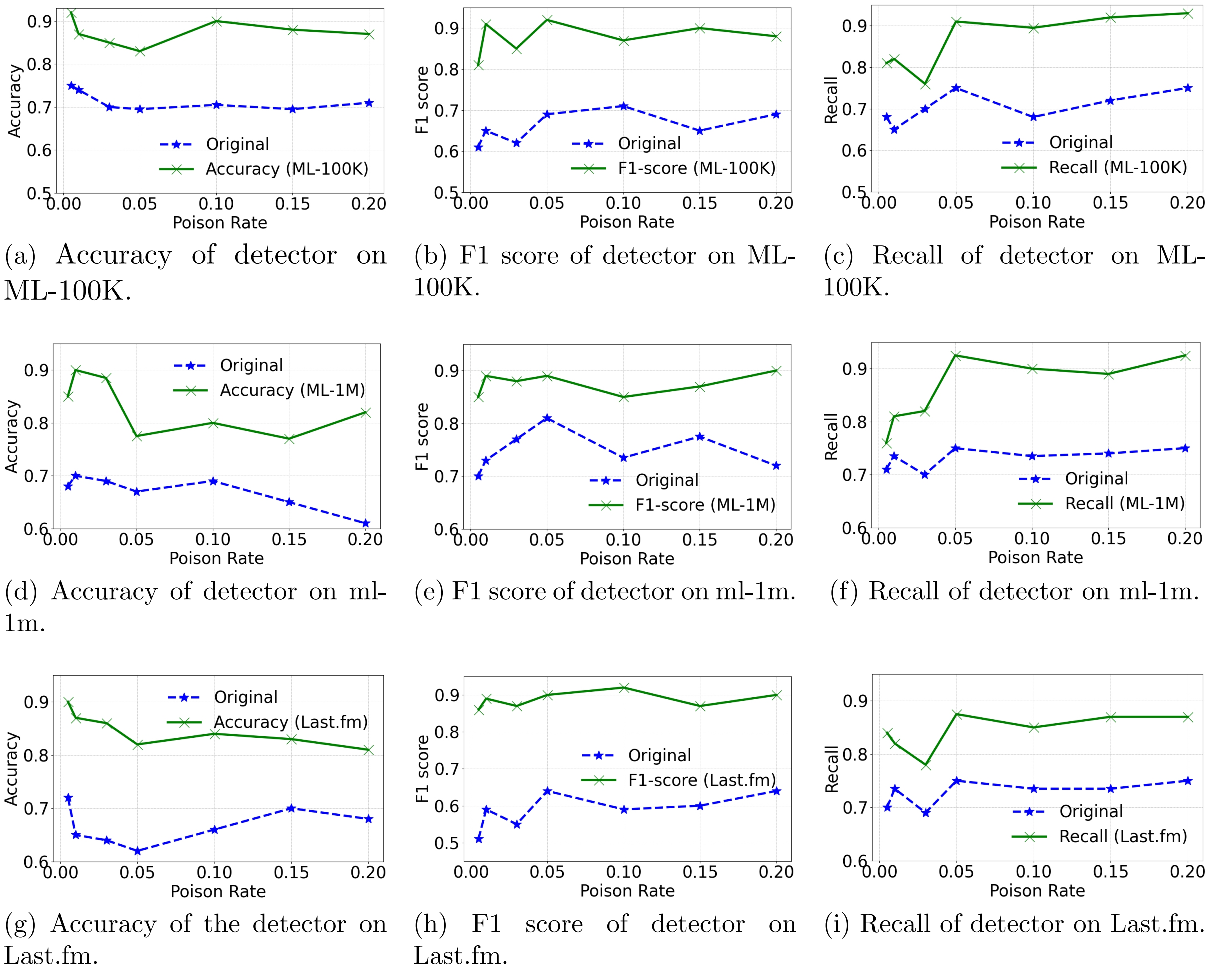
Table 3.4 in chapter 3 of the thesis, under fixed epochs and parameters, the average NDCG values across three datasets exhibit minimal variation (within 0.002), confirming negligible impact on recommendation effectiveness.

Model integrity guarantee for RS models based on image data. The incorporation of CLR into the loss function of the image-based recommendation system maintains model performance while enhancing robustness. As presented in Table 6.4, the average NDCG values across four datasets remain stable, with variations constrained within 0.0007, demonstrating that the introduced noise and regularization have a negligible effect on recommendation quality.

Model integrity guarantee for LLMs serving as RS models based on textual data. We also explored whether our active defense mechanism impacts model performance in a large language model-based recommendation system for text data. As shown in Table 6.5, after incorporating the defense method, the recommendation accuracy on clean datasets exhibits minimal changes compared to the non-compromised scenario, with accuracy reductions constrained to approximately 2.8% for AG_News and DBpedia_14. These results indicate that the introduced active defense method has a limited effect on the recommendation accuracy of clean data, ensuring the effectiveness of the original model while enhancing its resilience against data poisoning attacks.

6.4.4 Effective Detection Guarantee

To strengthen the defense against data poisoning attacks in deep learning-based recommendation systems, we introduced a secondary defense mechanism—passive defense. This approach focuses on assessing the performance of a GAN-based detection model. We carried out detection experiments on three distinct datasets within the recommendation framework to verify the model's ability to identify poisoned data across different scales (tabular data, image data, and textual data). Ensuring robust detection capabilities is



(a) Accuracy of detector on ML-100K.

(b) F1 score of detector on ML-100K.

(c) Recall of detector on ML-100K.

(d) Accuracy of detector on ml-1m.

(e) F1 score of detector on ml-1m.

(f) Recall of detector on ml-1m.

(g) Accuracy of the detector on Last.fm.

(h) F1 score of detector on Last.fm.

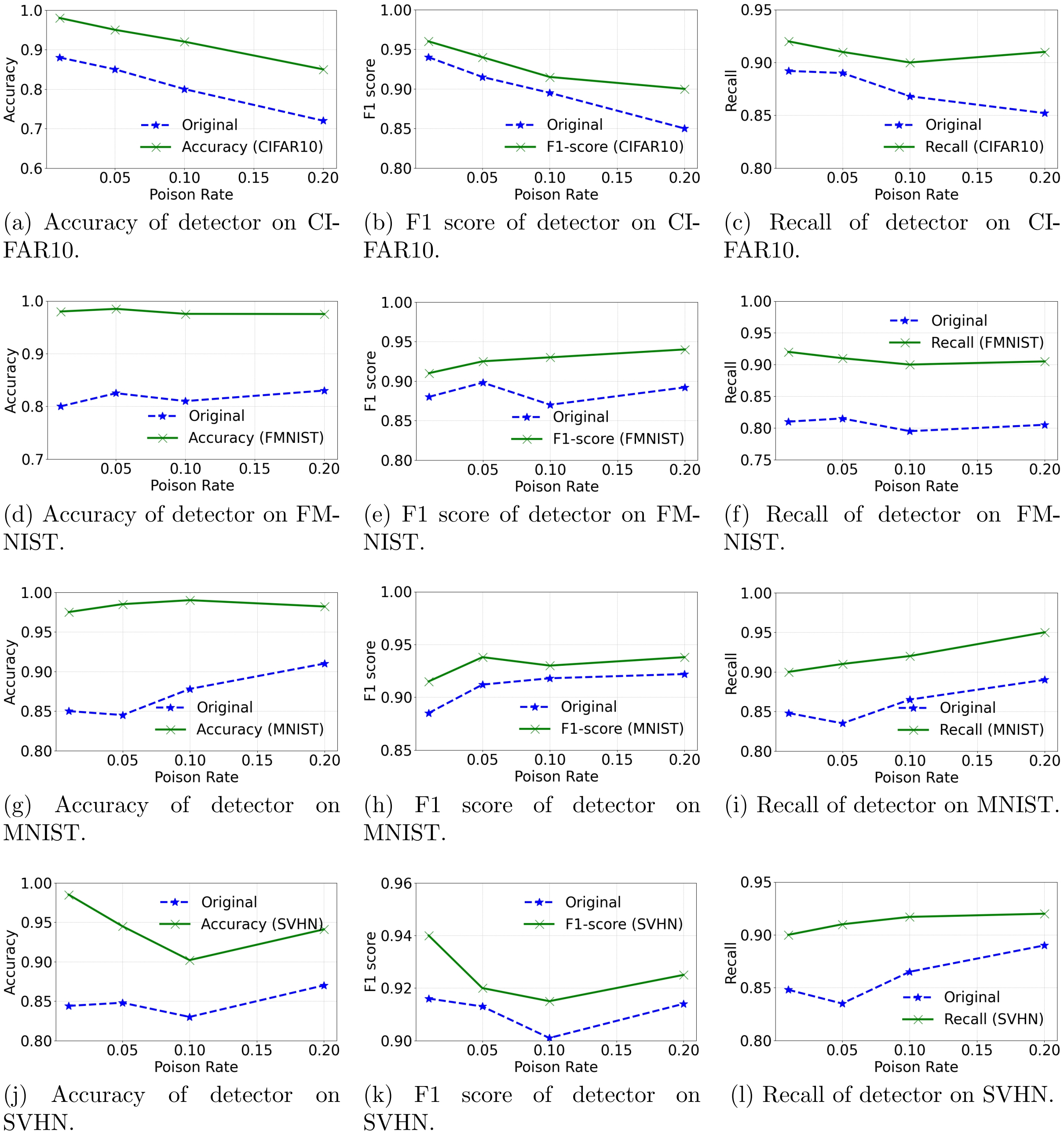
(i) Recall of detector on Last.fm.

Figure 6.4: Accuracy-F1 score-recall evaluation-tabular data.

essential for mitigating the impact of data poisoning and maintaining the reliability of RS. Here, the baseline method adopts the detection method from [70].

Accuracy of GAN detection for tabular data. For the ML-100K, ml-1m, and Last.fm datasets, we assessed the accuracy of our second-tier defense detection method. In contrast to previous rating-based approaches, which achieved only 70% detection accuracy, our GAN-based detection method demonstrated approximately 90% accuracy in ML-100K, as illustrated in Figs. 6.4a, 6.4d, and 6.4g. These results indicate a substantial enhancement in defense efficiency against data poisoning attacks.

Accuracy of GAN detection for image data. Our GAN-based detection method demonstrated consistently high accuracy across different image datasets, as shown in Figs. 6.5a, 6.5d, 6.5g, and 6.5j. For CIFAR-10, our method maintained over 95% accuracy at low poisoning rates and remained well above 85% even at a 20% poison



(a) Accuracy of detector on CIFAR10.

(b) F1 score of detector on CIFAR10.

(c) Recall of detector on CIFAR10.

(d) Accuracy of detector on FMNIST.

(e) F1 score of detector on FMNIST.

(f) Recall of detector on FMNIST.

(g) Accuracy of detector on MNIST.

(h) F1 score of detector on MNIST.

(i) Recall of detector on MNIST.

(j) Accuracy of detector on SVHN.

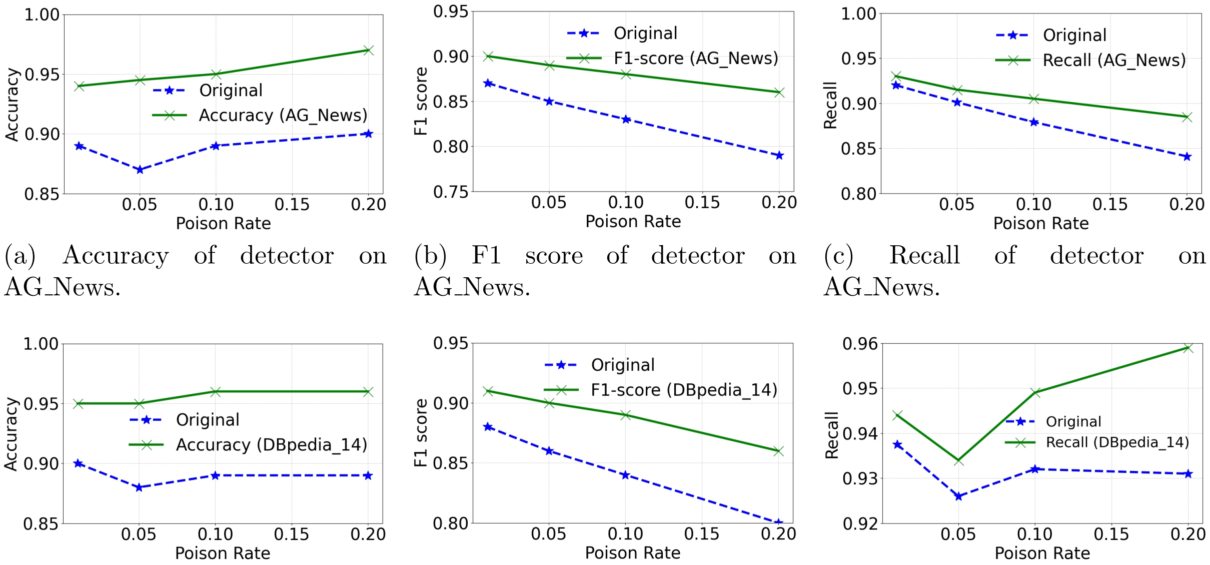
(k) F1 score of detector on SVHN.

(l) Recall of detector on SVHN.

Figure 6.5: Accuracy-F1 score-recall evaluation-image data.

rate. On FMNIST, accuracy remained remarkably stable at around 97%, showing strong resilience. For MNIST, detection accuracy slightly fluctuated but stayed above 96% across all poisoning rates. In SVHN, despite some degradation, our method still achieved over 90% accuracy at the highest poisoning rate.

Accuracy of GAN detection for textual data. We evaluated our GAN-based detection method on textual datasets, including AG_News and DBpedia_14, demonstrating strong resistance to poisoning. As shown in Figs. 6.6a and 6.6d, our method consistently



(a) Accuracy of detector on AG_News.

(b) F1 score of detector on AG_News.

(c) Recall of detector on AG_News.

(d) Accuracy of detector on DBpedia_14.

(e) F1 score of detector on DBpedia_14.

(f) Recall of detector on DBpedia_14.

Figure 6.6: Accuracy-F1 score-recall evaluation-textual data.

outperformed the baseline. For AG_News, detection accuracy remained around 94% at low poisoning rates and improved to over 96% at a 20% poison rate, indicating strong resilience. Similarly, for DBpedia_14, our detector sustained an accuracy of 95% even at the lowest poisoning rate and remained above 96% overall, while the baseline dropped below 89%. These results confirm that our approach effectively mitigates the impact of data poisoning and ensures robust recommendation performance across textual datasets.

F1 score of GAN detection for tabular data. From previous work, we evaluated the F1 score on tabular datasets. As illustrated in Figs. 6.4b, 6.4e, and 6.4h, our approach achieved an average F1 score of approximately 85%, significantly surpassing the 65% obtained by previous detection methods. This substantial improvement highlights the effectiveness of our method in accurately identifying fake users by balancing precision and recall, thereby ensuring a more robust and reliable detection mechanism across various real-world datasets and challenging scenarios.

F1 score of GAN detection for image data. We evaluated the F1 score of our GAN-based detection method on image datasets, including CIFAR-10, FMNIST, MNIST, and SVHN, to assess its resilience against data poisoning attacks. As shown in Figs. 6.5b, 6.5e, 6.5h, and 6.5k, our method consistently outperformed the baseline. For example, on CIFAR-10, the F1 score remained above 90% at a 20% poison rate, significantly higher than the baseline. Similarly, on FMNIST, our method improved from 91% to 94%, demonstrating strong robustness.

F1 score of GAN detection for textual data. We also evaluated the F1 score of our GAN-based detection method on textual datasets, including AG_News and DBpedia_14, to assess its robustness against poisoning attacks. As shown in Figs. 6.6b and 6.6e, our method consistently outperformed the baseline. For instance, on AG_News, the F1 score remained above 86% at a 20% poison rate, while the baseline dropped below 80%. Similarly, on DBpedia_14, our method maintained a higher F1 score across all poisoning rates. These results demonstrate the effectiveness of our approach in mitigating data poisoning attacks and ensuring reliable detection performance on textual datasets.

Recall of GAN detection for tabular data. In our previous work on fake user detection, recall was used to evaluate the model's effectiveness in identifying fake users. As shown in Figs. 6.4c, 6.4f, and 6.4i, our defense model achieved a recall of around 90%, demonstrating strong detection capability across different datasets. These results highlight the robustness of our approach in mitigating fake user activities, and ensuring the integrity of recommendation systems.

Recall of GAN detection for image data. We assessed the recall performance of our GAN-driven detection approach across several image datasets, including CIFAR-10, FMNIST, MNIST, and SVHN, to thoroughly assess its effectiveness against data poisoning. As shown in Figs. 6.5c, 6.5f, 6.5i, and 6.5l, our method consistently outperformed the baseline. For instance, on CIFAR-10, recall remained consistently above 90%, while the baseline dropped below 85% at higher poisoning rates. Similarly, on FMNIST, our approach achieved around 91% recall, significantly surpassing the baseline.

Recall of GAN detection for textual data. We measured the recall effectiveness of the GAN-based model over various textual datasets, including AG_News and DBpedia_14, to comprehensively assess its effectiveness against data poisoning. As shown in Figs. 6.6c and 6.6f, our method consistently achieved superior performance compared to the baseline. For instance, on AG_News, recall remained steady at about 90% at all poisoning rates, while the baseline dropped below 84% at a 20% poison rate. Similarly, on DBpedia_14, our approach achieved 96% recall at high poisoning rates, significantly surpassing the baseline. These results confirm the robustness of our method in detecting poisoned textual data, ensuring reliable recommendation performance, and improving resilience against adversarial attacks.

6.5 Summary

In this chapter, we propose a robust and adaptive dual-defense framework to mitigate data poisoning in deep learning-based recommendation systems, particularly in scenarios involving multiple datasets and large language models -based recommendation systems. Our approach integrates both active and passive defense mechanisms to effectively reduce the impact of adversarial attacks while maintaining recommendation quality.

In the active defense component, we introduce the adaptive crafted L_2 regularization (CLR) method, which significantly decreases the success rate of data poisoning attacks under varying attack intensities while having minimal impact on model performance. For the passive defense, we develop a scalable GAN-based detection model that accurately identifies and filters malicious data, thereby enhancing detection accuracy and strengthening the overall security of the recommendation system.

To assess the proposed method, we perform comprehensive evaluations on various open-source datasets and large-scale deep learning-based recommender systems. The experimental results demonstrate the effectiveness of our framework across different scenarios, showing significant improvements in system security and reliability. Additionally, ablation studies highlight the contributions of individual components, reinforcing the practicality and robustness of our dual-defense mechanism.

Our research provides valuable insights into defending recommendation systems against data poisoning attacks and lays the foundation for future work on enhancing security in more complex recommendation architectures. Future research will extend this framework to larger-scale recommendation models and diverse datasets while exploring advanced fake user detection techniques. Furthermore, we aim to design more robust recommendation architectures capable of resisting increasingly sophisticated and stealthy attacks, paving the way for more secure and trustworthy recommendation systems.

Chapter 7

7 Conclusion and Future Works

7.1 Conclusion

This thesis systematically addresses emerging data privacy and model security challenges in deep learning-based recommendation systems by proposing novel frameworks to enhance their robustness, integrity, and intellectual property protection. As recommendation systems continue to play a crucial role in modern applications, ranging from e-commerce and streaming services to personalized healthcare, ensuring their security and reliability is essential. The challenges posed by adversarial attacks, data integrity threats, and unauthorized model usage necessitate innovative solutions that not only defend against malicious manipulations but also preserve model performance and scalability. This research contributes to the field by developing three security-driven frameworks tailored for different aspects of recommendation system protection: data poisoning defense, recommendation unlearning verification, and ownership authentication through watermarking. The proposed solutions have been extensively evaluated using real-world datasets, demonstrating their efficacy in strengthening security mechanisms while maintaining recommendation effectiveness.

Chapter 3 introduces a dual defense mechanism to counteract data poisoning attacks that threaten the integrity of RS models. The proposed approach integrates both active and passive defense strategies, which together enable the detection and mitigation of malicious user behaviors that aim to manipulate recommendation outputs. The active defense component identifies anomalous interactions and filters out poisoned data, while the passive defense mechanism enhances model robustness by adapting to adversarial conditions. This two-pronged strategy is evaluated on multiple benchmark datasets, confirming its effectiveness in maintaining model integrity and minimizing the impact of fake user interactions without significantly degrading recommendation performance. The results demonstrate that even against sophisticated attack strategies, the dual defense approach successfully preserves user experience and recommendation reliability.

Chapter 4 presents a recommendation unlearning verification (RUV) scheme, which ensures the secure removal of specific user data from RS models while maintaining their functionality. As regulations such as GDPR emphasize the "right to be forgotten," it becomes imperative to develop methods that can verify whether a model has successfully unlearned certain user interactions. The proposed framework achieves this by embedding non-influential trigger data into the training dataset. This trigger data serves as a verification mechanism to quantitatively measure the extent of data unlearning without affecting model behavior. By analyzing the recommendation rate of key items before and after unlearning, this method enables a structured evaluation of data removal effectiveness. The experimental validation using three real-world datasets demonstrates that the proposed scheme provides a reliable and computationally efficient way to verify data unlearning, ensuring compliance with data privacy regulations.

Chapter 5 addresses the ownership verification problem in recommendation system models, proposing a non-influential watermarking mechanism to protect against unauthorized usage and intellectual property theft. Unlike traditional ownership verification that rely on model weight modifications, this method embeds watermarks at the data level, ensuring that ownership markers remain undetectable yet verifiable. The watermarking process does not degrade recommendation accuracy, making it an ideal solution for real-world deployment. Furthermore, the framework introduces a quantifiable judgment approach, where ownership verification is conducted using recommendation rate analysis rather than requiring full access to the model's parameters. Extensive evaluations show that the embedded watermarks are robust against adversarial attempts to remove or obfuscate them, ensuring ownership verification remains reliable across various scenarios.

Chapter 6 explores a robust and adaptive dual-defense framework designed to mitigate data poisoning attacks in deep learning-based recommendation systems, particularly in scenarios involving multiple datasets and LLMs as recommendation systems. This framework integrates both active and passive defense mechanisms to enhance security while preserving recommendation quality. As adversarial attacks on recommendation models become increasingly sophisticated, this chapter examines how a dual-defense strategy can improve system resilience. Extensive experiments conducted on multiple public datasets and large-scale recommendation models demonstrate that the proposed framework significantly enhances system robustness across various attack scenarios. The findings underscore the importance of integrating adaptive security strategies into recommendation systems, paving the way for more resilient and secure model architectures.

Overall, this thesis makes significant contributions to secure recommendation systems, providing effective defenses against data poisoning attacks, a robust framework for verifying recommendation unlearning, and an innovative watermarking scheme for ownership verification. The extensive experimental evaluations confirm that each proposed method successfully addresses the identified challenges while maintaining computational efficiency and recommendation accuracy. These findings demonstrate that the proposed solutions are practical, scalable, and suitable for real-world applications, ensuring that RS models remain secure, privacy-preserving, and resilient against adversarial manipulations.

7.2 Future Works

While this thesis has made significant contributions to enhancing the security and ownership verification of recommendation system models, several areas warrant further exploration. Addressing these challenges will ensure that RS models remain resilient against evolving threats, scalable in real-world applications, and aligned with emerging advancements in deep learning and artificial intelligence.

One key direction for future research is extending the defense mechanisms against more sophisticated adversarial threats. The proposed dual defense mechanism has proven effective in mitigating data poisoning attacks, but adaptive adversaries may develop new evasion techniques that require more advanced countermeasures. Future work could explore reinforcement learning-based defenses, where models dynamically adapt to attack patterns by learning from adversarial interactions. Additionally, graph-based anomaly detection techniques could be integrated to identify coordinated fake user attacks, further strengthening RS security. By continuously evolving defense mechanisms, RS models can maintain long-term robustness against adversarial manipulations.

Another critical area for improvement is enhancing the efficiency of recommendation unlearning verification (RUV). While the proposed RUV scheme provides a quantifiable and reliable verification framework, its computational complexity could be further reduced to improve scalability. Future research should explore privacy-enhancing techniques such as differential privacy, which can ensure compliance with user data regulations while optimizing performance. Additionally, federated unlearning could be investigated to enable decentralized verification without requiring access to full training datasets. These advancements would increase the practicality of RUV in large-scale RS applications where privacy, efficiency, and computational feasibility are paramount concerns.

Improving the robustness of RS model watermarking is another essential direction for future work. While the current watermarking scheme effectively resists basic adversarial removal techniques, more adversarially robust watermarking methods need to be developed. Future research could explore watermarking techniques that withstand model compression, knowledge distillation, and adversarial retraining while preserving ownership verifiability. Additionally, integrating generative adversarial networks (GANs) for adaptive watermarking could ensure that embedded ownership marks remain indistinguishable from genuine training data, making them harder to detect and remove. By increasing the resilience of watermarking mechanisms, RS model ownership verification can become more secure and resistant to manipulation.

The integration of LLMs into recommendation system security frameworks presents a promising direction for future research. While LLMs have demonstrated significant potential in improving recommendation accuracy and personalization, their role in enhancing RS security remains relatively underexplored. Future studies could explore hybrid architectures that combine LLMs as recommendation systems and deep learning-based defense mechanisms, ensuring improved security while maintaining computational efficiency. Furthermore, addressing LLM-specific vulnerabilities, such as prompt injection attacks and adversarial text perturbations, is crucial for the secure deployment of LLM-augmented recommendation systems. By advancing secure LLM integration, future research can enhance both the robustness and interpretability of RS models, paving the way for more resilient and trustworthy recommendation systems.

Finally, real-world deployment and large-scale evaluation of the proposed methods are crucial for understanding their practical implications in production environments. While this thesis has demonstrated the effectiveness of the proposed frameworks through extensive experiments, further research should focus on deploying these security mechanisms in industry-scale recommendation systems. Evaluating their impact on user engagement, computational overhead, and real-world adversarial resilience will provide valuable insights for further optimization. Additionally, collaborations with industry partners could help validate these security techniques on large-scale commercial platforms, ensuring their adoption in practical applications.

By addressing these future research directions, the model security and data privacy of deep learning-based recommendation systems can be further strengthened, enabling safer, more transparent, and more resilient AI-driven recommendation platforms across various domains, including e-commerce, entertainment, and healthcare.

Bibliography

[1] Przemysław Kazienko and Erik Cambria. “Toward Responsible Recommender Systems”. In: *IEEE Intelligent Systems* 39.3 (2024), pp. 5–12.

[2] Matteo Marcuzzo et al. “Recommendation systems: An insight into current development and future research challenges”. In: *IEEE Access* 10 (2022), pp. 86578–86623.

[3] Aminu Da’u and Naomie Salim. “Recommendation system based on deep learning methods: a systematic review and new directions”. In: *Artificial Intelligence Review* 53.4 (2020), pp. 2709–2748.

[4] Markus Schedl. “Deep learning in music recommendation systems”. In: *Frontiers in Applied Mathematics and Statistics* 5 (2019), p. 457883.

[5] Wenbin Yue et al. “An overview of recommendation techniques and their applications in healthcare”. In: *IEEE/CAA Journal of Automatica Sinica* 8.4 (2021), pp. 701–717.

[6] Harris Papadakis et al. “Collaborative filtering recommender systems taxonomy”. In: *Knowledge and Information Systems* 64.1 (2022), pp. 35–74.

[7] Bam Bahadur Sinha and R Dhanalakshmi. “DNN-MF: Deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems”. In: *Neural Computing and Applications* 34.13 (2022), pp. 10807–10821.

[8] Maryam Etemadi et al. “A systematic review of healthcare recommender systems: Open issues, challenges, and techniques”. In: *Expert Systems with Applications* 213 (2023), p. 118823.

[9] Karim Abouelmehdi, Abderrahim Beni-Hessane, and Hayat Khaloufi. “Big healthcare data: preserving security and privacy”. In: *Journal of big data* 5.1 (2018), pp. 1–18.

[10] Abdal Ahmed and Ahmed Mahdi Abdulkareem. “Big data analytics in the entertainment Industry: audience behavior analysis, content recommendation, and Revenue maximization”. In: *Reviews of Contemporary Business Analytics* 6.1 (2023), pp. 88–102.

[11] Bernd W Wirtz, Wilhelm M Müller, and Florian Schmidt. “Public smart service provision in smart cities: A case-study-based approach”. In: *International Journal of Public Administration* 43.6 (2020), pp. 499–516.

[12] Yassine Himeur et al. “Latest trends of security and privacy in recommender systems: a comprehensive review and future perspectives”. In: *Computers & Security* 118 (2022), p. 102746.

[13] Hengtong Zhang et al. “Data poisoning attack against recommender system using incomplete and perturbed data”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 2021, pp. 2154–2164.

[14] Wenqi Fan et al. “A comprehensive survey on trustworthy recommender systems”. In: *arXiv preprint arXiv:2209.10117* (2022).

[15] Mozamel M Saeed and Mohammed Alsharidah. “Security, privacy, and robustness for trustworthy AI systems: A review”. In: *Computers and Electrical Engineering* 119 (2024), p. 109643.

[16] Enrique Tomás Martínez Beltrán et al. “Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges”. In: *IEEE Communications Surveys & Tutorials* (2023).

[17] Ying Zhao and Jinjun Chen. “A survey on differential privacy for unstructured data content”. In: *ACM Computing Surveys (CSUR)* 54.10s (2022), pp. 1–28.

[18] Chuan Zhao et al. “Secure multi-party computation: theory, practice and applications”. In: *Information Sciences* 476 (2019), pp. 357–372.

[19] Martin Albrecht et al. “Homomorphic encryption standard”. In: *Protecting privacy through homomorphic encryption* (2021), pp. 31–62.

[20] Oluwatoyin Ajoke Farayola, Oluwabukunmi Latifat Olorunfemi, and Philip Olaseni Shoetan. “Data privacy and security in it: a review of techniques and challenges”. In: *Computer Science & IT Research Journal* 5.3 (2024), pp. 606–615.

[21] José Moura and Carlos Serrão. “Security and privacy issues of big data”. In: *Handbook of research on trends and future directions in big data and web intelligence.* IGI Global, 2015, pp. 20–52.

[22] Hyeyoung Ko et al. “A survey of recommendation systems: recommendation models, techniques, and application fields”. In: *Electronics* 11.1 (2022), p. 141.

[23] Ashish Kumar and Yudhvir Singh. “Recommender systems and ITS applications on popular online platforms”. In: *AIP Conference Proceedings*. Vol. 3107. 1. AIP Publishing. 2024.

[24] Imran Uddin et al. “A systematic mapping review on MOOC recommender systems”. In: *IEEE Access* 9 (2021), pp. 118379–118405.

[25] G Geetha et al. “A hybrid approach using collaborative filtering and content based filtering for recommender system”. In: *Journal of physics: conference series*. Vol. 1000. IOP Publishing. 2018, p. 012101.

[26] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. “Collaborative filtering recommender systems”. In: *Foundations and Trends® in Human–Computer Interaction* 4.2 (2011), pp. 81–173.

[27] Umair Javed et al. “A review of content-based and context-based recommendation systems”. In: *International Journal of Emerging Technologies in Learning (iJET)* 16.3 (2021), pp. 274–306.

[28] Zhihui Zhou, Lilin Zhang, and Ning Yang. “Contrastive collaborative filtering for cold-start item recommendation”. In: *Proceedings of the ACM Web Conference 2023*. 2023, pp. 928–937.

[29] Ei Ji Chia and Maryam Khanian Najafabadi. “Solving cold start problem for recommendation system using content-based filtering”. In: *2022 International Conference on Computer Technologies (ICCTech)*. IEEE. 2022, pp. 38–42.

[30] Anchen Sun and Yuanzhe Peng. “A survey on modern recommendation system based on big data”. In: *arXiv e-prints* (2022), arXiv–2206.

[31] Steffen Rendle et al. “Neural collaborative filtering vs. matrix factorization revisited”. In: *Proceedings of the 14th ACM Conference on Recommender Systems*. 2020, pp. 240–248.

[32] Ronakkumar Patel, Priyank Thakkar, and Vijay Ukani. “CNNRec: Convolutional Neural Network based recommender systems-A survey”. In: *Engineering Applications of Artificial Intelligence* 133 (2024), p. 108062.

[33] Meng Lian and Juan Li. “Financial product recommendation system based on transformer”. In: *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. Vol. 1. IEEE. 2020, pp. 2547–2551.

[34] Xiaoyao Zheng et al. “A matrix factorization recommendation system-based local differential privacy for protecting users’ sensitive data”. In: *IEEE Transactions on Computational Social Systems* 10.3 (2022), pp. 1189–1198.

[35] GDPR GDPR. “General data protection regulation”. In: *Regulation (EU)* 679 (2016).

[36] Eric Goldman. “An introduction to the california consumer privacy act (ccpa)”. In: *Santa Clara Univ. Legal Studies Research Paper* (2020).

[37] Shahriar Badsha et al. “Privacy preserving user-based recommender system”. In: *2017 IEEE 37th international conference on Distributed Computing Systems (ICDCS)*. IEEE. 2017, pp. 1074–1083.

[38] Caiwen Li et al. “Deep Learning-Based Recommendation System: Systematic Review and Classification”. In: *IEEE Access* (2023).

[39] Byeongjin Choe, Taegwan Kang, and Kyomin Jung. “Recommendation system with hierarchical recurrent neural network for long-term time series”. In: *IEEE Access* 9 (2021), pp. 72033–72039.

[40] Imran Ahmed et al. “A heterogeneous network embedded medicine recommendation system based on LSTM”. In: *Future Generation Computer Systems* 149 (2023), pp. 1–11.

[41] Abiodun E Onile et al. “Uses of the digital twins concept for energy services, intelligent recommendation systems, and demand side management: A review”. In: *Energy Reports* 7 (2021), pp. 997–1015.

[42] Bei Hui et al. “Personalized recommendation system based on knowledge embedding and historical behavior”. In: *Applied Intelligence* (2022), pp. 1–13.

[43] Shivangi Gheewala et al. “Exploiting deep transformer models in textual review based recommender systems”. In: *Expert Systems with Applications* 235 (2024), p. 121120.

[44] Ali Mostafaeipour et al. “Investigating the performance of Hadoop and Spark platforms on machine learning algorithms”. In: *The Journal of Supercomputing* 77 (2021), pp. 1273–1300.

[45] Abdul Awal Mintoo et al. “Adversarial Machine Learning In Network Security: A Systematic Review Of Threat Vectors And Defense Mechanisms”. In: *Innovatech Engineering Journal* 1.01 (2024), pp. 80–98.

[46] Yingqiang Ge et al. “A survey on trustworthy recommender systems”. In: *ACM Transactions on Recommender Systems* 3.2 (2024), pp. 1–68.

[47] Hengtong Zhang et al. “Practical data poisoning attack against next-item recommendation”. In: *Proceedings of the web conference 2020*. 2020, pp. 2458–2464.

[48] Peter Georg Picht and Florent Thouvenin. “AI and IP: Theory to policy and back again–policy and research recommendations at the intersection of artificial intelligence and Intellectual Property”. In: *IIC-International Review of Intellectual Property and Competition Law* 54.6 (2023), pp. 916–940.

[49] Jie Zhang et al. “Model watermarking for image processing networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 12805–12812.

[50] Bowen Li et al. “FedIPR: Ownership verification for federated deep neural network models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.4 (2022), pp. 4521–4536.

[51] Minxing Zhang et al. “Membership inference attacks against recommender systems”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, pp. 864–879.

[52] Sagedur Rahman. “Resisting Adversarial Manipulation in Collaborative Filtering: Techniques and Performance Analysis”. In: *Available at SSRN 4623214* (2023).

[53] Guansong Pang et al. “Deep learning for anomaly detection: A review”. In: *ACM computing surveys (CSUR)* 54.2 (2021), pp. 1–38.

[54] Abbas Acar et al. “A survey on homomorphic encryption schemes: Theory and implementation”. In: *ACM Computing Surveys (Csur)* 51.4 (2018), pp. 1–35.

[55] Xudong Zhang et al. “Targeted Data Poisoning Attack on News Recommendation System by Content Perturbation”. In: *arXiv preprint arXiv:2203.03560* (2022).

[56] Aiyuan Zhen and Xin Wang. “The deep learning-based physical education course recommendation system under the internet of things”. In: *Heliyon* 10.19 (2024).

[57] Qian Zhang, Jie Lu, and Yaochu Jin. “Artificial intelligence in recommender systems”. In: *Complex & Intelligent Systems* 7.1 (2021), pp. 439–457.

[58] Nasrin Dehbozorgi, Mourya Teja Kunuku, and Seyedamin Pouriyeh. “Personalized Pedagogy Through a LLM-Based Recommender System”. In: *International Conference on Artificial Intelligence in Education*. Springer. 2024, pp. 63–70.

[59] Kang Gu. "Towards Trustworthy LLMs: Understanding the Security and Privacy Risks of Large Language Models". In: (2024).

[60] Hao Peng et al. "Large-scale hierarchical text classification with recursively regularized deep graph-cnn". In: *Proceedings of the 2018 world wide web conference*. 2018, pp. 1063–1072.

[61] Mahdi Kherad and Amir Jalaly Bidgoly. "Recommendation system using a deep learning and graph analysis approach". In: *Computational Intelligence* 38.5 (2022), pp. 1859–1883.

[62] Bang Chen et al. "Neural-Symbolic Recommendation with Graph-Enhanced Information". In: *International Conference on Neural Information Processing*. Springer. 2023, pp. 411–423.

[63] Weiping Song et al. "AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks". In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2018).

[64] Qiang Chen et al. "Knowledge-enhanced Multi-View Graph Neural Networks for Session-based Recommendation". In: *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2023).

[65] C. Chen et al. "Efficient Neural Matrix Factorization without Sampling for Recommendation". In: *ACM Transactions on Information Systems (TOIS)* 38 (2020), pp. 1–28.

[66] Saurabh Sharma and Vishal Paranjape. "Experimental Hybrid Approach for Improving the Quality of Personalized Product Recommendation Systems with Deep Learning". In: *International Journal of Innovative Research in Computer and Communication Engineering* (2023).

[67] Qidong Liu et al. "Multimodal Recommender Systems: A Survey". In: *ArXiv* abs/2302.03883 (2023).

[68] Mingming Li et al. "Few-Shot Learning for Cold-Start Recommendation". In: *International Conference on Language Resources and Evaluation*. 2024.

[69] Zhaorui Zhang et al. "Towards Understanding the Overfitting Phenomenon of Deep Click-Through Rate Models". In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (2022).

[70] Hai Huang et al. “Data Poisoning Attacks to Deep Learning Based Recommender Systems”. In: *ArXiv* abs/2101.02644 (2021).

[71] Zongwei Wang et al. “Poisoning Attacks and Defenses in Recommender Systems: A Survey”. In: *ArXiv* abs/2406.01022 (2024).

[72] Jinyuan Jia et al. “PORE: Provably Robust Recommender Systems against Data Poisoning Attacks”. In: *ArXiv* abs/2303.14601 (2023).

[73] Mohan Li et al. “A Sampling-Based Method for Detecting Data Poisoning Attacks in Recommendation Systems”. In: *Mathematics* (2024).

[74] Zhibo Zhang et al. “Data Poisoning Attacks on EEG Signal-based Risk Assessment Systems”. In: 2023.

[75] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. “Certified Defenses for Data Poisoning Attacks”. In: *Neural Information Processing Systems.* 2017.

[76] Takahito Ino et al. “Data Poisoning Attack against Neural Network-Based On-Device Learning Anomaly Detector by Physical Attacks on Sensors”. In: *Sensors (Basel, Switzerland)* 24 (2024).

[77] Ye Li et al. “Blockfd: blockchain-based federated distillation against poisoning attacks”. In: *Neural Computing and Applications* 36.21 (2024), pp. 12901–12916.

[78] “Neural Computing and Applications to Marine Data Analytics”. In: *Frontiers Research Topics* (2022).

[79] Weilun Chen et al. “Boosting Decision-Based Black-Box Adversarial Attacks with Random Sign Flip”. In: *European Conference on Computer Vision.* 2020.

[80] Rishi Jha, Jonathan Hayase, and Sewoong Oh. “Label Poisoning is All You Need”. In: *ArXiv* abs/2310.18933 (2023).

[81] Behnam Ghavami et al. “Blind Data Adversarial Bit-flip Attack against Deep Neural Networks”. In: *2022 25th Euromicro Conference on Digital System Design (DSD)* (2022), pp. 899–904.

[82] Kuofeng Gao et al. “Clean-label Backdoor Attack against Deep Hashing based Retrieval”. In: *ArXiv* abs/2109.08868 (2021).

[83] Wencong You, Zayd Hammoudeh, and Daniel Lowd. “Large Language Models Are Better Adversaries: Exploring Generative Clean-Label Backdoor Attacks Against Text Classifiers”. In: *ArXiv* abs/2310.18603 (2023).

[84] Bolun Wang et al. “Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks”. In: *2019 IEEE Symposium on Security and Privacy (SP)* (2019), pp. 707–723.

[85] Xinyun Chen et al. “Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning”. In: *ArXiv* abs/1712.05526 (2017).

[86] Fanchao Qi et al. “Hidden Killer: Invisible Textual Backdoor Attacks with Syntactic Trigger”. In: *Annual Meeting of the Association for Computational Linguistics*. 2021.

[87] Mohamed Abdelaal, Christian Hammacher, and Harald Schoening. “REIN: A Comprehensive Benchmark Framework for Data Cleaning Methods in ML Pipelines”. In: *ArXiv* abs/2302.04702 (2023).

[88] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM Comput. Surv.* 41 (2009), 15:1–15:58.

[89] Rui Wen et al. “Is Adversarial Training Really a Silver Bullet for Mitigating Data Poisoning?” In: *International Conference on Learning Representations*. 2023.

[90] Jing Lin, Ryan S. Luley, and Kaiqi Xiong. “From adversarial examples to data poisoning instances: utilizing an adversarial attack method to poison a transfer learning model”. In: *ICC 2022 - IEEE International Conference on Communications* (2022), pp. 2351–2356.

[91] Massoud Mohsendokht et al. “Enhancing maritime transportation security: A data-driven Bayesian network analysis of terrorist attack risks.” In: *Risk analysis : an official publication of the Society for Risk Analysis* (2024).

[92] Chaoyi Zhu, Stefanie Roos, and Lydia Yiyu Chen. “LeadFL: Client Self-Defense against Model Poisoning in Federated Learning”. In: *International Conference on Machine Learning*. 2023.

[93] Eric A. Klein et al. “Clinical validation of a targeted methylation-based multi-cancer early detection test using an independent validation set.” In: *Annals of oncology : official journal of the European Society for Medical Oncology* (2021).

[94] Michael Price et al. “From Cadbury to Kay: discourse, intertextuality and the evolution of UK corporate governance”. In: *Accounting, Auditing & Accountability Journal* (2018).

[95] Yousef Farhaoui and Ahmad El Allaoui. “Deep Learning-Based Predictive Analytics for Anomaly Detection in Big Data Environments”. In: *The International Workshop on Big Data and Business Intelligence*. Springer. 2024, pp. 148–154.

[96] Karsten Roth et al. “Towards Total Recall in Industrial Anomaly Detection”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 14298–14308.

[97] Kexin Wu and Kun Chi. “Enhanced E-commerce Customer Engagement: A Comprehensive Three-Tiered Recommendation System”. In: *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* (2024).

[98] Sana Ben Hamida et al. “Assessment of data augmentation, dropout with L2 Regularization and differential privacy against membership inference attacks”. In: *Multim. Tools Appl.* 83 (2023), pp. 44455–44484.

[99] Manish Tongia. *Most Common Security Threats to Machine Learning Systems (Part 2)*. Dec. 2021. URL: https://heartbeat.comet.ml/most-common-security-threats-to-machine-learning-systems-part-2-40381eaf9fd8.

[100] Kun Shao et al. “Textual backdoor defense via poisoned sample recognition”. In: *Applied Sciences* 11.21 (2021), p. 9938.

[101] Xueluan Gong et al. “Kaleidoscope: Physical Backdoor Attacks Against Deep Neural Networks With RGB Filters”. In: *IEEE Transactions on Dependable and Secure Computing* 20 (2023), pp. 4993–5004.

[102] Shuo Wang et al. “Backdoor Attacks Against Transfer Learning With Pre-Trained Deep Learning Models”. In: *IEEE Transactions on Services Computing* 15 (2020), pp. 1526–1539.

[103] Orson Mengara, Anderson R. Avila, and Tiago H. Falk. “Backdoor Attacks to Deep Neural Networks: A Survey of the Literature, Challenges, and Future Research Directions”. In: *IEEE Access* 12 (2024), pp. 29004–29023.

[104] C. Chen et al. “Recommendation Unlearning”. In: *Proceedings of the ACM Web Conference 2022* (2022).

[105] Wenyan Liu et al. “Forgetting fast in recommender systems”. In: *arXiv preprint arXiv:2208.06875* (2022).

[106] Jing Long et al. “Decentralized collaborative learning framework for next POI recommendation”. In: *ACM Transactions on Information Systems* 41.3 (2023), pp. 1–25.

[107] Ziyao Liu et al. “Threats, attacks, and defenses in machine unlearning: A survey”. In: *arXiv preprint arXiv:2403.13682* (2024).

[108] Thanh Tam Nguyen et al. “A survey of machine unlearning”. In: *arXiv preprint arXiv:2209.02299* (2022).

[109] Wei Yuan et al. “Federated unlearning for on-device recommendation”. In: *Proceedings of the sixteenth ACM international conference on web search and data mining*. 2023, pp. 393–401.

[110] Thanveer Shaik et al. “Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[111] Yang Zhang et al. “Recommendation unlearning via influence function”. In: *ACM Transactions on Recommender Systems* 3.2 (2024), pp. 1–23.

[112] Jiahao Liu et al. “Recommendation unlearning via matrix correction”. In: *arXiv preprint arXiv:2307.15960* (2023).

[113] Youyang Qu et al. “Learn to unlearn: Insights into machine unlearning”. In: *Computer* 57.3 (2024), pp. 79–90.

[114] Yu Guo et al. “Verifying in the dark: Verifiable machine unlearning by using invisible backdoor triggers”. In: *IEEE Transactions on Information Forensics and Security* (2023).

[115] Yifan Yan et al. “Rethinking {White-Box} Watermarks on Deep Learning Models under Neural Structural Obfuscation”. In: *32nd USENIX Security Symposium (USENIX Security 23)*. 2023, pp. 2347–2364.

[116] Yiming Li et al. “Black-box dataset ownership verification via backdoor watermarking”. In: *IEEE Transactions on Information Forensics and Security* 18 (2023), pp. 2318–2332.

[117] Huili Chen et al. “Intellectual Property Protection of Deep Learning Systems via Hardware/Software Co-design”. In: *IEEE Design & Test* (2023).

[118] Sixiao Zhang et al. “Data Watermarking for Sequential Recommender Systems”. In: *arXiv preprint arXiv:2411.12989* (2024).

[119] Mingfu Xue et al. "Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations". In: *IEEE Transactions on Artificial Intelligence* 3.6 (2021), pp. 908–923.

[120] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[121] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. "Bert: Pretraining of deep bidirectional transformers for language understanding". In: *Proceedings of naacL-HLT*. Vol. 1. 2. Minneapolis, Minnesota. 2019.

[122] A Waswani et al. "Attention is all you need". In: *NIPS*. 2017.

[123] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[124] Aakanksha Chowdhery et al. "Palm: Scaling language modeling with pathways". In: *Journal of Machine Learning Research* 24.240 (2023), pp. 1–113.

[125] Peng Liu, Lemei Zhang, and Jon Atle Gulla. "Pre-train, Prompt, and Recommendation: A Comprehensive Survey of Language Modeling Paradigm Adaptations in Recommender Systems". In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 1553–1571.

[126] Zihuai Zhao et al. "Recommender systems in the era of large language models (llms)". In: *arXiv preprint arXiv:2307.02046* (2023).

[127] Likang Wu et al. "A survey on large language models for recommendation". In: *World Wide Web* 27.5 (2024), p. 60.

[128] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[129] Mohammad Shoeybi et al. "Megatron-lm: Training multi-billion parameter language models using model parallelism". In: *arXiv preprint arXiv:1909.08053* (2019).

[130] Tim Dettmers et al. "Qlora: Efficient finetuning of quantized llms". In: *Advances in Neural Information Processing Systems* 36 (2024).

[131] Emily M Bender et al. "On the dangers of stochastic parrots: Can language models be too big?" In: *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*. 2021, pp. 610–623.

[132] Lei Chen et al. “Enhancing ID-based Recommendation with Large Language Models”. In: *arXiv preprint arXiv:2411.02041* (2024).

[133] Peiyang Yu et al. “The application of large language models in recommendation systems”. In: *arXiv preprint arXiv:2501.02178* (2025).

[134] Sichun Luo et al. “Integrating large language models into recommendation via mutual augmentation and adaptive aggregation”. In: *arXiv preprint arXiv:2401.13870* (2024).

[135] Jianchao Ji et al. “Genrec: Large language model for generative recommendation”. In: *European Conference on Information Retrieval*. Springer. 2024, pp. 494–502.

[136] Nicholas Carlini et al. “Extracting training data from large language models”. In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021, pp. 2633–2650.

[137] Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. “Security and privacy challenges of large language models: A survey”. In: *ACM Computing Surveys* 57.6 (2025), pp. 1–39.

[138] Jinghao Zhang et al. “Stealthy attack on large language model based recommendation”. In: *arXiv preprint arXiv:2402.14836* (2024).

[139] Xiaocui Dang et al. “A Dual Defense Design Against Data Poisoning Attacks in Deep Learning-Based Recommendation Systems”. In: *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE. 2024, pp. 2115–2122.

[140] Xiangnan He et al. “Neural collaborative filtering”. In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 173–182.

[141] Yizhi Ren et al. “Semantic Shilling Attack against Heterogeneous Information Network Based Recommend Systems”. In: *IEICE TRANSACTIONS on Information and Systems* 105.2 (2022), pp. 289–299.

[142] Sanjay Seetharaman et al. “Influence based defense against data poisoning attacks in online learning”. In: *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE. 2022, pp. 1–6.

[143] Vasiliki Kelli et al. “Attacking and defending DNP3 ICS/SCADA systems”. In: *2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2022, pp. 183–190.

[144] Umar Islam et al. "Detection of distributed denial of service (DDoS) attacks in IOT based monitoring system of banking sector using machine learning models". In: *Sustainability* 14.14 (2022), p. 8374.

[145] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[146] H Hamidi and R Moradi. "Design of a dynamic and robust recommender system based on item context, trust, rating matrix and rating time using social networks analysis". In: *Journal of King Saud University-Computer and Information Sciences* 36.2 (2024), p. 101964.

[147] Javier Carnerero-Cano et al. "Regularisation can mitigate poisoning attacks: A novel analysis based on multiobjective bilevel optimisation". In: *arXiv preprint arXiv:2003.00040* (2020).

[148] Reda Yacouby and Dustin Axman. "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models". In: *Proceedings of the first workshop on evaluation and comparison of NLP systems*. 2020, pp. 79–91.

[149] Björn Bebensee. "Local differential privacy: a tutorial". In: *arXiv preprint arXiv:1907.11908* (2019).

[150] Minh-Hao Van, Alycia N Carey, and Xintao Wu. "HINT: Healthy Influential-Noise based Training to Defend against Data Poisoning Attacks". In: *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2023, pp. 608–617.

[151] Xiaocui Dang et al. "A Novel Scheme for Recommendation Unlearning Verification (RUV) Using Non-influential Trigger Data". In: *Proceedings of the IEEE Consumer Communications & Networking Conference (CCNC)*. 2025.

[152] Ruiqi Zheng et al. "Automl for deep recommender systems: A survey". In: *ACM Transactions on Information Systems* 41.4 (2023), pp. 1–38.

[153] Yang Li et al. "Recent developments in recommender systems: A survey". In: *IEEE Computational Intelligence Magazine* 19.2 (2024), pp. 78–95.

[154] David Marco Sommer et al. "Towards probabilistic verification of machine unlearning". In: *arXiv preprint arXiv:2003.04247* (2020).

[155] Jiyang Guan, Jian Liang, and Ran He. "Are you stealing my model? sample correlation for fingerprinting deep neural networks". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36571–36584.

[156] Gaoyang Liu et al. “Your Model Trains on My Data? Protecting Intellectual Property of Training Data via Membership Fingerprint Authentication”. In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 1024–1037.

[157] Zirui Peng et al. “Fingerprinting Deep Neural Networks Globally via Universal Adversarial Perturbations”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 13420–13429.

[158] Xiaocui Dang et al. “Recommendation System Model Ownership Verification via Non-Influential Watermarking”. In: *2024 17th International Conference on Security of Information and Networks (SIN)*. IEEE. 2024, pp. 1–8.

[159] Jie Zhang et al. “Deep Model Intellectual Property Protection via Deep Watermarking”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44 (2021), pp. 4005–4020.

[160] Sebastian Szyller et al. “DAWN: Dynamic Adversarial Watermarking of Neural Networks”. In: *Proceedings of the 29th ACM International Conference on Multimedia* (2019).

[161] Ruixiang Tang, Mengnan Du, and Xia Hu. “Deep Serial Number: Computational Watermark for DNN Intellectual Property Protection”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2023, pp. 157–173.

[162] Yihan Wu et al. *DiPmark: A Stealthy, Efficient and Resilient Watermark for Large Language Models*. 2024.

[163] Yusuke Uchida et al. “Embedding Watermarks into Deep Neural Networks”. In: *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval* (2017).

[164] Xiaocui Dang et al. “Robust and Adaptive Dual-Defense Against Data Poisoning Attacks in Recommendation Systems”. In: *Future Generation Computer Systems,* (2025).

[165] Shiyi Yang et al. “Review-Incorporated Model-Agnostic Profile Injection Attacks on Recommender Systems”. In: *2023 IEEE International Conference on Data Mining (ICDM)* (2023), pp. 1481–1486.

[166] Meiling Chao et al. “PATR: A Novel Poisoning Attack Based on Triangle Relations Against Deep Learning-Based Recommender Systems”. In: *International Conference on Collaborative Computing*. 2021.

[167] Zhenrui Yue et al. “Defending Substitution-Based Profile Pollution Attacks on Sequential Recommenders”. In: *Proceedings of the 16th ACM Conference on Recommender Systems* (2022).

[168] Abdulrahman Takiddin et al. “Robust Graph Autoencoder-Based Detection of False Data Injection Attacks Against Data Poisoning in Smart Grids”. In: *IEEE Transactions on Artificial Intelligence* 5 (2024), pp. 1287–1301.

[169] Xianfeng Tang et al. “Transferring Robustness for Graph Neural Network Against Poisoning Attacks”. In: *Proceedings of the 13th International Conference on Web Search and Data Mining* (2019).

[170] Xiaogang Xing et al. “A graph backdoor detection method for data collection scenarios”. In: *Cybersecur.* 8 (2025), p. 1.

[171] Chenwang Wu et al. “Fight Fire with Fire: Towards Robust Recommender Systems via Adversarial Poisoning Training”. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).

[172] Yulin Zhu et al. “BinarizedAttack: Structural Poisoning Attacks to Graph-based Anomaly Detection”. In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)* (2021), pp. 14–26.

[173] Shiyu Li et al. “NDRec: A Near-Data Processing System for Training Large-Scale Recommendation Models”. In: *IEEE Transactions on Computers* 73 (2024), pp. 1248–1261.

[174] Yunfan Wu et al. “Accelerating the Surrogate Retraining for Poisoning Attacks against Recommender Systems”. In: *ArXiv* abs/2408.10666 (2024).

[175] Augustinas Zinys, Bram van Berlo, and Nirvana Meratnia. “A Domain-Independent Generative Adversarial Network for Activity Recognition Using WiFi CSI Data”. In: *Sensors (Basel, Switzerland)* 21 (2021).