

REPUBLIC OF TÜRKİYE
AKDENİZ UNIVERSITY



**EXAMINING THE EFFECTS OF PROFILE INJECTION ATTACKS ON
RECOMMENDER SYSTEMS ON THE POPULARITY BIAS PROBLEM**

Özge GÜREL

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MSc THESIS

NOVEMBER 2024

ANTALYA

REPUBLIC OF TÜRKİYE
AKDENİZ UNIVERSITY



**EXAMINING THE EFFECTS OF PROFILE INJECTION ATTACKS ON
RECOMMENDER SYSTEMS ON THE POPULARITY BIAS PROBLEM**

Özge GÜREL

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MSc THESIS

NOVEMBER 2024

ANTALYA

REPUBLIC OF TÜRKİYE
AKDENİZ UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

**EXAMINING THE EFFECTS OF PROFILE INJECTION ATTACKS ON
RECOMMENDER SYSTEMS ON THE POPULARITY BIAS PROBLEM**

Özge GÜREL
DEPARTMENT OF COMPUTER ENGINEERING
MSc THESIS

NOVEMBER 2024

T.C.
AKDENİZ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

EXAMINING THE EFFECTS OF PROFILE INJECTION ATTACKS ON
RECOMMENDER SYSTEMS ON THE POPULARITY BIAS PROBLEM

Özge GÜREL
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI
YÜKSEK LİSANS TEZİ

Bu tez 01/11/2024 tarihinde jüri tarafından Oybirliği / Oyçokluğu ile kabul edilmiştir.

Doç. Dr. Alper BİLGE (Danışman)

Dr. Öğr. Üyesi Murat AK

Prof. Dr. Cihan KALELİ

ABSTRACT

EXAMINING THE EFFECTS OF PROFILE INJECTION ATTACKS ON RECOMMENDER SYSTEMS ON THE POPULARITY BIAS PROBLEM

Özge GÜREL

MSc Thesis in Computer Engineering

Supervisor: Assoc. Prof. Dr. Alper BİLGE

Co-Advisor: Assoc. Prof. Dr. Emre YALÇIN

November 2024; 59 pages

Digital platforms, in their quest to engage users with personalized recommendations, face an unforeseen challenge: popularity bias. Recommendation algorithms often favour popular content, limiting users' access to new and niche items, thereby reducing content diversity on these platforms. This thesis proposes an innovative approach to reduce popularity bias in recommendation systems, enhancing users' access to a broader range of content.

The unique aspect of this study is its strategic use of profile injection attack models, traditionally seen as security risks, to mitigate popularity bias. Attack models such as Average, Bandwagon, and Random Attack, typically used to manipulate systems, are here repurposed to restore diversity and balance in recommendations. In particular, the Random Attack model emerges as a strong alternative, promoting less popular content to achieve a fairer and more diverse recommendation system. Additionally, traditional debiasing techniques, such as XQUAD, ERPAug, ERPMul, and VAR, are analysed and compared against attack models for their effectiveness in reducing popularity bias.

This study is conducted using the widely adopted MovieLens dataset and employs the VAECF (Variational Autoencoder for Collaborative Filtering) algorithm. The experiments assess each method's impact on popularity bias and recommendation quality through core performance metrics such as APLT, F1-Score, nDCG, and Novelty. Results indicate that XQUAD is the most effective method for reducing popularity bias, while Random Attack (Attack Size 10, Filler Size 10) provides a balanced performance, preserving both recommendation accuracy and diversity. On the other hand, ERPMul and VAR offer high diversity but may compromise some recommendation accuracy. These findings serve as practical insights for digital platforms aiming to provide fairer, more diverse, and user-centric recommendations.

In conclusion, this thesis presents an innovative perspective on mitigating popularity bias in recommendation systems, contributing to strategies that enhance

fairness and diversity. The findings can serve as a foundation for future studies, which may broaden this approach by exploring various datasets and algorithms, ultimately advancing solutions for fair and diverse content delivery in recommendation systems.

KEYWORDS: Debiasing Methods, Diversity in Recommendations, Popularity Bias, Profile Injection Attacks, Recommender Systems, Unfairness

COMMITTEE: Assoc. Prof. Dr. Alper BİLGE

Asst. Prof. Murat AK

Prof. Dr. Cihan KALELİ



ÖZET

ÖNERİ SİSTEMLERİNE YAPILAN PROFİL ENJEKSİYON ATAKLARININ POPÜLERLİK YANLILIĞI SORUNU ÜZERİNDEKİ ETKİLERİNİN İNCELENMESİ

Özge GÜREL

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Alper BİLGE

Eş Danışman: Doç. Dr. Emre YALÇIN

Kasım 2024; 59 sayfa

Dijital platformlar, kullanıcıların ilgisini çekmek için sundukları içerikleri öneri sistemleriyle kişiselleştirirken, aynı zamanda öngörülemeyen bir sorun olan popülerlik yanlılığı ile karşı karşıya kalmaktadır. Öneri algoritmalarının popüler içeriklere yönelmesi, kullanıcıların yeni ve niş içeriklere ulaşma imkanını sınırlandırarak platformlardaki içerik çeşitliliğini daraltmaktadır. Bu tez, öneri sistemlerinde popülerlik yanlılığını azaltmak ve kullanıcıların daha çeşitli içeriklere erişimini sağlamak amacıyla yenilikçi bir yaklaşım sunmaktadır.

Çalışmanın özgün yanı, genelde güvenlik riski olarak bilinen profil enjeksiyon atak modellerini popülerlik yanlılığını azaltmada stratejik bir araç olarak kullanmasıdır. Average, Bandwagon ve Random Attack gibi saldırı modelleri, bu tezde, sistemin çeşitlilik dengesini sağlamak amacıyla farklı bir kullanım alanına taşınmıştır. Özellikle Random Attack modeli, popüler olmayan içerikleri öne çıkararak öneri sistemini daha dengeli ve adil hale getirmek için güçlü bir alternatif sunmaktadır. Aynı zamanda, popülerlik yanlılığını azaltma amacıyla önerilen geleneksel yöntemler olan XQUAD, ERPAug, ERPMul ve VAR gibi debiasing teknikleri de karşılaştırmalı olarak analiz edilmiştir.

Bu çalışma, yaygın kullanılan MovieLens veri seti üzerinde ve VAECF (Variational Autoencoder for Collaborative Filtering) algoritması kullanılarak gerçekleştirilmiştir. Deneylerde, her yöntemin popülerlik yanlılığını azaltma etkisi ve öneri kalitesini koruma düzeyi, APLT, F1-Score, nDCG ve Novelty gibi performans metrikleri ile ölçülmüştür. Sonuçlar, XQUAD yönteminin popülerlik yanlılığını azaltmada en etkili yöntem olduğunu ortaya koyarken, Random Attack (Attack Size 10, Filler Size 10) modelinin ise öneri doğruluğunu ve çeşitliliği bir arada koruyarak dengeli bir performans sunduğunu göstermektedir. Buna karşılık, ERPMul ve VAR yöntemleri yüksek çeşitlilik sağlarken öneri doğruluğunda bazı ödünler verebilmektedir. Bu

bulgular, dijital platformların adil, çeşitli ve kullanıcı odaklı öneriler sunmak için hangi yöntemi seçebileceğine dair rehber niteliğinde bilgiler sunmaktadır.

Sonuç olarak, bu tez öneri sistemlerinde popülerlik yanlılığını azaltmaya yönelik yenilikçi bir bakış açısı sunarak, adalet ve çeşitliliği artırmayı hedefleyen stratejik yaklaşımlara katkıda bulunmaktadır. Çalışmanın bulguları, gelecekte farklı veri setleri ve algoritmalarla genişletilerek öneri sistemlerinde adil ve çeşitlendirilmiş içerik sunma yönündeki çözümlerin geliştirilmesine ışık tutabilir.

ANAHTAR KELİMELELER: Adaletsizlik, Öneri Sistemleri, Önerilerde Çeşitlilik, Popülerlik Yanlılığı, Profil Enjeksiyon Saldırıları, Yanlılık Azaltma Yöntemleri

JÜRİ: Doç. Dr. Alper BİLGE

Dr. Öğr. Üyesi Murat AK

Prof. Dr. Cihan KALELİ

PREFACE

In today's digital world, recommender systems, designed to facilitate user access to information and provide personalized experiences, have become indispensable on platforms like Netflix, Amazon, and Spotify. These systems significantly enhance user satisfaction and drive business growth. However, as research into these systems advances, challenges surrounding their accuracy and fairness have surfaced. Among these challenges, popularity bias—the tendency of recommender systems to prioritize a small set of popular items—stands out, often limiting users' exposure to new or niche content and impacting overall system fairness.

This thesis addresses this critical issue by exploring not only the impact of popularity bias but also the potential of repurposing profile injection attacks as a debiasing method to counteract this bias. Traditionally viewed as security risks, profile injection attacks are innovatively utilized in this study to strategically reduce popularity bias, offering a novel perspective on balancing recommendation relevance and diversity. Through a comparative analysis of various debiasing methods and attack models, this research provides meaningful contributions toward developing fairer and more inclusive recommendation systems.

I would like to extend my sincere gratitude to Assoc. Prof. Dr. Alper BİLGE for his invaluable guidance, exceptional expertise, and continuous support throughout this journey. His profound knowledge and mentorship have been instrumental in shaping the direction and quality of this work. I am deeply thankful for his encouragement and the clarity he brought to every stage of this project. I am also grateful to Assoc. Prof. Dr. Emre YALÇIN for his insightful feedback and expertise, which greatly enriched this work. Finally, I am deeply thankful to my family, whose unwavering support and encouragement have been my constant source of strength during this project.

I hope that this thesis serves as a valuable resource for researchers and professionals in the field of recommender systems, contributing to advancements in building fairer and more user-centred recommendation technologies.

Özge Gürel

November 2024

TABLE OF CONTENTS

ABSTRACT	i
ÖZET	iii
PREFACE	v
TABLE OF CONTENTS	vi
ACADEMIC DECLARATION	viii
SYMBOLS AND ABBREVIATIONS	ix
LIST OF FIGURES	xi
LIST OF TABLES	xii
1. INTRODUCTION	1
1.1. Problem Statement	1
1.2. Research Objectives	1
1.3. Research Questions	2
1.4. Contributions of the Study	3
1.5. Thesis Structure	4
2. LITERATURE REVIEW	6
2.1. Recommender Systems and Algorithms	6
2.2. Popularity Bias	7
2.3. Profile Injection Attacks	9
2.4. Contributions of Previous Studies to the Thesis	10
2.5. Summary and Research Questions	12
3. MATERIALS AND METHODS	14
3.1. Dataset	14
3.2. Recommendation Algorithm	14
3.3. Profile Injection Attacks	15
3.4. Debiasing Methods	17
3.5. Evaluation Methods and Metrics	18
3.6. Experimental Setup	20
3.7. Dataset Analysis and Preparation	21
3.8. Procedure for Applying Profile Injection Attacks	24
3.9. Procedure for Applying Recommendation Algorithm	28
3.10. Procedure for Applying Debiasing Methods	33

3.11. Performance Evaluation.....	36
4. FINDINGS	42
4.1. Impact of Attack Models on Popularity Bias and Recommendation Quality	42
4.2. Comparison of Debiasing Methods with the Best Attack Model.....	45
4.3. Evaluation of Overall Performance Against the Base Model	47
4.4. Summary of Findings	48
5. DISCUSSION	52
5.1. Effectiveness of Debiasing Methods and Attack Models	52
5.2. Trade-Offs in Popularity Bias Reduction and Recommendation Quality.....	52
5.3. Impact of Metrics on Debiasing Goals.....	52
5.4. Comparison of Debiasing Methods and Their Practical Implications	53
5.5. Insights into Choosing Between Debiasing Methods and Attack Models	53
6. CONCLUSIONS.....	54
6.1. Summary of Key Findings	54
6.2. Practical Implications for Recommender System Design.....	54
6.3. Limitations and Future Directions.....	54
7. REFERENCES.....	56
8. APPENDICES.....	59
CURRICULUM VITAE	

ACADEMIC DECLARATION

I hereby declare that this work, titled “EXAMINING THE EFFECTS OF PROFILE INJECTION ATTACKS ON RECOMMENDER SYSTEMS ON THE POPULARITY BIAS PROBLEM”, which I have submitted as a Master's Thesis, has been written in accordance with academic rules and ethical values. I also declare that I have cited all sources for the information that is not my own in this thesis.

01/11/2024

Özge GÜREL



SYMBOLS AND ABBREVIATIONS

Symbols

corr : The correlation coefficient

$F1@N$: The harmonic mean of precision and recall at N

i : Item

I_F : Filler Items

I_S : Selected Items

$I_{\emptyset}$: Unrated Items

I_T : Targeted Item

I_u : The set of items rated by user u

N : The specified number of the most popular items

N_u : The list of recommended items to group g of which u belongs

p : The slope coefficient

$P@N$: Precision at N

$r_{u,i}$: The rating of u on item i

r_{max} : The maximum rating

r_{min} : The minimum rating

T : The set of tail items

top-N : The set of the most popular N items

u : The user profile

In this thesis, the decimal separator "21.01" is used.

Abbreviations

APLT : Average Percentage of Long-tail Items

DCG : Discounted Cumulative Gain

ERP	: The Enhanced Re-ranking Procedure
ERPAug	: The Enhanced Re-ranking Procedure - Augmentative
ERPMul	: The Enhanced Re-ranking Procedure - Multiplicative
IDCG	: Ideal Discounted Cumulative Gain
ML	: MovieLens
ML-1M	: MovieLens-1Million
nDCG	: Normalized Discounted Cumulative Gain
RSs	: Recommender Systems
VAECF	: Variational Autoencoders for Collaborative Filtering
VAR	: The Value-Aware Ranking
XQUAD	: The Explicit Query Aspect Diversification

LIST OF FIGURES

Figure 3.1. The long-tail distribution of item popularity in the MovieLens (ML) dataset.....	22
Figure 3.2. General form of attack profiles.....	25
Figure 3.3. Long tail distribution of MovieLens dataset under different attack models	28
Figure 3.4. Correlation between item popularity and recommendation frequency for VAE CF algorithm	31
Figure 3.5. Correlation between item popularity and recommendation frequency across debiasing methods.....	34
Figure 3.6. The impact of debiasing methods on VAE CF performance metrics	40



LIST OF TABLES

Table 3.1. Brief descriptions of profile injection attacks	16
Table 3.2. Brief descriptions of popularity debiasing methods.....	18
Table 3.3. Brief descriptions of performance metrics	20
Table 3.4. Statistics of the MovieLens (ML) dataset	22
Table 3.5. Cross-validation process overview.....	23
Table 3.6. Summary of attack profiles	25
Table 3.7. User-Item matrix	26
Table 4.1. Comparison of attack models on APLT, F1-Score, nDCG, and Novelty metrics	42
Table 4.2. Performance of debiasing methods compared to the best attack model and base model.....	45



1. INTRODUCTION

1.1. Problem Statement

Recommender systems (RSs) are critical technologies widely used in the digital world to facilitate user access to information and provide personalized experiences. These systems suggest products, services, or content to users based on their past behaviours and preferences. Recommender systems, commonly used on platforms such as Netflix, Amazon, and Spotify, play a significant role in enhancing user satisfaction and boosting business revenues (Adomavicius and Tuzhilin, 2005; Ricci, Rokach, Shapira, and Kantor, 2011).

However, the accuracy and fairness of recommender systems face various challenges. One of the primary challenges is the tendency known as popularity bias. Popularity bias can be defined as the propensity of recommender systems to consistently highlight a small number of popular items, reducing the likelihood of recommending less known or new items (Abdollahpouri, Burke, and Mobasher, 2017). This bias can negatively impact user experience and undermine the fairness of the system (Zhou et al., 2010).

Moreover, recommender systems are susceptible to various attacks, one of the most common being profile injection attacks. Profile injection attacks aim to deceive the algorithms of recommender systems by creating fake user profiles or manipulating existing ones (Lam and Riedl, 2004). Such attacks can severely jeopardize the accuracy and security of the system's recommendations (Burke, Mobasher, and Williams, 2006).

1.2. Research Objectives

This thesis has two primary objectives. The first is to examine the effects of profile injection attacks on popularity bias in recommender systems, with a focus on their potential as a debiasing method. The second objective is to investigate the effectiveness of established debiasing methods from the literature and compare them with the best-performing profile injection attack in reducing popularity bias.

To achieve these goals, this research will conduct experiments using both profile injection attacks and debiasing methods. Specifically, debiasing methods such as ErpMul, ErpAug, VAR, and XQUAD will be evaluated for their ability to mitigate popularity bias (Gantner et al., 2011). In terms of profile injection attacks, Average, Bandwagon, and Random attacks will be applied to assess their effectiveness in reducing bias (Harper and Konstan, 2015; Cantador, Brusilovsky, and Kuflik, 2011). These experiments will utilize the VAECF recommendation algorithm to provide a consistent evaluation framework.

The dataset used in this study will be the MovieLens dataset (Harper & Konstan, 2015), which is widely used in recommender system research. The recommender system's performance will be evaluated based on key metrics such as popularity bias, accuracy, diversity, and fairness (Zhou et al., 2010). This analysis aims to provide insights into how both profile injection attacks and debiasing methods influence the fairness and accuracy

of recommendation systems, offering a comparison between traditional debiasing methods and profile injection attacks in mitigating popularity bias.

1.3. Research Questions

This thesis aims to address the following key research questions, focused on understanding the impact of profile injection attacks on popularity bias and the effectiveness of debiasing methods in recommender systems.

One important aspect of the research investigates how different profile injection attacks, such as Average, Bandwagon, and Random, influence the level of popularity bias in recommender systems. It also examines whether these attacks can be used strategically to reduce popularity bias and to what extent they help in promoting less popular items.

How do different profile injection attacks, such as Average, Bandwagon, and Random, influence the level of popularity bias in recommender systems?

Can these attacks be used strategically to reduce popularity bias, and to what extent can they help in promoting less popular items?

Another focal point is to evaluate how effective established debiasing methods like ErpMul, ErpAug, VAR, and XQUAD are in mitigating popularity bias in the VAE CF recommendation algorithm. This includes a comparison of these debiasing methods with the best-performing profile injection attack in terms of their effectiveness in reducing popularity bias.

How effective are established debiasing methods like ErpMul, ErpAug, VAR, and XQUAD in mitigating popularity bias in the VAE CF recommendation algorithm?

How do these debiasing methods compare to the best-performing profile injection attack in terms of reducing popularity bias?

Finally, this research examines how profile injection attacks and debiasing methods affect the overall performance of the recommender system. Particular attention is given to metrics such as recommendation accuracy, diversity, and fairness, providing a comprehensive analysis of their impact on system performance.

How do profile injection attacks and debiasing methods affect the overall performance of the recommender system, particularly in terms of recommendation accuracy, diversity, and fairness?

These questions will guide the analysis in this research, providing insights into both the potential of profile injection attacks as debiasing strategies and the comparative effectiveness of traditional debiasing methods.

1.4. Contributions of the Study

This thesis makes several significant contributions to the field of recommender systems, particularly in addressing the challenge of popularity bias through innovative approaches. The first major contribution is the exploration of profile injection attacks as a potential debiasing strategy. Traditionally seen as detrimental to the performance of recommender systems, profile injection attacks are reconsidered in this study for their possible use in reducing popularity bias. By analysing the effects of various attack types, such as Average, Bandwagon, and Random, on the recommendation process, this research provides a novel approach to mitigating bias and promoting less popular items in the recommendation list.

A second key contribution is the comprehensive evaluation of traditional debiasing methods, including ErpMul, ErpAug, VAR, and XQUAD. These methods have been studied extensively in the literature for their ability to reduce popularity bias. This thesis critically assesses their performance when applied to the VAE CF recommendation algorithm, highlighting their strengths and limitations in improving fairness and diversity within the recommendation system. This detailed evaluation contributes valuable insights into the practical effectiveness of these methods.

The study's comparative analysis between profile injection attacks and traditional debiasing methods represents another significant contribution. By directly comparing these two approaches in terms of their impact on popularity bias, the research helps to illuminate the trade-offs between different strategies for enhancing recommendation fairness. This comparison provides a new perspective on the potential role of adversarial techniques in improving recommendation diversity, offering a fresh avenue for future research in the field.

Additionally, the application of the VAE CF algorithm to the MovieLens dataset further enhances the contribution of this research. The VAE CF algorithm is employed as a state-of-the-art recommendation framework, and the use of the well-known MovieLens dataset ensures that the findings of this research are relevant and comparable to other studies in the domain. This approach strengthens the external validity of the research, making it applicable to real-world recommender system scenarios.

Finally, the study introduces a multi-metric evaluation of system performance, focusing not only on popularity bias but also on other critical metrics such as accuracy, diversity, and fairness. By incorporating these multiple dimensions into the evaluation process, the research provides a more comprehensive understanding of the effects of both profile injection attacks and debiasing methods on overall system performance. This holistic analysis contributes to ongoing efforts to create fairer and more diverse recommender systems that maintain high accuracy levels.

1.5. Thesis Structure

This thesis is structured to thoroughly explore the impact of profile injection attacks on popularity bias in recommender systems and compare these attacks with traditional debiasing methods. The structure is designed to systematically address the research objectives and questions, analyse the methodologies used, and discuss the findings in relation to existing literature.

The Introduction chapter provides the context of the research, outlining the significance of recommender systems in the digital world and the challenges posed by popularity bias. It also highlights the threat of profile injection attacks, which have the potential to manipulate the fairness and accuracy of recommendations. This chapter clearly states the research objectives and poses the key research questions, such as how profile injection attacks impact popularity bias and how debiasing methods compare to these attacks in mitigating this bias.

The Literature Review delves into existing studies on recommender systems, popularity bias, and profile injection attacks. It offers a detailed background on key concepts, including the manifestation and effects of popularity bias, the types of profile injection attacks (e.g., Average, Bandwagon, and Random), and the effectiveness of debiasing methods like ErpMul, ErpAug, VAR, and XQUAD. This chapter also discusses how previous research has approached the balance between fairness, accuracy, and diversity in recommender systems.

The Materials and Methods section details the experimental setup, including the dataset (MovieLens) and the VAE CF recommendation algorithm, which is employed as the primary method for evaluating the effectiveness of both profile injection attacks and debiasing methods. This chapter outlines the methodology for applying profile injection attacks and the procedures for measuring their impact on popularity bias. It also explains the metrics used to assess the system's performance, such as accuracy, diversity, and fairness.

The Findings chapter presents the results of the experiments conducted, analysing the impact of different profile injection attacks on popularity bias within the VAE CF algorithm. It compares these results to the performance of traditional debiasing methods, offering insights into the strengths and weaknesses of each approach. The analysis focuses on how both attack and debiasing strategies influence key performance metrics and assesses the overall effectiveness of each method in reducing popularity bias.

The Discussion section provides a critical analysis of the results, comparing them with findings from existing literature. It reflects on the broader implications of using profile injection attacks as a debiasing strategy and discusses the challenges of balancing accuracy, diversity, and fairness in recommender systems. This section also explores

potential strategies for enhancing the robustness of recommender systems against biases and adversarial attacks.

Finally, the Conclusion summarizes the key findings of the research, emphasizing the contributions made to the field of recommender systems. It discusses the practical implications of using profile injection attacks for reducing popularity bias and suggests directions for future research, particularly in terms of developing more effective and secure recommendation systems.

By following this structured approach, the thesis aims to provide a comprehensive analysis of the interplay between profile injection attacks and popularity bias, contributing to the development of fairer and more secure recommender systems.



2. LITERATURE REVIEW

Recommender systems are crucial technologies developed to improve user experience by providing personalized recommendations based on past behaviours and preferences. These systems not only enhance user satisfaction but also significantly boost business revenues. Platforms such as Netflix, Amazon, and Spotify exemplify the widespread use and success of recommender systems. However, research on the accuracy and fairness of these systems reveals various challenges they face. This section will cover the fundamental concepts of recommender systems, popularity bias, profile injection attacks, and related studies in these areas.

2.1. Recommender Systems and Algorithms

Recommender systems are crucial for providing personalized experiences across various platforms, including e-commerce, streaming services, and social media. These systems aim to predict user preferences and recommend items such as products, movies, or services based on previous interactions. By filtering vast amounts of information, recommender systems enhance both user satisfaction and business outcomes (Ricci, Rokach, and Shapira, 2015). There are three primary types of recommender systems: collaborative filtering, content-based filtering, and hybrid approaches.

Collaborative filtering is one of the most commonly used methods in recommender systems, relying on the assumption that users with similar preferences in the past will continue to have similar tastes in the future (Schafer, Konstan, and Riedl, 2001). It works by analysing user-item interactions like ratings or clicks to generate recommendations. Collaborative filtering is typically categorized into two approaches: user-based and item-based. In user-based collaborative filtering, recommendations are generated by identifying users with similar preferences and suggesting items that these users have liked (Breese, Heckerman, and Kadie, 1998). In contrast, item-based collaborative filtering compares items based on their similarity and recommends items that are similar to those the user has interacted with (Sarwar et al., 2001). However, collaborative filtering faces challenges such as the cold-start problem, where the system struggles to provide recommendations for new users or items with limited data. Moreover, it tends to exacerbate popularity bias by frequently recommending popular items over lesser-known ones (Adomavicius and Tuzhilin, 2005).

In content-based filtering, the system recommends items based on their attributes and the user's past preferences (Pazzani and Billsus, 2007). For example, in a movie recommendation system, the algorithm might analyse features such as genre, director, or cast to suggest similar movies to those the user has already watched. One key advantage of content-based filtering is its ability to make recommendations even with limited user interaction data, addressing the cold-start problem more effectively. However, it also suffers from over-specialization, where the system only suggests items similar to those

the user has already seen, potentially limiting the discovery of new or diverse content (Lops, De Gemmis, and Semeraro, 2011).

Hybrid approaches combine both collaborative filtering and content-based filtering to overcome the limitations of each method (Burke, 2002). These systems take advantage of both user-item interaction data and item attributes to generate more accurate and diverse recommendations. For instance, Netflix employs a hybrid approach that integrates collaborative filtering with content-based features to recommend shows and movies based on both a user's viewing habits and the characteristics of the content (Gomez-Uribe and Hunt, 2016). While hybrid systems tend to perform better than using either method alone, they also come with increased complexity and computational requirements.

More recently, deep learning-based approaches have gained popularity in recommender systems due to their ability to model complex user-item interactions using neural networks. One prominent example is Variational Autoencoders for Collaborative Filtering (VAECF), which combines deep learning with collaborative filtering to provide more accurate recommendations (Liang et al., 2018). VAECF allows the system to capture hidden patterns and relationships between users and items by learning latent representations. This thesis employs the VAECF algorithm to examine the effects of profile injection attacks and debiasing methods. Although deep learning models have shown superior performance in many cases, they also bring challenges in terms of interpretability and higher computational demands, making their real-world application more difficult in some contexts.

2.2. Popularity Bias

Popularity bias is a well-known challenge in recommender systems, where the system disproportionately favours popular items over less popular or new items. This bias stems from the tendency of recommendation algorithms to promote items that have been highly rated or frequently interacted with by many users, leading to a feedback loop where popular items become even more prominent, while less popular items are often ignored (Abdollahpouri, Burke, and Mobasher, 2017).

The root of popularity bias lies in the collaborative filtering mechanisms commonly used in recommender systems. Since these algorithms rely heavily on historical user-item interaction data, items with a large amount of interactions (e.g., ratings or clicks) are more likely to be recommended than those with limited interactions. As a result, popular items, which have already garnered a lot of user attention, are continuously recommended, while niche or new items receive far fewer recommendations, limiting their exposure (Adomavicius and Kwon, 2011).

While popularity bias can be beneficial in some cases, such as recommending blockbuster movies or bestselling products, it can negatively affect both the user

experience and the overall fairness of the system. From a user perspective, this bias reduces the diversity of recommendations, as users are repeatedly exposed to a small subset of well-known items. For businesses and content creators, this bias can hinder the discovery of new or niche products, thereby limiting the exposure of items that might otherwise match specific user preferences (Celma and Cano, 2008).

Addressing popularity bias is particularly important in domains where diversity and fairness are critical. For instance, in music or movie recommendation systems, there is value in helping users discover lesser-known artists or films that align with their unique tastes. In these cases, an overly biased system towards popular items could reduce user satisfaction, as they are not provided with a broad enough selection of content to explore (Brynjolfsson, Hu, and Smith, 2006). Furthermore, popularity bias has the potential to perpetuate inequalities in content exposure, particularly for newer creators or smaller businesses who may not have the visibility enjoyed by more established players.

Several strategies have been proposed in the literature to mitigate the effects of popularity bias. One approach is to introduce debiasing methods, which adjust the recommendation process to give more visibility to less popular items. Techniques such as *ErpMul* (Gantner et al., 2011), *ErpAug* (Abdollahpouri et al., 2017), *VAR* (Chen et al., 2018), and *XQUAD* (Vargas and Castells, 2011) have been shown to improve recommendation diversity and fairness by incorporating measures that promote less popular items without significantly compromising recommendation accuracy. These methods aim to balance the trade-off between relevance and diversity, ensuring that users are exposed to a wider variety of items, including those that may not have initially been highly visible.

Popularity bias is also intertwined with the issue of long-tail items, which are products that cater to more specific or niche tastes and therefore have lower overall demand. While traditional recommendation systems tend to emphasize head items (those that are highly popular), reducing popularity bias can increase the visibility of long-tail items, thereby improving the diversity and personalization of recommendations (Park and Tuzhilin, 2008). Balancing the promotion of long-tail items with maintaining recommendation accuracy remains an ongoing challenge in the field of recommender systems.

In summary, popularity bias is a significant obstacle to fairness and diversity in recommender systems. While promoting popular items may lead to short-term gains in user satisfaction and business performance, addressing this bias is crucial for creating systems that can offer a diverse range of recommendations. This study explores the effects of both profile injection attacks and traditional debiasing methods on reducing popularity bias, contributing to the ongoing discourse on fairness in recommendation systems.

2.3. Profile Injection Attacks

Profile injection attacks, also known as shilling attacks, pose a significant threat to the integrity of recommender systems. These attacks involve the deliberate insertion of fake user profiles or manipulation of existing ones with the intent to alter the recommendations generated by the system. Attackers can use these strategies to either promote or demote specific items, which can severely compromise the fairness, accuracy, and overall reliability of the recommendation process (Lam and Riedl, 2004).

Profile injection attacks can be categorized into two primary objectives: push attacks and nuke attacks. Push attacks aim to increase the likelihood of a particular item being recommended by artificially inflating its popularity. In contrast, nuke attacks seek to diminish an item's visibility by lowering its ranking in the recommendation list (Burke, Mobasher, and Williams, 2006). Although both types of attacks exploit the weaknesses of recommendation algorithms, their specific goals and tactics differ based on whether they are trying to promote or penalize items.

Several types of profile injection attacks have been studied in the literature, each with distinct characteristics and strategies.

The Average Attack, is one of the most basic and well-known forms of profile injection attacks. In this type of attack, fake user profiles are generated with ratings that reflect the average ratings of legitimate users, except for the targeted item, which receives an artificially inflated or deflated rating. This strategy makes the fake profiles blend into the overall user population, making them harder to detect (Lam and Riedl, 2004).

The Bandwagon Attack, exploit the popularity of certain items to mask the presence of injected profiles. Attackers generate profiles that rate highly popular items alongside the targeted item, leveraging the fact that many legitimate users are likely to rate these popular items as well by associating the fake profiles with well-known items, the attacker aims to increase the credibility of their profiles and manipulate the recommendation process without raising suspicion (Mobasher, Burke, and Bhaumik, 2007).

The Random Attack, generate profiles with random ratings for non-targeted items while giving extreme ratings to the targeted item. Although this approach is less subtle than average or bandwagon attacks, it is often easier to implement. However, due to its randomness, it is more easily detectable by robust systems that monitor for anomalous behaviour (Lam and Riedl, 2004).

The effectiveness of profile injection attacks largely depends on the characteristics of the recommendation algorithm being targeted. Collaborative filtering algorithms, especially those based on user-item interactions, are particularly vulnerable to such attacks, as they heavily rely on user-provided data to generate recommendations. The

inclusion of fake profiles can distort the algorithm's perception of user preferences, leading to biased or inaccurate recommendations (O'Mahony et al., 2004).

In response to the growing threat of profile injection attacks, several attack detection and prevention mechanisms have been proposed in the literature. These mechanisms typically focus on identifying anomalous profiles by detecting inconsistencies in user behaviour, such as unusually high or low ratings for certain items or an abnormal rating pattern across many items (Mehta and Nejdl, 2008). Machine learning-based techniques have also been employed to classify and filter out suspicious profiles before they can influence the recommendation process.

Despite the advancements in attack detection, profile injection attacks continue to present a significant challenge to the robustness of recommender systems. This thesis specifically explores how profile injection attacks, such as Average, Bandwagon, and Random attacks, affect popularity bias in recommender systems. By examining the interplay between these attacks and the recommendation process, this research aims to provide insights into both the vulnerabilities of current systems and potential strategies for mitigating these risks.

Recent studies have also explored the possibility of using adversarial attacks, traditionally seen as harmful, for constructive purposes such as debiasing. For instance, Wu et al. (2022) introduced the concept of a debiasing backdoor attack, where backdoor attacks are used benignly to eliminate data bias in recommendation systems. While this approach uses backdoor attacks, it presents a parallel to how profile injection attacks could potentially be leveraged to address issues such as popularity bias by manipulating system behaviour in a way that promotes fairness and diversity.

2.4. Contributions of Previous Studies to the Thesis

The body of research on recommender systems, popularity bias, and profile injection attacks has provided a strong foundation for this thesis. Previous studies have explored various aspects of recommendation algorithms, adversarial attacks, and debiasing methods, offering valuable insights into the challenges and potential solutions for improving fairness and accuracy in recommendation systems. This thesis builds upon these contributions by focusing specifically on the interaction between profile injection attacks and popularity bias and how these attacks can be repurposed as tools for debiasing rather than as harmful manipulations.

One of the most influential areas of prior research is the extensive work on popularity bias in recommender systems. Studies by Abdollahpouri, Burke, and Mobasher (2017) have highlighted how popularity bias consistently skews recommendation outcomes towards highly popular items, thus limiting diversity and fairness. This thesis draws heavily on these findings, as it seeks to address this very issue by using profile injection attacks to potentially reduce the dominance of popular items.

The literature on debiasing techniques such as ErpMul (Gantner et al., 2011), ErpAug (Abdollahpouri et al., 2017), VAR (Chen et al., 2018), and XQUAD (Vargas and Castells, 2011) further informs the analysis by providing benchmark methods for comparison. These debiasing strategies have been widely studied in their ability to reduce popularity bias, and this thesis extends the conversation by exploring how profile injection attacks might offer a complementary or alternative approach.

Another critical contribution from prior research comes from the work on profile injection attacks. Studies by Lam and Riedl (2004) and Mobasher, Burke, and Bhaumik (2007) have extensively documented the nature and impact of such attacks on recommender systems, identifying Average, Bandwagon, and Random attacks as some of the most common and effective types. While these studies primarily focus on the detrimental effects of these attacks—such as reducing recommendation accuracy or causing security vulnerabilities—this thesis reimagines these attacks in a positive light, exploring their potential for reducing popularity bias. This shift in perspective is one of the key innovations of this thesis, as it seeks to repurpose adversarial techniques for constructive outcomes.

The work by Wu et al. (2022) on debiasing backdoor attacks provides further inspiration for this thesis. In their study, backdoor attacks—traditionally seen as harmful—are used as a debiasing mechanism to eliminate data bias and improve recommendation fairness. This research parallels the core idea of this thesis, where profile injection attacks are explored as tools to mitigate popularity bias. Wu et al.'s concept of a "benign attack" introduces a fresh approach to adversarial strategies, one that this thesis seeks to expand by applying the same logic to profile injection attacks.

The extensive research on collaborative filtering algorithms, which are highly susceptible to manipulation through profile injection attacks, has also significantly influenced this thesis. The vulnerabilities of collaborative filtering systems to such attacks have been well documented by O'Mahony et al. (2004) and others, who identified how profile injection attacks distort the recommendation process. By leveraging this understanding, this thesis examines whether these same vulnerabilities can be used strategically to improve recommendation diversity and reduce bias, offering a novel application of these adversarial techniques.

Finally, the attack detection and prevention mechanisms proposed by Mehta and Nejdil (2008) and other scholars provide important context for understanding the challenges that recommender systems face in defending against profile injection attacks. While these studies focus on identifying and mitigating malicious attacks, this thesis proposes that controlled and intentional profile injections could serve as a viable method for counteracting bias within the system, without compromising overall system integrity.

In summary, the previous research on popularity bias, profile injection attacks, debiasing methods, and collaborative filtering algorithms has laid the groundwork for this

thesis. By integrating these contributions, this thesis offers a unique perspective on how adversarial techniques can be leveraged to reduce bias and improve fairness in recommender systems. The exploration of profile injection attacks as a potential debiasing strategy not only builds upon the findings of prior research but also introduces an innovative approach to addressing a longstanding issue in the field.

2.5. Summary and Research Questions

The literature reviewed in this chapter has provided a comprehensive overview of key topics relevant to this thesis, including recommender systems, popularity bias, profile injection attacks, and debiasing methods. The studies on recommender systems and algorithms have highlighted the widespread use and importance of these systems in various digital platforms. Collaborative filtering, content-based filtering, and hybrid approaches have emerged as the most commonly employed methods, with each offering distinct advantages and challenges. However, one persistent issue across most recommendation algorithms is the prevalence of popularity bias, which skews recommendations towards highly popular items at the expense of diversity and fairness.

Several studies have explored strategies to mitigate popularity bias. Debiasing methods such as ErpMul, ErpAug, VAR, and XQUAD have been proposed as effective techniques to reduce the impact of popularity bias by promoting less popular items without significantly sacrificing accuracy. At the same time, research on profile injection attacks has demonstrated the vulnerability of recommender systems, particularly collaborative filtering algorithms, to adversarial manipulation. Average, Bandwagon, and Random attacks have been shown to distort recommendation outcomes, raising concerns about system reliability and fairness. However, this thesis explores a new perspective on these attacks by examining their potential to reduce popularity bias when applied strategically and constructively.

A key contribution of this thesis is the examination of profile injection attacks not as harmful manipulations, but as a potential debiasing strategy. The work by Wu et al. (2022) on debiasing backdoor attacks provides an important conceptual foundation for this idea, demonstrating how traditionally adversarial methods can be repurposed to correct bias in machine learning models. Similarly, this thesis seeks to explore whether profile injection attacks can be used to address popularity bias in recommender systems, thereby contributing to the broader discussion on fairness and diversity in algorithmic recommendations.

Given the gaps identified in the literature, this thesis seeks to explore several key research questions aimed at addressing the impact of profile injection attacks and debiasing methods on recommender systems.

The first set of questions examines the impact of profile injection attacks on popularity bias. Specifically, it investigates how different types of attacks, such as

Average, Bandwagon, and Random, influence the level of popularity bias in recommender systems. Additionally, it considers whether these attacks can be used strategically to reduce popularity bias and the extent to which they can help promote less popular items within recommendation lists.

How do different profile injection attacks, such as Average, Bandwagon, and Random, influence the level of popularity bias in recommender systems?

Can these attacks be used strategically to reduce popularity bias, and to what extent can they help in promoting less popular items?

The second focus of this research is on the effectiveness of established debiasing methods, including ErpMul, ErpAug, VAR, and XQUAD, in mitigating popularity bias within the VAECF recommendation algorithm. A critical part of this inquiry is comparing the performance of these debiasing methods to that of the best-performing profile injection attack in terms of reducing popularity bias.

How effective are established debiasing methods, such as ErpMul, ErpAug, VAR, and XQUAD, in mitigating popularity bias in the VAECF recommendation algorithm?

How do these debiasing methods compare to the best-performing profile injection attack in terms of reducing popularity bias?

Finally, the research evaluates the overall performance of the recommendation system when subjected to profile injection attacks and debiasing methods. This includes analysing how these approaches impact key metrics such as recommendation accuracy, diversity, and fairness, providing a comprehensive understanding of their effects on system functionality and user experience.

How do profile injection attacks and debiasing methods affect the overall performance of the recommender system, particularly in terms of recommendation accuracy, diversity, and fairness?

By addressing these research questions, this thesis aims to provide a deeper understanding of the potential role that profile injection attacks can play in mitigating popularity bias, as well as how these attacks compare with traditional debiasing techniques. The findings will contribute to the ongoing effort to develop fairer and more diverse recommender systems, ensuring that all items—not just the most popular ones—have the opportunity to be recommended to users.

3. MATERIALS AND METHODS

This chapter outlines the experimental framework used in this thesis to investigate the impact of profile injection attacks and debiasing methods on popularity bias in recommender systems. It describes the dataset, the recommendation algorithm employed, the profile injection attacks applied, and the evaluation metrics used to assess system performance. The objective of this chapter is to provide a comprehensive overview of the methods and materials used to systematically evaluate the research questions posed in this study.

3.1. Dataset

The experiments conducted in this thesis are based on the MovieLens dataset (Harper & Konstan, 2015), one of the most widely used datasets in recommender system research. MovieLens contains user ratings for a large collection of movies, which makes it ideal for evaluating both the recommendation performance and the effects of popularity bias.

The dataset includes user ratings on a scale from 1 to 5, movie metadata such as titles and genres, and user profile information like user IDs and movie IDs. The specific version of the MovieLens dataset used in this study contains 1,000,209 ratings from 6,040 users across 3,706 movies. This version, often referred to as MovieLens-1M (version details: 1 million), is particularly well-suited for this research due to its rich interaction data and its widespread use in the literature, which ensures comparability with previous studies on recommender systems and bias.

3.2. Recommendation Algorithm

In this thesis, the Variational Autoencoder for Collaborative Filtering (VAECF) algorithm is employed as the primary recommendation model. VAECF is a deep learning-based method that extends traditional collaborative filtering by utilizing variational autoencoders (VAEs) to model complex user-item interaction patterns through latent variables. Unlike traditional collaborative filtering, which typically depends on user-item interaction matrices, VAECF captures non-linear relationships between users and items, modelling hidden factors that represent user preferences and item characteristics (Liang et al., 2018).

Variational autoencoders are generative models that map input data into a probabilistic latent space, which is then used to generate outputs. In the context of recommendation systems, VAECF models user-item interactions as a generative process, where the latent variables capture the hidden preferences of users. These latent variables are learned by optimizing a variational lower bound, allowing the model to efficiently infer user preferences even in the presence of sparse or incomplete data. This makes VAECF particularly suitable for addressing challenges such as data sparsity and the cold-start problem, which are common in recommender systems.

The VAE CF model architecture consists of two main components: an encoder and a decoder. The encoder maps the input (user-item interaction matrix) into a latent space that represents user preferences, using a neural network to output a distribution over the latent variables, typically represented by a mean and variance vector. The decoder then samples from this latent space and reconstructs the user's interaction history, generating predictions for the user's potential interactions with items. The model is trained by minimizing a combination of reconstruction loss, which ensures the predicted interactions match actual interactions, and Kullback-Leibler divergence, which regularizes the latent variable distribution. This optimization process allows VAE CF to learn compact, informative representations of user preferences, enabling it to generate accurate personalized recommendations.

VAE CF offers several advantages over traditional collaborative filtering algorithms. First, its ability to learn a probabilistic latent space allows it to model user preferences effectively even in cases where user interaction data is sparse. This capability is particularly useful in situations with new users or items that have limited interaction history. Second, VAE CF's ability to capture non-linear relationships between users and items leads to improved recommendation accuracy compared to standard matrix factorization techniques. Additionally, the generative nature of VAE CF enables it to predict user preferences even in the absence of direct user-item interactions, making it robust in scenarios where data is missing or incomplete. Finally, VAE CF's deep learning framework allows it to model complex, high-dimensional relationships between users and items, producing more nuanced and personalized recommendations.

In this thesis, VAE CF serves as the foundational recommendation model for evaluating the effects of both profile injection attacks and debiasing methods on popularity bias. VAE CF provides a consistent evaluation framework, as it represents a state-of-the-art recommendation algorithm widely used in modern recommendation systems. The experiments conducted in this thesis use VAE CF to assess how different profile injection attacks, such as Average, Bandwagon, and Random attacks, affect the level of popularity bias in recommendations. Moreover, the VAE CF algorithm is used to evaluate the effectiveness of established debiasing methods like ErpMul, ErpAug, VAR, and XQUAD in mitigating popularity bias. Additionally, the algorithm enables an analysis of the trade-offs between recommendation accuracy, diversity, and fairness under different attack and debiasing scenarios.

By employing VAE CF, this study aims to provide insights into the interaction between adversarial attacks and debiasing strategies, contributing to the development of fairer and more diverse recommendation systems.

3.3. Profile Injection Attacks

Profile injection attacks, also referred to as shilling attacks, are adversarial strategies that involve the creation of fake user profiles with the intent to manipulate the

output of a recommender system. These attacks are typically used to promote or demote specific items, which can severely distort the recommendations generated by the system. In this thesis, however, profile injection attacks are explored from a different perspective: as potential tools to reduce popularity bias rather than their conventional malicious use.

There are three primary types of profile injection attacks studied in this thesis: Average Attack, Bandwagon Attack, and Random Attack. Each of these attacks operates by generating fake user profiles that interact with the recommendation system in specific ways to influence the prominence of certain items.

The Average Attack generates profiles that closely mimic the behaviour of real users by assigning average ratings to non-targeted items while giving extreme ratings to the targeted item. This makes the injected profiles blend into the user base, making them difficult to detect. The attack's goal is to manipulate the recommendation system into promoting the targeted items, which in this thesis focuses on less popular items, with the aim of mitigating popularity bias.

The Bandwagon Attack takes advantage of the popularity of certain well-known items. Fake profiles are created by assigning high ratings to both popular items and the targeted item, leveraging the fact that many legitimate users are likely to rate popular items. This strategy increases the credibility of the fake profiles and encourages the system to also recommend the targeted, less popular items.

The Random Attack is less sophisticated, as it assigns random ratings to non-targeted items while giving extreme ratings to the targeted item. This randomness makes the profiles easier to detect but still has the potential to influence the recommendation system by promoting less popular items, thus addressing popularity bias to some extent.

The following table provides a summary of the profile injection attacks used in this thesis, along with their descriptions and references.

Table 3.1. Brief descriptions of profile injection attacks

Attack Type	Description	Reference
Average Attack	Creates fake profiles with average ratings. These attacks can significantly reduce the recommendation accuracy of the system.	Lam & Riedl, 2004
Bandwagon Attack	Increases bias by giving high ratings to popular items, thereby reinforcing popularity bias.	Mobasher et al., 2007
Random Attack	Creates fake profiles with random values, jeopardizing the accuracy and security of the system.	Lam & Riedl, 2004

In this thesis, these profile injection attacks are studied in the context of the VAECF recommendation algorithm to evaluate their potential for debiasing the system. By

artificially inflating the visibility of less popular items through fake profiles, the goal is to examine whether these attacks can be repurposed to promote diversity in recommendations and reduce the system's tendency to overly favour popular items.

3.4. Debiasing Methods

Popularity bias is a well-documented issue in recommender systems, where a small set of highly popular items dominate recommendations, reducing both the diversity and fairness of the system. To address this issue, various debiasing methods have been proposed in the literature. These methods aim to reduce the over-promotion of popular items and increase the visibility of less popular items, leading to a more diverse and fair recommendation output. In this thesis, four well-known debiasing methods are employed: ErpMul, ErpAug, VAR, and XQUAD. Each of these methods approaches the problem of popularity bias differently, providing a diverse set of strategies to mitigate this bias.

ErpMul is a debiasing method that works by adjusting the recommendation process to give less weight to popular items while maintaining the relevance of recommendations (Gantner et al., 2011). The method applies a multiplicative factor to penalize highly popular items, thus reducing their chance of being recommended frequently. By applying this correction, ErpMul helps promote less popular items without significantly compromising the accuracy of the recommendations.

ErpAug extends the ErpMul approach by augmenting the user-item interaction data with additional debiasing factors (Abdollahpouri et al., 2017). This method incorporates external information or domain-specific features to further reduce the influence of popularity bias. ErpAug seeks to balance the trade-off between relevance and diversity by enhancing the recommendation process with more granular control over how items are prioritized. It allows for greater flexibility in controlling bias while maintaining user satisfaction with the recommendations.

VAR (The Value-Aware Ranking) is a method that promotes diversity by adjusting the recommendation scores to account for item popularity variance (Chen et al., 2018). VAR encourages the system to recommend items that are underrepresented or less popular, improving the overall diversity of the recommendations. By focusing on reducing the gap between popular and less popular items, VAR ensures that a broader range of items is recommended, thereby addressing popularity bias more effectively.

XQUAD is a widely used diversification algorithm that aims to explicitly promote diversity in recommendation lists (Vargas and Castells, 2011). XQUAD re-ranks the recommendation list by considering both the relevance of items and their contribution to the overall diversity of the recommendations. The algorithm dynamically adjusts the recommendation process to ensure that less popular items are included in the top recommendations, making the recommendation list more diverse while maintaining a reasonable level of accuracy.

In this thesis, these four debiasing methods—ErpMul, ErpAug, VAR, and XQUAD—are applied to the VAE CF algorithm to evaluate their effectiveness in mitigating popularity bias. The performance of each method is assessed based on how well it promotes less popular items without significantly affecting recommendation accuracy. Additionally, the trade-offs between diversity, fairness, and accuracy are analysed to provide insights into the practical applicability of these debiasing techniques in real-world recommender systems.

The following table summarizes the key debiasing methods employed in this thesis, along with brief descriptions and references.

Table 3.2. Brief descriptions of popularity debiasing methods

Algorithm	Description	Reference
ErpMul	Aims to increase the diversity of items in the recommendation list to reduce popularity bias.	Gantner et al., 2011
ErpAug	Aims to reduce popularity bias by adding various items to user profiles.	Abdollahpouri et al., 2017
VAR	Aims to increase the variance in the recommendation list to reduce popularity bias.	Chen et al., 2018
XQUAD	Aims to increase the diversity and inclusiveness of the recommendation list to reduce popularity bias.	Vargas & Castells, 2011

By employing these debiasing methods, this thesis aims to compare their performance with profile injection attacks as potential tools for reducing popularity bias. The findings will contribute to a deeper understanding of how different approaches to debiasing can influence recommendation outcomes and whether adversarial techniques like profile injection attacks can offer a viable alternative to traditional debiasing strategies.

3.5. Evaluation Methods and Metrics

To comprehensively evaluate the performance of the VAE CF algorithm, as well as the effects of profile injection attacks and debiasing methods on popularity bias, this thesis employs several key metrics. These metrics are designed to assess different dimensions of recommender system performance, including relevance, diversity, and novelty, ensuring a balanced evaluation across multiple aspects. The chosen metrics include Normalized Discounted Cumulative Gain (nDCG), F1-Score, Average Percentage of Long-tail Items (APLT), and Novelty.

nDCG, is a widely used metric in recommender systems that measures the ranking quality of the recommended items (Järvelin and Kekäläinen, 2002). It evaluates how well the system ranks relevant items higher in the recommendation list. The Discounted Cumulative Gain (DCG) assigns higher relevance scores to items appearing at the top of the recommendation list, with a logarithmic discount applied to items appearing lower in

the list. nDCG normalizes this score to allow for comparison between different recommendation lists. This metric is particularly useful in this thesis as it helps evaluate the relevance of the recommendations while also considering the ranking position of relevant items.

F1-Score, is a harmonic mean of precision and recall, providing a balanced measure of a system's accuracy in recommending relevant items (Van Rijsbergen, 1979). Precision measures the proportion of recommended items that are relevant, while recall measures the proportion of relevant items that are recommended. F1-Score is valuable in situations where both precision and recall are important, as it balances the trade-off between the two. In this thesis, F1-Score helps assess how well the VAE CF algorithm balances relevance (precision) and coverage of relevant items (recall), especially when evaluating the effects of debiasing methods and profile injection attacks.

Average Percentage of Long-tail Items (APLT), is a diversity-focused metric that measures the proportion of recommended items that belong to the long tail of the item distribution (Anderson, 2006). Long-tail items are typically less popular and receive fewer interactions, yet they are crucial for promoting diversity in recommendation systems. A higher APLT indicates that the system is successfully recommending less popular items, which is a central goal in mitigating popularity bias. In this thesis, APLT is used to evaluate the effectiveness of debiasing methods and profile injection attacks in promoting less popular, long-tail items.

Novelty, is a metric that measures the degree to which recommended items are new or unfamiliar to users (Vargas & Castells, 2011). A recommendation system that frequently suggests items the user has not yet interacted with is considered more novel. Novelty is critical for ensuring that the system provides users with fresh content, rather than repeatedly recommending items they are already familiar with. This metric is particularly relevant in this thesis, as it assesses how well the system introduces novel items, especially those from the long tail, which are typically less known.

While each of the metrics mentioned above provides valuable insights into different aspects of system performance, there are inherent trade-offs between them. For instance, increasing diversity (as measured by APLT) or novelty may reduce accuracy, as the system suggests less familiar items that might not perfectly align with the user's immediate preferences. Similarly, efforts to reduce popularity bias might affect ranking quality (nDCG) or relevance (F1-Score). This thesis will analyse these trade-offs to provide a balanced view of how different debiasing methods and profile injection attacks impact the overall performance of the recommender system.

The following table provides a brief summary of the key performance metrics used in this thesis, along with their descriptions and references.

Table 3.3. Brief descriptions of performance metrics

Metric	Description	Reference
Normalized Discounted Cumulative Gain (nDCG)	Measures the accuracy and quality of the ranking of recommendations. Ensures that the most interesting items to users are placed at the top.	Järvelin & Kekäläinen, 2002
F1-Score	The harmonic mean of precision and recall. F1-Score evaluates the overall accuracy and effectiveness of the recommendation system.	Rijsbergen, 1979
Average Percentage of Long-tail Items (APLT)	Measures the percentage of long-tail items in the recommendation list. A higher APLT value indicates that the system provides more diverse recommendations.	Anderson, 2006
Novelty	Measures the ability to recommend items that the user has not seen before or that have low popularity.	Vargas & Castells, 2011

By employing these metrics—nDCG, F1-Score, APLT, and Novelty—this thesis provides a comprehensive evaluation of the recommender system’s performance. The combination of these metrics ensures that the results reflect not only the system’s ability to generate accurate and relevant recommendations but also its capacity to promote diversity and novelty, which are crucial for mitigating popularity bias.

3.6. Experimental Setup

The experimental setup of this thesis is designed to evaluate the impact of profile injection attacks and debiasing methods on the performance of the VAE CF recommendation algorithm. The experiments aim to assess how these techniques affect popularity bias, accuracy, diversity, and novelty in recommendations. The following paragraphs detail the main components of the experimental process.

The first step involves preparing the MovieLens dataset, a widely used dataset in recommender system research due to its extensive user-item interaction data. The dataset is processed using a 5-fold cross-validation method. In this approach, the dataset is divided into five subsets. For each fold, the system is trained on four subsets and tested on the remaining subset, ensuring that each subset is used as the test set once. The final performance metrics are averaged across all five folds to ensure that the results are not biased by any specific data split. Additionally, the dataset is split into 80% training data and 20% test data in each fold to ensure a consistent evaluation process.

Once the dataset is split into training and test sets, the experiments proceed in two separate scenarios. In the first scenario, debiasing methods from the literature are directly applied to the VAE CF algorithm. The Cornac library is used to implement VAE CF, with its parameters set according to the original paper (Liang et al., 2018). After training the

model on the 80% training set, the performance is evaluated on the 20% test set. The system generates a list of top- N items for each user (where $N = 10$), which are then evaluated using the metrics discussed in Section 3.5. The performance metrics—nDCG, F1-Score, Average Percentage of Long-tail Items (APLT), and Novelty—are calculated to determine the effectiveness of each debiasing method in mitigating popularity bias and promoting diverse recommendations.

In the second scenario, profile injection attacks are applied to the system as an alternative debiasing strategy. In this approach, the MovieLens dataset is first modified by injecting fake user profiles according to the specific attack strategy (Average, Bandwagon, or Random attack). After the attack is applied, the modified dataset is processed using the same 5-fold cross-validation method, with the data split into 80% training and 20% test sets. The VAE CF algorithm is then applied in the same manner as in the first scenario. Once the model is trained on the 80% training set, the system generates top-10 recommendations for each user, and the same performance metrics are used to evaluate the impact of the profile injection attacks on recommendation quality and bias reduction.

By structuring the experiments in this way, the thesis compares the effects of traditional debiasing methods and adversarial profile injection attacks on the VAE CF algorithm. The comparison of results provides insights into the trade-offs between different approaches and their ability to reduce popularity bias while maintaining accuracy, diversity, and novelty in the recommendations.

3.7. Dataset Analysis and Preparation

This section provides a detailed analysis of the MovieLens (ML) dataset used in this thesis. The dataset's statistical properties, long-tail distribution of item popularity, and data preparation processes are outlined. The focus is on understanding the key characteristics of the dataset and the methods used for splitting the data into training and test sets for the experiments.

The MovieLens (ML) dataset is widely used in recommender system research due to its extensive user-item interaction data. In this study, the ML-1M version of the dataset is used, containing ratings from 6,040 users on 3,706 items, with a total of 1,000,209 ratings. This dataset covers a broad range of movies, making it a solid foundation for evaluating the performance of recommendation algorithms, especially in the context of popularity bias and diversity challenges.

One of the key features of the MovieLens dataset is its long-tail distribution of item popularity. This distribution is crucial for understanding the presence of popularity bias in recommendation systems. A small number of highly popular items receive the majority of user ratings, while the vast majority of items (those in the long tail) receive

significantly fewer ratings. This imbalance poses a challenge for creating diverse recommendation lists, as the system may favour popular items over less popular ones.

Figure 3.1 illustrates the long-tail distribution of item popularity in the MovieLens dataset. On the X-axis, items are ranked in descending order of popularity, and on the Y-axis, the number of ratings each item has received is shown. The visual highlights that only a few items are rated frequently, while most items have much lower interaction levels. This distribution reveals the system's tendency to promote a small set of popular items, limiting the exposure of long-tail items.

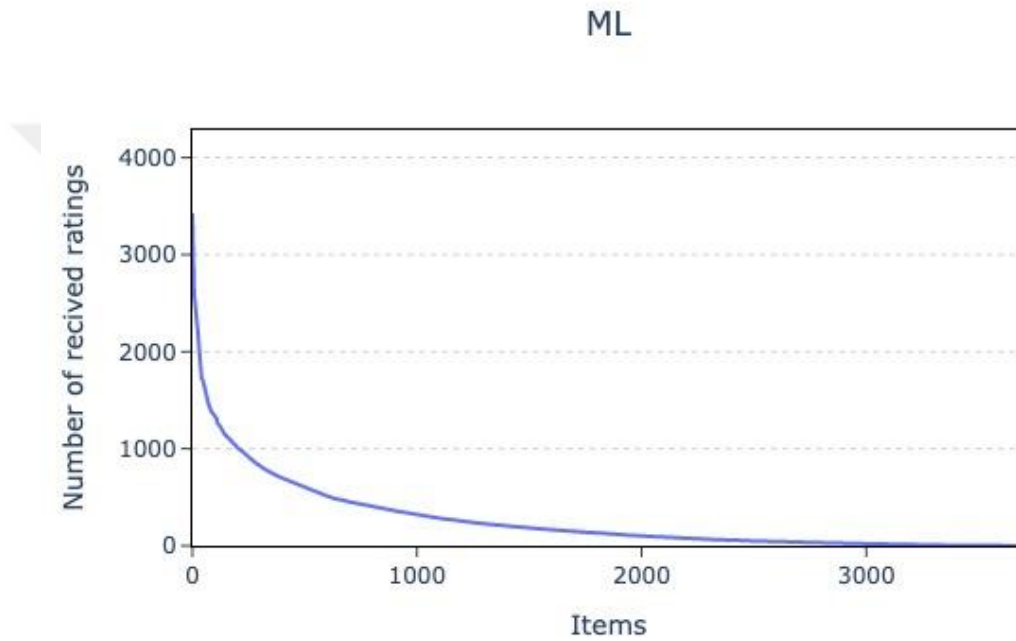


Figure 3.1. The long-tail distribution of item popularity in the MovieLens (ML) dataset

The dataset statistics table below presents a detailed breakdown of the fundamental characteristics of the MovieLens dataset. It includes the number of users, items, ratings, and other key metrics such as sparsity and average rating. These statistics provide valuable insights into the structure of the dataset, helping to identify challenges related to sparsity and the diversity of interactions.

Table 3.4. Statistics of the MovieLens (ML) dataset

Dataset	U	I	R	Sparsity	Avg_r	R / I	R / U
ML	6,040	3,706	1,000,209	0.955	3.582	269.889	165.598

The dataset has a sparsity of 0.955, meaning that only 4.5% of the possible user-item ratings are present. This high sparsity presents a challenge for the recommendation system, as it must predict user preferences based on limited data. The average rating is 3.582, indicating that users generally rate items positively. On average, each item receives around 269.889 ratings, while each user provides about 165.598 ratings, showcasing the density of user interactions.

To ensure a robust evaluation of the recommendation system, the dataset is split into 80% training data and 20% test data. This splitting method ensures that the system is trained on a substantial portion of the data while being evaluated on unseen interactions to test its generalization ability. In addition, 5-fold cross-validation is applied, providing a more reliable measure of performance. In this process, the dataset is divided into five subsets. For each fold, the system is trained on four subsets, and the remaining subset is used as the test set. The final performance metrics are averaged across all five folds. The table below provides an overview of the cross-validation process.

Table 3.5. Cross-validation process overview

Experiment	Training Set (80%)	Test Set (20%)	Evaluation Metrics
Experiment 1	Fold 2 + Fold 3 + Fold 4 + Fold 5	Fold 1	nDCG, F1-Score, APLT, Novelty
Experiment 2	Fold 1 + Fold 3 + Fold 4 + Fold 5	Fold 2	nDCG, F1-Score, APLT, Novelty
Experiment 3	Fold 1 + Fold 2 + Fold 4 + Fold 5	Fold 3	nDCG, F1-Score, APLT, Novelty
Experiment 4	Fold 1 + Fold 2 + Fold 3 + Fold 5	Fold 4	nDCG, F1-Score, APLT, Novelty
Experiment 5	Fold 1 + Fold 2 + Fold 3 + Fold 4	Fold 5	nDCG, F1-Score, APLT, Novelty

In each fold, 80% of the data is used for training, and the remaining 20% is used for testing. The system's performance is evaluated using metrics such as nDCG, F1-Score, APLT, and Novelty, and the results from each fold are averaged to provide a robust evaluation.

In experiments involving profile injection attacks, the MovieLens dataset is first modified by injecting fake user profiles designed to manipulate the recommendation process. These profiles are added to the training data, and then the system undergoes the same 5-fold cross-validation process. This ensures that the experiments are conducted in a consistent manner, allowing for a fair comparison between traditional debiasing methods and adversarial profile injection attacks.

3.8. Procedure for Applying Profile Injection Attacks

This section describes the methodology for applying profile injection attacks to the MovieLens dataset. Profile injection attacks, or shilling attacks, are used to manipulate recommendation algorithms by injecting fake user profiles into the system. In this study, a self-attack scenario is employed, where there is no specific targeted item. Instead, the goal is to influence the overall accuracy and fairness of the recommendation system, exploring how profile injection attacks affect its performance holistically.

The general form of attack profiles in recommendation systems involves manipulating user profiles to achieve specific objectives, such as promoting (push) or demoting (nuke) certain items. These profiles typically include selected items (I_S), filler items (I_F), unrated items ($I_\emptyset$), and targeted items (I_T) (Figure 3.2) (Gunes et al., 2014). The attack profiles can vary in complexity and are designed to exploit the weaknesses of recommendation algorithms. Below is an outline of the general form of attack profiles and their components:

Selected Items (I_S), are a set of items that define the characteristics of the attack. These can be popular items or items with specific features that the attacker wants to manipulate. The purpose of including selected items is to create a profile that blends well with genuine user profiles, making the attack less detectable.

Filler Items (I_F), are randomly chosen items that help disguise the malicious profile. These items are selected to fill up the profile and make it look more realistic. The selection of filler items is usually uniform and excludes the target item. The ratings assigned to these items can vary depending on the type of attack being performed.

Unrated Items ($I_\emptyset$), are those that remain unscored in the fake profile. Including unrated items helps mimic genuine user behaviour, as real users often do not rate every item they interact with. This aspect further aids in the disguise of the attack profile.

Targeted Item (I_T), is the focal point of the attack. In a push attack, the targeted item receives the highest possible rating to increase its recommendation probability. Conversely, in a nuke attack, the targeted item is given the lowest possible rating to decrease its recommendation probability.

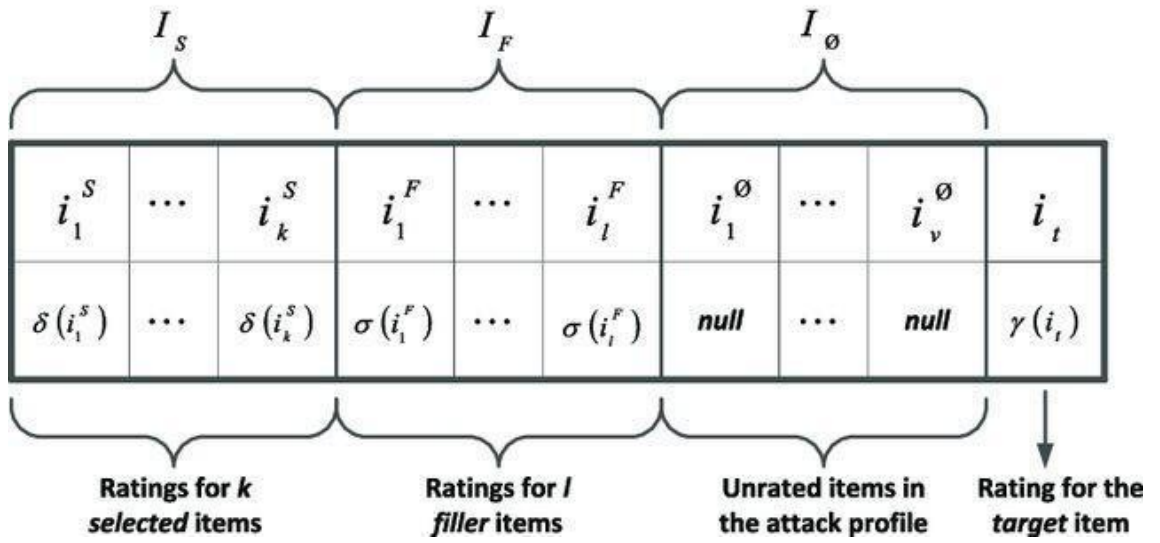


Figure 3.2. General form of attack profiles

The following table summarizes different types of attack profiles, their components, and the ratings assigned to each component.

Table 3.6. Summary of attack profiles

Attack Type	I_S		I_F		$I_\emptyset$	I_T (push/nuke)
	Items	Rating	Items	Rating		
Random Attack	null		randomly chosen	system mean	null	r_{max} / r_{min}
Average Attack	null		randomly chosen	item mean	null	r_{max} / r_{min}
Bandwagon Attack	popular items	r_{max} / r_{min}	randomly chosen	system mean	null	r_{max} / r_{min}

Each attack profile is meticulously designed to exploit the weaknesses of recommendation systems, such as over-reliance on user-provided ratings, and to influence the recommendation outcomes based on specific goals, such as pushing or nuking certain items.

The implementation of profile injection attacks involves adding fake user profiles to the user-item matrix. The user-item matrix represents ratings that users (rows) give to items (columns) depicted in Table 3.7. The injected fake profiles aim to influence the recommendation system by providing manipulated ratings for specific items.

The process of updating the user-item matrix with fake profiles begins with the creation of fake user profiles. These profiles are carefully structured based on the selected attack strategy, which typically involves defining targeted items, filler items, and unrated

items. The targeted items are those that the attack intends to promote or demote, while filler items are included to make the fake profiles appear realistic and less detectable.

Once the fake profiles are created, they are integrated into the user-item matrix by adding them as new rows. Ratings for the selected and filler items are then inserted into the corresponding cells of the matrix. This step ensures that the fake profiles are effectively embedded within the system's data, allowing them to influence the recommendation algorithm.

Finally, the recommendation algorithm is trained on the updated user-item matrix, which now includes the fake profiles. The ratings introduced by these fake profiles alter the algorithm's understanding of user-item relationships. As a result, the system's recommendation outcomes are influenced, with targeted items being either promoted or demoted, depending on the specific goals of the attack strategy. This manipulation demonstrates the potential impact of profile injection attacks on recommender systems.

By using a self-attack scenario, the attack does not focus on promoting or demoting any specific item but rather alters the system's overall recommendation patterns.

Table 3.7. User-Item matrix

User \ Item	Item1	Item2	Item3	Item4	Item5	Item6
User1	5	3	0	1	1	2
User2	4	0	0	1	2	1
User3	1	1	0	5	0	2
User4	0	3	4	4	1	3
FakeUser1	2	4	5	3	5	1
FakeUser2	3	3	2	1	0	0

The manipulated ratings from fake profiles affect how the recommendation system perceives certain items, which can lead to these items being recommended more or less frequently depending on the attack type.

Several parameters are crucial in shaping the impact of the profile injection attacks. The parameters used in this study include filler size, attack size, and selected size, which control the number of fake profiles and how they are structured.

Filler size represents the percentage of filler items in each fake profile. In this study, we use filler sizes of 3%, 5%, and 10%. A small filler size makes the fake profiles sparser and potentially less detectable but may reduce their overall impact. A larger filler size increases the realism of the profiles and their ability to influence the system, but it also increases the risk of detection.

Attack size refers to the percentage of fake profiles added to the user-item matrix. Attack sizes of 3%, 5%, and 10% are tested. A smaller attack size minimizes detectability

but has less influence on the system, while a larger attack size provides more manipulation power at the cost of a higher risk of detection.

Selected size refers to the proportion of items chosen to shape the fake profiles. A 20% selected size is used in this study, ensuring that a significant portion of items is included to blend the fake profiles with real user behaviour, increasing the attack's realism.

By configuring these parameters, the self-attack scenario can be tested in different conditions to assess its effect on the recommendation system's overall performance.

Although there is no specific targeted item in the self-attack scenario, the selection of filler items plays a crucial role in assessing how the attack affects the system's fairness and bias. In this study, filler items are selected from the long-tail (less popular) items. Popularity bias refers to the tendency of recommendation systems to favour popular items, which limits the exposure of less popular, long-tail items.

By choosing long-tail items as filler items in the fake profiles, the self-attack can help reduce popularity bias by increasing the visibility of these items in the recommendation lists. This approach promotes diversity in recommendations and challenges the system's tendency to over-promote popular items. In this way, the self-attack scenario not only tests the system's robustness against profile injection attacks but also evaluates its potential as a strategy to improve fairness and reduce bias in the recommendation process.

Long tail distribution visuals allow us to examine the distribution of popularity within recommendation systems. Especially in large datasets, these visuals help us understand how popular and lesser-known items (the long tail) are distributed across the system. In these distributions, popular items appear at the beginning of the graph, while long-tail items—those less frequently consumed—occupy the remaining portion. This distribution is crucial for analysing whether a recommendation system provides diversity and fairness; otherwise, continually highlighting popular items can limit the discovery of new and niche content. Long tail distribution visuals reveal the extent to which a recommendation system faces popularity bias and enable a visual evaluation of how effectively applied debiasing or attack models can mitigate this bias.

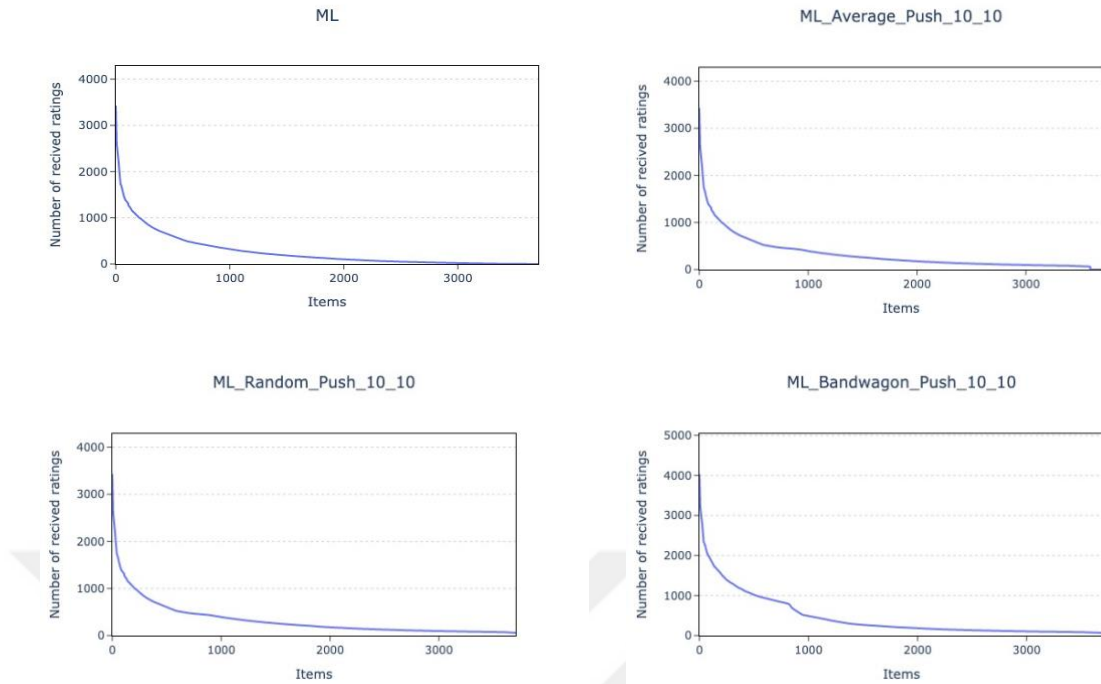


Figure 3.3. Long tail distribution of MovieLens dataset under different attack models

As shown in Figure 3.3, these visuals demonstrate the degree of popularity bias and the impact of different attack models on recommendation systems. The base long-tail distribution of the MovieLens dataset shows that popular items dominate while niche items in the long tail have limited visibility, highlighting the presence of popularity bias and the restricted capacity of the recommendation system to provide diverse and fair content.

When comparing the effects of different attack models, Random Attack proves to be the most effective in increasing the visibility of long-tail items, thus reducing popularity bias. Average Attack shows moderate effectiveness in mitigating this bias, whereas Bandwagon Attack tends to amplify the bias by promoting popular items even further. These comparisons visually demonstrate how attack models influence popularity bias, providing guidance for developing fairer and more diverse recommendation systems.

3.9. Procedure for Applying Recommendation Algorithm

This section outlines the application of the VAE CF recommendation algorithm on the MovieLens dataset. The Variational Autoencoder for Collaborative Filtering (VAECF) is a sophisticated model that leverages user-item interaction data to generate personalized recommendations. In this study, the algorithm is implemented using the

Cornac library, and the original hyperparameters proposed in the algorithm’s paper are applied to ensure consistency and optimal performance.

The VAE CF algorithm is a probabilistic model designed to learn latent representations of both users and items from sparse interaction data. By representing users and items in a shared latent space, the model can predict missing interactions, making it well-suited for datasets like MovieLens, where most user-item pairs are unrated. The algorithm maximizes the likelihood of the observed data while regularizing the learned latent factors to prevent overfitting. This enables the system to generate high-quality, personalized recommendations.

Before applying the VAE CF algorithm, the MovieLens dataset is split into 80% training data and 20% test data, as outlined in the Experimental Setup section. To further ensure the robustness of the results, 5-fold cross-validation is used. In each fold, the training data is used to train the VAE CF model, while the test data evaluates its performance on unseen interactions. This process ensures that the model’s generalization ability is thoroughly tested.

The implementation of the VAE CF model in this study utilizes the Cornac library, adhering to the hyperparameters suggested in the original paper. These hyperparameters, including the number of latent factors, learning rate, and regularization coefficients, are carefully optimized for collaborative filtering tasks, ensuring robust performance with the MovieLens dataset.

The process begins with the initialization of the VAE CF model. The Cornac library is employed to configure the model using the predefined hyperparameters from the original research. This setup ensures that the model is tailored to the characteristics of the MovieLens dataset, allowing it to effectively capture the underlying user-item interaction patterns.

During the training phase, the model optimizes the variational lower bound. This involves balancing the reconstruction of user-item interactions with the regularization of latent factors. As the training progresses, the model iteratively refines the latent representations of users and items, aiming to minimize the error between predicted ratings and actual user feedback. This step is critical for ensuring the accuracy of the recommendations.

Once the training is complete, the VAE CF model generates top-N recommendations for each user, where N is set to 10 in this study. The model predicts ratings for items and ranks them in descending order of relevance. The top 10 items for each user are selected based on these predicted ratings, forming the recommendation list. These recommendations are then used for further evaluation, assessing the model’s effectiveness in terms of accuracy, diversity, and fairness.

The VAE CF model’s performance is evaluated using the test set, which contains unseen interactions. The model is assessed based on the performance metrics outlined in Section 3.5, including nDCG, F1-Score, APLT, and Novelty. These metrics provide a comprehensive view of how well the model balances accuracy, diversity, and fairness in its recommendations.

The use of cross-validation ensures that the evaluation is reliable and robust. The results from each fold are averaged to provide a final assessment of the model’s performance.

The VAE CF model is evaluated across several distinct experimental scenarios to thoroughly assess its robustness and performance.

In the Baseline Scenario, the model is trained and evaluated without incorporating any additional debiasing methods or profile injection attacks. This setup provides a foundational performance metric, serving as a reference point for comparisons with other experimental scenarios. It highlights the unaltered behaviour of the VAE CF model under standard conditions.

The Debiasing Scenario introduces various debiasing methods into the VAE CF model. After applying these techniques, the model is retrained and re-evaluated. This scenario allows for the measurement of how debiasing methods influence key performance metrics, such as system accuracy, diversity, and fairness. By comparing these results with the baseline, the effectiveness of the debiasing approaches in mitigating popularity bias can be analysed.

In the Profile Injection Attack Scenario, the VAE CF model is tested in the presence of profile injection attacks, as detailed in Section 3.8. This scenario focuses on evaluating the system’s robustness against adversarial manipulations. By analysing the impact of these attacks on recommendation performance, this scenario provides insights into how the system responds to strategic adversarial behaviour and how such attacks influence the accuracy, diversity, and fairness of the recommendations.

For each scenario, the model’s ability to generate accurate, diverse, and fair recommendations is carefully measured and compared, providing insights into the effectiveness of the VAE CF algorithm under varying conditions.

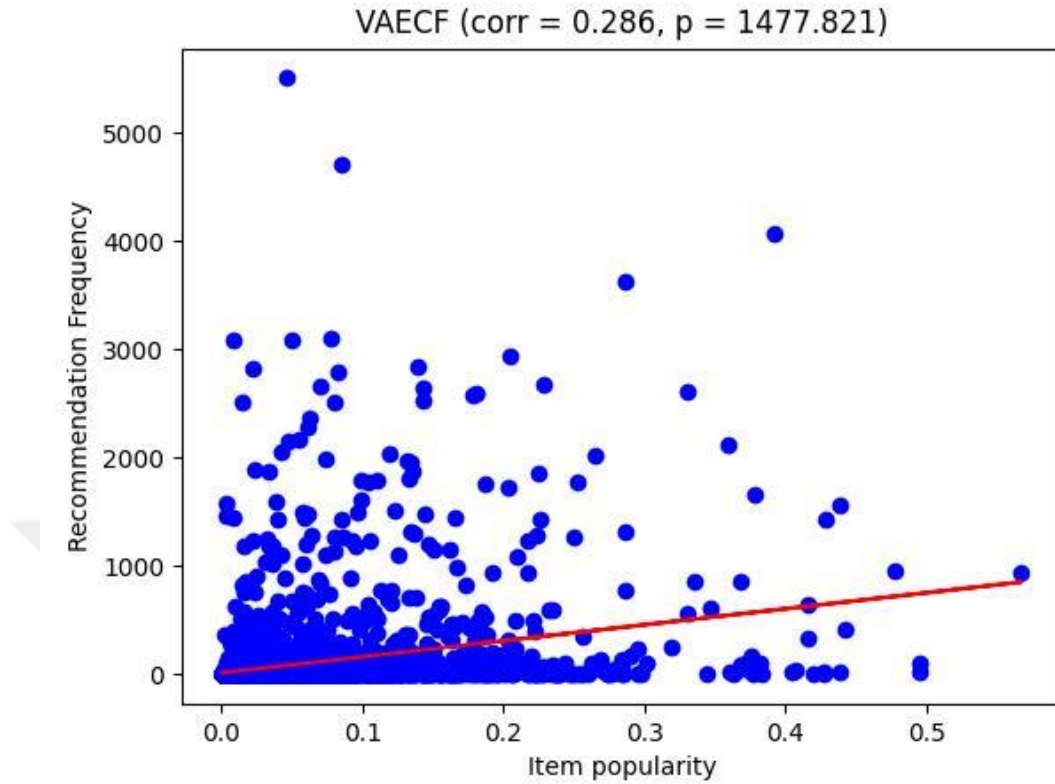


Figure 3.4. Correlation between item popularity and recommendation frequency for VAECF algorithm

Figure 3.4 illustrates the correlation between item popularity and recommendation frequency, analysing the relationship between how often items are rated and how frequently they appear in the top-10 recommendation lists generated by the recommendation system. This visualization sheds light on how recommendation algorithms tend to favour popular items, potentially propagating popularity bias. By examining the recommendation frequency of items with varying levels of popularity, this visualization provides valuable insights into how recommendation algorithms influence user experience and the fairness of content exposure.

Key insights from the visualization highlight various patterns in the recommendation system's behaviour.

The Bottom Left Region represents items that are seldom rated and seldom recommended. These items are largely overlooked by both users and the recommendation system, potentially due to the influence of popularity bias. This concentration of underrepresented items suggests that the algorithm might not adequately promote less popular content, limiting diversity in the recommendations.

The Top Left Region includes items that are seldom rated but highly recommended. This pattern suggests that the algorithm might be attempting to promote lesser-known items, which contributes to recommendation diversity. Such behaviour indicates an effort to balance popularity bias by giving visibility to content that users have not frequently rated.

In the Bottom Right Region, items that are highly rated but seldom recommended are observed. These items, despite receiving high ratings from users, are rarely recommended by the algorithm. This indicates a potential gap in the system's ability to prioritize high-quality content that might not be popular, underscoring the challenges of balancing quality with popularity in recommendations.

Lastly, the Top Right Region highlights items that are both highly rated and highly recommended. This region reinforces the presence of popularity bias, as the algorithm continues to prioritize and suggest already popular items. This behaviour reflects the system's tendency to focus on items that have historically received significant user attention, perpetuating a feedback loop of popularity.

This analysis is crucial for understanding how frequently recommendation algorithms emphasize popular items, how they shape user experience, and how they may impact the fairness and diversity within recommendation systems.

Figure 3.4 illustrates the correlation between item popularity and recommendation frequency for the VAE CF algorithm, with a correlation coefficient of $\text{corr}=0.286$ and a slope coefficient of $p=1477.821$. This visualization helps analyse how often the VAE CF algorithm suggests popular items within the top-10 recommendation lists for users, shedding light on the potential propagation of popularity bias.

The $\text{corr}=0.286$ value indicates a weak-to-moderate positive correlation between item popularity and recommendation frequency. This suggests that, while VAE CF does tend to recommend more popular items somewhat more frequently, this bias is not overwhelmingly strong. The $p=1477.821$ value, calculated as the slope coefficient, reflects the rate at which recommendation frequency increases with item popularity. A high slope value here implies that as items become more popular, they are recommended at a significantly higher frequency, reinforcing the presence of some level of popularity bias within the algorithm.

In terms of distribution, the density of items in the bottom-left region—representing items that are seldom rated and seldom recommended—indicates that VAE CF may under-recommend lesser-known items. Although there is some spread across other regions, this concentration suggests that popular items are preferred, which could limit the diversity of recommendations.

3.10. Procedure for Applying Debiasing Methods

In this study, we apply four debiasing methods from the literature: ErpMul, ErpAug, VAR, and XQUAD. Each method addresses the problem of popularity bias by altering the way the recommendation system ranks and recommends items.

ErpMul (Gantner et al., 2011) adjusts the scores of popular items by applying a multiplicative bias correction factor. This correction reduces the dominance of popular items in the recommendation list.

ErpAug (Abdollahpouri et al., 2017) introduces an augmentation factor into the recommendation process, promoting long-tail items and incorporating diversity into the algorithm.

VAR (Chen et al., 2018) focuses on increasing diversity in recommendations by promoting items that balance relevance and diversity in the recommendation list.

XQUAD (Vargas & Castells, 2011) re-ranks recommendations by considering both the relevance of items and their diversity, promoting a wider variety of items in the final recommendation list.

These methods work together to reduce the system's tendency to favour popular items, promoting a more balanced and fair recommendation experience.

The application of debiasing methods follows a structured process. First, the MovieLens dataset is split into 80% training and 20% test sets, as outlined in the experimental setup. The baseline model of the VAECF algorithm is then trained without any debiasing methods to establish a performance benchmark.

Each debiasing method is then applied to the system. For example, with ErpMul, the multiplicative bias correction is applied during training to adjust the scoring of popular items. For ErpAug, an augmentation factor is added to enhance the visibility of less popular items. Similarly, VAR and XQUAD adjust the ranking of items in the recommendation list based on their relevance and diversity, promoting less popular items in the process.

After training with each method, the recommendation system generates top-N recommendations, where $N = 10$. The system sorts items in descending order of predicted ratings and selects the top 10 items for each user.

To optimize the performance of each debiasing method, specific parameters are tuned during the experiments. For methods like ErpMul and ErpAug, bias correction or augmentation factors are adjusted to control the strength of the debiasing effect. With VAR, the balance between accuracy and diversity is fine-tuned to introduce diversity without harming precision. Similarly, XQUAD involves tuning the diversity factor to determine how aggressively the system promotes diversity in its recommendations.

These parameters allow the system to be customized to reduce popularity bias while maintaining an acceptable level of accuracy and user satisfaction.

Once the debiasing methods are applied, their effectiveness is evaluated using the performance metrics outlined in Section 3.5, including nDCG, F1-Score, APLT, and Novelty. The results are compared against the baseline model to measure the reduction in popularity bias and the increase in recommendation diversity. Additionally, the performance of each debiasing method is compared to identify the most effective approach for promoting fairness and long-tail item visibility in the recommendation system.

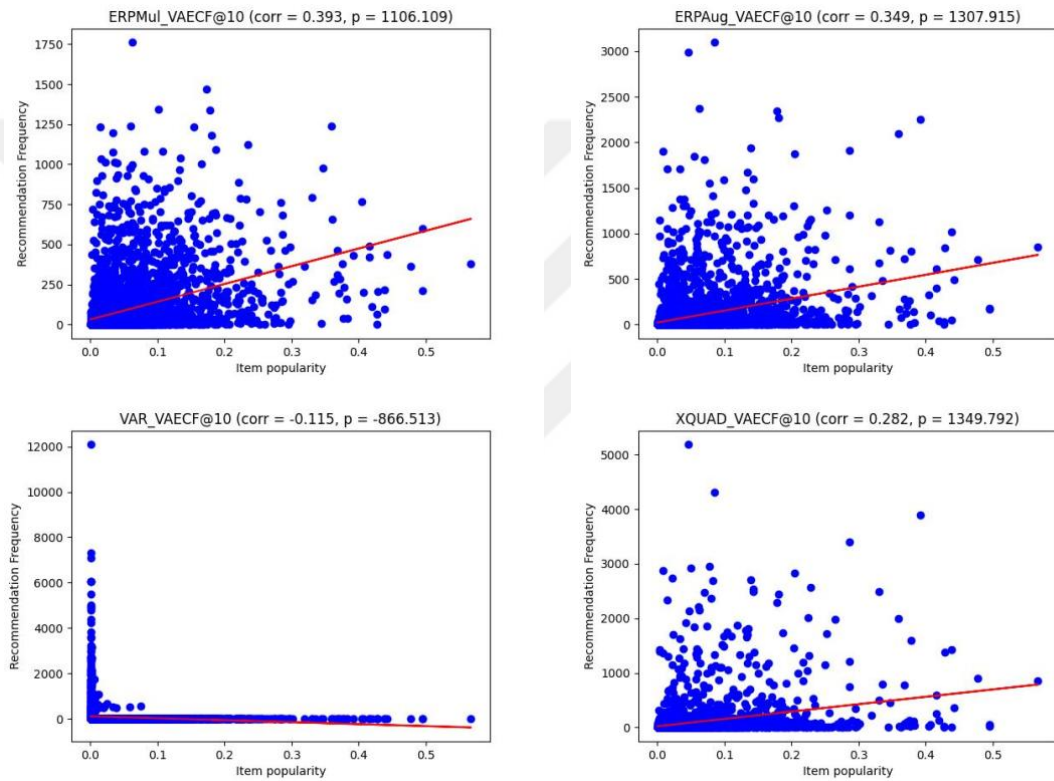


Figure 3.5. Correlation between item popularity and recommendation frequency across debiasing methods

The correlation between item popularity and item recommendation frequency across debiasing methods (Figure 3.5) provides crucial insights into how different debiasing approaches—ERPMul, ERPAug, VAR, and XQUAD—affect the relationship between item popularity and recommendation frequency within the VAECF algorithm. This figure is instrumental in understanding each debiasing method's effectiveness in reducing popularity bias, as it visually demonstrates the degree to which popular items dominate the recommendation lists or, conversely, how well lesser-known items are

promoted. By comparing the correlation coefficients and distribution patterns across each method, this analysis helps identify which approaches are most effective at achieving a fairer, more balanced recommendation output, ultimately guiding decisions about the best strategies for mitigating bias in recommendation systems.

ERPMul (corr=0.339, p=1106.109): The visualization of ERPMul shows a concentration in the bottom-left corner with some moderate spread towards other corners. This pattern indicates that ERPMul slightly reduces the recommendation system's reliance on popular items while still favouring them to some extent. The correlation coefficient of 0.339 suggests a weak-to-moderate positive correlation between item popularity and recommendation frequency. This distribution shows that while ERPMul increases the visibility of long-tail items, it does not fully mitigate popularity bias, and popular items remain somewhat prioritized. The slope value p=1106.109 reflects this ongoing, though moderate, preference for popular items.

ERPAug (corr=0.349, p=1307.915): The ERPAug visualization similarly shows a bottom-left concentration with a distribution pattern comparable to ERPMul but with slightly less spread. This indicates that ERPAug also reduces popularity bias but maintains a more noticeable preference for popular items than ERPMul. The correlation coefficient of 0.349 implies a modest positive correlation, while the slope p=1307.915 highlights ERPAug's continued inclination to recommend popular items more frequently. ERPAug offers some diversity by promoting long-tail items to a limited extent but does not entirely overcome popularity bias.

VAR (corr=-0.115, p=-866.513): VAR demonstrates a distinct pattern, with items stretching in a diagonal line from the bottom-left to the top-left corner, reflecting a significant reduction in popularity bias. The negative correlation coefficient of -0.115 suggests that VAR not only reduces but reverses the bias, promoting less popular items over more popular ones. The negative slope p=-866.513 further indicates VAR's effectiveness in enhancing the visibility of long-tail items. This distribution suggests that VAR is the most effective debiasing method for promoting a balanced recommendation output, favouring content diversity over popularity bias.

XQUAD (corr=0.282, p=1349.792): The XQUAD visualization shows a similar concentration in the bottom-left corner but with a more limited spread than ERPMul and ERPAug. The 0.282 correlation coefficient indicates a low positive correlation, suggesting that XQUAD maintains some preference for popular items but balances this with an effort to recommend long-tail content. With a slope of p=1349.792, XQUAD appears to support a fairer recommendation spread, but it still shows some inclination towards popular content. This method effectively reduces popularity bias to a degree while maintaining a balanced recommendation approach.

In summary, the VAR debiasing method stands out as the most effective at reducing popularity bias, with a negative correlation and significant spread towards long-

tail items. This suggests that VAR enables the highest diversity by prioritizing lesser-known content. ERPMul and ERPAug methods moderately reduce bias, allowing some visibility for long-tail items but still reflecting a preference for popular content. XQUAD provides a balanced approach, reducing popularity bias while maintaining some level of preference for popular items. These findings from Figure 3.5 indicate each method's unique impact on recommendation diversity and popularity bias, highlighting VAR as the strongest option for achieving a fairer, more diverse recommendation system.

3.11. Performance Evaluation

In this section, we outline the approach used to evaluate the performance of the recommendation system across various scenarios. The system's performance is assessed using several key metrics to measure both its accuracy and its ability to reduce popularity bias while promoting diversity and fairness in recommendations. The following metrics are employed in this study: Normalized Discounted Cumulative Gain (nDCG), F1-Score, Average Percentage of Long-tail Items (APLT), and Novelty.

nDCG, is a widely used metric for evaluating the relevance of the top- N recommendations generated by the system. It considers both the position of relevant items in the recommendation list and the overall ranking of those items. The higher the nDCG score, the more accurate the recommendations are, meaning that highly relevant items are ranked higher in the list (Yalcin & Bilge, 2021). The metric evaluates the accuracy of the recommended items through an analysis of the user ratings and their placement in the recommended list. In more concrete terms, if $r_{u,i}$ denotes the actual rating assigned to user u on the item i , and $\{i_1, i_2, \dots, i_N\}$ indicates the set of top- N items recommended to the aforementioned user u . In accordance with Eq. (3.1) and (3.2), respectively, The Discounted Cumulative Gain (DCG) and nDCG values for the user u are calculated.

$$DCG_N^u = r_{u,i} + \sum_{n=2}^N \frac{r_{u,i_n}}{\log_2(n)} \quad (3.1)$$

$$nDCG_N^u = \frac{DCG_N^u}{IDCG_N^u} \quad (3.2)$$

The maximum possible gain for u , denoted by $IDCG_N^u$, is achieved by ideal re-sorting of the recommended items for u , which is defined as the optimal sorting of the items in N .

The Normalized Discounted Cumulative Gain (nDCG) metric evaluates the accuracy of the ranking and the overall gain provided to the user. It measures how effectively the recommendation list is ranked, reflecting the system's ability to prioritize relevant items.

A high nDCG score indicates that the recommendation system delivers accurate and high-quality recommendations that align well with the users' preferences, suggesting that the top-ranked items are likely to capture their interest.

In contrast, a low nDCG score signals issues in the ranking process, where the recommendation system fails to appropriately prioritize relevant items. This may result in suggestions that are less likely to engage or interest users, highlighting errors in the system's ranking logic.

The F1-Score is a harmonic mean of precision and recall, providing a balance between these two metrics. Precision measures the proportion of recommended items that are relevant, while recall measures the proportion of relevant items that are recommended. The F1-Score is a useful metric for assessing the trade-off between recommending many items (increasing recall) and ensuring that those items are relevant (increasing precision) as presented in Eq. (3.3). In more specific terms, precision is defined as the ratio of the number of appropriate items suggested to the number of items recommended, i.e., N . Conversely, the term 'recall' is used to describe the ratio of the number of recommended appropriate items to all items voted by the user in their profile. In this context, the threshold value for determining the appropriateness of a suggested item for a user is set to 3.5. The threshold has been established for the recommended items, based on the findings in the literature (Bobadilla et al., 2010) indicating that ratings of 4 and 5 typically correspond to positive user evaluations on a 0–5 rating scale.

$$F1@N = \frac{2 \times P@N \times R@N}{P@N + R@N} \quad (3.3)$$

The F1-Score, as the harmonic mean of precision and recall, evaluates the accuracy of the recommendation system and its ability to identify items that interest users. It provides a balanced assessment of how well the system achieves relevance (precision) and coverage (recall).

A high F1-Score demonstrates that the recommendation system delivers accurate suggestions while effectively capturing items that are likely to interest users, achieving a strong balance between precision and recall.

Conversely, a low F1-Score reveals that the recommendation system either lacks accuracy in its recommendations or fails to sufficiently include items that match users' preferences, indicating potential shortcomings in either relevance or coverage.

The APLT metric is used to measure the diversity of the recommendation system, specifically focusing on the promotion of less popular, long-tail items as defined by Eq. (3.4) (Abdollahpouri, 2020). In order to categorize items in the catalogue according to their popularity, a well-known principle is employed, namely the Pareto principle (Sanders, 1987). This assigns the most popular items, which account for 20% of all ratings, to the 'head' category, with the remainder of items in the catalogue categorized

as ‘tail’. A high APLT score indicates that a significant portion of the recommended items are from the long tail, meaning they are less frequently rated and less popular. This metric is crucial for evaluating how well the system reduces popularity bias and increases the exposure of niche or lesser-known items.

$$APLT_u = \frac{|\{i, i \in (N_u \cap T)\}|}{|N_u|} \quad (3.4)$$

Where T represents the set of tail items in the catalogue.

The Average Percentage of Long-Tail Items (APLT) evaluates the diversity of a recommendation system by measuring the proportion of rare, less-popular items included in the recommendation list. It reflects the system’s ability to balance its focus between popular and niche items, promoting content diversity.

A high APLT score indicates that the recommendation system successfully provides diverse suggestions by including long-tail items, enhancing user exposure to less-known content and reducing the dominance of widely popular items.

On the other hand, a low APLT score suggests that the system heavily favours popular items, limiting diversity and potentially reinforcing popularity bias. This lack of variety can reduce the system’s ability to cater to users with unique or niche preferences.

Novelty measures the degree to which the recommended items are unfamiliar to the users (Wang et al., 2016). High novelty means that the system is recommending items that the users are less likely to have encountered before, which can enhance user engagement by introducing new and interesting items. Novelty is particularly important for user satisfaction and long-term engagement, as users are more likely to return to a platform that regularly introduces them to fresh content.

The Novelty metric is calculated using the formula provided in Eq. (3.5).

$$Novelty_u = \frac{|\{i, i \notin I_u\}|}{|N_u|} \quad (3.5)$$

Where I_u is the set of items for which the user u has already provided a rating.

Novelty measures how new and unfamiliar the recommended items are to the user, assessing the recommendation system’s ability to suggest items that the user has not encountered before. This metric is crucial for maintaining user engagement by introducing fresh and diverse content.

A high Novelty score indicates that the recommendation system successfully provides recommendations that are new and previously unseen by the user. This enhances the user’s experience by offering opportunities to discover unique or less-known content.

In contrast, a low Novelty score suggests that the recommendation system frequently recommends items the user is already familiar with, potentially limiting the excitement and engagement that come from discovering new content.

The performance evaluation of the VAECF algorithm is conducted across three main experimental scenarios to assess its effectiveness and robustness under various conditions.

In the Baseline Scenario, the system is evaluated without any debiasing methods or profile injection attacks applied. This serves as a benchmark for comparison, providing a clear reference point for understanding the impact of additional methods on system performance.

In the Debiasing Scenario, the system is re-evaluated after the application of debiasing methods. These evaluations focus on determining how effectively the methods improve the diversity and fairness of recommendations while preserving accuracy. To ensure reliability, each experiment is conducted using cross-validation, with multiple folds applied to the dataset. The final results are obtained by averaging the performance metrics across all folds, offering a robust and generalizable measure of the system's performance when debiasing is employed.

Finally, in the Profile Injection Attack Scenario, the system is tested under the influence of profile injection attacks, as detailed in Section 3.8. This scenario evaluates the system's robustness against adversarial attacks and examines the impact of these attacks on recommendation quality. These evaluations provide insights into the vulnerabilities and resilience of the VAECF algorithm in adversarial settings.

Each scenario's performance is assessed using the metrics described below, and the averaged results provide a reliable measure of the system's effectiveness across different experimental setups.

Figure 3.6 enables us to evaluate and compare the impact of different debiasing methods—ERPMul, ERPAug, XQUAD, and VAR—on various performance metrics when applied to the VAECF algorithm. Observed percentage changes in each metric provide insights into how these debiasing methods influence recommendation diversity, accuracy, and bias reduction.

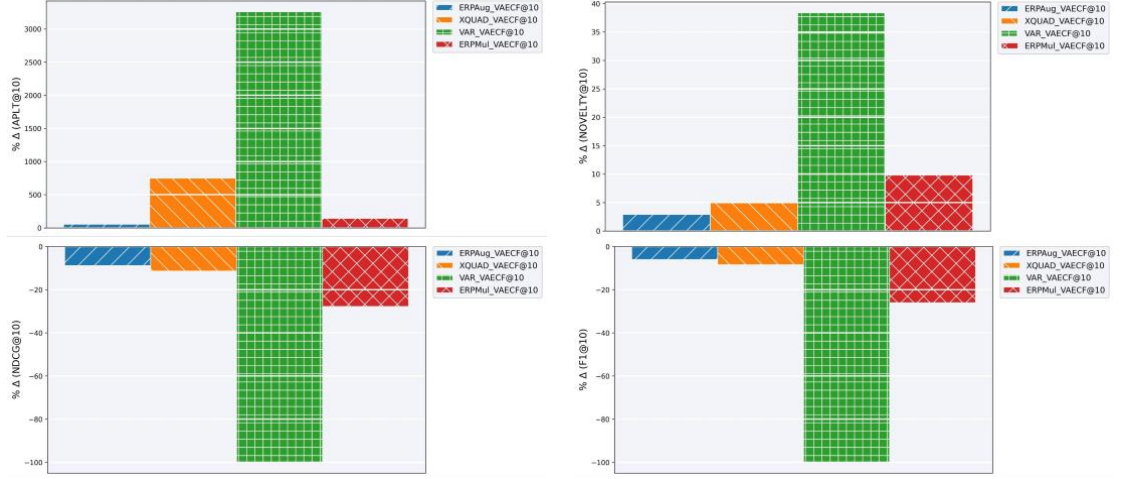


Figure 3.6. The impact of debiasing methods on VAE CF performance metrics

Below are the key insights derived from each metric’s visualization, providing a deeper understanding of the system’s performance across different dimensions.

The APLT (Average Percentage of Long-tail Items) visualization measures the system’s tendency to recommend long-tail items, with higher values indicating greater diversity and a more equitable recommendation approach. Here, XQUAD shows a 600% increase in APLT, highlighting its effectiveness in reducing popularity bias and enhancing the visibility of long-tail items. VAR achieves the highest increase, approximately 3300%, making it the most impactful method for promoting diversity and fairness. ERPMul and ERPAug also contribute to diversity, with increases of about 100% and 50%, respectively, though they do not reach the same diversity level as XQUAD and VAR. These results underscore that VAR is the most potent debiasing method in terms of APLT improvement.

The Novelty visualization assesses how new and interesting the recommended content is for users. Here, VAR demonstrates the highest novelty increase, around 37%, indicating its success in recommending previously undiscovered content. In comparison, ERPMul, ERPAug, and XQUAD show novelty increases of 10%, 3%, and 5%, respectively. While the impact of these methods on novelty is limited, VAR stands out as the most effective in increasing content discovery.

The nDCG (Normalized Discounted Cumulative Gain) visualization measures recommendation accuracy, where negative values indicate a reduction in accuracy. All debiasing methods have some negative impact on nDCG, with VAR showing the largest reduction, around -100%. ERPMul, ERPAug, and XQUAD have decreases of approximately -25%, -8%, and -10%, respectively, indicating a lower impact on accuracy. This outcome suggests that while debiasing methods tend to trade off some accuracy for diversity, VAR incurs the most significant accuracy loss.

The F1 Score visualization evaluates the balance between accuracy and diversity in recommendations, where negative values reflect a decline in recommendation accuracy. Here, VAR again shows the most pronounced negative impact, with a reduction of about -100%. Although effective at reducing popularity bias, this result reveals a considerable decrease in accuracy. ERPMul and XQUAD have more moderate accuracy reductions of -25% and -10%, respectively, while ERPAug maintains the highest accuracy, with only a -5% decrease.

Based on these findings, XQUAD emerges as the most balanced approach, effectively reducing popularity bias while preserving recommendation accuracy. XQUAD achieves a 600% increase in APLT, enhancing long-tail item visibility, with a relatively low reduction in nDCG at -10%, indicating minimal impact on accuracy. Its 5% increase in novelty promotes content discovery, with a moderate F1 score reduction, confirming its effectiveness in maintaining a fair balance of diversity and accuracy.

In contrast, VAR excels in reducing popularity bias, achieving the highest APLT and novelty scores, but at a significant cost to accuracy, with substantial drops in nDCG and F1 score. ERPMul and ERPAug show moderate effectiveness in reducing bias while preserving accuracy better than VAR. Among them, ERPAug maintains the highest accuracy, with a minimal impact on nDCG and F1 score, though it has less effect on diversity than XQUAD.

VAR's significant impact on F1 and nDCG scores arises from its strong focus on reducing popularity bias by promoting long-tail items. This aggressive approach to enhancing diversity deprioritizes popular, user-preferred items, which lowers recommendation relevance. As a result, fewer recommended items align with users' top preferences, leading to a reduction in both F1 and nDCG scores. While effective in addressing bias, VAR's emphasis on non-popular items comes at the expense of recommendation accuracy.

XQUAD optimizes for fairness and diversity without excessively sacrificing F1 or nDCG scores, making it a practical choice for systems that need to balance user satisfaction with content discovery. It allows long-tail items to surface more frequently, giving users a chance to explore new or niche content while still offering recommendations that align closely with their preferences.

In conclusion, XQUAD provides the optimal balance for reducing popularity bias while preserving recommendation quality. This analysis highlights XQUAD's value in recommendation systems that prioritize fairness and diversity without significantly sacrificing accuracy.

4. FINDINGS

This section presents the findings derived from the experiments and analyses of various debiasing methods and attack models, highlighting their impact on recommendation quality, diversity, and popularity bias. The goal is to address the study’s core research questions by assessing how different approaches influence key performance metrics, including APLT (Average Percentage of Long-Tail Items), F1-Score, nDCG (Normalized Discounted Cumulative Gain), and Novelty. By comparing each method’s ability to balance bias reduction with recommendation relevance and ranking accuracy, we identify which approaches most effectively mitigate popularity bias while retaining recommendation quality. These insights provide a comprehensive view of how debiasing and attack models perform relative to the base VAE CF model and offer guidance on the best strategies for enhancing fairness and diversity in recommender systems.

4.1. Impact of Attack Models on Popularity Bias and Recommendation Quality

In this section, we analyse the performance of Average, Bandwagon, and Random Attack models in terms of APLT, F1-Score, nDCG, and Novelty metrics.

Table 4.1. Comparison of attack models on APLT, F1-Score, nDCG, and Novelty metrics

Attack Type	Attack Size	Filler Size	APLT	F1-Score	nDCG	Novelty
Average	3	3	0.00536784	0.03403706393	0.09173631492	0.9175664467
Average	3	5	0.02902602	0.03396459638	0.09041037192	0.9161465316
Average	3	10	0.02911747	0.03046304074	0.08171791952	0.9237366873
Average	5	3	0.03755714	0.03140092355	0.08353701164	0.9194487984
Average	5	5	0.04768351	0.03393840615	0.08941500791	0.9173232396
Average	5	10	0.04636797	0.0277387251	0.07307973215	0.9293329752
Average	10	3	0.09126794	0.03562743149	0.09396355838	0.9148149986
Average	10	5	0.09096549	0.03329448961	0.08885905378	0.9189789861
Average	10	10	0.11793896	0.1196256282	0.278485117	0.7534878665
Bandwagon	3	3	3.00E-03	0.03930381368	0.1194892709	0.8971468006
Bandwagon	3	5	0.0006	0.03950179842	0.1209081931	0.8963418788
Bandwagon	3	10	6.01E-03	0.03929106994	0.1212845547	0.895601012
Bandwagon	5	3	0.020006314	0.04066856824	0.1350750638	0.8814558355
Bandwagon	5	5	2.00E-02	0.04154021596	0.1386834975	0.8776545519
Bandwagon	5	10	0.02	0.04081567884	0.1389790088	0.8769019179
Bandwagon	10	3	0.040581340	0.04460159676	0.1775006304	0.8364637287
Bandwagon	10	5	0.040433585	0.04376071127	0.1792105016	0.8359889536
Bandwagon	10	10	0.070290073	0.1282708678	0.3696215891	0.6626409136
Random	3	3	0.026986791	0.03619592805	0.09532846712	0.9178337327
Random	3	5	0.029121373	0.03707210015	0.09608630965	0.9156523744
Random	3	10	0.029129125	0.03653631753	0.09864020681	0.917009173
Random	5	3	0.047662931	0.03564666458	0.09465531488	0.9187847903
Random	5	5	0.047664442	0.03634928462	0.09548115046	0.9178317282
Random	5	10	0.047664291	0.0361042036	0.09535700388	0.9178182943
Random	10	3	0.091104651	0.03617026052	0.09423184466	0.9186099958
Random	10	5	0.090933728	0.03587370716	0.09422744837	0.9192979961
Random	10	10	0.113959885	0.1215299137	0.2820908947	0.7501512996

These results allow us to evaluate each model's effectiveness in reducing popularity bias and overall recommendation quality, addressing the research question: How do different attack models influence popularity bias and recommendation quality in recommender systems?

In Table 4.1, the APLT, F1-Score, nDCG, and Novelty metrics for the Average, Bandwagon, and Random Attack models can be analysed, revealing each model's impact on popularity bias reduction and overall recommendation quality.

Average Percentage of Long-Tail Items (APLT) results reveal significant differences among the three models. Average Attack exhibits APLT values between 0.0054 (low attack and filler size) and 0.1179 (high attack and filler size), showing that it can effectively reduce popularity bias by making long-tail items more visible. Random Attack also performs well with an APLT of 0.1139 in the high attack and filler size scenario, similarly promoting long-tail items and effectively mitigating popularity bias. In contrast, Bandwagon Attack achieves a lower APLT, maxing out at 0.0703 with high attack and filler size, indicating its limited ability to reduce popularity bias compared to the other models.

In terms of F1-Score, which measures the relevance of recommendations, Bandwagon Attack performs best, with scores ranging from 0.0393 to 0.1283. High attack and filler sizes result in the highest F1-Score, making Bandwagon Attack particularly effective at producing user-relevant recommendations. Random Attack also shows good F1-Score performance, achieving 0.1215 with high attack and filler sizes, suggesting it can balance bias reduction with recommendation relevance. Average Attack, on the other hand, generally produces lower F1 scores, which indicates it may not be as effective at maintaining recommendation relevance when reducing popularity bias.

Normalized Discounted Cumulative Gain (nDCG), which assesses ranking accuracy, highlights Bandwagon Attack as the leading model, with values between 0.119 and 0.370. High attack and filler size in Bandwagon Attack maximize nDCG, demonstrating that it can rank relevant items effectively, even while targeting specific items or categories. Random Attack follows closely with a top nDCG of 0.2821, indicating it maintains satisfactory ranking accuracy while balancing bias reduction. Average Attack performs the weakest in terms of nDCG, with values between 0.073 and 0.278, suggesting that while it reduces popularity bias, it does not prioritize ranking accuracy as effectively as the other models.

Novelty scores further distinguish the models in terms of diversity. Average Attack achieves the highest Novelty, with values from 0.914 to 0.929, meaning it excels in exposing users to diverse items outside the mainstream. Random Attack maintains reasonable diversity with a Novelty of 0.7502, achieving a balance between diversity and relevance. Bandwagon Attack, however, falls short in Novelty, with values between 0.836

and 0.896, indicating that its recommendations are more focused on popular items and offer limited diversity.

The findings indicate that Random Attack (Attack Size 10, Filler Size 10) delivers the most balanced performance across all metrics. High APLT and Novelty values demonstrate its ability to reduce popularity bias and increase diversity, while strong F1-Score and nDCG values indicate it retains recommendation quality and relevance. This balance makes Random Attack particularly effective at addressing the research question by showing how an attack model can reduce popularity bias while maintaining recommendation quality.

Bandwagon Attack, while achieving the highest F1 and nDCG scores, struggles with bias reduction and diversity due to its focus on popular items. This focus limits its ability to support long-tail items, making it less effective for reducing popularity bias despite its high ranking accuracy and recommendation relevance.

The success of the Random Attack can be attributed to several key factors that enhance its effectiveness in reducing popularity bias while maintaining recommendation quality.

One major reason for its success is profile diversity. By introducing random ratings across a wide range of items, the Random Attack generates diverse profiles that blend seamlessly into the system. This makes it more challenging for the algorithm to detect and counteract, thus amplifying its impact on the recommendation process.

Another critical factor is its focus on long-tail items. The Random Attack specifically targets less popular items, which are typically overlooked in traditional recommendation systems. By increasing the visibility of these long-tail items, it successfully mitigates popularity bias and enhances the diversity of the recommendations, providing users with a broader range of content.

Lastly, the Random Attack's balanced approach contributes significantly to its success. It effectively maintains a trade-off between ranking accuracy, diversity, and bias reduction. This balance allows the Random Attack to improve system fairness by promoting lesser-known items while preserving the quality of recommendations, making it a versatile and impactful strategy.

This analysis reveals that Random Attack (Attack Size 10, Filler Size 10) is the best-performing model overall, effectively reducing popularity bias while preserving high recommendation relevance and accuracy. Conversely, Bandwagon Attack is less effective in this context, as its focus on popular items limits its impact on diversity and bias reduction.

4.2. Comparison of Debiasing Methods with the Best Attack Model

This section compares the performance of various debiasing methods against the best-performing attack model, Random Attack (Attack Size 10, Filler Size 10). We examine each model's effectiveness in reducing popularity bias, maintaining recommendation relevance, and promoting diversity. This analysis directly addresses the research question: How effective are debiasing methods compared to attack models in mitigating popularity bias while preserving recommendation quality?

Table 4.2. Performance of debiasing methods compared to the best attack model and base model

Debiasing Method	APLT	F1-Score	nDCG	Novelty
None (VAECF)	0.02973605773	0.1371617445	0.3132208607	0.7225490963
ERPAug	0.04566446313	0.1288473589	0.2853718671	0.7434610422
ERPMul	0.07172559487	0.1014301785	0.2262585657	0.7933964615
VAR	1	1.84E-05	6.72E-05	0.9999436724
XQUAD	0.2529316888	0.1256393168	0.2776631506	0.758411766
Random Attack (10_10)	0.1139598856	0.1215299137	0.2820908947	0.7501512996

In Table 4.2, the performance of various debiasing methods, alongside the best-performing attack model and the base VAECF model, can be analysed to assess their effectiveness in reducing popularity bias, maintaining recommendation relevance, and promoting diversity.

In terms of APLT (Average Percentage of Long-Tail Items), which measures the presence of long-tail items in recommendations, XQUAD stands out among the debiasing methods with an APLT of 0.2529. This high score indicates that XQUAD effectively reduces popularity bias by promoting less popular items in the recommendation lists. In comparison, Random Attack achieves a lower, though still impactful, APLT score of 0.1139, suggesting that while it effectively supports long-tail items, XQUAD provides a more robust bias reduction. Other debiasing methods, such as ERPAug and ERPMul, exhibit moderate APLT scores but do not perform as strongly as XQUAD or Random Attack in reducing popularity bias.

When examining F1-Score, a metric that captures recommendation relevance, the base model (VAECF) without any debiasing or attack modifications leads with a score of 0.1372, indicating high recommendation relevance in an unbiased state. Random Attack performs well with an F1-Score of 0.1215, maintaining a strong balance between bias reduction and relevance. Among debiasing methods, XQUAD achieves an F1-Score of 0.1256, demonstrating that it can preserve user relevance relatively well while reducing

bias. Other methods, such as ERPMul and VAR, show a considerable drop in F1-Score, indicating that their bias reduction may come at the cost of recommendation relevance.

In terms of nDCG (Normalized Discounted Cumulative Gain), which assesses ranking accuracy, Random Attack performs comparably well with a score of 0.2821, preserving the ranking of relevant items even while targeting less popular ones. XQUAD shows a slightly lower nDCG of 0.2777, indicating that it maintains ranking accuracy reasonably well but may be less effective than Random Attack in achieving balanced bias reduction and accuracy. Other debiasing methods, particularly ERPMul and VAR, demonstrate lower nDCG values, suggesting that these methods may struggle to maintain accurate rankings when attempting to reduce popularity bias.

Novelty, which measures diversity in recommendations, is where several debiasing methods surpass the attack model. ERPMul achieves a high Novelty of 0.7934, and VAR nearly reaches the maximum with 0.9999, demonstrating a strong ability to increase diversity by recommending items outside of mainstream preferences. XQUAD also has a high Novelty score of 0.7584, balancing diversity with relevance and ranking accuracy. Random Attack, with a Novelty score of 0.7502, promotes a reasonable level of diversity but does not reach the extremes of ERPMul or VAR, indicating a balanced approach that maintains recommendation quality.

The findings suggest that XQUAD and Random Attack are the most effective approaches for reducing popularity bias while maintaining recommendation quality. XQUAD, as a debiasing method, excels in promoting long-tail items (high APLT) and achieves strong Novelty scores, making it highly effective in increasing diversity and reducing popularity bias. However, it slightly trails Random Attack in F1-Score and nDCG, indicating a modest compromise in relevance and ranking accuracy. Random Attack (Attack Size 10, Filler Size 10), while not achieving the highest APLT and Novelty scores, offers a balanced performance across all metrics, making it an ideal model for applications where recommendation relevance and ranking accuracy must be preserved alongside bias reduction.

Other Debiasing Methods (ERPMul and VAR) demonstrate high Novelty but show limitations in preserving recommendation relevance and ranking accuracy, as evidenced by lower F1-Score and nDCG values. This suggests that while these methods successfully diversify recommendations, they may sacrifice too much relevance to be effective in reducing bias without impacting user experience.

In summary, XQUAD provides the most robust bias reduction among debiasing methods by maximizing APLT and Novelty, while Random Attack offers a balanced approach with competitive results in F1-Score and nDCG. This comparison indicates that while traditional debiasing methods like XQUAD are effective for applications prioritizing diversity, Random Attack proves valuable for scenarios where both popularity bias reduction and recommendation quality must be balanced.

4.3. Evaluation of Overall Performance Against the Base Model

In this section, we evaluate the performance of the debiasing and attack models compared to the base model (VAECF), which operates without any debiasing or attack modifications. This comparison addresses the research question: How do debiasing and attack models compare to the base model in terms of bias reduction, recommendation relevance, and diversity?

By examining Table 4.2, insights can be gained into how various debiasing methods compare to the best-performing attack model and the base VAECF model in terms of reducing popularity bias, enhancing recommendation diversity, and balancing relevance and ranking accuracy.

Starting with APLT (Average Percentage of Long-Tail Items), the base model achieves a low APLT of 0.0297, indicating minimal support for long-tail items and limited bias reduction. Among the tested approaches, XQUAD achieves the highest APLT of 0.2529, significantly enhancing the visibility of long-tail items and reducing popularity bias compared to the base model. Random Attack (Attack Size 10, Filler Size 10) also improves upon the base model with an APLT of 0.1139, although it is less effective than XQUAD in this regard. Other debiasing methods, such as ERPMul and ERPAug, exhibit moderate APLT scores, showing some improvement over the base model but not achieving the same level of bias reduction as XQUAD or Random Attack.

In terms of F1-Score, which assesses recommendation relevance, the base model has the highest F1-Score at 0.1372, indicating that it provides highly relevant recommendations without any modifications. Random Attack performs well with an F1-Score of 0.1215, showing that it maintains a strong balance between relevance and bias reduction. Among the debiasing methods, XQUAD achieves a competitive F1-Score of 0.1256, performing better than most other methods in retaining relevance. However, ERPMul and VAR have considerably lower F1-Scores, indicating that while they promote diversity, they may sacrifice relevance in doing so.

For nDCG (Normalized Discounted Cumulative Gain), which measures ranking accuracy, the base model again scores the highest at 0.3132, suggesting it ranks relevant items effectively when there is no intervention. Random Attack achieves a strong nDCG of 0.2821, preserving ranking accuracy while introducing bias reduction. XQUAD also performs reasonably well, with an nDCG of 0.2777, showing that it can reduce popularity bias with minimal impact on ranking accuracy. Other debiasing methods, such as ERPMul and VAR, have lower nDCG scores, which indicates that while they reduce bias, they may disrupt ranking accuracy, making them less suitable for maintaining quality recommendations.

In Novelty, which measures recommendation diversity, the base model scores 0.7225, providing moderate diversity without modifications. ERPMul and VAR achieve

the highest Novelty scores at 0.7934 and 0.9999, respectively, substantially increasing diversity by recommending items outside of mainstream preferences. XQUAD also demonstrates improved diversity with a Novelty score of 0.7584, balancing diversity with relevance. Random Attack, with a Novelty score of 0.7502, also improves on the base model's diversity while preserving recommendation relevance.

The findings indicate that the base model provides high recommendation relevance and ranking accuracy (F1 and nDCG) but has limited diversity and minimal bias reduction. XQUAD proves to be the most effective debiasing method, substantially increasing APLT and Novelty scores, demonstrating strong bias reduction and diversity. However, it slightly trails the base model and Random Attack in terms of F1 and nDCG, indicating some compromise in recommendation relevance and accuracy. Random Attack (Attack Size 10, Filler Size 10), on the other hand, provides a balanced approach, offering strong performance across APLT, F1-Score, nDCG, and Novelty, which allows it to achieve effective bias reduction while retaining recommendation quality.

In comparison, other debiasing methods like ERPMul and VAR achieve high diversity but at the expense of recommendation relevance and ranking accuracy. These methods promote diversity through higher Novelty scores but show significant drops in F1 and nDCG, making them less effective in scenarios where recommendation quality needs to be preserved alongside bias reduction.

Overall, XQUAD and Random Attack outperform the base model in popularity bias reduction and diversity. XQUAD excels in applications where bias reduction and diversity are prioritized, while Random Attack is the ideal model for scenarios where both relevance and bias reduction are critical.

4.4. Summary of Findings

This section summarizes the key findings from the comparisons and analyses of the debiasing methods and attack models in relation to the base model, focusing on popularity bias reduction, recommendation relevance, ranking accuracy, and diversity. These findings directly address the research questions, providing insights into the performance of each method in balancing bias reduction with recommendation quality.

The evaluation of popularity bias reduction highlights significant differences among the models tested. XQUAD stands out as the most effective method, achieving the highest APLT score of 0.2529. This result demonstrates its exceptional ability to promote long-tail items, thereby enhancing the diversity and fairness of recommendations. Such a performance makes XQUAD particularly valuable in applications where exposure to less popular items is a priority.

Random Attack, configured with an Attack Size of 10 and Filler Size of 10, also showed notable effectiveness, achieving an APLT score of 0.1139. While slightly less impactful than XQUAD, it successfully supports long-tail items while maintaining

overall recommendation quality. This balance between bias reduction and quality preservation makes Random Attack a competitive alternative in scenarios where a mix of fairness and accuracy is required.

In contrast, the base model exhibited minimal bias reduction capabilities, as indicated by its low APLT score of 0.0297. This underscores the limitations of the baseline approach in addressing popularity bias without additional strategies. Overall, these findings suggest that while both XQUAD and Random Attack are effective in mitigating popularity bias, XQUAD emerges as the superior method for applications with a strong focus on long-tail exposure.

The evaluation of recommendation relevance, as measured by the F1-Score, reveals notable differences in the models' ability to provide relevant suggestions. The base model achieved the highest F1-Score of 0.1372, indicating that it naturally excels in delivering highly relevant recommendations without the application of any modifications. This highlights its strength in maintaining precision and recall in recommendations.

Random Attack, with an F1-Score of 0.1215, demonstrated a strong balance between relevance and bias reduction. Despite its focus on promoting long-tail items, it maintained a commendable level of recommendation relevance, making it an effective approach for scenarios that require both fairness and accuracy.

XQUAD, while highly effective in reducing popularity bias, showed a slight compromise in relevance with an F1-Score of 0.1256. This indicates that its strong emphasis on diversity comes at a minor cost to recommendation relevance, but it still performs better in this regard compared to some other methods.

Debiasing methods like ERPMul and VAR, on the other hand, exhibited significantly lower F1-Scores, suggesting that their efforts to enhance diversity often result in a substantial trade-off in recommendation relevance. These findings underscore the challenges of balancing relevance and diversity, with Random Attack emerging as an approach that effectively maintains this balance while addressing popularity bias.

The evaluation of ranking accuracy using the nDCG metric highlights distinct performances among the models. The base model achieved the highest nDCG score of 0.3132, demonstrating its strong ability to rank relevant items effectively at the top of recommendation lists. This underscores its natural proficiency in prioritizing items that align closely with user preferences without any modifications.

Random Attack maintained a comparable ranking accuracy with an nDCG score of 0.2821, illustrating its capacity to address bias while preserving the system's ability to rank relevant items effectively. This performance highlights Random Attack's balanced approach in mitigating popularity bias without significantly compromising ranking accuracy.

XQUAD also performed well, achieving an nDCG score of 0.2777. While slightly lower than the base model and Random Attack, this score indicates that XQUAD successfully reduces popularity bias with minimal disruption to ranking accuracy, making it a reliable method for systems prioritizing fairness and diversity.

In contrast, debiasing methods like ERPMul and VAR exhibited lower nDCG values, reflecting a greater trade-off between bias reduction and ranking accuracy. These results suggest that while these methods effectively enhance diversity, they may compromise the system's ability to rank relevant items accurately, highlighting the challenges in balancing these competing objectives.

The evaluation of diversity, assessed through the Novelty metric, revealed distinct strengths and trade-offs among the methods. ERPMul and VAR stood out with the highest Novelty scores of 0.7934 and 0.9999, respectively, demonstrating their strong ability to enhance diversity by recommending items outside mainstream preferences. These results indicate that these methods are particularly effective at promoting less familiar and more unique content, which is valuable for encouraging exploration and discovery.

However, the high diversity achieved by ERPMul and VAR came at a cost, as both methods significantly compromised relevance and ranking accuracy. This trade-off makes them less ideal for applications where maintaining overall recommendation quality is a priority.

XQUAD and Random Attack, on the other hand, achieved balanced Novelty scores of 0.7584 and 0.7502, respectively. These scores indicate that both methods effectively promote diversity while preserving a reasonable level of relevance and ranking accuracy. Their ability to enhance recommendation diversity without excessively disrupting user satisfaction or system performance makes them more suitable for scenarios that require a balance between diversity and quality.

The findings indicate that XQUAD and Random Attack (Attack Size 10, Filler Size 10) are the most effective methods for achieving a balance across key metrics: popularity bias reduction, recommendation relevance, ranking accuracy, and diversity.

XQUAD stands out as the most robust debiasing method for scenarios that prioritize long-tail exposure and diversity. It significantly enhances the visibility of less popular items while maintaining acceptable levels of relevance and ranking accuracy. However, compared to the base model, XQUAD does entail a slight compromise in these metrics, making it a trade-off between diversity and user satisfaction.

Random Attack, on the other hand, delivers a well-rounded performance across all evaluation metrics. It achieves a commendable balance between reducing popularity bias and maintaining recommendation relevance and ranking accuracy. This makes Random Attack an excellent choice for systems that need to address bias without significantly sacrificing user experience.

In contrast, the base model excels in relevance and ranking accuracy, offering the highest scores in these metrics without any interventions. However, its limited ability to reduce popularity bias and promote diversity highlights its shortcomings in addressing fairness in recommendations.

These findings suggest that XQUAD and Random Attack can be strategically employed based on the specific goals of the recommendation system. XQUAD is better suited for applications with a strong emphasis on diversity and fairness, while Random Attack is ideal for systems aiming for a balanced approach that reduces bias while maintaining relevance and ranking quality.



5. DISCUSSION

In this section, we interpret the findings from the experimental evaluation of debiasing methods and attack models, discussing their impacts on recommendation quality, popularity bias reduction, and diversity. This analysis addresses the trade-offs involved in applying these methods, examining their effectiveness and limitations in balancing user relevance and diversity in recommendations. The findings offer insights into the strengths and weaknesses of each approach, suggesting guidelines for selecting the best method depending on the goals of the recommendation system.

5.1. Effectiveness of Debiasing Methods and Attack Models

The results indicate that XQUAD and Random Attack (Attack Size 10, Filler Size 10) are the most effective methods for reducing popularity bias, each excelling in different areas. XQUAD achieves the highest APLT score, significantly increasing long-tail item exposure, which is critical for diversifying recommendations and mitigating popularity bias. However, XQUAD's impact on F1-Score and nDCG suggests a slight compromise in recommendation relevance and ranking accuracy. Random Attack, by contrast, maintains a balanced approach, achieving strong scores in APLT, F1, and nDCG, making it highly suitable for applications where both relevance and bias reduction are essential. This balance highlights Random Attack's versatility, especially in contexts where recommendation quality needs to be preserved alongside bias mitigation.

5.2. Trade-Offs in Popularity Bias Reduction and Recommendation Quality

One of the main trade-offs identified in the findings is the balance between popularity bias reduction and recommendation relevance. While methods such as XQUAD and ERPMul improve diversity by emphasizing long-tail items, they may cause slight declines in F1-Score and nDCG, reflecting a potential impact on recommendation accuracy and user satisfaction. VAR maximizes diversity with an exceptionally high Novelty score but compromises heavily on relevance and ranking accuracy, as shown by its low F1-Score and nDCG. This suggests that debiasing methods aimed at extreme diversity may negatively impact recommendation quality, indicating that choosing the right method involves assessing the acceptable trade-offs between user relevance and long-tail exposure.

5.3. Impact of Metrics on Debiasing Goals

The performance metrics used in this study, namely APLT, F1-Score, nDCG, and Novelty, highlight different aspects of popularity bias reduction and recommendation quality. APLT and Novelty emphasize diversity and long-tail exposure, indicating each method's ability to reduce bias and promote less popular items. However, methods that prioritize high APLT and Novelty values, such as XQUAD and VAR, may also show lower F1-Score and nDCG values, underscoring a reduction in recommendation relevance. This observation reinforces that each metric captures a specific goal of the

recommendation system, suggesting that the metric prioritization should align with the intended application. For example, a platform focused on user discovery might prioritize Novelty, while a commercial recommendation system might require a balance across all metrics.

5.4. Comparison of Debiasing Methods and Their Practical Implications

The analysis reveals that debiasing methods like XQUAD and ERPMul offer distinct advantages for recommender system applications that prioritize diversity and bias reduction. XQUAD's ability to mitigate popularity bias while preserving recommendation relevance makes it highly effective for settings where balanced recommendations are needed. In contrast, ERPMul and VAR, with their strong emphasis on diversity, are ideal for systems that prioritize novelty over relevance. However, the decreased F1-Score and nDCG of ERPMul and VAR imply that these methods are better suited for applications that can tolerate lower relevance in favour of increased diversity. Random Attack emerges as a powerful alternative for applications that require a balance between recommendation quality and popularity bias reduction, underscoring the potential of attack models to enhance system fairness.

5.5. Insights into Choosing Between Debiasing Methods and Attack Models

Overall, this discussion underscores that selecting an optimal debiasing or attack method requires understanding the trade-offs each approach presents. For applications seeking high diversity with some bias reduction, debiasing methods like XQUAD and ERPMul are effective options. For balanced performance across diversity, relevance, and ranking accuracy, Random Attack provides a robust solution, making it ideal for contexts where a comprehensive approach is needed. The findings suggest that combining debiasing and attack models can be a promising avenue for future research, especially in complex recommendation environments where multiple objectives must be met.

6. CONCLUSIONS

This thesis set out to address the challenge of popularity bias in recommender systems by evaluating various debiasing methods and attack models and analysing their effects on recommendation quality, diversity, and bias reduction. Through systematic experimentation, we aimed to determine how different approaches impact key metrics—APLT, F1-Score, nDCG, and Novelty—and to identify the most effective strategies for achieving fairness and diversity in recommendations without compromising relevance. Based on the findings, we can draw the following conclusions.

6.1. Summary of Key Findings

Our results demonstrated that XQUAD and Random Attack (Attack Size 10, Filler Size 10) were the most effective models for mitigating popularity bias, each excelling in different areas. XQUAD achieved the highest APLT score, significantly increasing long-tail item exposure and demonstrating superior performance in reducing popularity bias, particularly in applications requiring high diversity. However, the method’s impact on F1-Score and nDCG indicated a slight compromise in recommendation relevance, which may affect user experience in some contexts. In contrast, Random Attack achieved a balanced performance across APLT, F1-Score, and nDCG, making it an ideal approach for applications needing to retain recommendation quality while minimizing bias. Other debiasing methods, such as ERPMul and VAR, showed strong performance in promoting novelty but at the cost of lower relevance and ranking accuracy, indicating that they are more suited for applications prioritizing extreme diversity.

6.2. Practical Implications for Recommender System Design

The study’s findings offer valuable insights for recommender system design. XQUAD is particularly useful for platforms focused on content discovery or fairness, where long-tail item exposure and diversity are prioritized. For applications where relevance and bias reduction must be balanced, Random Attack presents a promising alternative to traditional debiasing methods, demonstrating that attack models can be repurposed to enhance fairness effectively. Commercial and user-centric platforms that seek to maximize engagement while promoting fairness could consider combining methods like XQUAD with attack models, such as Random Attack, to achieve a customized approach that balances all metrics. The findings underscore the importance of aligning debiasing strategy with the platform’s objectives and user experience goals, tailoring the choice of method to the desired balance of diversity and relevance.

6.3. Limitations and Future Directions

This study has several limitations that suggest future research directions. First, the experiments were limited to the MovieLens dataset and one recommendation algorithm (VAECF), which may constrain the generalizability of the findings across diverse datasets and models. Future research could broaden this scope by examining multiple datasets and

recommendation algorithms, offering a more comprehensive understanding of how different methods perform across varied contexts. Additionally, while we explored the potential of attack models as debiasing tools, combining multiple debiasing methods with attack-based strategies could yield further improvements in popularity bias reduction without sacrificing recommendation quality. Exploring these combinations and examining the effects of other debiasing and attack techniques could provide new pathways for building fairer and more effective recommendation systems.

In conclusion, this thesis demonstrates that different debiasing methods and attack models have distinct strengths in reducing popularity bias and enhancing recommendation diversity. Each method serves a unique role depending on the specific goals of the recommender system. XQUAD and Random Attack offer the most balanced approaches, achieving meaningful bias reduction while maintaining recommendation quality. This study provides practical insights and a foundation for future research on enhancing fairness, diversity, and user satisfaction in recommendation technologies.

7. REFERENCES

- Abdollahpouri, H. 2020. Popularity bias in recommendation: A multi-stakeholder perspective. In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20)*, pp. 1-5.
- Abdollahpouri, H., Burke, R., Mobasher, B. 2017. Controlling popularity bias in learning-to-rank recommendation. In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*, ACM, New York, pp. 42-46.
- Adomavicius, G., & Kwon, Y. 2011. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, 24(5): 896–911.
- Adomavicius, G., & Tuzhilin, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749.
- Anderson, C. 2006. *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion, New York, 267 p.
- Bobadilla, J., Serradilla, F., & Bernal, J. 2010. Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems*, 23(6): 550–554.
- Breese, J.S., Heckerman, D., Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Francisco, pp. 43-52.
- Brynjolfsson, E., Hu, Y.J., & Smith, M.D. 2006. From niches to riches: Anatomy of the long tail. *Sloan Management Review*, 47(4): 67-71.
- Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4): 331-370.
- Burke, R., Mobasher, B., Williams, C. 2006. Classification features for attack detection in collaborative recommender systems. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, pp. 542-547.
- Cantador, I., Brusilovsky, P., Kuflik, T. 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec 2011). In: *Proceedings of the 5th ACM Conference on Recommender Systems*, ACM, New York, pp. 387-388.
- Celma, O., & Cano, P. 2008. From hits to niches: Recommender systems and their impact on sales diversity. In: *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, ACM, New York, pp. 1-8.
- Chen, L., Wu, W., & He, L. 2018. VAR: Value-aware ranking in recommender systems. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI*, Stockholm, pp. 3877-3883.
- Gantner, Z., Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. 2011. MyMediaLite: A free recommender system library. In: *Proceedings of the 5th ACM Conference on Recommender Systems*, ACM, New York, pp. 305-308.

- Gomez-Uribe, C.A., Hunt, N. 2016. The Netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4): 13.
- Gunes, I., Kaleli, C., Bilge, A., & Polat, H. 2014. Shilling attacks against recommender systems: A comprehensive survey. *Artificial Intelligence Review*, 42(4): 767–799.
- Harper, F.M., & Konstan, J.A. 2015. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4): 19.
- Järvelin, K., & Kekäläinen, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4): 422-446.
- Lam, S.K., Riedl, J. 2004. Shilling recommender systems for fun and profit. In: *Proceedings of the 13th International Conference on World Wide Web*, ACM, New York, pp. 393-402.
- Liang, D., Krishnan, R.G., Hoffman, M.D., & Jebara, T. 2018. Variational autoencoders for collaborative filtering. *Proceedings of the 2018 World Wide Web Conference (WWW 2018)*, ACM, New York, pp. 689-698.
- Lops, P., De Gemmis, M., Semeraro, G. 2011. Content-based recommender systems: State of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (Eds.), *Recommender Systems Handbook*, Springer, Boston, pp. 73-105.
- Mehta, B., & Nejdl, W. 2008. Attack resistant collaborative filtering. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New York, pp. 75-82.
- Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. 2007. Effective attack models for shilling item-based collaborative filtering systems. *Proceedings of the 2007 WebKDD and 1st SNA-KDD Workshop on Web Mining and Social Network Analysis*, ACM, New York, pp. 21-30.
- O'Mahony, M. P., Hurley, N. J., & Silvestre, G. C. 2004. Recommender systems: Attack types and strategies. *Proceedings of the 20th National Conference on Artificial Intelligence*, 334-339.
- Park, Y.J., & Tuzhilin, A. 2008. The long tail of recommender systems and how to leverage it. In: *Proceedings of the 2008 ACM Conference on Recommender Systems*, ACM, New York, pp. 11-18.
- Pazzani, M.J., Billsus, D. 2007. Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (Eds.), *The Adaptive Web*, Springer, Berlin, pp. 325-341.
- Ricci, F., Rokach, L., Shapira, B. 2015. *Recommender Systems Handbook*. Springer, Boston, 1050 s.
- Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. 2011. *Recommender Systems Handbook*. Springer, Boston, 842 s.
- Sanders, R. 1987. The Pareto Principle: Its Use and Abuse. *Journal of Services Marketing*, 1(2): 37-40.
- Sarwar, B., Karypis, G., Konstan, J., Riedl, J. 2001. Item-based collaborative filtering

- recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, ACM, New York, pp. 285-295.
- Schafer, J.B., Konstan, J.A., Riedl, J. 2001. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1): 115-153.
- Van Rijsbergen, C.J. 1979. *Information Retrieval*. Butterworth-Heinemann, London, 208 p.
- Vargas, S., & Castells, P. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, ACM, New York, pp. 109-116.
- Wang, S., Gong, M., Li, H., & Yang, J. 2016. Multi-objective optimization for long tail recommendation. *Knowledge-Based Systems*, 104: 145–155.
- Wu, X., Lin, T., Chen, W., Chen, T., & Yang, Q. 2022. Adversarial debiasing: A new perspective on tackling bias in recommender systems. In: *Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, pp. 1330-1340.
- Yalcin, E., & Bilge, A. 2021. Enhanced re-ranking procedure for mitigating popularity bias in recommender systems. *Information Processing & Management*, 58(3): 102506.
- Zhou, T., Kuscsik, Z., Liu, J.G., Medo, M., Wakeling, J.R., Zhang, Y.C. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10): 4511-4515.

8. APPENDICES

Appendix 1. Code and Implementation

The GitHub repository contains all relevant code and resources for this study: <https://github.com/ozgegurel/personalized-recommendation-system>

The F1_score, nDCG evaluation metrics, and VAE CF recommendation algorithm are implemented using the Cornac library: <https://github.com/PreferredAI/cornac>



CURRICULUM VITAE

ÖZGE GÜREL

EDUCATION

Master's degree 2021-2024	Akdeniz University Institute of Natural and Applied Sciences, Department of Computer Engineering, Antalya
Bachelor's degree 2012-2016	Akdeniz University Faculty of Engineering, Department of Electrical & Electronics Engineering, Antalya

WORK EXPERIENCE

Software Developer 06/2018 - 07/2021	Bilkav Eğitim Danışmanlık A.Ş. Antalya
Electrical & Electronics Engineer 02/2018 - 06/2018	Imecar Electronics Antalya

PROJECTS

1. Using Virtual Reality Technology to Improve the Efficiency of the Visual Evoked Potential (VEP) Test; Project Manager; Supported by TUBITAK 2209-A Support Program for Domestic Research Projects of University Students.

HONOURS AND AWARDS

1. Boeing - Istanbul Technical University 2015 Unmanned Aerial Vehicle Design and Competition; 1st Place Award.
2. Turkish Education Foundation (TEV) Scholarship; Awarded during undergraduate studies at Akdeniz University for academic excellence and financial need.

ORGANIZATIONAL SKILLS

1. Vice-President, R&D Society, Akdeniz University; Successfully led and organized the 1st InnoProject Mediterranean Award-winning Project Competition in 2015 and the 2nd InnoProject Mediterranean Award-winning Project Competition in 2016.