People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Chadli Bendjedid University - El Tarf

Computer Science Department

THESIS

For the award of the Doctoral Degree (Third Cycle)

Diversity in Recommandation Systems for E-commerce

Field: Computer Science

Specialization: Artificial Intelligence and its Applications

Prepared by: Abderaouf BAHI

Defended on: April 21, 2026

Before the jury composed of:

President:	Dr. Amel Ziani	Chadli Bendjedid University - El Tarf
Supervisor:	Dr. Ibtissem Gasmi	Chadli Bendjedid University - El Tarf
Co-Supervisor:	Dr. Sassi Bentrad	National Higher School of Cyber Security
Examiner:	Prof. Mohamed Tarek Khadir	Badji Mokhtar University - Annaba
Examiner:	Dr. Sabri Ghazi	Badji Mokhtar University - Annaba
Examiner:	Dr. Amina Makhlouf	Chadli Bendjedid University - El Tarf

Academic Year: 2025/2026

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Chadli Bendjedid - El Tarf

Département d'Informatique

THÈSE

Pour l'obtention du diplôme de docteur 3ème cycle

Diversité dans les Systèmes de Recommandation pour le Commerce Èlectronique

Filière: Informatique

Spécialité: Intelligence Artificielle et ses Applications

Préparé par: Abderaouf BAHI

Soutenue le: 21 avril 2026

Devant le jury composé de:

Président :	Dr. Amel Ziani	Université Chadli Bendjedid - El Tarf
Rapporteur :	Dr. Ibtissem Gasmi	Université Chadli Bendjedid - El Tarf
Co-Rapporteur :	Dr. Sassi Bentrad	École Nationale Supérieure - Cybersécurité
Examinateur :	Prof. Mohamed Tarek Khadir	Université Badji Mokhtar - Annaba
Examinateur :	Dr. Sabri Ghazi	Université Badji Mokhtar - Annaba
Examinateur :	Dr. Amina Makhlouf	Université Chadli Bendjedid - El Tarf

Année Universitaire: 2025/2026

Abstract

Recommender systems have become essential components of modern digital platforms, particularly in e-commerce, online services, and personalized content delivery. Their primary objective is to infer user preferences from historical interactions and provide accurate, relevant, and engaging suggestions. However, conventional recommendation models still struggle to balance accuracy with diversity, often reinforcing repetitive content, and they offer limited guarantees regarding the integrity and confidentiality of user data.

This dissertation introduces two complementary contributions that address these challenges from both algorithmic and security perspectives. The first contribution, *MycGNN*, is a mycelium-inspired graph neural network designed to enhance recommendation diversity without compromising accuracy. By dynamically reweighting edges during the recommendation stage and employing a probabilistic exploration–exploitation mechanism, MycGNN mimics adaptive foraging behaviors observed in natural mycelial networks. Experiments on the Retail Rocket and MovieLens-1M datasets show that while some state-of-the-art GNN models surpass MycGNN in raw accuracy, MycGNN consistently delivers superior intra-list diversity while maintaining competitive accuracy.

The second contribution, *SFNN*, proposes a security-aware recommendation framework integrating accuracy, diversity, and data protection. SFNN combines a graph neural network with a regularized variational autoencoder whose loss function encourages latent-space dispersion to promote diversity. An adaptive fusion mechanism consolidates the strengths of both networks. Additionally, blockchain-based storage ensures data integrity and confidentiality. Experiments on the Retail Rocket, clothing, and MovieLens-1M datasets show SFNN achieving high accuracy (78.13%, 82.44%, and 86.16%) along with substantial diversity (46.82%, 37.78%, and 47.65%).

Together, these contributions demonstrate how biologically inspired modeling, graph learning, and decentralized security can jointly advance next-generation recommender systems that are robust, diverse, and trustworthy.

Keywords: recommender systems; bio-inspired algorithms; diversity; data security; accuracy; e-commerce.

ملخص

أصبحت أنظمة التوصية من الركائز الأساسية في المنصّات الرقمية الحديثة، خاصة في مجال التجارة الإلكترونية، والخدمات عبر الإنترنت، وتقديم المحتوى المخصّص. ويتمثّل هدفها الرئيسي في استنتاج تفضيلات المستخدمين اعتمادًا على تفاعلاتهم السابقة، وتقديم اقتراحات دقيقة وملائمة وجذّابة. غير أنّ النماذج التقليدية لأنظمة التوصية لا تزال تعاني من صعوبة تحقيق توازن فعّال بين الدقّة والتنوّع، إذ غالبًا ما تؤدي إلى تعزيز المحتوى المتكرر، كما توفّر ضمانات محدودة فيما يخص نزاهة بيانات المستخدمين وسريّتها.

تقدّم هذه الأطروحة مساهمتين متكاملتين لمعالجة هذه الإشكاليات من منظورين خوارزمي وأمني في آنٍ واحد. تتمثّل المساهمة الأولى في نموذج قائم على الشبكات العصبية البيانية، مستوحى من البنية الحيوية لشبكات الفطريات الأرضية، ويهدف إلى تعزيز تنوّع التوصيات دون الإخلال بمستوى الدقّة. يعتمد هذا النموذج على إعادة وزن العلاقات بين العناصر بشكل ديناميكي أثناء مرحلة التوصية، إلى جانب آلية احتمالية لتحقيق توازن بين الاستكشاف والاستغلال، بما يحاكي سلوكيات البحث التكيفية الملاحظة في الأنظمة الحيوية الطبيعية. وقد أظهرت التجارب التطبيقية على مجموعات بيانات خاصة بالتجارة الإلكترونية وتقييم الأفلام أنّ هذا النموذج، رغم تفوّق بعض النماذج المتقدّمة من حيث الدقّة الصرفة، يقدّم باستمرار مستوى أعلى من التنوّع داخل قائمة التوصيات مع الحفاظ على أداء تنافسي من حيث الدقّة.

أمّا المساهمة الثانية، فتقترح إطارًا آمنًا لأنظمة التوصية يدمج بين الدقّة، والتنوّع، وحماية البيانات. يجمع هذا الإطار بين شبكة عصبية بيانية ونموذج ترميز تلقائي احتمالي منظّم، حيث تم تصميم دالة الخسارة بطريقة تشجّع على تباعد التمثيلات الكامنة بهدف تعزيز التنوّع. كما يعتمد الإطار على آلية دمج تكيفية توحّد نقاط القوّة في كلا النموذجين. إضافة إلى ذلك، يضمن تخزين البيانات باستخدام تقنية السجلّات الموزّعة سلامة المعلومات وحمايتها من التلاعب، مع الحفاظ على سريّة بيانات المستخدمين. وقد بيّنت النتائج التجريبية على عدّة مجموعات بيانات واقعية أنّ هذا الإطار يحقّق مستويات مرتفعة من الدقّة، إلى جانب درجات معتبرة من التنوّع.

تُظهر هذه الأطروحة، من خلال مساهماتها المختلفة، كيف يمكن لتكامل النمذجة المستوحاة من الأنظمة البيولوجية، وتعلّم الرسوم البيانية، وآليات الأمان اللامركزية أن يساهم في تطوير جيل جديد من أنظمة التوصية، يتميّز بالمتانة، والتنوّع، والموثوقية، ويستجيب لمتطلّبات التطبيقات الرقمية الحديثة.

الكلمات المفتاحية: أنظمة التوصية؛ الخوارزميات المستوحاة بيولوجيًا؛ التنوّع؛ أمن البيانات؛ الدقّة؛ التجارة الإلكترونية.

Résumé

Les systèmes de recommandation jouent un rôle fondamental dans les plateformes numériques modernes, notamment dans le commerce électronique et les services en ligne. Leur objectif principal est d'identifier les préférences des utilisateurs à partir de leurs interactions passées afin de fournir des suggestions précises, pertinentes et engageantes. Toutefois, les approches traditionnelles rencontrent encore des difficultés à concilier la précision et la diversité, tout en offrant des garanties limitées en matière de sécurité et de confidentialité des données.

Cette thèse présente deux contributions complémentaires visant à dépasser ces limites. La première contribution, *MycGNN*, est un réseau neuronal graphique inspiré des réseaux mycéliens. Il introduit un réajustement dynamique des arêtes et un mécanisme probabiliste exploration–exploitation permettant d'améliorer significativement la diversité des recommandations tout en maintenant une précision compétitive. Les expérimentations menées sur Retail Rocket et MovieLens-1M montrent que, même si certains modèles surpassent MycGNN en précision brute, celui-ci offre une diversité nettement supérieure.

La seconde contribution, *SFNN*, propose un cadre de recommandation conscient des enjeux de sécurité, combinant précision, diversité et protection des données. Le modèle fusionne un réseau neuronal graphique avec un autoencodeur variationnel régularisé afin de favoriser un espace latent plus dispersé et donc plus diversifié. Un mécanisme de fusion adaptatif est utilisé pour agréger les prédictions, tandis qu'une couche basée sur la blockchain assure l'intégrité et la confidentialité des données. Les expériences sur trois jeux de données montrent que SFNN atteint des niveaux élevés de précision et de diversité simultanément.

Ces deux contributions démontrent comment les approches bio-inspirées, l'apprentissage sur graphes et les technologies décentralisées peuvent ensemble faire progresser la nouvelle génération de systèmes de recommandation robustes, diversifiés et fiables.

Mots-clés : systèmes de recommandation ; algorithmes bio-inspirés ; diversité ; sécurité des données ; précision ; commerce électronique.

Publications Included in this Thesis

Publication Included in Chapter 3

Title: *MycGNN: Enhancing Recommendation Diversity in E-Commerce Through Mycelium-Inspired Graph Neural Network*

Authors: Abderaouf Bahi, Ibtissem Gasmi, Sassi Bentrad, and Ramzi Khantouchi

Published: 21 October 2024

DOI: 10.1007/s10660-024-09911-9

Journal: *Electronic Commerce Research*

Publisher: Springer

ISSN (Online): 1572-9362

H-index: 58

Impact Factor: 4.7

Quartile: Q1

Publication Included in Chapter 4

Title: *SFNN: A Secure and Diverse Recommender System Through Graph Neural Network and Regularized Variational Autoencoder*

Authors: Abderaouf Bahi, Ibtissem Gasmi, Sassi Bentrad, Mohamed Walid Azizi, Ramzi Khantouchi, and Meryem Uzun-Per

Published: 29 November 2025

DOI: 10.1016/j.knosys.2025.114983

Journal: *Knowledge-Based Systems*

Publisher: Elsevier

ISSN (Online): 1872-7409

H-index: 188

Impact Factor: 7.6

Quartile: Q1

Acknowledgments

Praise be to Allah who granted me the strength, perseverance, and clarity to complete this dissertation. Peace and blessings be upon His final Messenger, Mohamed.

First and foremost, I extend my deepest appreciation to my supervisor, Dr. Ibtissem Gasmi, for her continuous guidance, scientific rigor, availability, and unwavering support throughout this research journey. Her insightful comments, constructive feedback, and high academic standards have significantly shaped the direction and quality of this thesis.

I am equally grateful to my co-supervisor, Dr. Sassi Bentrad, for his valuable advice, encouragement, and thoughtful discussions that greatly enriched this work. His expertise and trust were essential to the accomplishment of this research.

I would also like to express my sincere thanks to the President of the dissertation committee, Dr. Amel Ziani, for accepting to chair the jury and for the time and attention devoted to evaluating this thesis.

My heartfelt gratitude goes to the examiners:

- Prof. Mohamed Tarek Khadir
- Dr. Sabri Ghazi
- Dr. Amina Makhlouf

Their expertise, critical analysis, and recommendations have greatly strengthened this thesis.

Finally, I would my gratitude to the faculty members of the Department of Computer Science for providing a stimulating academic environment, as well as all colleagues and collaborators who, directly or indirectly, contributed to this research through discussions, exchanges, and support.

Dedication

This work is lovingly dedicated to my parents, Prof. Tahar Bahi and Prof. Amel Ourici, whose unwavering love, constant prayers, sacrifices, and continuous support have been the foundation of my achievements. Their belief in me has always been my greatest source of strength and motivation.

To my brother, Islem Bahi, for his encouragement and presence throughout this journey.

To my friends and close companions, whose support, motivation, and sincere wishes accompanied me during every stage of this work.

I also wish to honor the memory of my late grandmother, *Fatma Alirachedi*, whose kindness, prayers, and unwavering spirit remain a lasting source of inspiration. May Allah grant her mercy and the highest rank in Paradise.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations and Symbols

M_t	The model at time t.
M_{t-1}	The model at time $t-1$.
$F(M_{t-1})$	The function applied to the model at time $t-1$ to update it.
$h_v^{(l+1)}$	The hidden state of a node v at layer $l+1$.
$N(v)$	The neighborhood of the node v.
$c_{u,v}$	The connection strength or feature between nodes u and v.
$W^{(l)}$	The weight matrix at layer l.
$h_u^{(l)}$	The hidden state of a node u at layer l.
v_j	The item characteristic vector for node j.
β	The parameter controlling the extent of dynamic edge reweighting.
$P(i)$	The probability of recommending item i.
$Q(i)$	The quality score of item i.
I	The set of all items.
$H^{(l+1)}$	The output of the $(l+1)$-th layer in the graph convolutional network.
W_k	The learnable weight matrix corresponding to the k-th redundant pathway.
A_k	The adjacency matrix corresponding to the k-th pathway.
$w_{ij}(t)$	The weight of the edge between nodes i and j at iteration t.
u_i	The user interest vector for node i.
x_i	The original value of the feature.
$\min(X)$	The minimum value of the feature across the dataset.
$\max(X)$	The maximum value of the feature across the dataset.
x'_i	The imputed value of the feature.
n	The number of observed values in the dataset.
x	The categorical feature.
$\mathrm{Mode}(x)$	The mode, or most frequent category, for the attribute.
t_i	The original timestamp.
$\min(T)$	The earliest timestamp in the dataset.
$\max(T)$	The latest timestamp in the dataset.
D	Preprocessed data to be encrypted.
C	Encrypted data.
C'	Doubly encrypted data.

CI'	The ii-th fragment of the doubly encrypted data.
K	The private key for data encryption.
K_2	The second encryption key used for the additional layer of encryption.
Encrypt_K	The data encryption function with key K.
Decrypt_K	The data decryption function with key K.
$\text{Price}(C_i')$	The cost of storing the i-th fragment.
StorageTime	The duration of time the data is stored.
B_{new}	The new block added to the blockchain.
M	The set of miners validating the transaction.
$h_i^{(l+1)}$	The updated feature vector of node v_i at layer $l+1$.
σ	A non-linear activation function.
AGG	The aggregation function that combines the features of the neighboring nodes $N(v_i)$.
$h_j^{(l)}$	The feature vector of a neighboring node v_j at layer l.
z	The latent variable sampled from the probabilistic distribution.
μ	The mean vector from the encoder.
ϵ	A random noise vector sampled from a standard normal distribution.
z_i, z_j	Latent representations of different items in the recommendation set.
w_{GNN}	The weight given to the GNN prediction score.
w_{VAE}	The weight given to the VAE prediction score.
FP	False Positives, negative cases incorrectly predicted as positive.
FN	False Negatives, positive cases incorrectly predicted as negative.
TP	True Positives, correctly predicted positive cases.
TN	True Negatives, correctly predicted negative cases.
item-j	The j-th recommended item in the list.
$T-i$	Timestamp of the i-th recommended item.
$T-j$	Mean timestamp of all recommended items.

General Introduction

Recommender systems have become indispensable components of modern digital ecosystems, shaping user experiences across e-commerce platforms, streaming services, social networks, and numerous other environments. Their primary role is to identify and suggest relevant items by exploiting heterogeneous sources of information, including behavioral patterns, item descriptors, and contextual signals. Over the last decade, the field has witnessed a remarkable evolution, transitioning from traditional content-based and collaborative filtering paradigms to sophisticated models powered by deep learning, graph representation learning, and biologically inspired mechanisms. As recommender systems mature, however, new challenges emerge that transcend the conventional pursuit of accuracy.

A central issue concerns the limited diversity produced by many recommendation algorithms. By optimizing primarily for precision, these systems repeatedly expose users to highly similar items, reinforcing redundancy and creating filter bubbles that restrict novelty and discovery. Such behavior can hinder user satisfaction, reduce long-term engagement, and limit the potential for serendipitous interactions. Ensuring a balanced trade-off between relevance and diversity therefore represents a fundamental requirement for the next generation of recommender systems, motivating the need for algorithms capable of dynamic exploration, adaptive learning, and context-aware decision making.

Simultaneously, the growing dependence on personal data raises significant concerns regarding security, privacy, and the integrity of recommendation processes. As systems accumulate sensitive information and operate in environments increasingly exposed to adversarial threats, the risk of data leakage, manipulation, and unauthorized inference becomes critical. Ensuring trustworthy recommendations requires robust, transparent, and tamper-resistant architectures. Advances in decentralized technologies—particularly blockchain—offer promising avenues to enhance the reliability and auditability of recommendation pipelines by enabling immutable logs, distributed consensus, and privacy-preserving interactions.

The convergence of these two dimensions—diversity and security—constitutes a complex yet essential frontier for modern recommendation research. Addressing both simultaneously requires models capable of reasoning over large-scale relational data, adapting to evolving user preferences, and withstanding real-world constraints such as sparsity, scalability, and adversarial environments. The present thesis explores this multifaceted landscape by introducing new approaches inspired by biological networks, graph-based learning, and decentralized security frameworks to develop recommender systems that are not only effective but also responsible and resilient.

Within this perspective, the work undertaken in this thesis proposes two original contributions. The first is a mycelium-inspired Graph Neural Network, denoted *MycGNN*, which integrates adaptive edge reweighting and a controlled exploration–exploitation mechanism to naturally promote diversity while maintaining recommendation relevance. This design draws inspiration from the adaptive foraging behavior of mycelium networks, known for their ability to efficiently explore their environment and dynamically adjust their structural pathways. The second contribution is a security-oriented recommendation framework, *SFNN*, which embeds cryptographic primitives and decentralized components to ensure that user–item interactions remain confidential, verifiable, and resistant to tampering. Together, these contributions aim to demonstrate that diversity enhancement and security enforcement can be simultaneously achieved through principled algorithmic design.

The motivations underlying this work stem from the growing expectation for personalized yet varied recommendations, coupled with a pressing need for transparent, secure, and privacy-compliant systems. Technically, many existing algorithms still struggle with the exploration–exploitation dilemma, especially in large and sparse datasets where discovering new relevant items is essential. Graph-based approaches—particularly GNNs—offer a powerful framework to address this challenge by capturing complex relational dependencies. Meanwhile, traditional centralized architectures remain vulnerable to manipulation and data exposure, calling for secure-by-design solutions that integrate decentralized verification, encrypted communication, and robust neural architectures. Responding to these gaps, this thesis aims to contribute to the development of recommendation systems that uphold user trust while ensuring high-quality, diverse, and ethically aligned recommendations.

The thesis is organized as follows. The first chapter introduces the foundations of recommender systems and highlights the scientific and practical motivations driving this research. The second chapter provides the necessary theoretical background, covering classical and modern recommendation techniques, diversity-oriented methods, graph learning principles, nature-inspired frameworks, and evaluation methodologies. The third chapter develops the proposed *MycGNN* model, detailing its mycelium-inspired mechanisms and its capacity to enhance diversity through adaptive graph learning. The fourth chapter presents the *SFNN* framework, focusing on its ability to integrate secure filtering mechanisms and decentralized technologies to strengthen the integrity of recommendation processes. Finally, the thesis concludes with a synthesis of the achieved contributions and outlines future directions for advancing secure and diverse recommender systems.

Chapter 1: Foundations and Challenges of Recommender Systems

Contents

1.1 Introduction

Recommender systems (RSs) are essential components of modern intelligent information systems, designed to assist users in coping with the massive growth of digital content [1,2]. By exploiting user preferences, interaction histories, and contextual information, RSs provide personalized recommendations that facilitate decision-making and information discovery [3]. These systems are widely used in domains such as e-commerce and digital media platforms, where they significantly enhance user engagement and satisfaction, and are increasingly applied in areas including education, healthcare, and scientific research [4–6].

This chapter introduces the fundamental concepts of recommender systems, including data representations, main algorithmic approaches, and evaluation criteria. It establishes the theoretical and methodological foundations necessary for understanding the advanced recommendation models proposed in the following chapters.

1.2 General Motivation

The rapid proliferation of online content has given rise to a pervasive challenge known as *information overload* [7]. Modern users are confronted with an overwhelming quantity of choices, from millions of products on e-commerce platforms to vast repositories of media, research articles, and digital services. In this environment, decision-making becomes cognitively taxing, often leading to user fatigue, suboptimal choices, or disengagement from the platform [8]. Recommender systems provide a principled approach to mitigate these issues by intelligently filtering and presenting items that are most relevant to the user, thereby reducing cognitive load and enhancing overall satisfaction [9].

From an operational and business perspective, RSs represent a strategic capability. Empirical analyses consistently demonstrate that a significant fraction of user interactions on major platforms can be traced back to recommendation engines [4]. By tailoring product and content exposure to individual interests, RSs directly contribute to increased click-through rates, higher conversion rates, and extended user retention [10]. Furthermore, they facilitate the discovery of long-tail items, allowing platforms to monetize niche or less popular content while maintaining a high level of user satisfaction [11].

Beyond commercial advantages, the societal implications of RSs are profound. Personalized recommendations can enhance learning outcomes by guiding learners to appropriate educational resources [12], improve healthcare delivery through tailored treatment suggestions [6], and support scientific progress by surfacing highly relevant publications and datasets [13]. As such, recommender systems are not merely technical artifacts but are also catalysts of economic, educational, and societal value, bridging the gap between massive digital information and human decision-making [14].

In light of these motivations, it becomes evident that the study of RSs is both timely and essential. Understanding their theoretical underpinnings, practical implementations, and evaluation methodologies is crucial for designing systems that are effective, fair, and aligned with user-centric objectives [15]. This work seeks to contribute to this understanding by exploring state-of-the-art methodologies, emerging trends, and challenges in the development and deployment of modern recommender systems.

1.3 Foundations of Recommender Systems

1.3.1 Definition

A recommender system can be formally defined as an intelligent software system whose primary objective is to estimate the relevance of items for individual users and generate personalized suggestions [16, 17]. Let $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users and $I = \{i_1, i_2, \ldots, i_n\}$ the set of items. The core task of a recommender system is to approximate an unknown utility function [18]

$$r : U \times I \to \mathbb{R}, \tag{1.1}$$

which maps user-item pairs to preference scores. The predicted preference or rating can be expressed as:

$$\hat{r}_{u,i} = f(u, i, X), \tag{1.2}$$

where $\hat{r}_{u,i}$ denotes the estimated preference of user u for item i, $f(\cdot)$ represents the recommendation model, and X contains all auxiliary information, including historical interactions, user attributes, item features, and contextual data [19].

In practice, recommender systems typically generate a ranked list of top-N items for each user by sorting the predicted scores [20]:

$$\text{Top-N}(u) = \text{argmax}_{i \in I}\, \hat{r}_{u,i}, \quad |\text{Top-N}(u)| = N. \tag{1.3}$$

The system architecture usually follows a multi-stage pipeline comprising data collection, preprocessing, feature extraction, model training, and recommendation generation [10], as illustrated in Figure 1.1.

Figure 1.1: General Architecture of a Recommender System. The system integrates user and item data, interactions, preprocessing steps, and multiple recommendation algorithms to generate top-N personalized outputs.

1.3.2 Input Data Types

Recommender systems rely on heterogeneous data sources that capture complementary aspects of user behavior and item characteristics [21]. Accurate modeling requires both proper representation and integration of these diverse data types.

Behavioral Data

Behavioral data captures the explicit and implicit interactions between users and items [22]. Explicit feedback includes ratings, likes, and reviews, while implicit feedback includes clicks, views, purchases, or dwell time. These interactions can be represented as a user-item interaction matrix:

$$R = [r_{u,i}]_{m \times n}, \quad r_{u,i} \in \mathbb{R} \cup \{\emptyset\}, \tag{1.4}$$

where $r_{u,i}$ denotes the observed interaction of user u with item i, and $\emptyset$ indicates missing interactions. In real-world scenarios, R is typically sparse, as most users engage with only a small subset of items [23]. Techniques such as matrix factorization, low-rank approximation, neighborhood-based methods, or graph propagation are employed to address sparsity and predict missing interactions [24].

Demographic Data

Demographic information provides structured user attributes such as age, gender, location, education level, and profession [25]. These can be encoded as a feature vector:

$$\mathbf{d}_u = [d_{u,1}, d_{u,2}, \ldots, d_{u,p}]^\top \in \mathbb{R}^p, \tag{1.5}$$

where p is the number of demographic attributes. Categorical variables can be transformed via one-hot encoding, embeddings, or binning. Demographic data is particularly valuable in cold-start scenarios, enabling the system to infer initial preferences for new users based on population-level patterns [26].

Content-Based Data

Content-based data describes intrinsic properties of items, such as textual descriptions, images, audio signals, or categorical metadata [21]. Each item i can be represented by a feature vector:

$$\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]^\top \in \mathbb{R}^d, \tag{1.6}$$

where d is the dimensionality of the item feature space. Modern systems often leverage representation learning techniques, including neural embeddings, convolutional networks, or autoencoders, to automatically extract high-level features [27]. The relevance score for content-based recommendations is often calculated via similarity functions, such as the cosine similarity [28]:

$$\mathrm{sim}(\mathbf{p}_u, \mathbf{x}_i) = \frac{\mathbf{p}_u \cdot \mathbf{x}_i}{\|\mathbf{p}_u\| \, \|\mathbf{x}_i\|}, \tag{1.7}$$

where $\mathbf{p}_u$ represents the user profile vector derived from past interactions or preferences.

Social and Contextual Data

Social and contextual data augment traditional user-item interactions [29]. Social data captures relationships among users, such as friendships, trust, or group memberships, typically modeled as a graph [30]:

$$G = (U, E), \quad E \subseteq U \times U, \tag{1.8}$$

where edges E encode social links. Contextual data encompasses temporal information (timestamps, seasonality), spatial location, device type, or current user intent [31]. Incorporating such data allows context-aware and socially-informed recommendations, improving both relevance and user trust.

For instance, a context-aware recommender can adjust predicted scores according to time of day or location [29]:

$$\hat{r}_{u,i}^{(c)} = f(u, i, X, C), \tag{1.9}$$

where C represents contextual variables. Similarly, social propagation techniques can enhance predictions [30]:

$$\hat{r}_{u,i}^{(s)} = \alpha \hat{r}_{u,i} + (1 - \alpha) \frac{1}{|N(u)|} \sum_{v \in N(u)} \hat{r}_{v,i}, \tag{1.10}$$

where $N(u)$ is the neighborhood of user u in the social graph, and $\alpha \in [0, 1]$ balances personal vs. social influence.

Table 1.2 show the overview of recommender system input data types.

Table 1.2: Overview of Recommender System Input Data Types.

Data Type	**Description**
Behavioral	User interactions with items, explicit (ratings, likes) or implicit (clicks, views, purchases).
Demographic	User attributes such as age, gender, location, education, profession; used in cold-start scenarios.
Content-Based	Item features, including text, images, audio, metadata; used for similarity-based recommendation.
Social	Relationships among users, including friendships and trust networks; modeled as a graph.
Contextual	Temporal, spatial, device, or situational information influencing user preferences.

1.3.3 Recommendation Techniques

Content-Based Filtering (CBF)

Content-Based Filtering recommends items by matching user profiles with item attributes [21,32]. A user profile $\mathbf{p}_u$ is typically constructed by aggregating the features of items previously consumed by the user, often weighted by interaction strength:

$$\mathbf{p}_u = \frac{1}{|I_u|} \sum_{i \in I_u} r_{u,i} \mathbf{x}_i \tag{1.11}$$

The relevance of an item is computed using a similarity measure, most commonly cosine similarity [28]:

$$\text{sim}(\mathbf{p}_u, \mathbf{x}_i) = \frac{\mathbf{p}_u \cdot \mathbf{x}_i}{\|\mathbf{p}_u\| \|\mathbf{x}_i\|} \tag{1.12}$$

Items with the highest similarity scores are ranked and recommended. Content-based approaches are interpretable and effective in cold-start item scenarios; however, they may suffer from over-specialization, limiting recommendation diversity [33].

Collaborative Filtering (CF)

Collaborative Filtering exploits collective user behavior to generate recommendations, under the assumption that users with similar preferences in the past will exhibit similar preferences in the future [16].

Memory-based collaborative filtering   Memory-based CF directly operates on the interaction matrix by computing similarities between users or items [34]. In user-based CF, the predicted rating is computed as:

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{sim}(u,v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in N(u)} |\text{sim}(u,v)|} \tag{1.13}$$

where $\bar{r}_u$ is the average rating of user u, and $N(u)$ denotes the set of nearest neighbors of u. Despite their simplicity, memory-based methods face scalability and sparsity issues in large-scale systems [35].

Model-based collaborative filtering   Model-based CF addresses scalability and sparsity by learning latent representations of users and items [36]. Matrix factorization approximates the interaction matrix as:

$$R \approx U \cdot V^T \tag{1.14}$$

where $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$ are low-dimensional latent factor matrices, with $k \ll \min(m,n)$. The predicted rating is then given by:

$$\hat{r}_{u,i} = \mathbf{u}_u^\top \mathbf{v}_i + b_u + b_i \tag{1.15}$$

where b_u and b_i are user and item bias terms to capture global tendencies [23]. This framework forms the basis of many state-of-the-art recommender systems [24]. As illustrated in Figure 1.2, matrix factorization is a model-based collaborative filtering technique that approximates the sparse user–item interaction matrix by decomposing it into low-dimensional latent user and item representations. This approach allows the recommender system to predict unknown ratings by capturing hidden patterns in user preferences and item characteristics.

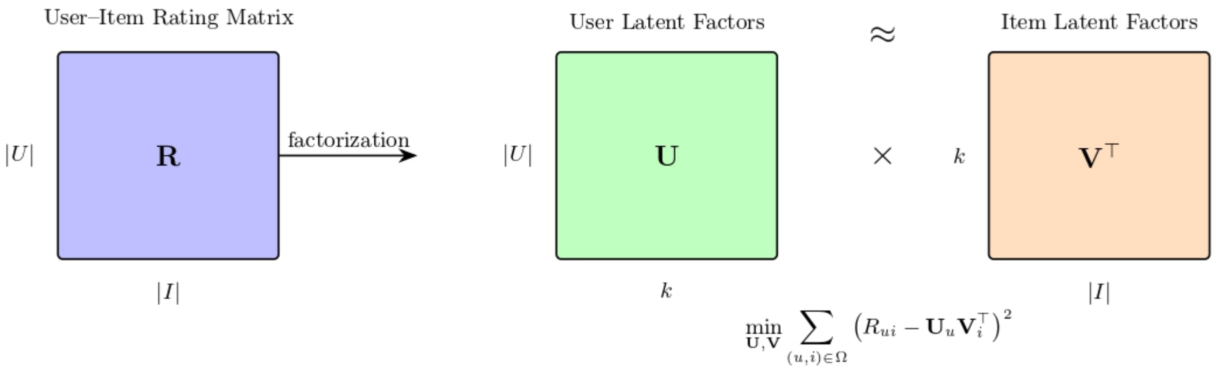

Figure 1.2: Matrix Factorization for Collaborative Filtering. The original user–item rating matrix $\mathbf{R}$ *is approximated by the product of the user latent factor matrix* $\mathbf{U}$ *and the item latent factor matrix* $\mathbf{V}^\top$*, with* k *latent dimensions.*

As illustrated in Figure 1.3, content-based and collaborative filtering represent two fundamental paradigms in recommender systems, each exploiting different sources of information to generate recommendations [2].

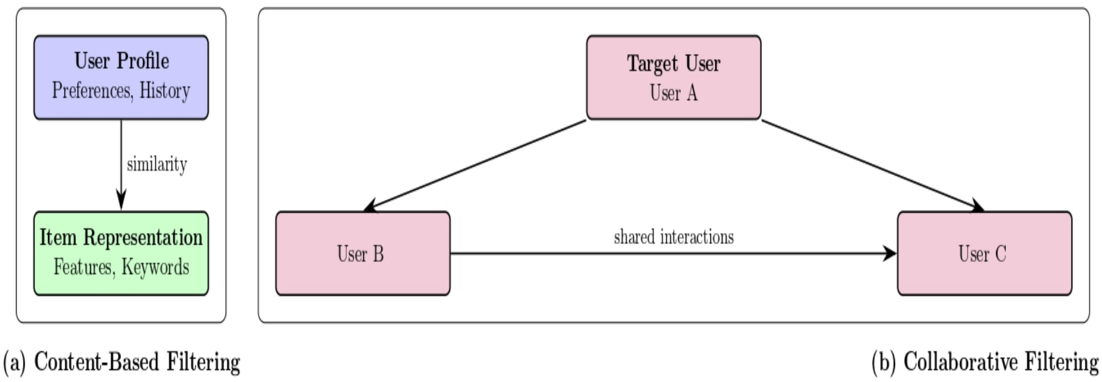

Figure 1.3: Comparison of Content-Based Filtering (CBF) and Collaborative Filtering (CF).

Hybrid Recommendation Approaches

Hybrid recommender systems combine multiple techniques to overcome the limitations of individual approaches, such as sparsity, cold-start, and lack of diversity [18]. Table 1.3 summarizes the most common hybridization strategies, widely adopted in industrial systems [37]. As illustrated in Figure 1.4, hybrid recommender systems combine multiple recommendation models to improve accuracy, coverage, or diversity [10]. Different hybridization strategies, such as weighted, switching, and cascade approaches, determine how the outputs of content-based and collaborative filtering models are combined to produce personalized recommendations.

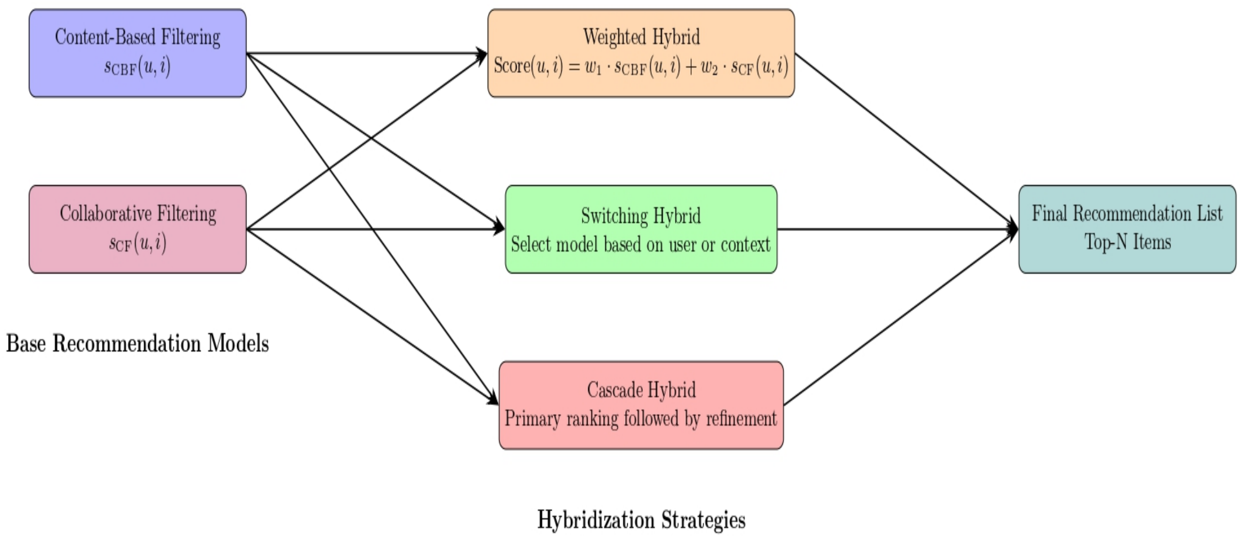

Figure 1.4: Typology of Hybrid Recommender Systems. Weighted hybrids combine model scores linearly, switching hybrids select a model based on context, and cascade hybrids apply multiple models in a sequential refinement process.

Table 1.3: Types of Hybrid Recommender Systems

Hybrid Strategy	**Description**
Weighted	Combined predictions
Switching	Context-based selection
Mixed	Multiple systems together
Feature combination	Model features as input
Cascade	Sequential refinement

1.3.4 Recommendation Goals and Quality Dimensions

Primary Objective of the Recommender System

The primary objective of a recommender system is to deliver highly relevant, personalized recommendations that maximize user satisfaction and engagement [38,39]. This can be formalized as the optimization of an expected utility function over the set of users U and items I:

$$\max_{\hat{R}} \sum_{u \in U} \sum_{i \in I} \mathcal{U}(u, i, \hat{r}_{u,i}), \tag{1.16}$$

where $\mathcal{U}(u, i, \hat{r}_{u,i})$ represents the utility of recommending item i to user u, based on the predicted rating or relevance $\hat{r}_{u,i}$, and $\hat{R}$ is the predicted rating matrix. The utility function $\mathcal{U}$ can be defined in various ways, reflecting not only immediate accuracy but also long-term engagement, user satisfaction, and trustworthiness [40, 41].

In practical deployments, recommender systems often aim to optimize multi-objective functions that balance several factors [42]:

$$\mathcal{U}_{\text{total}} = \lambda_1 \mathcal{U}_{\text{accuracy}} + \lambda_2 \mathcal{U}_{\text{diversity}} + \lambda_3 \mathcal{U}_{\text{novelty}} + \lambda_4 \mathcal{U}_{\text{serendipity}}, \tag{1.17}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in [0, 1]$ are weighting coefficients allowing the system designer to prioritize different recommendation goals.

Quality Criteria of the Recommendation Output

The quality of recommendation output is multi-dimensional, reflecting the complex trade-offs inherent in modern recommender system design [43]. Beyond accuracy, other critical quality dimensions include:

- **Diversity**: Measures how dissimilar the recommended items are within a single list, preventing monotony and broadening user exposure [44]. The intra-list diversity (ILD) is commonly used:

$$\text{ILD}(L_u) = \frac{2}{|L_u|(|L_u| - 1)} \sum_{i,j \in L_u, i<j} (1 - \text{sim}(i, j)), \tag{1.18}$$

  where L_u denotes the top-N recommendation list for user u, and $\text{sim}(i, j)$ is a similarity measure between items i and j.

- **Novelty**: Captures the unexpectedness or newness of recommended items for a user [45]. A common measure is the inverse popularity:

$$\text{Novelty}(i) = -\log \frac{|U_i|}{|U|}, \tag{1.19}$$

  where $|U_i|$ is the number of users who have previously interacted with item i.

- **Coverage**: Represents the fraction of items or users for which the system can provide recommendations [46]:

$$\text{Coverage} = \frac{|\{i \in I : \exists u \in U, \hat{r}_{u,i} \neq \emptyset\}|}{|I|}. \tag{1.20}$$

- **Serendipity**: Reflects the system's ability to recommend items that are both relevant and unexpected, introducing users to content they would not have discovered independently [47].

- **Robustness**: Measures the stability of the recommendation output under noisy, incomplete, or adversarial input data [48]. Formally, robustness can be quantified as the variance in predictions when small perturbations are introduced to the input:

$$\text{Robustness}(u,i) = 1 - \frac{\text{Var}(\hat{r}_{u,i}^{\text{perturbed}})}{\text{Var}(\hat{r}_{u,i})}. \tag{1.21}$$

These quality dimensions often involve trade-offs. For instance, increasing diversity or novelty can sometimes reduce predictive accuracy, highlighting the need for multi-objective optimization in modern RSs [49].

1.3.5 Real-World Applications of Recommender Systems

E-commerce

In e-commerce, recommender systems enable personalized product discovery, targeted marketing, and cross-selling strategies [50, 51]. By analyzing user browsing patterns, purchase histories, and demographic profiles, RSs improve conversion rates, revenue, and customer retention. For instance, a platform may compute a personalized purchase probability:

$$P(\text{purchase}_{u,i}) = \sigma(\hat{r}_{u,i} + \mathbf{d}_u^\top \mathbf{x}_i), \tag{1.22}$$

where σ is a logistic function, $\mathbf{d}_u$ is the user demographic vector, and $\mathbf{x}_i$ is the item feature vector. Empirical reports indicate that personalized recommendations may account for 35% of total sales on major platforms [50].

Online Media and Entertainment

Media platforms utilize RSs to personalize content consumption, including movies, music, and streaming videos [52, 53]. Accurate recommendations reduce user churn and increase metrics like watch-time or session duration. Hybrid approaches, combining collaborative filtering with content-based and graph-based techniques, are widely adopted in platforms like Netflix and Spotify [53].

Education and E-learning

Recommender systems in education enable adaptive learning paths, suggesting courses, exercises, or resources tailored to a learner's current knowledge state [54, 55]. For example, intelligent tutoring systems use performance metrics to update user profiles dynamically and recommend the next best learning activity. Mathematically, an adaptive learning RS may maximize a utility function:

$$\hat{r}_{u,i}^{\text{learning}} = f(\mathbf{p}_u, \mathbf{x}_i, t), \tag{1.23}$$

where t represents temporal progress, $\mathbf{p}_u$ is the learner's proficiency vector, and $\mathbf{x}_i$ is the skill or knowledge content vector of item i [54].

Healthcare, News, and Others

In healthcare, RSs assist clinicians and patients by recommending treatments, medications, or lifestyle interventions based on patient profiles, historical responses, and contextual factors [56, 57]. News platforms leverage RSs to deliver personalized information feeds, improving engagement and retention [58]. Additional applications extend to travel, social networking, professional networking, and smart city services, demonstrating the ubiquity and societal impact of recommender systems [59].

1.4 Evaluation of Recommender Systems

1.4.1 Evaluation Protocols

Offline Evaluation

Offline evaluation relies on historical datasets and predefined train-test splits [38, 60]. Standard metrics include:

- **Root Mean Squared Error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{u,i} - r_{u,i})^2}\,, \tag{1.24}$$

  where T is the test set, $\hat{r}_{u,i}$ the predicted rating, and $r_{u,i}$ the true rating of user u for item i.
- **Precision@K**:

$$\text{Precision@K} = \frac{|L_u \cap T_u|}{K}, \tag{1.25}$$

  where L_u is the top-K recommendation list for user u and T_u the set of relevant items.
- **Recall@K**:

$$\text{Recall@K} = \frac{|L_u \cap T_u|}{|T_u|}, \tag{1.26}$$

  where L_u is the top-K recommendation list for user u and T_u the set of relevant items.
- **Normalized Discounted Cumulative Gain (NDCG@K)**:

$$\text{NDCG@K} = \frac{1}{Z} \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{1.27}$$

  where rel_i is the relevance score of the i-th item in the top-K list and Z is a normalization constant for the ideal ranking [61].

Online Evaluation

Online evaluation measures system performance in real-world environments using A/B testing or interleaving methods [51, 58]. Metrics include click-through rate (CTR), conversion rate, session duration, or user retention. Online evaluation captures actual user behavior and satisfaction, complementing offline metrics to validate the practical effectiveness of recommender systems.

1.4.2 Accuracy-Oriented Metrics

Accuracy-oriented metrics focus on measuring how well a recommender system predicts user preferences or retrieves relevant items. These metrics are central to traditional RS evaluation and are widely used in both academic research and industrial benchmarking. The most common metrics are summarized below.

Precision

Precision measures the proportion of recommended items that are actually relevant to the user. It reflects the exactness of the recommendation list and is particularly important when the cost of presenting irrelevant items is high. Formally, it is defined as:

$$\text{Precision} = \frac{|\text{Relevant} \cap \text{Recommended}|}{|\text{Recommended}|}, \tag{1.28}$$

where Relevant is the set of relevant items for the user and Recommended is the set of items recommended.

In practice, Precision@K is often computed, considering only the top-K recommended items.

Recall

Recall measures the ability of the recommender system to retrieve all relevant items for a user, emphasizing completeness over exactness:

$$\text{Recall} = \frac{|\text{Relevant} \cap \text{Recommended}|}{|\text{Relevant}|}, \tag{1.29}$$

where Relevant is the set of relevant items and Recommended is the set of recommended items.

Recall@K is commonly used in top-K recommendation scenarios.

F1-Score

The F1-score provides a harmonic mean of precision and recall, offering a balanced evaluation when both metrics are equally important:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{1.30}$$

where Precision and Recall are defined as above.

NDCG (Normalized Discounted Cumulative Gain)

NDCG evaluates ranking quality by considering both the relevance of items and their positions in the recommendation list:

$$\text{NDCG} = \frac{DCG}{IDCG}, \quad DCG = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)}, \tag{1.31}$$

where rel_i is the relevance score of the item at rank i, N is the number of recommended items, and $IDCG$ is the ideal DCG.

RMSE / MAE

For explicit-feedback scenarios, rating prediction accuracy is often measured using RMSE or MAE:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i} (\hat{r}_{u,i} - r_{u,i})^2}, \quad MAE = \frac{1}{N} \sum_{u,i} |\hat{r}_{u,i} - r_{u,i}|, \tag{1.32}$$

where N is the total number of observed ratings, $\hat{r}_{u,i}$ is the predicted rating, and $r_{u,i}$ is the true rating of user u for item i.

Table 1.4 summarizes the most common accuracy-oriented evaluation metrics.

Table 1.4: Summary of Accuracy-Oriented Evaluation Metrics

Metric	**Focus**	**Typical Use Case**
Precision@K	Exactness	Top-K recommendation
Recall@K	Completeness	Retrieval-oriented tasks
F1-score	Balance	Comparative evaluation
NDCG@K	Ranking quality	Ordered recommendation lists
RMSE / MAE	Rating accuracy	Explicit feedback prediction

1.4.3 Diversity-Oriented Metrics

While accuracy is essential, it alone is insufficient to capture user satisfaction. Diversity-oriented metrics assess how varied, novel, and exploratory the recommendation lists are.

Intra-List Diversity

Intra-List Diversity (ILD) measures the dissimilarity among items within a recommendation list L:

$$ILD = \frac{2}{|L|(|L|-1)} \sum_{i,j \in L, i<j} (1 - sim(i,j)), \tag{1.33}$$

where L is the recommendation list and $sim(i,j)$ denotes the similarity between items i and j.

Coverage

Coverage evaluates the proportion of items that appear in at least one recommendation list across all users:

$$\text{Coverage} = \frac{|\bigcup_{u \in U} L_u|}{|I|}, \tag{1.34}$$

where U is the set of users, L_u the recommendation list for user u, and I the set of all items.

Novelty

Novelty quantifies how unfamiliar or unexpected recommended items are to users, often based on item popularity:

$$\text{Novelty}(i) = -\log_2 P(i), \tag{1.35}$$

where $P(i)$ is the probability that item i has been previously observed or interacted with by users.

Serendipity

Serendipity measures the extent to which recommendations are both relevant and surprising. It can be computed as the proportion of recommended items that are relevant but unexpected based on user history. While there is no universal formula, it combines novelty and accuracy to enhance discovery and long-term engagement.

Table 1.5 summarizes the most common diversity-oriented evaluation metrics.

Table 1.5: Summary of Diversity-Oriented Evaluation Metrics

Metric	Focus	Typical Use Case
Intra-List Diversity (ILD)	Heterogeneity	Variety in top-K
Coverage	Item-space utilization	Item exposure
Novelty	Unexpectedness	New items
Serendipity	Surprise	Unexpected relevance

1.5 Issues and Limitations in Recommender Systems

1.5.1 Data-Related Issues

Sparsity

User-item interaction matrices $R = [r_{u,i}]_{m\times n}$ are inherently sparse, especially in large-scale systems where most users interact with only a small subset of items. Sparsity reduces the availability of co-occurrence information, which is crucial for computing similarity measures [62]:

$$\text{sim}(u, v) = \frac{\sum_{i\in I_{uv}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i\in I_{uv}} r_{u,i}^2}\sqrt{\sum_{i\in I_{uv}} r_{v,i}^2}} \tag{1.36}$$

Here, I_{uv} represents the set of items co-rated by users u and v. Eq. (1) demonstrates that sparsity can lead to unreliable similarity estimates, degrading model performance [63].

Cold-Start Problem

The cold-start problem occurs when new users or items lack sufficient interaction history. Let U_{new} and I_{new} denote new users and items. Traditional collaborative filtering cannot compute predictions for these entities due to missing $r_{u,i}$ values [62]. Mitigation strategies include:

- Incorporating side information: $\mathbf{d}_u$ for users and $\mathbf{x}_i$ for items.
- Hybrid models combining content-based and collaborative approaches [64].
- Active learning to solicit initial feedback from new users.

1.5.2 Algorithmic Issues

Scalability

Scalability concerns arise when $|U|$ and $|I|$ are large. Matrix factorization, for instance, requires optimization over $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{n\times k}$, with complexity $O(k(m+n))$ per iteration.

Distributed computing frameworks and approximate nearest neighbor techniques are commonly used to handle millions of users and items efficiently [63].

Popularity Bias

Popularity bias occurs when recommender systems favor items with high global interaction counts [65]:

$$P(i) = \frac{\sum_{u \in U} \mathbf{1}_{r_{u,i} \neq \emptyset}}{|U|}, \tag{1.37}$$

where U is the set of all users, $r_{u,i}$ is the rating or interaction of user u with item i, and $\mathbf{1}_{r_{u,i} \neq \emptyset}$ is an indicator function equal to 1 if user u interacted with item i, 0 otherwise [66].

Overfitting

Overfitting happens when models capture noise instead of underlying preference patterns. In matrix factorization, regularization is applied:

$$\min_{U,V} \sum_{(u,i) \in \Omega} (r_{u,i} - \mathbf{u}_u^\top \mathbf{v}_i)^2 + \lambda(\|U\|_F^2 + \|V\|_F^2), \tag{1.38}$$

where Ω is the set of observed user-item interactions, $\mathbf{u}_u$ and $\mathbf{v}_i$ are latent vectors for user u and item i, U and V are matrices of all user and item vectors, and λ is the regularization parameter [63].

1.5.3 User-Centric Issues

Filter Bubbles and Lack of Diversity

Over-personalization may trap users in "filter bubbles", exposing them only to similar items [65]. Intra-list diversity (Eq. 6) and serendipity metrics can be used to quantify and mitigate this effect.

Lack of Explainability

Black-box models, particularly deep neural networks, often lack interpretability. Techniques such as attention visualization, feature importance scores, or natural language explanations can improve transparency and user trust [67].

Trust and User Satisfaction

Trust depends on perceived relevance, fairness, and explanation quality [68]. User satisfaction S_u can be modeled as a combination of accuracy, diversity, novelty, and explainability:

$$S_u = \alpha \cdot \text{Accuracy}_u + \beta \cdot \text{Diversity}_u + \gamma \cdot \text{Novelty}_u + \delta \cdot \text{Explainability}_u, \tag{1.39}$$

where S_u is the satisfaction of user u, $\alpha, \beta, \gamma, \delta$ are weighting coefficients, and Accuracy_u, Diversity_u, Novelty_u, Explainability_u are the corresponding metric values for user u.

1.5.4 Security and Ethical Concerns

Adversarial Attacks

Recommender systems are vulnerable to shilling and poisoning attacks, where malicious users inject fake interactions to manipulate predictions [69]. Detection often relies on anomaly scoring or graph-based analysis.

Privacy Issues

RSs rely on sensitive user data (demographics, behavior). Privacy-preserving approaches include differential privacy, federated learning, and data anonymization to prevent unauthorized profiling [69].

Bias and Fairness

Biases present in training data can propagate and amplify in predictions. Metrics such as demographic parity or equal opportunity can quantify fairness [70]:

$$\text{Fairness} = 1 - |\Pr(\hat{r}_{u,i} \geq \tau | g_u = a) - \Pr(\hat{r}_{u,i} \geq \tau | g_u = b)|, \tag{1.40}$$

where $\hat{r}_{u,i}$ is the predicted rating for user u and item i, τ is a relevance threshold, and $g_u \in \{a, b\}$ denotes the sensitive group attribute of user u.

Table 1.6 summarizes the key recommender system issues and mitigation strategies.

Table 1.6: Summary of Key Recommender System Issues and Mitigation Strategies

Issue	Impact	Mitigation Strategies
Sparsity	Poor similarity estimation	Side info, matrix factorization
Cold-start	Inability to recommend new items	Hybrid, content features
Scalability	High computation cost	Distributed, approximate methods
Popularity bias	Reduced diversity	Re-ranking, normalization
Overfitting	Poor generalization	Regularization, cross-validation
Filter bubbles	Limited exposure	Diversity, serendipity
Lack of explainability	Low user trust	Attention, NLP explanations
Adversarial attacks	Manipulated recommendations	Anomaly detection, robust models
Privacy issues	Data leakage	Federated, anonymization
Bias	Unfair outcomes	Fairness learning, post-processing

1.6 Emerging Trends in Recommender Systems

1.6.1 Context-Aware and Sequential Recommendation

Sequential models capture evolving user preferences over time. Let $S_u = \{i_1, i_2, ..., i_t\}$ be a sequence of past interactions. Recurrent neural networks, transformers, or temporal embeddings model $P(i_{t+1}|S_u)$, enhancing recommendation relevance [71, 72]. Transformers such as BERT4Rec have become influential for modeling bidirectional context in item sequences, enabling more effective next-item prediction [72]. Adaptive attention mechanisms and temporal contexts further improve sequential modeling [73].

1.6.2 Fair and Federated Recommendation

Federated learning (FL) enables decentralized model training directly on user devices, mitigating privacy risks by avoiding centralized data collection. Let each user device u possess local data D_u. FL optimizes a global objective:

$$\min_{\theta} F(\theta) = \sum_{u \in U} \frac{|D_u|}{|D|} F_u(\theta) \tag{1.41}$$

where $F_u(\theta)$ is the local loss on device u and $|D| = \sum_u |D_u|$. Eq. (1) shows the aggregation of local updates without sharing raw data. Federated recommendation research has studied privacy, security, and system heterogeneity challenges in decentralized recommender systems [74, 75].

Fairness-aware objectives can be incorporated by adding constraints or regularization terms to balance performance across user groups. For example, extending Eq. (5) to the federated setting:

$$\min_{\theta} F(\theta) + \lambda \sum_{g \in G} \left| \text{Accuracy}_g - \bar{\text{Accuracy}} \right| \tag{1.42}$$

where G represents demographic groups, Accuracy_g is the accuracy for group g, and λ balances fairness vs. overall accuracy [76].

Table 1.7 summarizes the key Advantages and Challenges in Fair and Federated Recommendation.

Table 1.7: Key Advantages and Challenges in Fair and Federated Recommendation

Aspect	**Advantages**	**Challenges**
Privacy	Data remains local	Communication overhead, heterogeneous devices
Fairness	Reduces group disparities	Requires fairness metrics and trade-offs
Scalability	Parallel training across devices	Limited local data, stragglers effect

1.6.3 Graph-Based Models (GNNs)

Graph Neural Networks (GNNs) leverage the user-item interaction graph $G = (U \cup I, E)$ to capture higher-order dependencies. Each node $v \in U \cup I$ has an embedding $\mathbf{h}_v^{(l)}$ at layer l, which is updated as:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \frac{1}{\sqrt{|\mathcal{N}(v)|}\sqrt{|\mathcal{N}(u)|}} W^{(l)} \mathbf{h}_u^{(l)}\right), \tag{1.43}$$

where $\mathcal{N}(v)$ is the set of neighbors of node v, $W^{(l)}$ is the weight matrix at layer l, and σ is a non-linear activation function. GNNs have been shown to outperform traditional collaborative filtering in capturing intricate patterns of interaction data by aggregating multi-hop neighborhood information [62, 77].

GNN-based RSs often outperform traditional CF in capturing complex patterns. Node embeddings can be used to predict ratings or generate top-K recommendation lists:

$$\hat{r}_{u,i} = \mathbf{h}_u^{(L)\top} \mathbf{h}_i^{(L)} \tag{1.44}$$

where L is the number of GNN layers.

As illustrated in Figure 1.5, graph-based recommender systems leverage the user-item interaction graph to capture higher-order dependencies between users and items. Each node in the graph is represented by an embedding $\mathbf{h}_v$, which is iteratively updated through neighborhood aggregation in Graph Neural Networks (GNNs). By stacking multiple layers, GNNs can model complex patterns across the graph, enabling accurate rating predictions and top-K recommendation generation.

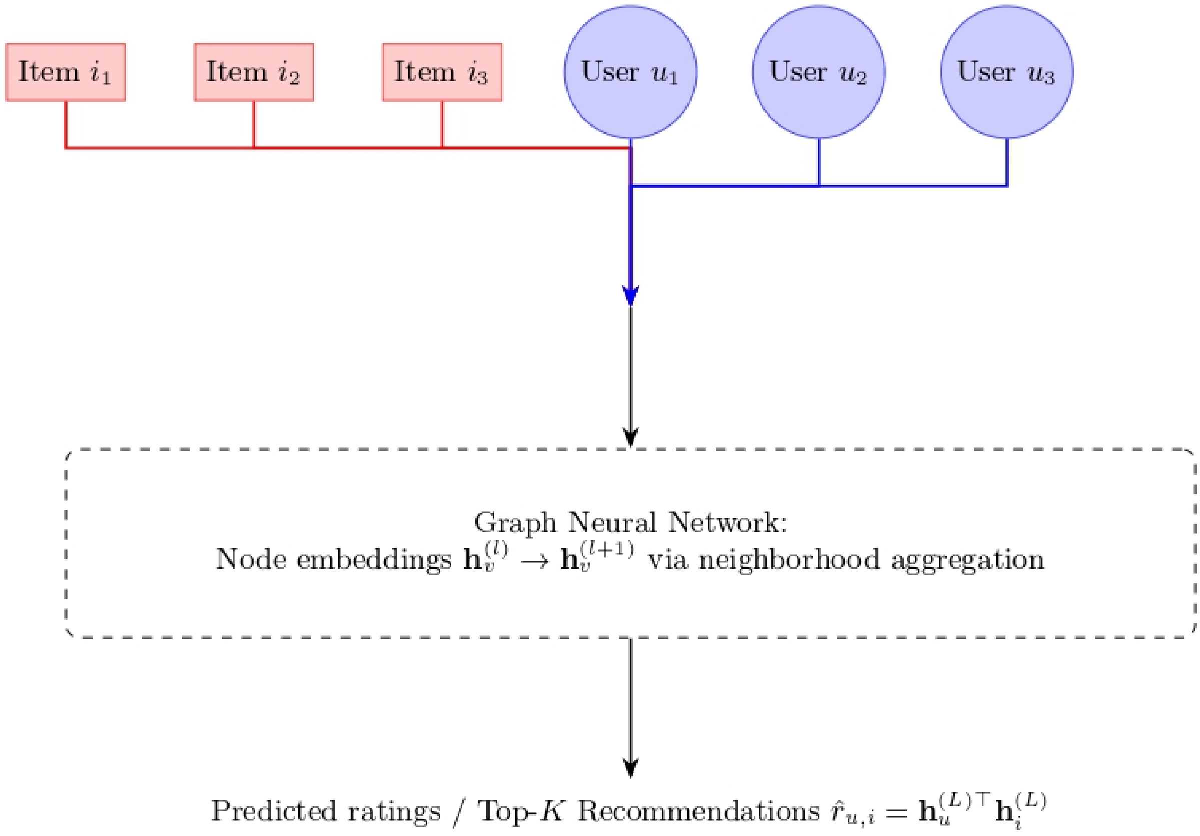

Figure 1.5: User-item bipartite graph with message passing in graph-based recommender systems. Node embeddings are iteratively updated through neighborhood aggregation to predict ratings or generate top-K recommendations.

1.6.4 Bio-Inspired Algorithms

Bio-inspired algorithms mimic natural phenomena to balance exploration and exploitation in recommendation tasks. Examples include:

- **Ant Colony Optimization (ACO):** Uses pheromone trails to guide item selection probabilities. For item i, the probability of recommendation is:

$$P(i) = \frac{\tau_i^\alpha \eta_i^\beta}{\sum_{j \in I} \tau_j^\alpha \eta_j^\beta} \tag{1.45}$$

  where τ_i is the pheromone level, η_i the heuristic desirability, and (α, β) control exploration vs. exploitation [78].

- **Mycelium-Inspired Models:** Mimic fungal networks to propagate influence through a sparse graph efficiently, adapting to changing user behavior while promoting diversity. As shown in Figure 1.6, the mycelium network propagates influence efficiently across sparse connections.

Table 1.8 summarizes the most common diversity-oriented evaluation metrics.

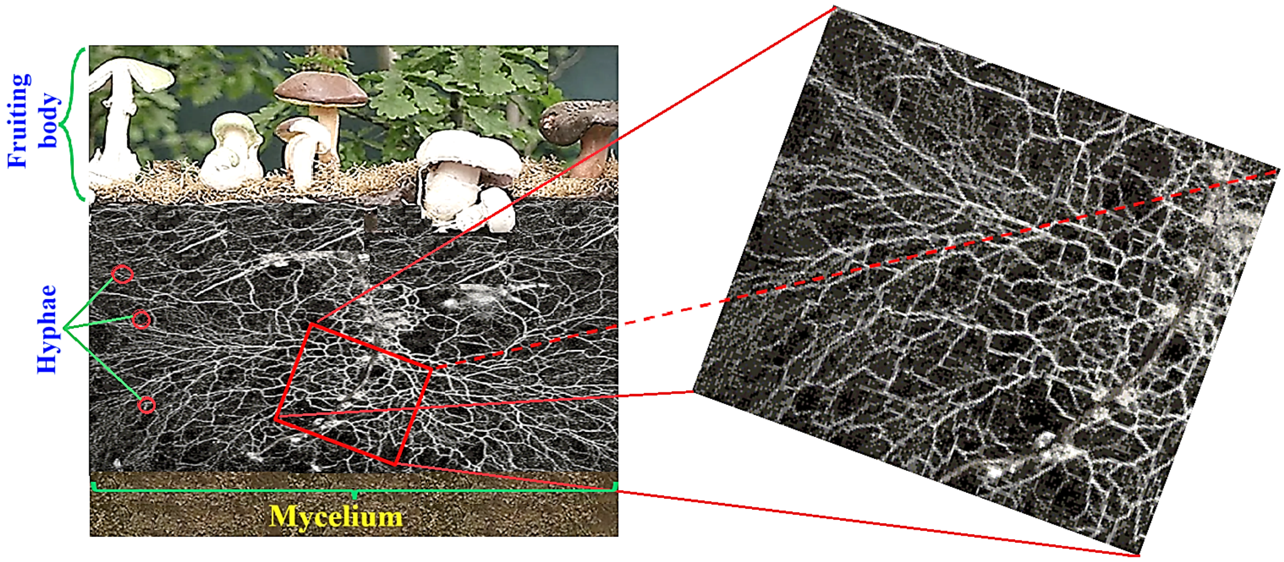

Figure 1.6: Real-world mycelium network structure, illustrating the highly connected yet sparse and adaptive nature of fungal hyphae. This biological inspiration informs the Mycelium-Inspired Recommender System.

Table 1.8: Comparison of Bio-Inspired Recommender Mechanisms

Algorithm	**Mechanism**	**Strengths**
ACO	Pheromone-based probabilistic selection	Balances exploration/exploitation
Mycelium-inspired	Diffusive network propagation	Promotes diversity, scalability

1.6.5 LLMs and Explainability

Large Language Models (LLMs) have recently emerged as powerful tools for enhancing recommender systems by providing not only high-quality predictions but also interpretable reasoning in natural language [71]. For a given user u and item i, an explanation $E_{u,i}$ can be generated as:

$$E_{u,i} = \mathrm{LLM}(\mathrm{UserProfile}_u, \mathrm{ItemFeatures}_i), \tag{1.46}$$

where $\mathrm{UserProfile}_u$ represents the historical interactions, preferences, and demographic information of user u, and $\mathrm{ItemFeatures}_i$ encodes textual descriptions, categorical metadata, and other relevant attributes of item i. LLMs offer contextual reasoning abilities that can be tailored to produce natural language explanations for recommendation decisions.

In addition to generating textual explanations, LLMs can be integrated with recommendation scores to create a weighted hybrid explanation-recommendation function:

$$\hat{r}_{u,i}^{\mathrm{hybrid}} = \alpha \cdot \hat{r}_{u,i}^{\mathrm{model}} + (1-\alpha) \cdot \hat{r}_{u,i}^{\mathrm{LLM}}, \tag{1.47}$$

where $\hat{r}_{u,i}^{\mathrm{model}}$ is the predicted score from the underlying recommendation model, $\hat{r}_{u,i}^{\mathrm{LLM}}$ is a relevance score inferred by the LLM reasoning, and $\alpha \in [0,1]$ balances accuracy and explainability.

Finally, the integration of LLMs enables interactive explainable recommendation. Users can query the system with natural language requests, and the LLM generates explanations while updating the recommendation list in real-time:

$$\hat{R}_u^{\text{interactive}} = f(\text{LLM}(Q_u, \text{UserProfile}_u, \text{ItemFeatures})), \tag{1.48}$$

where Q_u represents a user query in natural language, and $f(\cdot)$ maps the generated reasoning to a ranked recommendation list $\hat{R}_u^{\text{interactive}}$, enhancing transparency and user trust.

1.7 Conclusion

This chapter established the fundamental principles of recommender systems by presenting their core concepts, data foundations, evaluation criteria, and inherent challenges. Through the analysis of both accuracy- and diversity-oriented metrics, we highlighted the multidimensional nature of recommendation quality and the trade-offs involved in system design. Key issues such as sparsity, cold-start, popularity bias, and explainability were discussed to underline their practical impact on real-world deployments. In addition, recent advances including graph-based models and large language models were introduced to illustrate how modern recommender systems increasingly combine performance, adaptability, and interpretability. The figures, tables, and mathematical formulations provided in this chapter offer a structured framework for understanding these systems from both theoretical and practical perspectives. Building upon this foundation, the next chapter, *Chapter 2: Advances and Challenges in Recent Recommender System Models*, reviews the state of the art in recommender system research. It focuses on recent algorithmic developments, emerging trends, and open challenges, providing a critical analysis of related works that motivates the methodological choices adopted in this thesis.

Chapter 2: Advances and Challenges in Recent Recommender System Models

Contents

2.1 Introduction

Recommender systems (RSs) have become a cornerstone of modern information retrieval, e-commerce, and personalized digital services. Traditional approaches, such as collaborative filtering and content-based methods, while effective in capturing user preferences, often suffer from limitations including sparsity, cold-start problems, and popularity bias. In response, recent research has shifted towards hybrid and advanced models that aim to balance accuracy, diversity, and fairness in recommendations.

Recent advances in RSs leverage a range of sophisticated techniques. Graph-based models, particularly Graph Neural Networks (GNNs), enable the representation of complex user-item interactions and capture high-order relationships beyond immediate neighbors. Bio-inspired algorithms, drawing from natural phenomena such as swarm intelligence and mycelium growth, provide mechanisms for adaptive and dynamic exploration of the recommendation space. Concurrently, decentralized and privacy-preserving solutions, including federated learning and blockchain technologies, address critical concerns regarding data security, user privacy, and trustworthiness of recommendations.

This chapter systematically reviews state-of-the-art RS models, focusing on their underlying principles, methodological innovations, and real-world applicability. Highlighting emerging research directions that aim to advance the next generation of intelligent recommender systems.

2.2 Accuracy and Diversity in Recommender Systems

Balancing accuracy and diversity is crucial in RSs. Accuracy ensures recommendations are relevant to users, while diversity increases exposure to novel and less popular items. Consider a recommendation list L of length N. Accuracy-oriented metrics and Diversity-oriented metrics are defined as in Chapter 1. As illustrated in Figure 2.1, recommender systems inherently face a trade-off between accuracy and diversity [49]. While highly accurate models tend to focus on a narrow set of popular or highly relevant items, increasing diversity often requires exploring less similar or less frequently recommended items, which may reduce predictive accuracy [44].

Table 2.1: Techniques for Balancing Accuracy and Diversity in RSs

Technique	**Description**
Re-ranking	Adjust ranking to improve diversity while maintaining accuracy
Multi-objective optimization	Optimize for multiple metrics simultaneously
Hybrid filtering	Combine content-based and collaborative methods
Exploration strategies	Introduce stochastic item selection

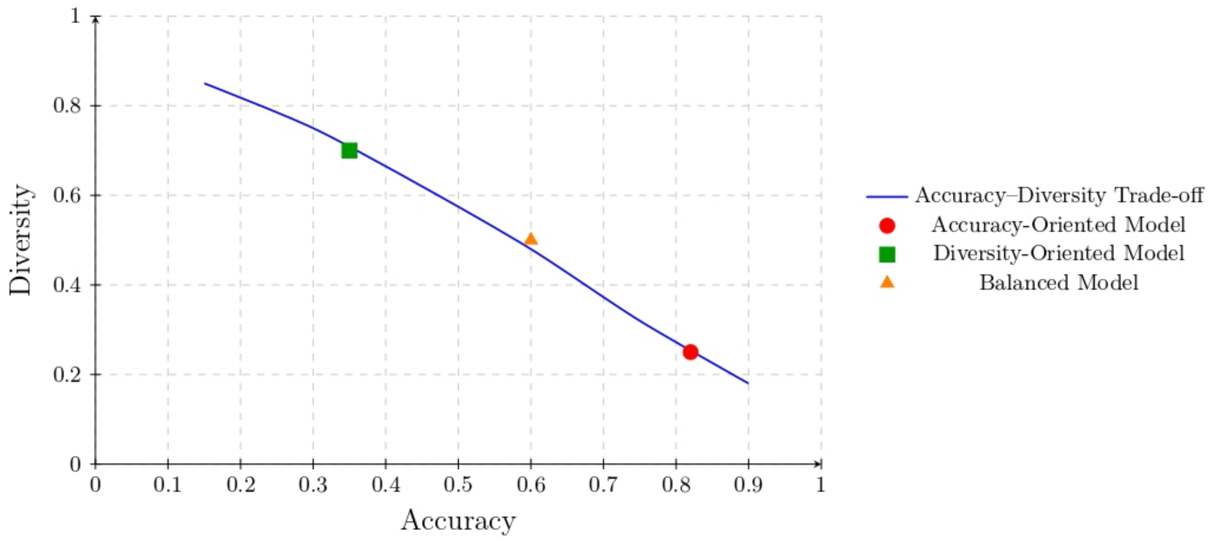

Figure 2.1: Trade-off between accuracy and diversity in recommender systems. Improving accuracy often leads to homogeneous recommendation lists, whereas enhancing diversity promotes exploration at the potential cost of reduced precision.

Improving user interaction with e-commerce platforms relies heavily on the effective deployment of various recommender system techniques [79]. Among the most widely adopted approaches are content-based filtering, collaborative filtering, and hybrid methods [80], all of which have been extensively applied in both research and practical scenarios to enhance recommendation accuracy and user satisfaction. To address the inherent limitations of traditional models, [81] proposed **COFIS**, a novel framework specifically designed for sequential implicit feedback-based collaborative filtering. This model integrates two distinct learning paradigms and two representation methodologies, rendering it adaptable to both factorization-based and deep learning-based techniques. By effectively managing noisy negative samples and accurately capturing dynamic user preferences, COFIS demonstrates superior performance on real-world datasets when compared to conventional methods. Similarly, [82] introduced **UI2vec**, a collaborative filtering method that embeds both users and items into a shared latent space via a feature extraction network, employing similarity measures to predict user interests. Building upon this, the authors developed **VUI2vec**, a generative model leveraging variational inference to represent users and items as separate Gaussian distributions. Extensive experiments on datasets such as TaFeng, MovieLens1M, and Netflix validate that UI2vec and VUI2vec consistently outperform baseline models, while also highlighting the significant role of hyperparameter tuning in model performance. In another contribution, [83] proposed **Rating Singular Value Decomposition (RSVD)**, a recommendation technique grounded in singular value decomposition (SVD). RSVD constructs user and item profiles that faithfully replicate the matrix structures typical of collaborative filtering, exploiting the mathematical properties of SVD. Comparative evaluations against standard collaborative filtering methods demonstrate RSVD's effectiveness in mitigating information overload and improving recommendation accuracy. Recommendation systems play a crucial role in enhancing user experience and practical e-commerce outcomes. Techniques such as collaborative filtering (CF) [84], user-based filtering [85], and content-based (CB) approaches [86] form the foundational toolkit, while more advanced hybrid strategies have been developed to overcome traditional limitations. For instance, [87] proposed a hybrid recommendation approach incorporating sentiment analysis, effectively addressing challenges such as cold-start problems and data sparsity while improving the relevance of recommendations in e-commerce contexts. Likewise, [88] introduced a matrix factorization method for predicting consumer preferences; although effective, this approach exhibits high computational complexity and limited generalizability. To further enhance recommendation quality, [89] developed **Multi-Factor Ranking (MF-R)**, which balances accuracy, novelty, diversity, and coverage by considering both predicted ratings and item popularity. In a similar vein, [90] proposed **CSSVD**, a contextual recommendation algorithm designed to tackle cold-starts and data insufficiency. CSSVD constructs user and item attribute matrices based on similarity metrics, employing the CWP similarity metric to integrate contextual information into the recommendation process. Addressing the challenge of long-tail item recommendation, [91] introduced an adaptive correlation clustering method that emphasizes adaptive clustering and timestamp prediction, thereby enhancing visibility for less popular items. Complementarily, [92] explored movie recommender systems and noted the limitations of conventional collaborative filtering in capturing evolving user preferences. Their proposed user segmentation strategy, based on movie enjoyment patterns and validated on the MovieLens-10M dataset, demonstrated substantial improvements in recommendation relevance. Further empirical studies reinforce the significance of advanced recommendation strategies. For example, [93] analyzed content-based filtering (CBF) recommendation systems in mobile app markets, utilizing social network analysis and econometric modeling. Their results indicate that CBF effectively promotes niche products, mitigating the dominance of highly popular apps and supporting long-tail market diversity. Additionally, the study highlights how quality signals from CBF systems influence consumer decision-making, potentially generating demand spillover from mainstream to less-known apps. In a related hybrid approach, [90] proposed a recommendation algorithm integrating user comment sentiment analysis with matrix decomposition techniques. By first calculating sentiment tendencies via an LSTM model and then incorporating these signals with user ratings, the method achieved more accurate and personalized recommendations. Experiments on BeerAdvocate, Modcloth, and Amazon datasets confirmed improvements in recommendation quality, while also revealing potential limitations in computational efficiency and dependence on sentiment polarity. A 2024 study by [94] examined how the length of recommendation lists affects network connectivity, retrieval accuracy, and diversity during user navigation. Using datasets such as Steam, MovieLens, and Amazon, the study demonstrated that while longer lists enhance object diversity, they may compromise retrieval accuracy by diverting attention from highly relevant items. This research emphasizes the delicate balance required in designing recommendation interfaces to optimize both exploration and precision. Lastly, [95] investigated the impact of financial restatements on the accuracy of sell-side recommendations across BRICS countries. Their analysis over an eleven-year period revealed that restatements generally reduce target price prediction accuracy, particularly in buy-and-hold recommendations, while occasionally improving sell recommendation accuracy. These findings highlight the critical influence of reliable financial data on investment decision-making and recommendation validity. Overall, these studies collectively underscore the pivotal role of recommender systems in e-commerce and beyond, demonstrating that the integration of advanced modeling techniques, hybrid approaches, and context-aware adaptations significantly enhances user engagement, recommendation relevance, and system effectiveness.

Table 2.2 summarizes the general aspects of the related works described above.

Diversity in recommendation systems refers to the variety of items presented to users, ensuring that suggestions are both relevant and cover a broad spectrum of options [96]. This aspect is crucial, as it prevents over-specialization, where systems repeatedly recommend items that are too similar or closely aligned with a user's previous interactions [97]. By integrating diversity into recommendations, systems enhance user engagement and satisfaction by exposing users to content they might not have discovered otherwise [98]. This enriches the user experience, encourages exploration, and increases users' time on the platform.

The importance of diversity is further underscored by its capacity to mitigate the *echo chamber effect*, wherein users are continuously presented with content reinforcing existing preferences, thereby limiting exposure to new ideas and options. Effective diversity implementation balances recommendation relevance with novelty [99, 100], fostering more dynamic and enriching interactions between users and the system. In contemporary applications, diverse recommendations are particularly valued in domains such as e-commerce and digital media, where content catalogs are vast and user preferences evolve rapidly.

In 2024, Dunlu Peng and Yi Zhou [101] proposed **LAP-SR**, a post-processing framework for session-based recommendation systems designed to mitigate the long-tail effect by incorporating personalized diversity. LAP-SR dynamically adjusts recommendation lists to include more long-tail items based on the computed diversity of user sessions, enhancing user engagement and addressing the filter bubble issue. The framework demonstrated effectiveness across multiple datasets, improving the exposure of less popular items without significantly compromising overall recommendation performance.

Similarly, Zihao Li et al. [102] introduced **Teddy**, a sequential recommendation model

Table 2.2: Accuracy in recommendation systems works summary

Reference	Year	Approach	Advantages	Limitations	Dataset
Hou [94]	2024	Recommendation network connectivity	Enhanced user navigation through better connectivity	Balance between recommendation depth and breadth needed	Steam, MovieLens, Amazon
Umar [95]	2023	Financial restatement impact analysis	Increased accuracy in financial environments	Specific to financial data, not generalizable	BRICS stock markets
Lin [81]	2023	Sequential Recommendation	Utilizing implicit feedback from learning user preferences	Scalability and generalization across domains	MovieLens1M, Steam, Gowalla, CDsVinyl
Alharbe [82]	2023	Word Embedding	Simultaneously learns the embedded representation of users and items	Lack of diversity	TaFeng, MovieLens1M, Netflix
Colace [83]	2022	Singular Value Decomposition	Exploit SVD in a Content-Based recommendation method	High computational complexity	MovieLens100k, MovieLens10M
Karn [87]	2023	Hybrid sentiment analysis	Improves targeting of recommendations, reduces cold start	Lack of diversity, impacted by polarity	Amazon fine foods reviews, Amazon movie reviews
Wang [88]	2023	Matrix factorization	In-depth analysis of sales records	Lack of generalization, high computational complexity	UCI UC Irvine Machine Learning Database
Jozani [93]	2023	User comment sentiment analysis	Improved accuracy through sentiment analysis	High computational cost, sensitivity to data quality	BeerAdvocate, Modcloth, Amazon
Rodpysh [90]	2023	CSSVD	Efficient combination of matrix techniques with similarity criteria	Scalability concerns with large datasets	IMDB, STS datasets
Lakshmi [91]	2023	Adaptive correlation clustering	Intelligent highlighting of neglected elements	High complexity	MovieLens 20M

that disentangles and models both dominant interest trends and scattered interest diversity. Combining a Temporal Convolutional Network (TCN) with a Multilayer Perceptron (MLP), Teddy processes user interactions to capture long-term interests and less frequent preferences separately. This dual-path approach provides nuanced, personalized recommendations, validated across several datasets, demonstrating superiority over state-of-the-art methods in capturing a comprehensive range of user preferences. However, managing dual pathways in real-time environments presents challenges in scalability and computational efficiency.

Huaizhen Kou et al. [103] proposed **DI-RAR**, a diversity-driven framework for automated web API recommendations focusing on implicit requirements. DI-RAR models functional dependencies among APIs using a weighted Mashup correlation graph and applies a dynamic planning retrieval system leveraging the Steiner tree algorithm. This method uncovers non-core functional requirements while enriching recommendation diversity. Although effective, the complexity of differentiating core versus non-core requirements and computational overhead pose challenges for large-scale deployment.

Alvise De Biasio et al. [104] addressed diversity in recommender systems for sensitive users, focusing on mitigating the influence of specific items that could bias user behavior. Their methodology refines both diversity and calibration of recommendations, ensuring fair exposure while reducing reinforcement of harmful behaviors in social network environments. Validation on real-world datasets demonstrated improvements in ethical recommendation outcomes and healthier user interactions.

Diversity, in general, ensures exposure to a wide range of novel items, countering over-specialization [105]. It fosters exploration, enhances engagement, and contributes to long-term user retention and satisfaction [106–108]. In e-commerce environments, with rapidly evolving preferences and extensive item catalogs, maintaining diversity is critical [109].

To address diversity challenges, multiple innovative models have been proposed. LAP-SR [101] constructs long-tail item sets and leverages session-level diversity via item embeddings. Teddy [102] disentangles general and scattered interests through TCNs and MLPs, achieving balanced recommendations. DI-RAR [103] employs self-attention and weighted graphs to diversify API recommendations, while clustering-based methods powered by evolutionary algorithms [110] balance intra-cluster similarity and inter-cluster diversity, enhancing personalization.

Finally, fairness-driven strategies [104] ensure that sensitive users receive diverse and calibrated exposure to influential items, promoting equitable and ethically responsible recommendations. Collectively, these works demonstrate the central role of diversity in enhancing user experience, mitigating echo chambers, and supporting fair and comprehensive recommendation practices.

Table 2.3 summarizes the general aspects of the related works described above.

Table 2.3: Diversity in recommendation systems works summary

Reference	Year	Approach	Advantages	Limitations	Dataset
Peng [101]	2024	Long-tail alleviation with personalized diversity	Enhances long-tail item exposure, improves user engagement	Complexity in adjusting recommendation algorithms	E-commerce datasets
Li [102]	2024	Sequential recommendation system	Separates and models interest trend and diversity	Complexity in managing dual pathways	E-commerce and social media datasets
Kou [103]	2023	Automated web API recommendations	Focuses on implicit requirements for API diversity	Computational complexity in real-time scenarios	Web API datasets
De Biasio [104]	2023	Recommendation for sensitive users	Balances influential item impact on user behavior	High specificity to user sensitivities, scalability challenges	Social network data
Peng [101]	2024	Long-tail alleviation with personalized diversity	Enhances long-tail item exposure, improves user engagement	Complexity in adjusting recommendation algorithms	Yoochoose, Diginetica, Tmall
Li [102]	2024	Sequential recommendation system	Separates and models interest trend and diversity	Complexity in managing dual pathways	Amazon Beauty, Amazon Sports, Amazon Toys, Steam
Kou [103]	2023	Automated web API recommendations	Focuses on implicit requirements for API diversity	Computational complexity in real-time scenarios	ProgrammableWeb API Dataset
Berbague [110]	2021	Genetic Algorithm-based Clustering	Enhances recommendation diversity and scalability	Potential increase in computational cost due to the optimization process	MovieLens100k, MovieLens1M
De Biasio [104]	2023	Recommendation for sensitive users	Balances influential item impact on user behavior	High specificity to user sensitivities, scalability challenges	myPersonality
Alhijawi [89]	2023	MF-R	Enhances novelty, diversity and coverage	High computational costs	MovieLens Latest, MovieLens100k, HotelExpedia
Banerjee [92]	2023	Timestamp-based movie recommendation	More in tune with current user tastes	Lack of adaptability	MovieLens10M

2.3 Bio-Inspired Algorithms in Recommender Systems

2.3.1 Introduction to Bio-Inspired Concepts

Bio-inspired algorithms are computational methods that mimic mechanisms observed in natural systems, such as ant colony foraging, bee communication, and mycelial network growth. These algorithms offer robust solutions for complex optimization challenges commonly encountered in recommender systems (RSs). In particular, they are effective for balancing exploration and exploitation, enhancing recommendation diversity, and mitigating issues associated with sparse or incomplete data.

By emulating collective intelligence and self-organizing behaviors found in nature, bio-inspired approaches provide adaptive, flexible, and scalable strategies for recommendation tasks. For instance, ant-inspired algorithms can model user-item interactions as paths to be optimized, while fungal-inspired mechanisms can propagate influence across a network, enabling novel connections between users and items. Such algorithms extend beyond traditional heuristic or matrix-based methods, introducing dynamic adaptability that aligns with evolving user preferences and system constraints.

2.3.2 Application of Bio-Inspired Algorithms in Recommender Systems

2.4 Bio-Inspired Recommender Systems

Bio-inspired recommender systems leverage principles derived from biological, ecological, and evolutionary processes to enhance the adaptability, personalization, and robustness of recommendation algorithms. These systems aim to address persistent challenges in recommender systems, such as cold-start problems, heterogeneous user preferences, and the need for flexible recommendation strategies, by emulating behaviors observed in natural systems [111]. Techniques inspired by swarm intelligence [112], genetic algorithms [113], and ant colony optimization [114] provide heuristic approaches that effectively navigate the complex interactions between users and items. As highlighted by Jain [115], bio-inspired models introduce adaptive solutions that transcend conventional accuracy-focused methods, accommodating the multifaceted nature of recommendation tasks.

Kilitcioglu et al. [116] proposed **Pyrorank**, a nature-inspired ranking algorithm designed to enhance diversity in recommender systems. The model draws inspiration from the ecological concept of pyrodiversity, which emphasizes biodiversity recovery following natural fire disturbances. Pyrorank mitigates systemic bias by promoting user-centered diversity while preserving predictive accuracy. Empirical evaluations demonstrated that Pyrorank achieves superior diversity outcomes while maintaining competitive accuracy compared to existing state-of-the-art methods.

Logesh et al. [117] developed a bio-inspired clustering ensemble for user-based collaborative filtering, integrating fuzzy clustering techniques with swarm intelligence. Experiments conducted on Yelp and TripAdvisor datasets indicated that swarm-based clustering notably improves both the accuracy and stability of recommendations. Extending this approach, the authors introduced the **Hybrid Swarm Intelligence Clustering Ensemble (HSICE)** for personalized travel recommendations. HSICE combines multiple swarm optimization algorithms, including BrainStorm Optimization (BSO) for the user clustering phase, and a novel variant, QBSO, designed to enhance clustering quality. By further integrating QBSO with an immune genetic algorithm, HSICE generates efficient and diverse user clusters, ultimately enabling highly accurate and personalized recommendations for travel applications.

Collectively, these studies illustrate that bio-inspired methods provide a robust framework for developing recommender systems that are both adaptive and capable of balancing accuracy with diversity, offering innovative solutions beyond traditional algorithmic approaches.

2.4.1 Mycelium-Inspired Modeling

Biological Principles of Mycelium Networks

Mycelium networks, formed by fungal hyphae, represent highly efficient natural systems capable of exploring and exploiting complex environments. The biological mechanisms underlying these networks provide valuable inspiration for designing adaptive and robust computational models. Key principles include:

- **Distributed sensing and resource allocation**: Each hyphal tip independently senses local environmental conditions, enabling decentralized decision-making and efficient allocation of nutrients and resources across the network.
- **Adaptive edge formation based on environmental stimuli**: Connections within the mycelium dynamically form or dissolve in response to local stimuli, allowing the network to prioritize high-value resources while minimizing energy expenditure.
- **Efficient information propagation through sparse but highly connected networks**: Despite a sparse structural configuration, mycelium networks maintain high connectivity,

facilitating rapid signal transmission and robust adaptation to environmental changes.

These principles have been effectively translated into computational models for recommendation systems and other graph-based applications, enabling adaptive edge reweighting, dynamic exploration-exploitation strategies, and resilience against data sparsity or evolving user behavior.

Review of Existing Mycelium-Based Models

The decentralized, adaptive, and highly resilient organization of natural mycelium networks has increasingly attracted attention across computational sciences, engineering, and sustainable design. In biological systems, mycelium exhibits a remarkable capacity to self-organize, continuously reconfigure its connectivity, and efficiently allocate resources in response to environmental constraints. These properties emerge from local interactions rather than centralized control, allowing fungal networks to maintain functionality under perturbations, balance exploration and exploitation, and optimize nutrient transport. Consequently, mycelium-inspired models aim to translate these principles into artificial systems in order to enhance robustness, diversity, scalability, and energy efficiency in complex and dynamic environments.

Within computational and networked systems, mycelium-inspired approaches have been particularly explored in the context of adaptive routing and distributed optimization. A representative example is the **HyphaNet** model proposed by da Costa Bento and Wille [118], which introduces a biologically grounded routing mechanism for mobile ad hoc networks (MANETs). HyphaNet is inspired by the growth and reinforcement dynamics of fungal hyphae, where nutrient flow naturally gravitates toward regions of higher biomass density. In the algorithmic formulation, this biological concept is abstracted through the notion of *attractiveness*, which quantifies the desirability of a communication path based on its historical success, transmission cost, and reliability. Paths that successfully deliver packets are reinforced through increased virtual biomass, while less efficient routes gradually decay, enabling continuous adaptation to changes in network topology.

A key strength of HyphaNet lies in its fully decentralized decision-making process, where each node updates its routing preferences based solely on local information, closely mirroring the local sensing and response mechanisms observed in real mycelial networks. Extensive simulations conducted using the NS-2 platform demonstrated that this bio-inspired strategy achieves superior packet delivery ratios and reduced routing overhead when compared to classical protocols such as AODV, as well as bio-inspired baselines like the ant colony-based SARA algorithm [118]. These results highlight the suitability of mycelium-inspired mechanisms for highly dynamic and resource-constrained environments, where adaptability and resilience are critical.

Beyond digital networks, mycelium-inspired research has also gained significant momentum in the domain of sustainable materials and ecological engineering. Mycelium-based composites (MBCs) represent a novel class of bio-fabricated materials that leverage the natural binding properties of fungal mycelium to form lightweight yet mechanically viable structures. Alaneme [119] provided a comprehensive analysis of MBCs as sustainable alternatives to conventional construction materials, emphasizing their low environmental footprint, biodegradability, and reduced dependence on fossil fuel-derived inputs. From a systems perspective, these materials embody the same principles of efficiency and adaptability observed in computational mycelium models, as their properties can be tuned by controlling growth conditions, substrate composition, and fabrication processes.

Extending this line of research toward architectural and structural applications, Abdelhady et al. [120] proposed a modular design framework for mycelium-based building components. Their approach integrates computational form-finding techniques with interlocking modular geometries, enabling the generation of structurally stable assemblies without the need for additional reinforcement. By fabricating and evaluating a prototype composed of five distinct module types, the authors demonstrated how mycelium-based systems can support diverse spatial configurations while maintaining structural coherence. This work illustrates how bio-inspired design principles can be operationalized at scale, bridging the gap between biological growth processes and engineered architectural systems.

Collectively, these studies demonstrate that mycelium-inspired models offer a unifying paradigm for designing decentralized, adaptive, and sustainable systems across both digital and physical domains. While existing works primarily focus on routing optimization and material engineering, they collectively underscore the potential of mycelium-inspired mechanisms to address challenges related to robustness, diversity, and efficiency. This body of literature provides a strong conceptual foundation for extending mycelium-based principles to other domains, such as recommendation systems, where dynamic connectivity, adaptive reinforcement, and exploration–exploitation trade-offs are equally essential.

Table 2.4 summarizes the general aspects of the bio-inspired algorithms discussed in this section.

Table 2.4: Bio-inspired algorithms in recommendation systems works summary

Reference	Year	Approach	Advantages	Limitations	Dataset
Kilitcioglu [116]	2023	Bio-inspired re-ranking algorithm	High improvement of recommendation diversity	Difficult to implement, does not account for data privacy	-
Logesh [117]	2020	Hybrid swarm intelligence clustering	Enhances diversity and accuracy	Challenges in adaptability and scalability	TripAdvisor
da Costa Bento [118]	2020	Bio-inspired Routing Algorithm for MANETs	Adaptive and resilient routing inspired by fungal networks	Previously restricted to wired networks	NS-2.35 Simulations
Abdelhady [120]	2023	Mycelium-Based Composites for Modular Systems	Sustainable, biodegradable, lightweight, promotes eco-friendly architecture	Scalability and long-term durability untested	Initial prototype of five modules

2.5 Graph Neural Networks in Recommender Systems

2.5.1 Introduction to GNNs and Graph Modeling

Graph Neural Networks (GNNs) have emerged as a powerful paradigm for modeling relational data, capturing complex dependencies between entities in graph-structured datasets. In the context of recommender systems, users and items can naturally be represented as nodes in a bipartite graph, with edges encoding interactions such as ratings, clicks, or purchases. This graph-based formulation enables the modeling of both direct interactions and higher-order relationships, which are often neglected in traditional collaborative filtering and content-based approaches.

GNNs operate by iteratively aggregating and transforming information from neighboring nodes, producing embeddings that encapsulate both local and global structural properties of the graph. Formally, given a graph $G = (V, E)$ with nodes V and edges E, each node $v \in V$ is associated with a feature vector $\mathbf{x}_v$. At each layer of a GNN, node embeddings are updated via an aggregation function:

$$\mathbf{h}_v^{(k)} = \text{UPDATE}^{(k)}\Big(\mathbf{h}_v^{(k-1)}, \text{AGGREGATE}^{(k)}\Big(\{\mathbf{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}\Big)\Big), \tag{2.1}$$

where $\mathcal{N}(v)$ denotes the set of neighbors of node v, $\mathbf{h}_v^{(0)} = \mathbf{x}_v$, and k is the layer index. This neighborhood aggregation mechanism allows GNNs to capture both structural and feature-based similarities, making them particularly suitable for recommendation tasks where user-item interactions are complex and sparse.

In recommender systems, GNN-based models have demonstrated superior performance in capturing user preferences, improving recommendation accuracy, and incorporating auxiliary information such as social relations, item attributes, and temporal dynamics. By leveraging graph structures, these models also facilitate advanced functionalities, including explainable recommendations, diversity-aware ranking, and sequential or session-based recommendations.

2.5.2 Review of GNN-Based Recommendation Models

Graph Neural Networks (GNNs) have emerged as a foundational paradigm for learning over graph-structured data, offering a principled way to model relational dependencies through message passing, neighborhood aggregation, and representation propagation. Unlike traditional machine learning or matrix factorization approaches, GNNs explicitly encode the topology of interactions, making them particularly well suited for domains where entities and their relationships jointly influence outcomes. As a result, GNNs have demonstrated strong empirical performance in a wide range of applications, including social network analysis [121], molecular and chemical property prediction [122], and intelligent transportation systems [123].

In the context of e-commerce recommender systems, user–item interactions naturally form bipartite or heterogeneous graphs, often enriched with side information such as product attributes, temporal signals, and contextual behaviors. GNN-based recommendation models exploit this structure to capture not only direct interactions (e.g., purchases or clicks) but also higher-order collaborative signals, such as shared neighborhoods, transitive preferences, and implicit community structures. By propagating information across multiple hops, GNNs enable more expressive user and item embeddings, leading to improved recommendation accuracy, robustness, and explainability when compared to classical collaborative filtering methods.

Despite their advantages, early GNN-based recommender systems face several practical challenges, including semantic sparsity, scalability to industrial-scale graphs, bias toward popular items, and limited effectiveness in cold-start and long-tail scenarios. To address these issues, Xv et al. [124] proposed **CC-GNN**, a scalable framework designed to overcome four major limitations: inaccurate semantic representations, poor scalability on large graphs, degraded performance for long-tail queries, and the cold-start problem for newly introduced products. CC-GNN introduces adaptive representation learning mechanisms that better capture semantic context while maintaining efficiency on massive graphs. Evaluated on an industrial-scale dataset containing approximately 100 million nodes, CC-GNN achieved more than a 10% performance improvement over strong baselines. However, the authors also observed that the model's effectiveness remains sensitive to its ability to dynamically adapt to evolving data distributions, underscoring the importance of resilience and adaptability in real-world recommender deployments.

Cold-start mitigation and new product exposure have also motivated the design of intent-aware GNN architectures. Jin et al. [125] introduced a **dual-intent network** that explicitly models user decision-making processes by jointly analyzing historical behavior and simulated judgment patterns. By disentangling different user intents, the model improves the discovery of newly introduced products while offering enhanced interpretability. Although the approach demonstrates strong performance in cold-start settings, its reliance on rich product attributes and its limited scalability restrict its applicability across highly diverse and large-scale e-commerce platforms.

Beyond model design, the deployment of GNN-based recommenders in industrial environments introduces additional concerns related to reliability, privacy, and system integration. Zhang et al. [126] conducted an in-depth investigation into these practical challenges, proposing a framework that facilitates the integration of GNN-based models into real-world recommendation pipelines. While their study confirms the feasibility of industrial adoption, it also highlights persistent difficulties in achieving sufficient recommendation diversity and generalization across heterogeneous domains, pointing to open research questions at the intersection of performance, trustworthiness, and fairness.

Hybridization with collaborative filtering (CF) techniques represents another important research direction. In this context, Alharbe et al. [82] proposed **UI2vec**, a representation learning framework that embeds users and items using feature extraction networks combined with similarity-based objectives. Building upon this foundation, they introduced **VUI2vec**, a generative extension that integrates variational learning principles. Extensive experiments across multiple benchmark datasets demonstrated that VUI2vec outperforms several state-of-the-art baselines, illustrating the effectiveness of hybrid GNN–CF models in capturing both relational structure and latent preference distributions.

Addressing the long-standing trade-off between recommendation accuracy and diversity, Berbague et al. [110] proposed an evolutionary algorithm-based clustering strategy. Their approach partitions users into primary and secondary clusters, enabling personalized recommendation generation through user-based collaborative filtering while preserving coverage and diversity. Although not purely GNN-based, this work is closely related, as it emphasizes the importance of structured user grouping and scalable optimization—principles that align well with graph-based learning frameworks.

More recently, the integration of advanced GNN architectures into recommendation systems has further expanded modeling capabilities. Wu et al. [127] characterized this integration as a paradigm shift, enabling unified modeling of heterogeneous entities and interactions. Within this line of research, Xia et al. [128] proposed the **Temporal Graph Transformer (TGT)**, which explicitly models multi-behavior user interactions over time. TGT captures both short-term and long-term dynamics by extracting type-specific relational contexts and identifying implicit temporal dependencies, resulting in significant improvements in sequential recommendation accuracy.

Inter-item relationships and explainability have also gained increasing attention. Liu et al. [129] introduced the **Item Relationship GNN (IRGNN)**, which leverages recursive node embeddings and edge relational networks to model multi-hop product associations. By explicitly encoding semantic and structural relationships among items, IRGNN enhances both recommendation performance and interpretability. Experimental results on large-scale sparse datasets demonstrate its robustness and superiority over conventional GNN-based recommenders.

Beyond traditional preference prediction tasks, GNNs have been applied to more specialized recommendation scenarios. Shi et al. [130] proposed a personalized image aesthetics assessment framework that integrates GNNs with collaborative filtering. Their model jointly captures internal attribute interactions (within images) and external relational dependencies (across users and items), enabling accurate prediction of personalized aesthetics scores. While the framework outperforms prior methods, its adaptability to datasets with heterogeneous or weakly defined attribute interactions remains an open challenge.

Finally, GNNs have proven effective in cross-domain and multi-task recommendation settings. Zhao et al. [131] introduced the **Search and Recommendation Joint Graph (SRJGraph)** for click-through rate (CTR) prediction. By constructing a heterogeneous graph composed of users, items, and search queries, and employing an intention- and upstream-aware aggregation mechanism, SRJGraph captures high-order relational dependencies and facilitates knowledge transfer between search and recommendation tasks. Large-scale experiments on the Taobao dataset confirmed its superiority over existing CTR prediction models, highlighting the potential of joint graph modeling for unified recommendation systems.

Overall, existing GNN-based recommendation models demonstrate substantial progress in modeling complex user item interactions, temporal dynamics, and heterogeneous relationships. Nevertheless, persistent challenges remain, particularly with respect to scalability, adaptability, diversity enhancement, and robustness under data sparsity and distribution shifts. These limitations motivate the exploration of novel bio-inspired and adaptive graph learning paradigms that can complement GNN architectures and further improve recommendation performance in dynamic real world environments.

Table 2.5 summarizes the general aspects of the bio-inspired algorithms discussed in this section.

Table 2.5: Graph neural networks in recommendation systems works summary

Reference	Year	Approach	Advantages	Limitations	Dataset
Xv [124]	2023	CC-GNN	Effective management of GNN challenges	Lack of adaptability and robustness	IPQS, Amazon Sport Dataset
Jin [125]	2023	Dual-intention network	Interpretability and reasoning on new items	Dependence on attribute availability	Amazon Grocery, Amazon Gourmet Food, Yelpsmall
Zhang [126]	2024	Word Embedding	Simultaneously learns embedded representations	Lack of diversity	TaFeng, MovieLens1M, Netflix
Berbague [110]	2021	Genetic Algorithm-based Clustering	Enhances recommendation diversity and scalability	Potential increase on computational cost	MovieLens100k, MovieLens1M
Xia [128]	2022	Multi-typed user behaviors	Reveals intricate relationship frameworks a	Complexity of implementation, limited adaptability	Taobao, IJCAI-Context
Liu [129]	2022	Multilabel link prediction	Captures multi-hop product relationships	Dependence on attribute availability	Amazon, Taobao
Zhao [131]	2022	CTR model	Aggregator aware of upstream dynamics	Very high computational complexity	Taobao
Shi [130]	2024	Graph-based aesthetic assessment	Enhanced visual recommendation	Requires high-quality image data, computationally intensive	Various image datasets

2.6 Blockchain-Based Models for Secure Recommendation

2.6.1 Introduction to Blockchain Technology

Blockchain technology is a decentralized and distributed ledger system designed to record transactions in a secure, transparent, and immutable manner. Initially introduced as the underlying infrastructure for cryptocurrencies, blockchain has evolved into a general-purpose technology with applications spanning finance, healthcare, supply chain management, and digital platforms. Its core characteristics—decentralization, cryptographic security, consensus mechanisms, and tamper resistance—make it particularly suitable for environments where trust, data integrity, and transparency are critical.

A blockchain consists of a sequence of blocks, each containing a set of transactions, a timestamp, and a cryptographic hash of the previous block. This chained structure ensures that any attempt to alter stored data requires modifying all subsequent blocks, which is computationally infeasible in large-scale networks. Consensus protocols, such as Proof of Work (PoW) or Proof of Stake (PoS), govern how new blocks are validated and added, enabling agreement among distributed participants without relying on a central authority.

In the context of data-driven systems, blockchain offers a novel paradigm for secure data sharing and decentralized computation. By allowing multiple parties to interact over a trustless network, blockchain mitigates risks related to data manipulation, unauthorized access, and single points of failure. These properties are increasingly relevant for modern recommendation systems, which rely on large volumes of sensitive user data and often suffer from privacy concerns, data silos, and lack of user control.

As a result, blockchain has emerged as a promising enabling technology for next-generation recommender systems, providing a secure and transparent foundation for data ownership, incentive mechanisms, and collaborative model training in decentralized environments.

2.6.2 Application of Blockchain Technology in Recommender Systems

Blockchain technology is a decentralized and distributed digital ledger that enables secure, transparent, and immutable recording of transactions across a peer-to-peer network through cryptographic hashing and consensus mechanisms [132]. While initially designed as the underlying infrastructure for cryptocurrencies such as Bitcoin, blockchain has rapidly evolved into a general-purpose technology with significant implications for data management, trust establishment, and decentralized system design. Its intrinsic properties—namely decentralization, immutability, auditability, and resistance to tampering—make blockchain particularly attractive for applications where trust, security, and data integrity are critical.

In e-commerce recommender systems, these requirements are especially pronounced. Modern recommendation pipelines rely heavily on large volumes of sensitive user data, including browsing behavior, transaction histories, preferences, and contextual signals. Centralized architectures, which dominate current industrial deployments, expose such systems to privacy breaches, data manipulation, single points of failure, and opaque decision-making processes. Blockchain technology offers an alternative paradigm by enabling decentralized data ownership, verifiable computation, and transparent interaction logging, thereby fostering user trust while mitigating security risks.

Early research efforts have primarily focused on leveraging blockchain to enhance privacy preservation in recommender systems. Ying [133] introduced *Share-fMF*, a privacy-aware matrix factorization framework that combines federated learning with secret sharing techniques. By distributing model training across multiple participants and preventing direct access to raw user data, Share-fMF achieves strong privacy guarantees while maintaining scalability on large e-commerce datasets. However, the framework introduces non-trivial system complexity and requires sophisticated coordination mechanisms, which can hinder its practical deployment in real-world platforms.

Similarly, Villarán et al. [134] proposed *Privacy Advisor*, a user-centric recommendation framework that empowers users to control how their personal data are utilized within recommendation processes. The model emphasizes transparency and informed consent, aligning with emerging data protection regulations. Despite these advantages, its effectiveness depends heavily on active user participation and suffers from limited adaptability across heterogeneous recommendation environments and platforms.

Recognizing the broader implications of privacy and trust challenges, Himeur et al. [135] conducted a comprehensive survey of recommender systems across domains such as e-commerce, healthcare, and social media. Their analysis underscores that privacy leakage, lack of transparency, and susceptibility to data manipulation represent major obstacles to user adoption and regulatory compliance. The authors identify blockchain as a promising technological enabler capable of ensuring data integrity, supporting decentralized identity management, and enabling privacy-preserving recommendation pipelines without significantly compromising predictive performance.

More recent studies have moved beyond conceptual analyses to propose concrete blockchain-based architectures for recommender systems. Javadi et al. [136] introduced a novel hybrid framework that combines blockchain smart contracts with off-chain recommender engines. In their approach, user interaction data are stored on-chain following a structure defined within smart contracts, ensuring immutability and transparency. Computationally intensive recommendation algorithms are executed off-chain, and the resulting recommendations are written back to the blockchain during transaction execution. This design significantly reduces gas consumption and transaction costs while preserving recommendation accuracy and scalability, effectively addressing one of the major limitations of fully on-chain recommender implementations.

Blockchain-enabled recommender systems have also been explored in domain-specific applications. Sun et al. [137] proposed β**FSCM**, an enhanced food supply chain management system that integrates a hybrid blockchain architecture with recommender systems. By combining public and private blockchain components with access control mechanisms, the system achieves a balance between transparency and decentralization. The integrated recommender module leverages machine learning to personalize product offerings and optimize inventory management, leading to measurable improvements in customer engagement, supply chain monitoring, and operational efficiency. This work demonstrates how blockchain-based recommendation frameworks can extend beyond classical e-commerce toward complex multi-stakeholder ecosystems.

Privacy and data authenticity challenges are further amplified in cross-domain recommendation settings, where data originate from multiple platforms and organizations. To address these issues, Wang et al. [138] proposed **BAM_CRS**, a blockchain-based anonymous model for cross-domain recommendation systems. Their architecture employs a three-chain structure to separately store users, items, and interaction relationships, thereby ensuring anonymity and preventing linkage attacks. Additionally, the model introduces contribution-based incentive mechanisms and signature-based verification to protect against data poisoning attacks and ensure data authenticity. Experimental evaluations on real-world datasets demonstrate that BAM_CRS not only enhances privacy protection but also achieves approximately 2% improvement in precision and F1-score compared to baseline approaches.

Blockchain has also been leveraged to support recommendation-driven decision-making in e-commerce logistics and shipping policies. Kumar et al. [139] proposed a History Informed Shipping (HISHIP) framework that integrates blockchain-based transaction storage with machine learning-based recommender systems. By utilizing immutable order histories stored on-chain, HISHIP dynamically adjusts shipping fee exemptions based on customer purchasing behavior rather than static thresholds. The integration of a recommender module further assists customers in selecting additional products to qualify for shipping benefits. Experimental results demonstrate significant improvements in customer participation and revenue generation, highlighting the practical value of blockchain-backed recommender systems in operational decision support.

Overall, existing studies demonstrate that blockchain technology can play a pivotal role in addressing critical challenges faced by modern recommender systems, particularly with respect to privacy preservation, data integrity, transparency, and trust. Nevertheless, several open issues remain, including the trade-off between decentralization and computational efficiency, interoperability between on-chain and off-chain components, and the integration of advanced machine learning models within blockchain-constrained environments. These challenges motivate further research into hybrid architectures and adaptive frameworks that combine blockchain technology with intelligent recommender systems.

Table 2.6 summarizes the main characteristics of representative blockchain-based recommender system approaches discussed in this section.

2.7 Critical Review and Research Positioning

Recent advances in Graph Neural Networks (GNNs), bio-inspired algorithms, and blockchain-based models have significantly contributed to improving the accuracy, diversity, robustness,

Table 2.6: Blockchain technology in recommendation systems works summary

Reference	Year	Approach	Advantages	Limitations	Dataset
Javadi et al. [136]	2025	Hybrid on-chain/off-chain RS	Low gas cost, scalability	Off-chain dependency	Not specified
Sun et al. [137]	2025	Hybrid blockchain + RS for FSCM	Transparency, inventory optimization	Domain-specific	FSCM data
Wang et al. [138]	2025	BAM_CRS (Anonymous CDR)	Privacy, robustness, incentive mechanism	System complexity	Real-world datasets
Kumar et al. [139]	2025	Blockchain-based shipping + RS	Revenue increase, trust	Policy-specific	TPC-H, All Beauty
Villarán [134]	2022	User-centric Privacy Advisor	User-controlled privacy	Dependence on user consent	Identity federation data
Rajasekaran [132]	2020	Share-fMF (Federated MF + Secret Sharing)	Strong privacy guarantees, scalability	High system complexity	MovieLens

and security of modern recommender systems. Despite these notable achievements, a number of fundamental challenges continue to hinder their large-scale and real-world deployment.

- **Scalability and computational cost:** GNN-based recommender systems often suffer from high computational and memory overheads when applied to large-scale user–item graphs, limiting their applicability in industrial environments with strict latency constraints.
- **Modeling complexity and parameter sensitivity:** Bio-inspired recommender systems, while effective in promoting diversity and adaptability, frequently rely on complex heuristics and sensitive hyperparameter tuning, which can affect model stability and reproducibility across different datasets.
- **Integration and efficiency constraints:** Although blockchain-based solutions provide strong guarantees in terms of security, transparency, and privacy preservation, their integration into low-latency recommendation pipelines remains challenging due to transaction overhead and synchronization delays.

These limitations highlight the need for *hybrid and adaptive recommendation frameworks* capable of leveraging the complementary strengths of these paradigms while mitigating their respective weaknesses. Positioning research in this space therefore requires a unified approach that combines graph-based representation learning, bio-inspired adaptive mechanisms, and privacy-aware decentralized architectures.

In this context, the solutions proposed in **Chapter 3** and **Chapter 4** address the identified research gaps by introducing novel hybrid models. These contributions aim to bridge the gap between theoretical advances and practical deployment, offering a coherent research direction for next-generation recommender systems.

2.8 Conclusion

This chapter provided an in-depth analysis of recent recommender system models, with particular emphasis on accuracy-oriented techniques, diversity-aware strategies, bio-inspired approaches, graph neural networks, and blockchain-based secure frameworks. Collectively, these paradigms illustrate the evolving landscape of recommender systems and highlight their potential to deliver more personalized, reliable, and resilient recommendations in complex and large-scale environments.

Despite these advances, the critical review conducted in this chapter reveals persistent limitations related to scalability, adaptability, diversity control, and practical deployment.

Motivated by these observations, the next chapter introduces **MycGNN**, a novel recommender system model inspired by the adaptive and decentralized behavior of mycelium networks. By integrating graph neural networks with bio-inspired mechanisms, MycGNN aims to overcome the identified limitations and provide a principled solution that enhances diversity without sacrificing accuracy. **Chapter 3** details the theoretical foundations, architectural design, and core mechanisms of the proposed MycGNN model, laying the groundwork for its experimental validation and comparative evaluation.

Chapter 3: MycGNN: Enhancing Recommendation Diversity Through Mycelium-Inspired Graph Neural Network

Contents

3.1 Introduction

This chapter introduces **MycGNN** [140], a novel hybrid Graph Neural Network inspired by the adaptive exploration and growth mechanisms of mycelium networks. The proposed model addresses the inherent trade-off between accuracy and diversity in recommender systems by dynamically reweighting graph edges during the recommendation phase based on user interactions and item features.

MycGNN further incorporates a probabilistic exploration–exploitation strategy that enables the generation of more personalized and diverse recommendations compared to traditional GNN-based approaches. The effectiveness of the proposed model is evaluated on the *Retail Rocket* and *MovieLens1M* datasets. Experimental results show that while some state-of-the-art models may achieve higher accuracy, MycGNN consistently outperforms them in terms of intra-list diversity while maintaining competitive overall accuracy.

3.2 Motivation and Design Objectives

Despite the strong predictive performance of Graph Neural Networks (GNNs) in recommender systems, existing models often suffer from popularity bias and limited recommendation diversity. GNNs effectively capture complex relational structures between users and items but typically prioritize highly connected or frequently interacted items, leading to over-specialized recommendations and reduced item discovery.

To address these limitations, **MycGNN** combines the expressive power of GNNs with the adaptive and resilient principles observed in natural mycelium networks. Mycelium-inspired systems can dynamically adjust connectivity and balance exploration and exploitation in response to environmental stimuli, motivating their integration into graph-based recommendation models to improve both personalization and diversity.

In MycGNN, two complementary mechanisms are applied during the recommendation phase:

- **Dynamic edge reweighting:** After GNN training, edge weights are adjusted based on user interactions and item attributes to reduce the dominance of popular items and promote long-tail exposure.
- **Exploration–exploitation function:** A probabilistic mechanism balances familiar and novel recommendations by dynamically adjusting the ratio of exploratory items in the recommendation list.

The design objectives of MycGNN are:

- Integrate mycelium-inspired adaptability with GNN relational modeling for enhanced recommendation quality.

- Mitigate popularity bias while maintaining competitive accuracy.
- Increase recommendation diversity through controlled exploration without compromising personalization.

3.3 Design and Implementation of the MycGNN Framework

The MycGNN framework integrates the decentralized and adaptive properties of mycelium networks with the relational learning capabilities of Graph Neural Networks (GNNs) to enhance recommendation diversity. The framework consists of two primary components: the mycelium-inspired growth mechanism and the GNN message-passing process. Their combination allows dynamic adaptation of edge weights during recommendation, promoting long-tail exposure while maintaining high accuracy.

3.3.1 Mycelium-Inspired Growth

The decentralized growth and information propagation in mycelium networks is captured by:

$$M_t = M_{t-1} + F(M_{t-1}) \tag{3.1}$$

where M_t represents the mycelium state at time t, M_{t-1} is the state at the previous time step, and $F(M_{t-1})$ models resource allocation, adaptive edge formation, and information exchange. This mechanism allows dynamic adjustment of edge influence based on user interactions and item attributes, introducing diversity and exploration.

3.3.2 Graph Neural Network Message Passing

The GNN component updates node representations through neighborhood aggregation:

$$h_v^{(l+1)} = \sum_{u \in N(v)} \frac{1}{c_{u,v}} W^{(l)} h_u^{(l)} \tag{3.2}$$

where $h_v^{(l+1)}$ is the updated representation of node v at layer $l+1$, $N(v)$ denotes its neighbors, $c_{u,v}$ is a normalization constant for edge (u,v), $W^{(l)}$ is a learnable weight matrix, and $h_u^{(l)}$ is the representation of neighbor node u. This captures structural dependencies among users and items.

3.3.3 Integration of Mycelium and GNN

The MycGNN model combines mycelium-inspired dynamics with GNN aggregation:

$$h_v^{(l+1)} = \sum_{u \in N(v)} \frac{1}{c_{u,v}} W^{(l)} h_u^{(l)} + M_t \tag{3.3}$$

M_t is introduced as an additive factor to adjust edge influence based on structural relations and adaptive mycelium dynamics. This fusion enables a balance between exploration and exploitation, improving long-tail item exposure while preserving recommendation relevance.

3.3.4 Implementation Overview

The implementation of MycGNN involves the following stages:

- Data preprocessing and graph construction to represent user-item interactions.
- Training the GNN to learn base node embeddings.
- Applying the mycelium-inspired dynamic edge reweighting and exploration–exploitation strategies during recommendation generation.
- Evaluating the framework on standard datasets such as Retail Rocket and MovieLens1M.

This design provides a flexible and adaptive recommendation framework that addresses the common limitations of traditional GNNs, particularly regarding popularity bias and lack of diversity. Figure 3.1 presents an overview of the proposed model's workflow.

3.3.5 Data Preprocessing

Before implementing MycGNN, we assessed the quality and relevance of the datasets. Large-scale datasets often contain missing, erroneous, or duplicated entries. To ensure accurate and unique interactions, records lacking critical information (e.g., user or item IDs) were removed, and duplicates were deleted. Text data was normalized, tokenized, and encoded into numerical vectors. Numerical features were scaled or normalized, and Principal Component Analysis (PCA) was applied for dimensionality reduction, retaining the most informative features. The overall preprocessing workflow is summarized in Algorithm 3.1.

3.3.6 Graph Construction

The recommendation graph encodes users, items, and interactions as a structured network. Users and items are represented as nodes with high-dimensional attribute vectors, while edges capture interactions weighted by importance. The graph evolves dynamically based on user behavior and item popularity. Algorithm 3.2 summarizes the construction procedure.

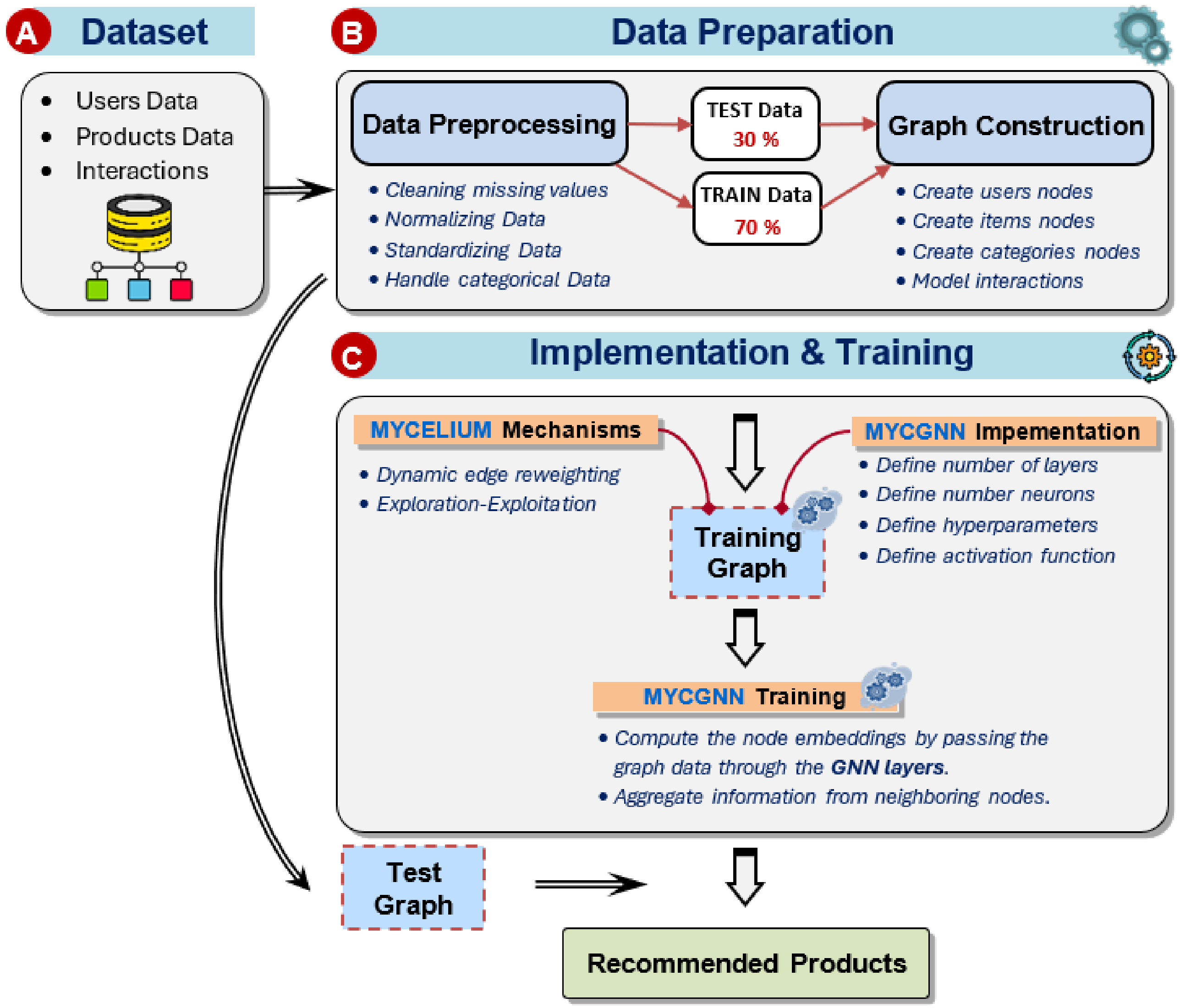

Figure 3.1: MycGNN General Workflow

3.3.7 Integration of Mycelium-Inspired Mechanisms

MycGNN introduces two primary bio-inspired mechanisms:

Dynamic Edge Reweighting. Inspired by nutrient flow in mycelium networks, edge weights are updated during recommendations based on user interactions and item attributes:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(u_i v_j - w_{ij}(t)) \tag{3.4}$$

where $w_{ij}(t+1)$ is the updated edge weight, η is the learning rate, u_i is the user's interest vector, and v_j is the item's attribute vector. A parameter β allows partial adjustment:

$$w_{ij}(t+1) = (1-\beta)w_{ij}(t) + \beta(u_i v_j) \tag{3.5}$$

Algorithm 3.1: MycGNN Data Preprocessing

Step 1: Input: Raw data $\mathcal{D} = \{(u_i, v_j, x_k)\}$, where u_i are users, v_j are items, and x_k represents attributes.
Step 2: Output: Preprocessed_Data $\mathcal{D}'$
Step 3: Clean Data:
  3.1: Identify and remove corrupt entries where $u_i = \emptyset \vee v_j = \emptyset$
  3.2: Fill missing attributes x_k using $x_k \leftarrow \text{mean}(x_k)$ or default value
Step 4: Remove Duplicates:
  4.1: For entries $e_a, e_b \in \mathcal{D}$, if $e_a = e_b$ then mark e_b for removal
  4.2: Retain a single instance: $\mathcal{D} \leftarrow \mathcal{D} \setminus \{e_b\}$
Step 5: Text Normalization:
  5.1: Convert text $t \in x_k$ to lowercase: $t \leftarrow \text{lower}(t)$
  5.2: Remove punctuation/special chars: $t \leftarrow t \setminus \{\text{punctuation}\}$
  5.3: Apply stemming/lemmatization: $t \leftarrow \text{stem}(t)$
Step 6: Tokenization:
  6.1: Tokenize text $t \rightarrow \{w_1, w_2, \ldots, w_n\}$
Step 7: Numerical Encoding:
  7.1: Encode categorical variables $c \in x_k$ as one-hot vectors $\mathbf{v}_c \in \{0,1\}^{|C|}$
  7.2: Apply text vectorization: TF-IDF $\mathbf{v}_t = \text{TF-IDF}(t)$ or embeddings $\mathbf{v}_t = \text{Embed}(t)$
Step 8: Scaling and Normalization:
  8.1: Normalize numerical features $x \in \mathbb{R}$: $x' = \frac{x - \min(x)}{\max(x) - \min(x)}$
Step 9: Dimensionality Reduction:
  9.1: Apply PCA: $\mathbf{Z} = \mathbf{X}\mathbf{W}_{\text{PCA}}$, retaining top d principal components
Step 10: Finalize:
  10.1: Output Preprocessed_Data: $\mathcal{D}' = \{(u_i, v_j, \mathbf{z}_k)\}$

Algorithm 3.2: MycGNN Graph Construction

Step 1: Input: Preprocessed_Data $\mathcal{D}' = \{(u_i, v_j, \mathbf{z}_k)\}$, where u_i are users, v_j are items, and $\mathbf{z}_k$ are feature vectors
Step 2: Output: Training Graph $G = (V, E)$, Testing Graph $G' = (V', E')$
Step 3: Split Data:
  3.1: Divide $\mathcal{D}'$ into Train_Data $\mathcal{D}_{\text{train}}$ (70%) and Test_Data $\mathcal{D}_{\text{test}}$ (30%)
Step 4: Initialize Graphs:
  4.1: $G \leftarrow (\emptyset, \emptyset)$, an empty graph for training
  4.2: $G' \leftarrow (\emptyset, \emptyset)$, an empty graph for testing
Step 5: Populate Training Graph:
  5.1: For each record $(u_i, v_j, \mathbf{z}_k) \in \mathcal{D}_{\text{train}}$:
    5.1.1: Add user node u_i to V if $u_i \notin V$
    5.1.2: Add item node v_j to V if $v_j \notin V$
    5.1.3: Add edge $e_{ij} = (u_i, v_j)$ to E with weight $w_{ij} = f(\mathbf{z}_k)$, where f maps features to edge importance
Step 6: Populate Testing Graph:
  6.1: For each record $(u_i, v_j, \mathbf{z}_k) \in \mathcal{D}_{\text{test}}$:
    6.1.1: Add user node u_i to V' if $u_i \notin V'$
    6.1.2: Add item node v_j to V' if $v_j \notin V'$
    6.1.3: Add edge $e'_{ij} = (u_i, v_j)$ to E' with weight $w'_{ij} = f(\mathbf{z}_k)$
Step 7: Finalize:
  7.1: Output the constructed Training Graph $G = (V, E)$ and Testing Graph $G' = (V', E')$

Exploration–Exploitation Balance. To encourage diverse recommendations, MycGNN probabilistically balances exploration and exploitation:

$$P(i) = \frac{e^{Q(i)/\alpha}}{\sum_{j \in I} e^{Q(j)/\alpha}} \tag{3.6}$$

where $Q(i)$ is the item quality score, α controls the exploration–exploitation trade-off, and I is the set of all items.

Resilient Graph Convolution. Multiple redundant paths in graph convolution layers enhance robustness against noisy or incomplete data:

$$H^{(l+1)} = \sigma\left(\sum_{k=1}^{K} W_k H^{(l)} A_k + b\right) \tag{3.7}$$

where $H^{(l)}$ is the input, W_k are learnable weights for redundant pathways, A_k are adjacency matrices, b is bias, and σ is the activation function.

3.3.8 MycGNN Training

The input graph contains nodes for users, items, and categories, with edges representing their relationships. During training, node embeddings are iteratively updated through neighborhood aggregation, nonlinear activation, and normalization. MycGNN applies both dynamic edge reweighting and exploration–exploitation mechanisms, allowing adaptation to evolving user preferences and balancing accuracy with diversity.

3.3.9 Computational Complexity Analysis

MycGNN is designed to be scalable and efficient:

- Embedding layer: $O(N \times d)$, with N nodes and $d = 128$ embedding dimensions.
- Graph convolution layer: $O(E \times f^2)$, where E is the number of edges and $f = 64$ feature size.
- Attention mechanism: $O(h \times N \times f^2)$ for h attention heads.
- Fully connected layer: $O(f)$ for mapping features to predictions.

This design ensures that MycGNN maintains performance while processing large-scale graphs efficiently.

3.4 Experiments and Results

3.4.1 Datasets

To evaluate **MycGNN**, we selected two distinct datasets and applied the preprocessing and graph construction methodology described in Section 4.

Retail Rocket Dataset: This dataset contains users' behavior data from a real-world e-commerce website. It consists of three primary files: behavioral data (`events.csv`), item properties (`itemproperties.csv`), and the category tree (`categorytree.csv`). It captures a wide range of e-commerce activities, including click rates, add-to-cart actions, and checkout events.

MovieLens1M Dataset: This benchmark dataset contains 1,000,000 ratings and has been widely used since its first release in February 2003 for machine learning and recommender system research. It includes detailed user and movie information, making it suitable for collaborative filtering and other data mining approaches.

3.4.2 Graph Analysis and Visualization

Before the modeling phase, understanding the structural characteristics of the data is essential to construct the recommendation graph on which the GNN operates. Table 3.1 summarizes the key graph statistics.

Table 3.1: Graph characteristics of the Retail Rocket and MovieLens1M datasets

Attributes	**Retail Rocket**	**MovieLens1M**
Number of graph nodes	1,644,510	9,941
Number of graph edges	286,514,406	1,006,617
Number of user nodes	1,512,458	6,040
Number of item nodes	130,383	3,883
Number of category nodes	1,669	18

For graphical visualization, 20 random nodes were selected to illustrate subgraphs from each dataset. Figure 3.2 shows the distribution of different types of nodes: users, items, and categories. Subfigure (a) represents a subgraph of the MovieLens1M dataset, while subfigure (b) shows a subgraph of the Retail Rocket dataset.

3.4.3 Experimental Study of the MycGNN Architecture

The architecture of **MycGNN** was determined through an iterative trial-and-error approach, testing multiple configurations by varying one parameter at a time while keeping others constant.

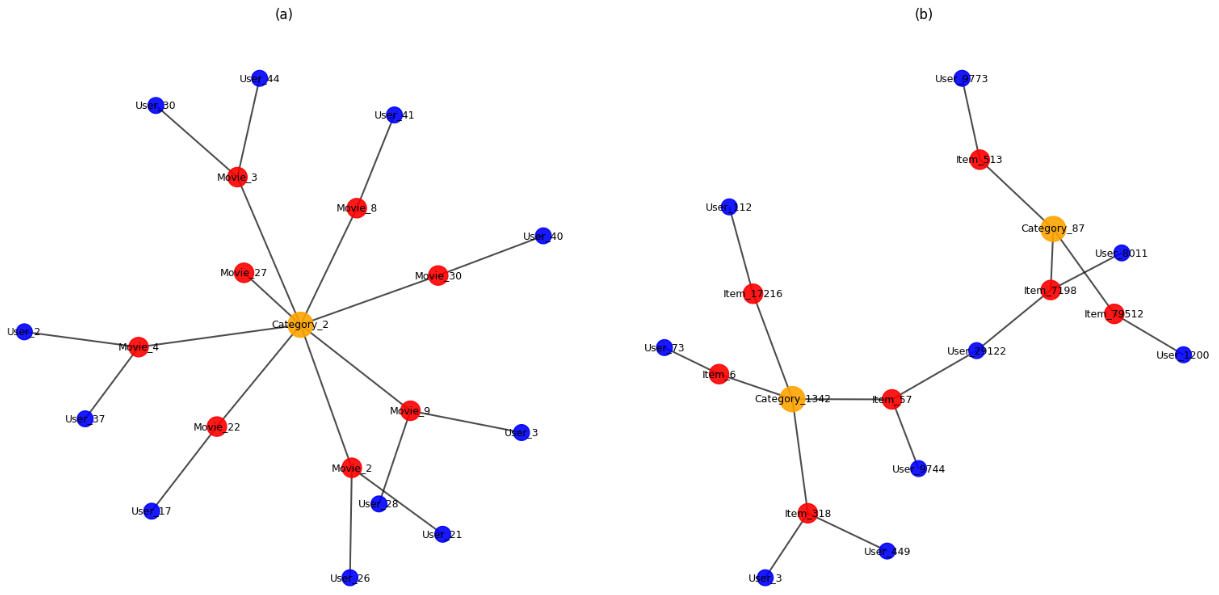

Figure 3.2: Subgraphs of the MovieLens1M (a) and Retail Rocket (b) datasets, illustrating the distribution of users, items, and categories.

Figure 3.3 and Table 3.2 summarizes the optimal configuration.

Initially, the focus was on maximizing model accuracy, followed by the application of mycelium-inspired mechanisms to increase recommendation diversity without compromising performance.

The embedding dimension of 128 was chosen to adequately capture the semantic complexity of user-item interactions. Higher dimensions led to overfitting and increased computational cost, whereas lower dimensions risked oversimplifying node features.

The first convolutional layer with 64 dimensions performs initial feature extraction from multiple neighborhoods, ensuring rich node representations. The second convolutional layer of 64 dimensions continues this process, using ReLU activation to address vanishing gradients while maintaining computational efficiency.

A 64-dimensional attention layer with LeakyReLU activation ($\alpha = 0.2$) and 4 attention heads allows MycGNN to focus on different types of relationships within the graph. The attention mechanism preserves small non-zero gradients for inactive neurons, supporting robust weight updates.

Hyperparameters such as a learning rate of 0.01 and dropout of 0.5 were selected to balance learning stability and overfitting prevention. Finally, the output layer is fully connected with a sigmoid activation function to produce a probability score for each user-item interaction.

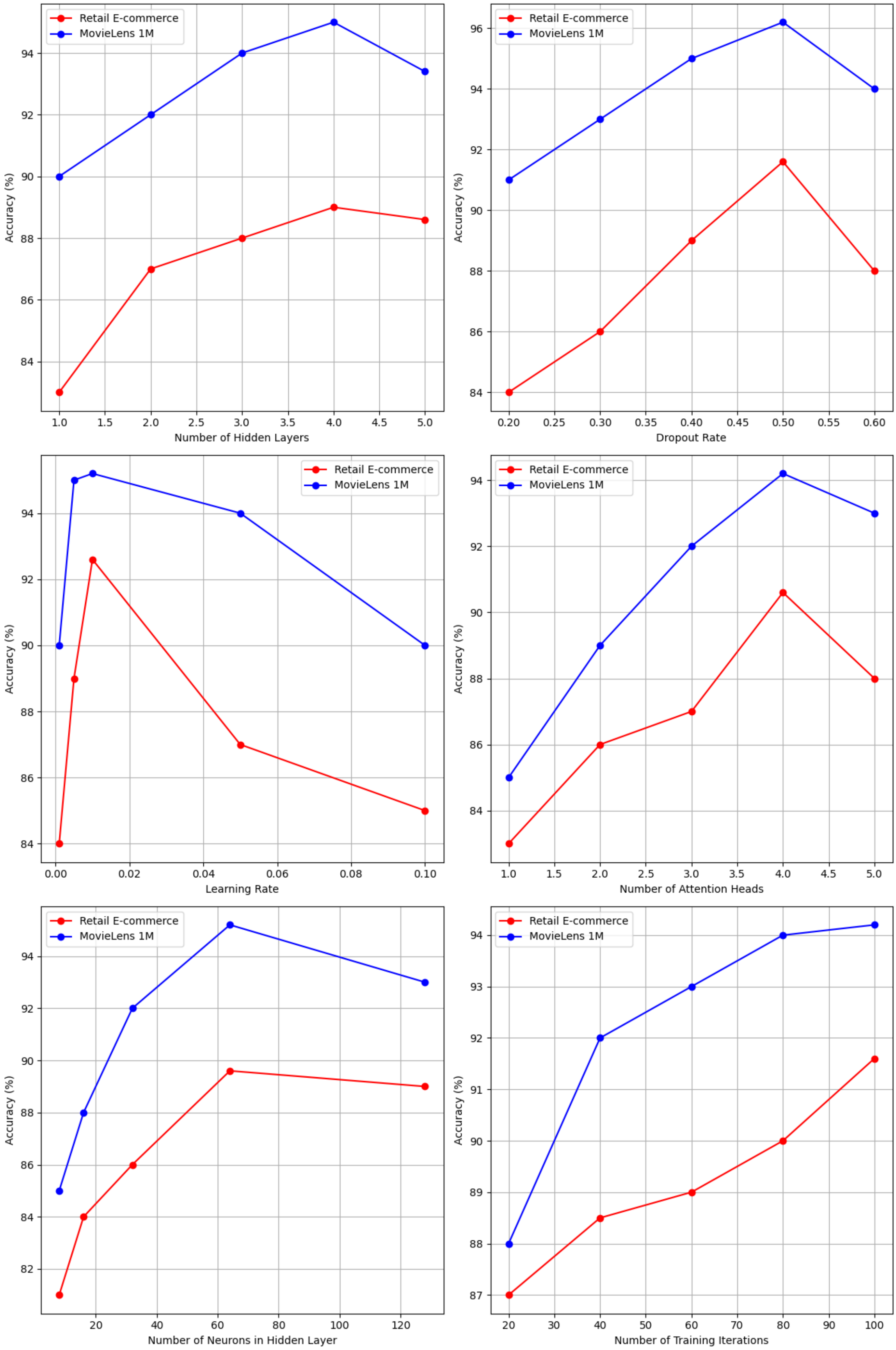

Figure 3.3: Parameter variation tests to determine the optimal MycGNN configuration.

3.4.4 Evaluation of the Mycelium Strategy

To evaluate the impact of each mycelium-inspired mechanism, multiple model variants were created:

Table 3.2: Optimal Architecture of MycGNN

Layer	Type	Dimension	Activation Function	Hyperparameters
Input	Embedding nodes	128	-	-
1	Convolutional Layer	64	ReLU	Learning rate: 0.01
2	Convolutional Layer	64	ReLU	Dropout: 0.5
3	Attention Layer	64	LeakyReLU ($\alpha = 0.2$)	Attention heads: 4
4	Convolutional Layer	64	ReLU	Category parameters
Output	Fully Connected Layer	1	Sigmoid	-

- **Full model (MycGNN):** includes all mycelium-inspired mechanisms.
- **MycGNN@1:** dynamic edge reweighting removed.
- **MycGNN@2:** exploration-exploitation balance strategy removed.

The models were evaluated using accuracy, recall, F1-score, and intra-list diversity. Tables 3.3 and 3.4 summarize the results.

Table 3.3: Performance comparison of mycelium-inspired models on Retail Rocket Dataset

Model	Accuracy (%)	Recall (%)	F1-Score (%)	Intra-list Diversity (%)
MycGNN@1	87.75	80.45	85.17	25.59
MycGNN@2	87.40	79.94	84.46	28.35
MycGNN	86.49	78.67	83.65	32.87
Improvement	-1.07	-1.21	-0.6	+17.78

Table 3.4: Performance comparison of mycelium-inspired models on MovieLens1M Dataset

Model	Accuracy (%)	Recall (%)	F1-Score (%)	Intra-list Diversity (%)
MycGNN@1	83.18	76.05	82.11	30.79
MycGNN@2	85.55	76.56	82.59	35.37
MycGNN	84.59	77.53	83.34	37.62
Improvement	-1.24	-1.41	-0.62	+16.70

These results clearly demonstrate that mycelium-inspired mechanisms significantly enhance recommendation diversity while maintaining competitive accuracy, recall, and F1-score.

3.4.5 Overview of MycGNN Performance

In this section, we evaluate the MycGNN model according to the parameters β and γ. Tables 3.5 and 3.6 present the evaluation results of MycGNN on the Retail Rocket and MovieLens1M datasets, respectively, showing the impact of the parameters β and γ on accuracy, recall, F1-Score, and ILD.

As shown in Figure 3.4, the MycGNN model can balance accuracy and diversity by adjusting β and γ. Lower β and γ values emphasize diversity, producing a more varied recommendation list. Conversely, higher values focus on accuracy, providing more precise recommendations. This

Table 3.5: *Evaluation of MycGNN on Retail Rocket dataset for varying β parameters*

$\beta = 0$, $\gamma = 0$	**Accuracy (%)**	**Recall (%)**	**F1-Score (%)**	**ILD (%)**
0	86.49	78.67	83.65	32.87
0.2	88.41	80.41	85.32	32.22
0.4	90.33	82.17	86.99	31.55
0.6	92.26	83.91	88.67	30.90
0.8	94.18	85.66	90.34	30.24
1	96.10	87.41	92.94	29.58

Table 3.6: *Evaluation of MycGNN on MovieLens1M dataset for varying β parameters*

$\beta = 0$, $\gamma = 0$	**Accuracy (%)**	**Recall (%)**	**F1-Score (%)**	**ILD (%)**
0	84.59	77.53	83.34	37.62
0.2	86.46	79.26	85.19	36.57
0.4	88.34	80.98	87.04	35.89
0.6	90.22	82.70	88.90	35.21
0.8	92.10	84.43	90.75	34.54
1	93.98	86.15	92.60	33.86

flexibility is essential to adapt recommendations to the specific requirements of each user and the global objectives of application domains such as e-commerce.

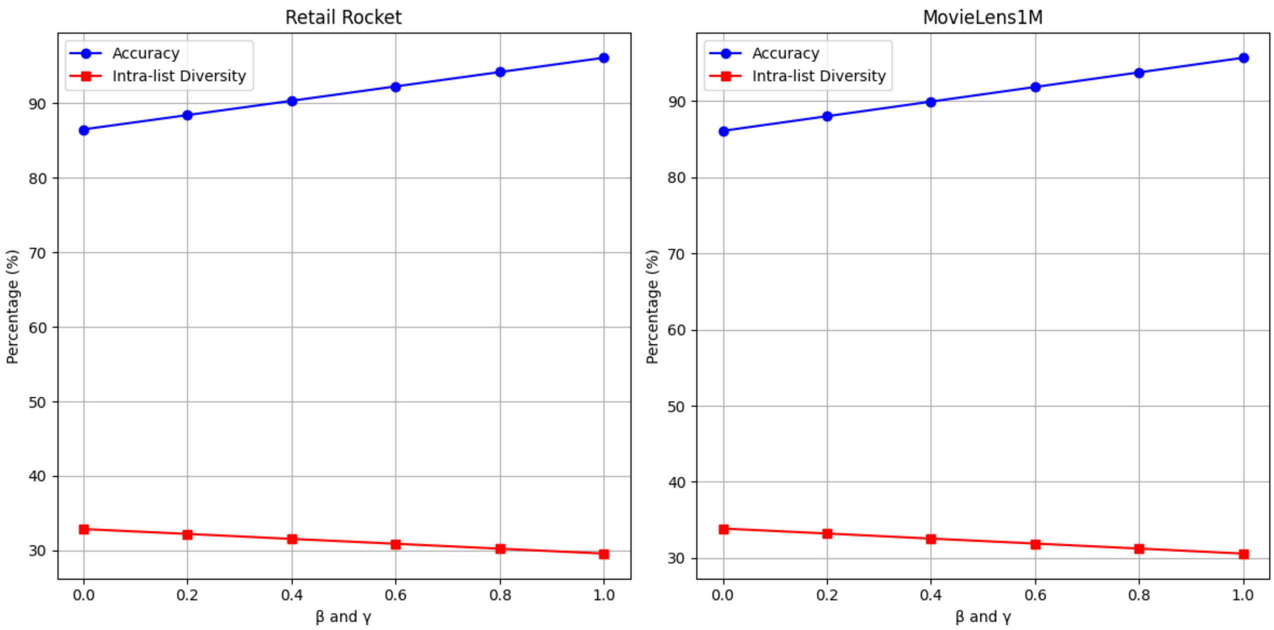

Figure 3.4: *Balancing accuracy and diversity in MycGNN recommendations.*

Tables 3.7a and 3.7b present the performance of MycGNN on top K recommendations for the Retail Rocket and MovieLens1M datasets, respectively, highlighting the accuracy and ILD values when $\beta = 1$ and $\gamma = 1$. As expected, accuracy decreases as intra-list diversity increases. For Retail Rocket, accuracy drops from 95.63% at $K = 1$ to 66.12% at $K = 10$, while intra-list diversity rises from 0% to 56.97%. For MovieLens1M, accuracy drops from 98.47% to 67.04%, while diversity increases from 0% to 59.31%.

Table 3.7: MycGNN performance on top K recommendations for Retail Rocket and MovieLens1M datasets when $\beta = 1$ and $\gamma = 1$

Top K	**Accuracy (%)**	**ILD (%)**
1	95.63	0.00
2	92.76	19.34
3	91.11	23.62
4	87.98	28.19
5	84.45	33.83
6	79.12	37.27
7	77.28	39.45
8	74.90	42.10
9	69.42	52.85
10	66.12	56.97

(a) Retail Rocket dataset, $\beta = 1$, $\gamma = 1$

Top K	**Accuracy (%)**	**ILD (%)**
1	98.47	0.00
2	96.12	7.37
3	94.38	12.49
4	91.93	18.25
5	86.43	30.92
6	81.09	38.12
7	77.32	44.16
8	74.31	47.59
9	70.98	56.42
10	67.04	59.31

(b) MovieLens1M dataset, $\beta = 1$, $\gamma = 1$

Tables 3.8a and 3.8b present the performance of MycGNN on top K recommendations for the Retail Rocket and MovieLens1M datasets, respectively, highlighting accuracy and ILD values when $\beta = 0$ and $\gamma = 0$. Here, intra-list diversity is emphasized, leading to lower accuracy values. For Retail Rocket, accuracy decreases from 86.07% at $K = 1$ to 59.51% at $K = 10$, while diversity increases from 0% to 54.16%. Similarly, for MovieLens1M, accuracy declines from 88.62% to 68.34%, while diversity rises from 0% to 51.26%.

Table 3.8: MycGNN performance on top K recommendations for Retail Rocket and MovieLens1M datasets when $\beta = 0$ and $\gamma = 0$

Top K	**Accuracy (%)**	**ILD (%)**
1	86.07	0.00
2	83.48	23.78
3	81.99	29.29
4	79.18	34.13
5	76.01	38.19
6	71.21	41.45
7	69.55	44.20
8	67.41	48.82
9	62.48	52.57
10	59.51	54.16

(a) Retail Rocket dataset, $\beta = 0$, $\gamma = 0$

Top K	**Accuracy (%)**	**ILD (%)**
1	88.62	0.00
2	86.50	17.41
3	84.94	21.26
4	82.73	25.37
5	79.73	30.45
6	77.90	33.54
7	74.67	35.51
8	72.34	37.89
9	70.98	47.57
10	68.34	51.26

(b) MovieLens1M dataset, $\beta = 0$, $\gamma = 0$

These results highlight MycGNN's flexibility in balancing accuracy and diversity, making it adaptable to different application scenarios in e-commerce.

3.4.6 Comparative Evaluation with State-of-the-Art Models

Some research studies in the field of recommender systems prioritize increasing accuracy over diversity. The MycGNN model is distinguished not only by its focus on recommendation diversity but also by the high accuracy it can achieve, enabling flexible control over the trade-off between accuracy and diversity.

To further verify the performance of the proposed algorithm, Table 3.9 shows a comparison of MycGNN with state-of-the-art method on the MovieLens1M dataset. MycGNN produces a significant improvement in Accuracy, Recall, and F1-Score.

Table 3.9: Comparison of MycGNN ($\beta = 0$, $\gamma = 0$) with state-of-the-art methods on MovieLens1M dataset

Metrics / Ref	**MycGNN**	**[82]**	**[110]**
Accuracy (%)	84.59	-	-
Precision (%)	90.09	28.41	60.90
Recall (%)	77.53	13.27	57.80
F1-Score (%)	83.34	18.09	59.30

Additionally, we compared MycGNN with previously proposed bio-inspired techniques, namely Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO), on the same datasets. Table 3.10a and Table 3.10b presents the obtained results, showing that MycGNN outperforms both ACO- and ABC-based models in terms of Accuracy, Recall, and F1-Score, while providing higher intra-list diversity.

Table 3.10: Comparison of MycGNN ($\beta = 0$, $\gamma = 0$) with nature-inspired models on Retail Rocket and MovieLens1M datasets

Metrics	**MycGNN**	**ACO**	**ABC**
Accuracy (%)	86.49	97.52	97.48
Recall (%)	78.67	88.94	89.15
F1-Score (%)	83.65	94.21	94.01
ILD (%)	32.87	29.12	31.78

(a) Retail Rocket dataset, $\beta = 0$, $\gamma = 0$

Metrics	**MycGNN**	**ACO**	**ABC**
Accuracy (%)	84.59	96.94	95.42
Recall (%)	77.53	87.67	87.23
F1-Score (%)	83.34	93.87	93.89
ILD (%)	37.62	33.67	34.12

(b) MovieLens1M dataset, $\beta = 0$, $\gamma = 0$

3.4.7 Interpretation of Results

Figures 3.4 and Tables 3.5, 3.6, 3.7a, 3.7b , 3.8a, 3.8b illustrate the evaluation of MycGNN on the Retail Rocket and MovieLens1M datasets, highlighting its suitability across diverse contexts.

On the Retail Rocket dataset, MycGNN demonstrates strong results in Accuracy, Recall, and F1-Score, reflecting its ability to identify relevant products while balancing precision and recall. However, increasing intra-list diversity leads to a gradual decrease in accuracy, indicating that prioritizing diversity may slightly reduce recommendation precision. Thus, an optimal trade-off must be found to provide diverse yet accurate recommendations.

On the MovieLens1M dataset, the model maintains high precision with slightly lower Recall and F1-Score compared to Retail Rocket. Diversity trends similarly increase with larger recommendation lists. These results reflect MycGNN's ability to recommend both diverse and relevant movies despite more complex user preferences.

The differences in dataset characteristics explain performance variations: Retail Rocket features frequent, direct user interactions, providing rich modeling data, while MovieLens1M captures more abstract user preferences through movie ratings, which may require the model to capture subtler relationships.

Moreover, the mycelium-inspired mechanisms substantially enhance recommendation diversity. Tables 3.9 and 3.10a, 3.10b show that MycGNN maintains a strong balance between diversity and accuracy, outperforming both conventional and nature-inspired methods in terms of overall recommendation quality.

3.4.8 Discussion and Limitations

While MycGNN achieves high accuracy and diversity on both datasets, it faces several limitations:

- **Dataset Size and Sparsity:** The model relies on high-dimensional graphs; performance may degrade on smaller datasets or datasets with sparse interactions.
- **Dropout and Information Loss:** Dropout mitigates overfitting but can ignore potentially valuable signals, possibly affecting performance in some scenarios.
- **Dynamic Environments:** In real-world settings with evolving user behavior, model performance may decrease without continuous adaptation.
- **Interpretability:** As a deep learning-based GNN, MycGNN has inherent limits in interpretability, making it challenging to understand specific decision processes.
- **Scalability:** Although computationally manageable, large-scale, real-time deployments may face challenges due to the model's complexity.

In summary, MycGNN demonstrates strong capabilities in exploiting complex user-item interactions for accurate and diverse recommendations. Future work should focus on improving efficiency, interpretability, and real-time applicability to ensure practical deployment in dynamic e-commerce environments.

3.5 Conclusion

This chapter presented MycGNN, a mycelium-inspired graph neural network that enhances recommender systems by integrating dynamic edge reweighting and exploration-exploitation strategies. Experimental results on the Retail Rocket and MovieLens1M datasets show that MycGNN achieves a strong balance between accuracy, F1-Score, Recall, and intra-list diversity, demonstrating its ability to provide both relevant and diverse recommendations.

Despite its effectiveness, the model faces limitations related to high-dimensional graphs, interpretability, and real-time deployment. Building on these insights, the next chapter introduces SFNN, a *Secure Fusion Neural Network*-based recommender system that ensures data integrity and confidentiality while optimizing both accuracy and diversity through the integration of a GNN and a regularized variational autoencoder.

Chapter 4: SFNN: A Secure and Diverse Recommender System through Graph Neural Network and Variational Autoencoder

Contents

4.1 Introduction

This chapter presents the **Secure Fusion Neural Network (SFNN)** [141], a unified recommendation framework designed to simultaneously address accuracy, diversity, and data security. SFNN combines graph-based learning with generative modeling in a single architecture and leverages blockchain technology to ensure both the confidentiality and integrity of user data. By jointly capturing user–item interactions and generating diversity-aware latent representations, SFNN provides an adaptive and principled approach suitable for contemporary recommendation scenarios.

Extensive evaluations on multiple real-world datasets demonstrate that SFNN maintains a strong balance between recommendation relevance and diversity while enforcing robust security measures, highlighting its potential for practical deployment in large-scale applications.

4.2 Motivation and Design Objectives

Although recommender systems have seen notable advancements, many existing solutions treat accuracy, diversity, and security independently or rely on post-processing strategies such as re-ranking. This decoupled approach often results in trade-offs: increasing accuracy can reduce diversity, while security mechanisms are typically applied as an external layer rather than being integrated. Such limitations restrict the effectiveness of recommendation systems in real-world contexts, where high-quality personalization, user trust, and data protection are all essential.

The primary motivation of this work is to develop an end-to-end recommender system that simultaneously maximizes recommendation relevance and diversity while ensuring the integrity and confidentiality of data. To achieve this, the **SFNN** framework integrates complementary learning paradigms into a cohesive architecture.

SFNN is founded on three principal components. First, a graph neural network (GNN) models user–item interactions by leveraging the global structure of the interaction graph, capturing both explicit and implicit relationships to enhance recommendation accuracy. Second, a regularized variational autoencoder (Reg-VAE) generates stochastic latent representations that promote diversity. Its learning is guided by a modified evidence lower bound (ELBO) loss function augmented with a diversity-promoting regularization term, reducing redundancy and mitigating similarity bias in recommended items. Third, blockchain technology is employed to secure the data through encryption and decentralized validation, providing a tamper-resistant and trustworthy storage mechanism.

To integrate the outputs of the GNN and Reg-VAE, SFNN uses an adaptive weighted fusion mechanism. Fusion parameters are optimized using a desirability function, allowing the system to dynamically balance accuracy and diversity according to application needs and overcoming the inflexibility of static fusion methods.

Based on these motivations, SFNN aims to:

- Provide a unified, adaptive recommendation framework that simultaneously optimizes accuracy, diversity, and security.
- Enhance recommendation diversity through generative latent representations regulated by a diversity-aware ELBO term.
- Implement an adaptive fusion mechanism that dynamically balances the complementary strengths of GNN and Reg-VAE.
- Optimize fusion parameters via a desirability function to achieve an effective trade-off between relevance and diversity.
- Guarantee strong data confidentiality and integrity using blockchain-based encryption and decentralized storage.
- Validate the effectiveness of the framework through comprehensive experiments, including ablation studies and benchmarking against state-of-the-art models.

4.3 SFNN Framework

This section presents the overall architecture of the **Secure Fusion Neural Network (SFNN)**, a unified recommendation framework designed to jointly address the trade-off between accuracy and diversity while enforcing data security and integrity. As illustrated in Figure 4.1, SFNN combines two complementary neural components—a graph neural network (GNN) and a regularized variational autoencoder (Reg-VAE)—within a blockchain-enabled security infrastructure.

The recommendation process starts with the encryption of user and item information, which is subsequently stored on a blockchain ledger to prevent unauthorized access and ensure data immutability. On top of this secure data layer, the GNN component models the complex relational structure of user–item interactions through graph representations, enabling accurate preference learning and facilitating the discovery of less-exposed items.

In parallel, the Reg-VAE component promotes recommendation diversity by learning stochastic latent representations guided by a modified evidence lower bound (ELBO). This objective function is augmented with a diversity-aware regularization term, which encourages latent-space dispersion and reduces redundancy among recommended items.

To exploit the complementary strengths of both components, SFNN employs an adaptive weighted fusion mechanism that combines the prediction scores of the GNN and Reg-VAE. The fusion weights are optimized using a desirability function, allowing the system to dynamically balance recommendation relevance and diversity. This adaptive integration enables SFNN to

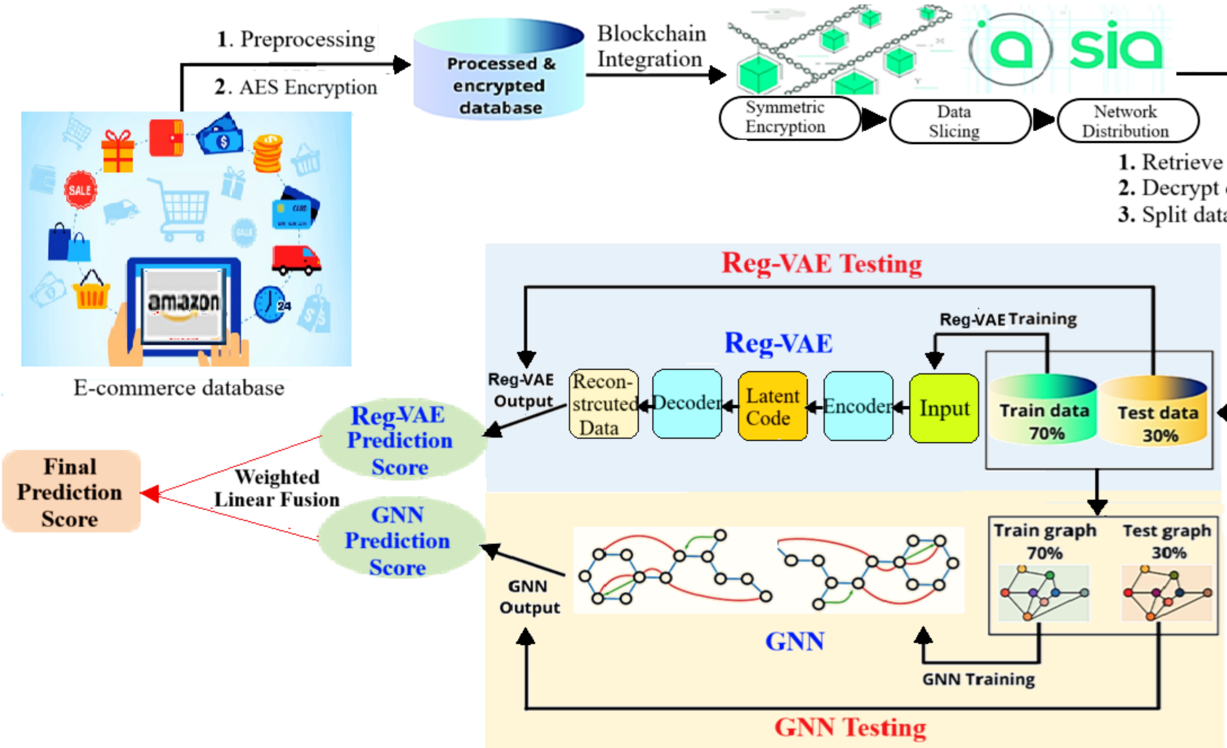

Figure 4.1: Proposed SFNN-based recommendation framework

align personalized recommendations with user preferences while maintaining overall system efficiency.

4.3.1 Data Preprocessing

Data quality and consistency play a critical role in the performance of the proposed SFNN framework. Therefore, a dedicated preprocessing pipeline was applied prior to model training to ensure the reliability and suitability of the input data. As expected in large-scale datasets, incomplete, duplicated, or inconsistent records may be present. Entries lacking essential identifiers, such as user or item IDs, were removed, and duplicate interactions were eliminated to preserve the uniqueness of each observation.

To avoid bias caused by scale disparities among numerical features, normalization and feature scaling were applied. Specifically, Min–Max normalization was employed to rescale numerical attributes into the interval $[0, 1]$, as defined in Equation (1):

$$x'_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

Missing values were handled according to the nature of the corresponding attributes. For numerical features, missing entries were imputed using the mean value computed from the available observations, as shown in Equation (2). For categorical attributes, the most frequent category was used to ensure consistency and limit bias, as expressed in Equation (3):

$$x_i' = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.2}$$

$$x_i' = \mathrm{Mode}(x) \tag{4.3}$$

Categorical features, including user-related attributes and item categories, were subsequently transformed using one-hot encoding. This encoding strategy enables the representation of nominal variables without introducing artificial ordinal relationships, as illustrated in Equation (4):

$$Y = \begin{cases} (1,0,0) & \text{if } y = A \\ (0,1,0) & \text{if } y = B \\ (0,0,1) & \text{if } y = C \end{cases} \tag{4.4}$$

Finally, interaction timestamps were normalized to account for the temporal dynamics of user behavior. This step ensures that recent interactions are not overshadowed by older ones and allows the model to better capture evolving user preferences. The normalization process is defined in Equation (5):

$$t_i' = \frac{t_i - \min(T)}{\max(T) - \min(T)} \tag{4.5}$$

4.3.2 Data Security and Integrity

To ensure the confidentiality and integrity of user and item data, SFNN employs a two-layered security approach. First, the preprocessed dataset D is locally encrypted using AES 256-bit symmetric encryption, producing an encrypted dataset C:

$$C = \mathrm{Encrypt}_K(D) \tag{4.6}$$

Next, the encrypted data is stored on the decentralized Sia storage platform, which leverages blockchain technology for secure and tamper-proof storage. Before distribution across the network, a second encryption layer is applied using a new key K_2, yielding a doubly encrypted dataset C':

$$C' = \mathrm{Encrypt}_{K_2}(C) \tag{4.7}$$

The dataset C' is then divided into n fragments for distributed storage:

$$C_i', \quad i = 1, \ldots, n = \mathrm{Split}(C') \tag{4.8}$$

Each transaction is validated by miners M on the Sia blockchain, and a new block B_{new} is appended to ensure immutable record-keeping:

$$B_{\text{new}} = \text{AddBlock}(T, M) \tag{4.9}$$

Data integrity is further verified using SHA-256, generating a unique 256-bit hash at each stage. The AES 256-bit keyspace (2^{256}) ensures resistance against brute-force attacks, providing strong data protection.

4.3.3 GNN Implementation

Once retrieved, the encrypted dataset C is decrypted locally using the original key K:

$$D = \text{Decrypt}_K(C) \tag{4.10}$$

The decrypted data is split into training and test sets. One copy is used for the Reg-VAE, while the other constructs training and test graphs for the GNN.

The GNN represents user-item interactions as an undirected graph $G = (V, E)$, where nodes V include both users and items. Each node $v_i \in V$ is assigned a type $t(v_i)$:

$$t(v_i) = \begin{cases} \text{User}, & \text{if node represents a user} \\ \text{Item}, & \text{if node represents an item} \end{cases} \tag{4.11}$$

Edges E represent interactions between users and items, optionally annotated with timestamps $(v_i, v_j) \mapsto t$. Sensitive user information is omitted, focusing only on relational and temporal data.

Node features h_i are initialized based on relevant attributes. Node representations are updated using message-passing operations over neighboring nodes $N(v_i)$:

$$h_i^{(l+1)} = \sigma\Big(W^{(l)}\, \text{AGG}\Big(h_j^{(l)} : v_j \in N(v_i)\Big)\Big) \tag{4.12}$$

Here, $h_i^{(l+1)}$ is the updated feature vector of node v_i, σ is a non-linear activation function (ReLU), $W^{(l)}$ is the trainable weight matrix, and AGG is the aggregation function over neighbors. Stacking multiple layers allows the GNN to capture complex user-item interactions, producing embeddings suitable for high-accuracy recommendations.

4.3.4 Reg-VAE Implementation

To enhance recommendation diversity, SFNN incorporates a **Regularized Variational Autoencoder (Reg-VAE)**, which integrates a diversity-promoting regularization term into the loss function during training represented in Figure 4.2. This encourages the model to generate a broader set of recommendations.

The Reg-VAE encoder maps input vectors x, representing a user's interaction history, into a probabilistic latent representation z. The encoder outputs the mean μ and variance σ^2 vectors that define the distribution of z. Sampling from this latent space is performed using the reparameterization trick:

$$z = \mu + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0,1) \tag{4.13}$$

The decoder reconstructs the user-item interaction vector x' from the latent representation z, predicting the likelihood of user interactions with items:

$$x' = \text{Decoder}(z) \tag{4.14}$$

By sampling multiple latent vectors z per user, the Reg-VAE introduces controlled stochasticity, promoting diversity by including items that may be less popular but consistent with the user's preferences.

The Reg-VAE is trained by maximizing the evidence lower bound (ELBO), which balances reconstruction accuracy and latent-space regularization:

$$\text{ELBO} = \mathbb{E}_{q(z|x)}\Big[\log p(x|z)\Big] - \text{KL}\Big(q(z|x)\|p(z)\Big) \tag{4.15}$$

Here, the first term measures the reconstruction likelihood, while the Kullback-Leibler divergence (KL) regularizes $q(z|x)$ to remain close to the prior $p(z)$, typically $\mathcal{N}(0,1)$.

To explicitly encourage diversity, a regularization term based on pairwise similarity of latent representations is added. For a set of recommended items with latent vectors $\{z_1, z_2, \ldots, z_n\}$, the diversity-promoting term is defined as:

$$\ell_{\text{diversity}} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{sim}(z_i, z_j) \tag{4.16}$$

where $\text{sim}(z_i, z_j)$ measures the similarity between latent vectors z_i and z_j. A high value indicates that items are too similar in the latent space.

The total Reg-VAE loss combines the ELBO and the diversity regularization term:

$$\mathcal{L}_{\text{Reg-VAE}} = \text{ELBO} + \lambda\,\ell_{\text{diversity}} \tag{4.17}$$

The hyperparameter λ controls the influence of the diversity term on learning: larger values encourage greater diversity in the recommendations. The encoder parameters θ and decoder parameters ϕ are updated via gradient descent:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{Reg-VAE}}, \quad \phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{\text{Reg-VAE}} \tag{4.18}$$

where η is the learning rate controlling the step size during optimization. This training procedure ensures that the Reg-VAE learns both accurate and diverse latent representations for recommendation generation.

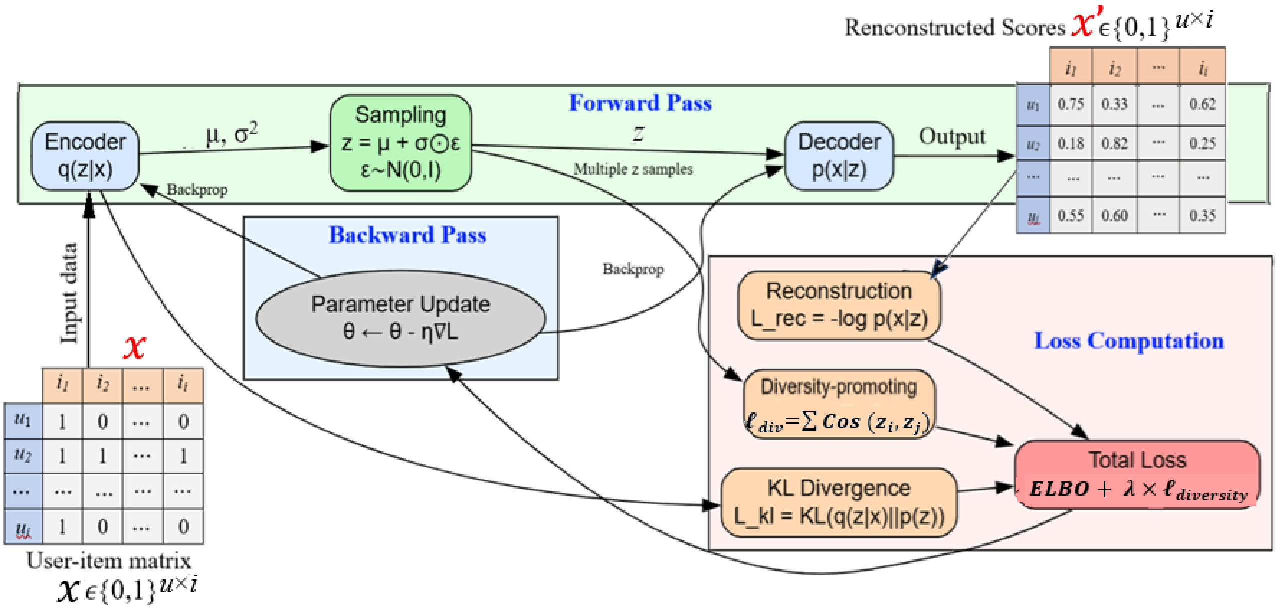

Figure 4.2: Overview of the Diversity-Regularized VAE training process.

4.3.5 Weighted Linear Fusion Mechanism

Within the SFNN framework, the GNN and Reg-VAE components generate separate prediction scores for each item, reflecting complementary aspects of the recommendation task. The GNN emphasizes accuracy by leveraging the user-item interaction graph to estimate the likelihood of user engagement based on historical interactions and temporal information. Conversely, the Reg-VAE encourages diversity by exploring underrepresented items in the user preference space and capturing latent factors that reveal less obvious preferences.

To integrate these complementary predictions, we employ an adaptive weighted linear fusion strategy. Prior to fusion, the output scores from both models are normalized to a common scale. The final prediction score for each item is computed as a weighted sum of the normalized GNN and Reg-VAE scores:

$$\text{FinalPredictionScore} = \alpha \cdot \text{GNNPredictionScore} + \beta \cdot \text{Reg-VAE PredictionScore} \tag{4.19}$$

Here, α and β are weighting factors that control the trade-off between accuracy and diversity. These parameters are optimized using a desirability function (DF) [141], allowing the system to adaptively balance highly relevant items with less explored, diverse recommendations.

Algorithm 4.1: SFNN Recommendation Process

Step 1: Input: Raw dataset $D = \{(u_i, i_j, t_k, \dots)\}$, symmetric keys (AES-256), blockchain key, hyperparameters α, β
Step 2: Output: Top-N secure and diverse personalized recommendations
Step 3: Data Preprocessing:
  3.1: Remove duplicates
  3.2: Normalize numerical features using
  3.3: Impute missing values using and one-hot encode categorical features
  3.4: Scale features to [0,1]
Step 4: AES Encryption:
  4.1: Encrypt local data using AES-256
Step 5: Blockchain Integration:
  5.1: Re-encrypt data
  5.2: Split data into fragments
  5.3: Record and validate transactions
Step 6: Retrieve and Decrypt Data:
  6.1: Decrypt blockchain blocks
  6.2: Split dataset into train/test sets
Step 7: GNN Training:
  7.1: Construct graphs from training data
  7.2: Initialize GNN parameters and train the network
Step 8: Reg-VAE Training:
  8.1: Encode user history
  8.2: Decode preferences
  8.3: Optimize Reg-VAE loss
Step 9: Weighted Linear Fusion:
  9.1: Compute final prediction scores using
Step 10: Recommendation Generation:
  10.1: Rank items and select top-N recommendations per user

The main steps of SFNN are summarized in Algorithm 4.1.

4.3.6 Computational Complexity Analysis

The computational cost of SFNN is driven by its three primary components: GNN, Reg-VAE, and the blockchain security layer.

GNN. The complexity of GNN computations is as follows:

- Embedding layer: $O(N \times d)$
- Each graph convolutional layer: $O(E \times f^2)$
- Attention mechanism (if applied): $O(h \times N \times f^2)$

where N is the number of nodes, E is the number of edges, d is the embedding size, f is the feature dimension, and h is the number of attention heads.

Reg-VAE. For diversity modeling, the complexity is:

$$O(N \times d_e) + O(N \times d_d) \tag{4.20}$$

where d_e and d_d are the dimensions of the encoder latent space and decoder output, respectively.

Blockchain Layer. Ensuring data integrity and confidentiality adds:

$$O(D \times k) + O(B \times H) \tag{4.21}$$

where D is the dataset size, k is the encryption key length, B is the block size, and H is the number of hashes per block.

Overall Complexity. Combining all components, the SFNN computational complexity can be approximated as:

$$O(Nd + Ef^2 + Nd_e + Dk + BH) \tag{4.22}$$

This analysis demonstrates that SFNN is scalable for large-scale recommender systems, maintaining high accuracy and diversity while ensuring robust security for user data.

4.4 Experimental Evaluation

To assess the effectiveness and practical applicability of the proposed SFNN model, extensive experiments were performed. These experiments aim to answer the following research questions:

- **RQ1:** Are the user data securely stored, and is data integrity maintained across all operations?
- **RQ2:** What computational overhead is introduced by the security and integrity mechanisms?
- **RQ3:** How effective is the hybrid SFNN system compared to standalone models (GNN or Reg-VAE)?
- **RQ4:** How do variations in the weighting factors α and β affect recommendation accuracy and diversity?
- **RQ5:** How does SFNN perform in comparison with state-of-the-art baseline?

4.4.1 Datasets

To comprehensively evaluate SFNN, experiments were conducted on three widely-used datasets. All datasets were preprocessed and transformed into graph representations. Table 4.1 provides a summary of their characteristics.

Retail Rocket Dataset. This dataset captures consumer interactions on a real-world e-commerce platform over approximately 4.5 months. It contains item metadata, category hierarchies, and behavioral logs, including page visits, cart actions, and purchases. These interactions allow the analysis of user behavior and item relationships in a practical e-commerce environment.

Amazon Product Dataset (Clothing Subset). Extracted from Amazon, this dataset contains user reviews, ratings, and product metadata. The subset considered in this study focuses on the "Clothing" category, providing insights into user preferences, item popularity, and interaction patterns suitable for recommendation tasks.

MovieLens 1M Dataset. This benchmark dataset contains one million ratings from 6,040 users on 3,706 movies. It has been widely adopted in recommender systems research for evaluating personalization algorithms and analyzing user behavior in a controlled, standardized environment.

Table 4.1: Summary of the datasets used for evaluating SFNN

Dataset	**#Users**	**#Items**	**Avg. Actions/User**	**#Actions**
Retail Rocket	1,407,580	417,053	1.9	2,756,101
Amazon Clothing	39,387	23,033	9.5	371,686
MovieLens 1M	6,040	3,706	163.6	1,000,209

These datasets vary in size, sparsity, and domain, enabling the evaluation of SFNN under diverse and realistic conditions. The preprocessing ensures that interactions, timestamps, and item/user features are properly aligned for the GNN and Reg-VAE components while maintaining data security through encryption and blockchain storage.

4.4.2 Model Architecture

The architecture of the SFNN model was optimized through an iterative process in which one parameter was tuned at a time while keeping the others fixed. This systematic approach allowed the identification of an optimal configuration for both the GNN and Reg-VAE components, as summarized in Tables 4.2 and 4.3.

Table 4.2: Graph Neural Network (GNN) Architecture

Layer	**Type**	**Dimension**	**Activation**	**Hyperparameters**
Input	Node Embedding	128	-	-
1	Graph Convolution	64	ReLU	Learning rate: 0.01
2	Graph Convolution	64	ReLU	Dropout: 0.5
3	Attention Layer	64	LeakyReLU (0.2)	Attention heads: 4
Output	Fully Connected	1	Sigmoid	-

4.4.3 Data Security and Integrity Tests (RQ1)

To validate the security of the stored datasets, we conducted theoretical evaluations of AES-256 encryption robustness under various scenarios. Table 4.4 shows the estimated time to brute-force AES-256 encryption, demonstrating the model's resistance to unauthorized access.

Table 4.3: Regularized Variational Autoencoder (Reg-VAE) Architecture

Layer	Type	Dimension	Activation	Hyperparameters
Input	User-item interaction matrix	#users × #items	-	-
Encoder 1	Fully Connected	128	ReLU	Learning rate: 0.001
Encoder 2	Fully Connected	64	ReLU	Dropout: 0.3
Latent Space	Mean and Variance	64	-	Reparameterization
Decoder 1	Fully Connected	64	ReLU	-
Decoder 2	Fully Connected	128	ReLU	-
Output	Reconstructed interaction vector	#users × #items	Sigmoid	-
Loss	Reg-VAE			$\lambda = 0.01$
Optimizer	Adam			Learning rate: 0.001

The pre-encryption dataset requires approximately 3.67×10^{63} years to compromise, which decreases slightly after encryption and blockchain integration but remains astronomically high, emphasizing the effectiveness of the security layer.

Table 4.4: Estimated time to break AES-256 encryption under different scenarios

Scenario	Keys Tested/sec	Time to Break AES-256 (years)
Pre-Encryption	10^6	3.67×10^{63}
Post-Encryption	10^{12}	3.67×10^{57}
Post-Blockchain Integration	10^{15}	3.67×10^{54}

Data integrity was further verified using SHA-256 hashing at critical stages. Table 4.5 demonstrates that hash values remain consistent across encryption, blockchain integration, and decryption, ensuring the trustworthiness and immutability of user data.

Table 4.5: SHA-256 hash values for the datasets at different stages

Dataset	Process Step	SHA-256 Hash
Retail Rocket	Pre-Encryption	b4d9c8e8f3e9c7d2a124f5e789ab65e7dfc4a5b6c8d9e8f7b9a2c7d6e5f3f1f0
	Post-Encryption	b4d9c8e8f3e9c7d2a124f5e789ab65e7dfc4a5b6c8d9e8f7b9a2c7d6e5f3f1f0
	Post-Blockchain	b4d9c8e8f3e9c7d2a124f5e789ab65e7dfc4a5b6c8d9e8f7b9a2c7d6e5f3f1f0
	Post-Decryption	b4d9c8e8f3e9c7d2a124f5e789ab65e7dfc4a5b6c8d9e8f7b9a2c7d6e5f3f1f0
Clothing	Pre-Encryption	a3f5f4c8bf1bed0a4c36e9e3ab2b978efc570a5e8c5e8b4a8e4a7e65c1b2c7c6
	Post-Encryption	a3f5f4c8bf1bed0a4c36e9e3ab2b978efc570a5e8c5e8b4a8e4a7e65c1b2c7c6
	Post-Blockchain	a3f5f4c8bf1bed0a4c36e9e3ab2b978efc570a5e8c5e8b4a8e4a7e65c1b2c7c6
	Post-Decryption	a3f5f4c8bf1bed0a4c36e9e3ab2b978efc570a5e8c5e8b4a8e4a7e65c1b2c7c6
MovieLens1M	Pre-Encryption	c6d7e8f9a0b1c2d3e4f5a6b7c8d9e0f1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7
	Post-Encryption	c6d7e8f9a0b1c2d3e4f5a6b7c8d9e0f1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7
	Post-Blockchain	c6d7e8f9a0b1c2d3e4f5a6b7c8d9e0f1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7
	Post-Decryption	c6d7e8f9a0b1c2d3e4f5a6b7c8d9e0f1a2b3c4d5e6f7g8h9i0j1k2l3m4n5o6p7

4.4.4 Security Overhead (RQ2)

We measured the additional computational cost induced by AES-256 encryption/decryption and blockchain verification across all datasets. Results in Table 4.6 indicate that AES operations add 2.3–2.9 ms per transaction, while blockchain verification introduces 3.8–4.5 ms. The total overhead remains modest (6.1–7.4 ms per operation), resulting in only 1.7–2.1% relative slowdown, demonstrating that SFNN integrates strong security measures without significantly affecting computational efficiency.

Table 4.6: Computational overhead introduced by AES-256 encryption and blockchain verification

Dataset	**AES-256 (ms)**	**Blockchain (ms)**	**Total Overhead (ms)**	**Relative Slowdown (%)**
Retail Rocket	2.3	3.8	6.1	1.7
Clothing	2.5	4.1	6.6	1.9
MovieLens1M	2.9	4.5	7.4	2.1

4.5 Results and Discussion

4.5.1 Ablation Study (RQ3)

An ablation analysis is conducted to quantify the individual contribution of each component within the proposed SFNN framework. The complete SFNN model jointly exploits the structural learning capabilities of the GNN and the generative diversity mechanisms of the Reg-VAE, where the latter is guided by a diversity-aware regularization term embedded in its loss function. This unified training strategy allows the model to simultaneously capture user–item relational patterns and encourage diverse recommendation outputs.

To better understand the role of each module, three simplified variants of SFNN are evaluated by selectively removing specific components. The first variant, denoted **SFNN-GNN**, excludes the Reg-VAE entirely and relies solely on the graph neural network for recommendation generation. The second variant, **SFNN-VAE (with reg)**, removes the GNN and retains only the Reg-VAE enhanced with the diversity-promoting regularization term defined in Eq. (16). The third variant, referred to as **SFNN-VAE**, keeps the Reg-VAE architecture but removes the diversity regularization term from the objective function.

The quantitative results reported in Table 4.7 highlight clear performance trade-offs. The SFNN-GNN variant achieves the highest accuracy across all datasets, with improvements ranging from approximately 17.6% to 19.1% compared to the full SFNN model. This confirms the effectiveness of graph-based relational modeling in improving predictive accuracy. However, this gain is accompanied by a substantial reduction in intra-list diversity, exceeding 64% on all datasets, indicating a strong tendency toward homogeneous and popularity-biased recommendations. Conversely, the SFNN-VAE (with reg) variant achieves the highest diversity scores,

Table 4.7: Ablation study results on Retail Rocket, Clothing, and MovieLens1M datasets

Dataset	**Model**	**Accuracy**	**Recall**	**F1-Score**	**Intra-list Diversity**
Retail Rocket	SFNN	0.7832	0.8157	0.7926	0.4768
	SFNN-GNN	0.9211	0.7307	0.8081	0.1619
	SFNN-VAE	0.5127	0.6625	0.5924	0.3982
	SFNN-VAE (with reg)	0.4464	0.6333	0.5632	0.6208
Clothing	SFNN	0.7428	0.7711	0.7547	0.4211
	SFNN-GNN	0.8844	0.7078	0.7665	0.1326
	SFNN-VAE	0.4783	0.6341	0.5619	0.3727
	SFNN-VAE (with reg)	0.4165	0.5965	0.5382	0.5944
MovieLens1M	SFNN	0.8138	0.8472	0.8170	0.5160
	SFNN-GNN	0.9601	0.7609	0.8259	0.1853
	SFNN-VAE	0.5851	0.7028	0.6412	0.4039
	SFNN-VAE (with reg)	0.5244	0.6722	0.6247	0.6967

with relative gains between 30.2% and 41.2%. This result demonstrates the effectiveness of the diversity regularization term in encouraging exploration and reducing redundancy. Nevertheless, this improvement in diversity comes at the cost of a significant decrease in accuracy, ranging from 35.6% to 43.9%, highlighting the inherent trade-off between relevance and diversity when generative models are used in isolation. These results provide clear empirical evidence that the regularization term plays a pivotal role in shaping the latent representation space by encouraging dispersion and exploration.

Figure 4.3 provides a comprehensive visual summary of the ablation study results. The figure clearly demonstrates that the complete SFNN configuration achieves the most effective trade-off between accuracy and diversity across all evaluated datasets.

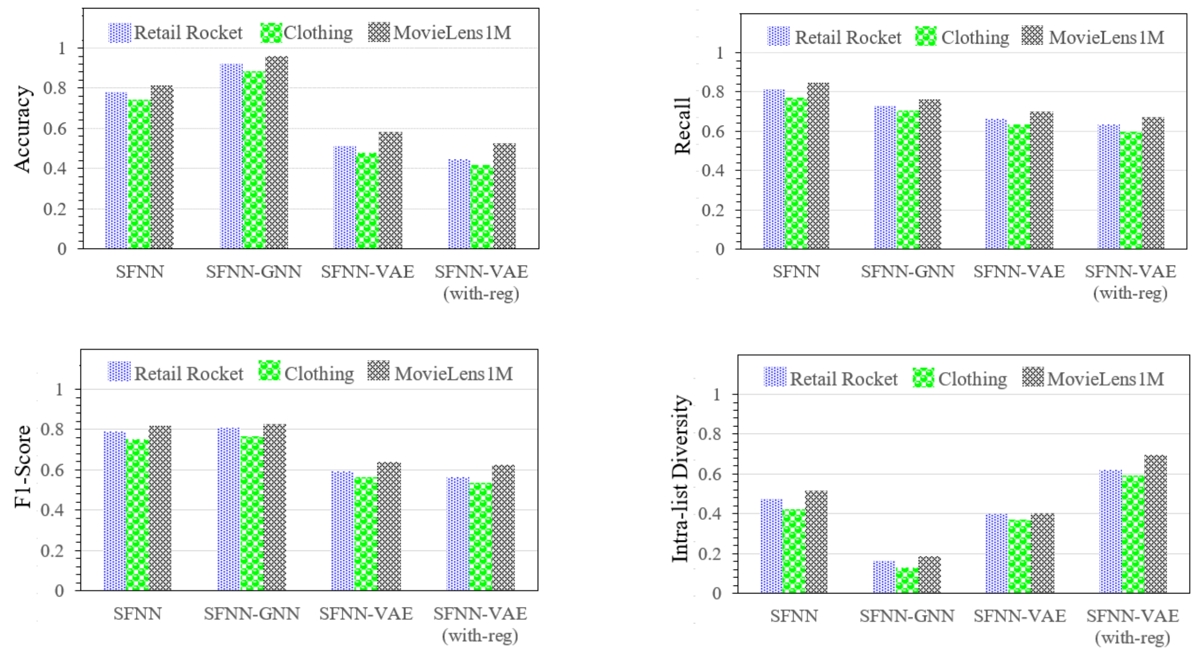

Figure 4.3: Ablation study results visualization

4.5.2 Impact of α and β Parameters on SFNN's Performance (RQ4)

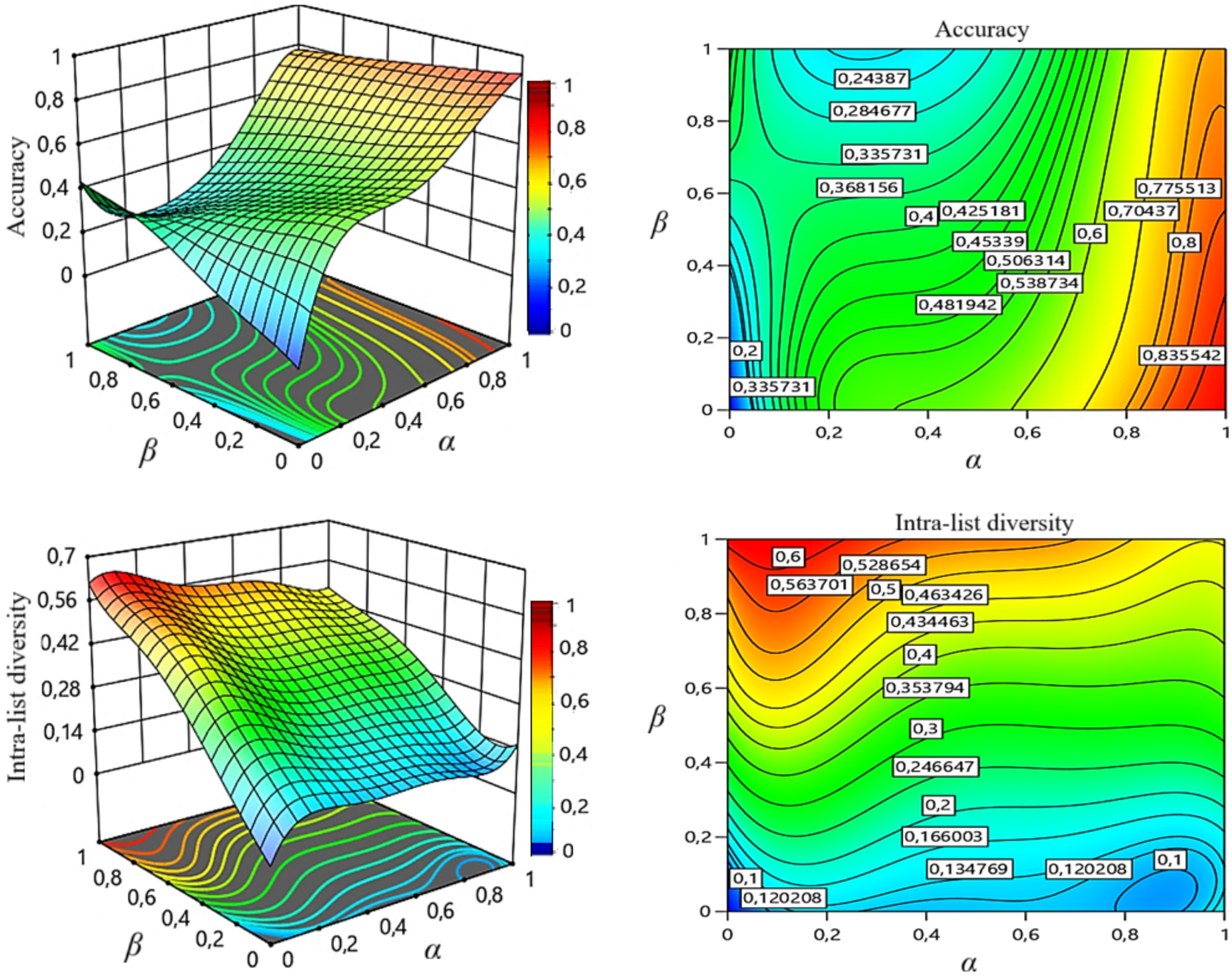

Figure 4.4: Impact of the parameters α and β on accuracy and intra-list diversity for Retail Rocket Dataset.

To analyze the influence of the weighting parameters α and β on the performance of SFNN, three-dimensional response surface plots and contour maps were generated using a full factorial experimental design. The relationship between the control parameters and the evaluation metrics—namely accuracy and intra-list diversity—was modeled using a second-order quadratic regression for the Retail Rocket, Clothing, and MovieLens1M datasets.

Across all datasets, a consistent trend is observed: increasing the value of β leads to higher intra-list diversity, while simultaneously causing a degradation in accuracy. This behavior highlights the inherent trade-off between relevance and diversity within the proposed framework. Specifically, when $\alpha = 0$ and $\beta = 1$, intra-list diversity reaches its maximum values of 0.62, 0.59, and 0.69 for the Retail Rocket, Clothing, and MovieLens1M datasets, respectively. However, under these conditions, accuracy decreases to 0.44, 0.41, and 0.52, confirming that diversity-oriented optimization negatively impacts predictive precision.

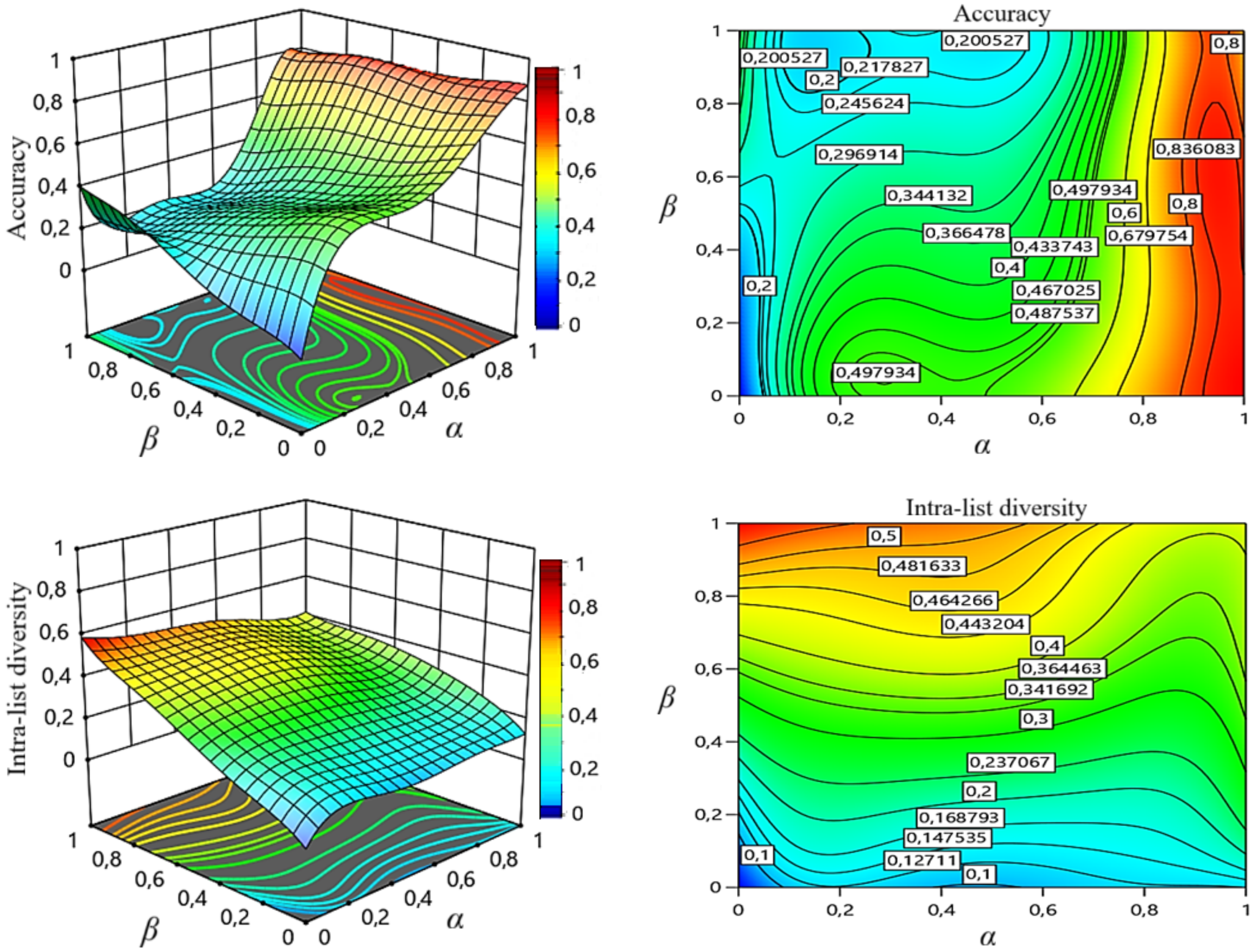

Figure 4.5: Impact of the parameters α and β on accuracy and intra-list diversity for Clothing Dataset.

In contrast, higher values of α strongly favor accuracy. When $\alpha = 1$, accuracy peaks at 0.92 for Retail Rocket, 0.88 for Clothing, and 0.96 for MovieLens1M. Nevertheless, these gains are accompanied by a substantial reduction in intra-list diversity, which drops to 0.16, 0.13, and 0.18, respectively. This observation confirms that α primarily controls accuracy-driven learning, whereas β governs diversity enhancement.

Figure 4.4 illustrates the joint effect of α and β on accuracy and intra-list diversity for the Retail Rocket dataset. It is evident that α has a dominant influence on accuracy compared to β. At higher values of α, accuracy remains consistently high regardless of the β setting. The highest accuracy value of 0.92 is achieved at $\alpha = 1$ and $\beta = 0$. Conversely, intra-list diversity shows limited sensitivity to α in low β regimes, but increases significantly as β grows, particularly when α decreases. The maximum diversity value of 0.64 is obtained at $\alpha = 0.1$ and $\beta = 1$.

Figure 4.5 presents the corresponding analysis for the Clothing dataset. Similar to the previous case, α is identified as the most influential parameter for accuracy. High accuracy values are consistently achieved for any β level when α lies in the range $[0.8, 1]$. On the other hand, β has the strongest effect on intra-list diversity. The highest diversity score of 0.59 is observed at $\alpha = 0$ and $\beta = 1$, confirming the role of β as the primary diversity control parameter.

The impact of α and β on the MovieLens1M dataset is illustrated in Figure 4.6. For high values of α and β values in the range $[0, 0.4]$, the model achieves particularly strong accuracy performance. As observed in the other datasets, α exhibits a stronger influence on accuracy than β, with the highest accuracy value of 0.96 obtained at $\alpha = 1$ and $\beta = 0$. Intra-list diversity remains nearly constant for very small values of β across all α levels, but increases noticeably as β grows. The highest diversity score of 0.69 is achieved when β is close to 1 and α approaches 0.

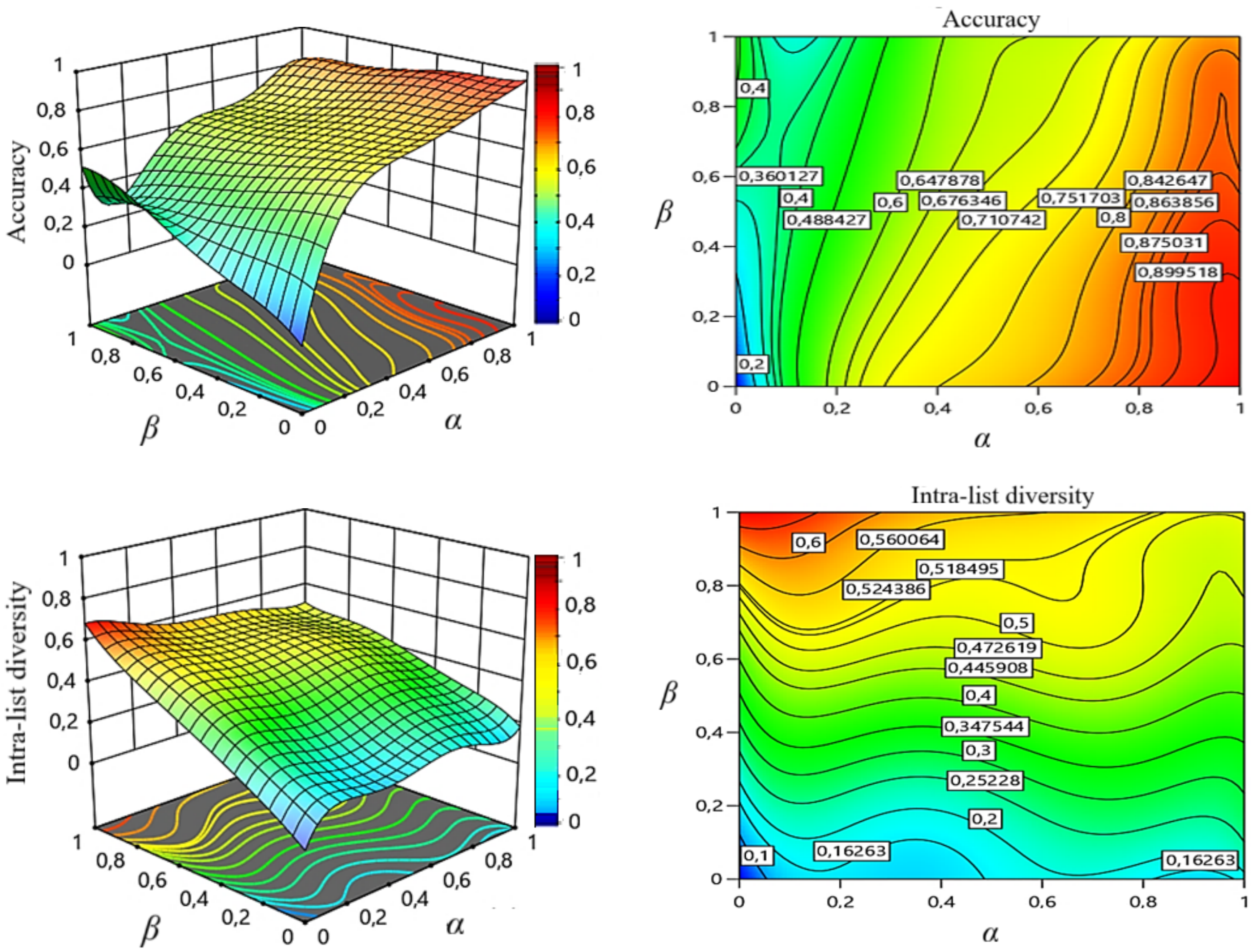

Figure 4.6: Impact of the parameters α and β on accuracy and intra-list diversity for MovieLens1M Dataset.

4.5.3 Optimization of α and β Parameters

To determine the optimal trade-off between recommendation accuracy and intra-list diversity, the desirability function (DF) approach was employed. This multi-objective optimization technique converts each performance metric into a dimensionless desirability score ranging from 0 (completely undesirable) to 1 (fully desirable). The goal is to identify the optimal values of α and β within the interval $[0, 1]$ that simultaneously satisfy both accuracy and diversity objectives.

For each response variable y_i, a one-sided desirability function $d_i(y_i)$ is defined as follows:

$$d_i(y_i) = \begin{cases} 0, & y_i < y_i^{\min} \\ \left(\dfrac{y_i - y_i^{\min}}{y_i^{\max} - y_i^{\min}} \right)^w, & y_i^{\min} \le y_i \le y_i^{\max} \\ 1, & y_i > y_i^{\max} \end{cases} \tag{4.23}$$

where $y_i^{\min}$ and $y_i^{\max}$ represent the lower and upper acceptable bounds of the response variable, and w is a weighting parameter controlling the curvature of the desirability function. In this study, w was set to 0.5 to assign equal importance to accuracy and intra-list diversity.

The overall desirability function is computed as the geometric mean of the individual desirability scores associated with each objective:

$$DF = \left(\prod_{i=1}^{n} d_i(y_i) \right)^{\frac{1}{n}}, \tag{4.24}$$

where $0 \le d_i(y_i) \le 1$. Since two objectives are considered, the aggregated desirability function becomes:

$$DF = \left(d_{\text{Accuracy}}(x), d_{\text{Intra-list Diversity}}(x) \right), \tag{4.25}$$

which can be rewritten as:

$$DF = \left(d_{\text{Accuracy}} \cdot d_{\text{Intra-list Diversity}} \right)^{\frac{1}{2}}. \tag{4.26}$$

The optimization objective is therefore defined as maximizing DF under the following constraints:

$$\max_{\alpha, \beta} DF \quad \text{s.t.} \quad 0 \le \alpha \le 1,\ 0 \le \beta \le 1. \tag{4.27}$$

Table 4.8 reports the optimal values of α and β that achieve the best compromise between accuracy and intra-list diversity for each dataset. Among the evaluated datasets, MovieLens1M achieves the most balanced trade-off, yielding the highest combined performance across both metrics.

Table 4.8: Optimal trade-off between accuracy and intra-list diversity

Dataset	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$	**Accuracy**	**Intra-list Diversity**
Retail Rocket	0.997	1.000	0.7813	0.4682
Clothing	1.000	0.647	0.8244	0.3778
MovieLens1M	0.976	0.875	0.8616	0.4765

For clearer interpretation, the desirability scores associated with accuracy and intra-list diversity are illustrated in Figure 4.7. The bar charts highlight dataset-specific trade-offs between relevance and diversity. MovieLens1M achieves the highest combined desirability score (0.8885), reflecting its strong balance between high accuracy (0.8972) and substantial intra-list diversity (0.6902).

Retail Rocket exhibits high intra-list diversity (0.7557), attributable to its heterogeneous user behavior, but at the cost of reduced accuracy (0.8445), resulting in the lowest overall desirability score (0.8008). The Clothing dataset demonstrates the highest accuracy (0.9288) combined with competitive diversity (0.6969), yielding a total score (0.8046) comparable to Retail Rocket. These results underline the domain-dependent nature of the accuracy–diversity trade-off.

The overlay contour plots shown in Figure 4.8 provide a visual representation of the feasible regions where both accuracy and intra-list diversity requirements are simultaneously satisfied. The contours correspond to predefined performance thresholds: $0 \leq \text{Accuracy} \leq 0.92$ and $0 \leq \text{ILD} \leq 0.62$ for Retail Rocket, $0 \leq \text{Accuracy} \leq 0.88$ and $0 \leq \text{ILD} \leq 0.59$ for Clothing, and $0 \leq \text{Accuracy} \leq 0.96$ and $0 \leq \text{ILD} \leq 0.69$ for MovieLens1M. The shaded regions indicate the admissible parameter space, while the highlighted points correspond to the optimal (α, β) configurations.

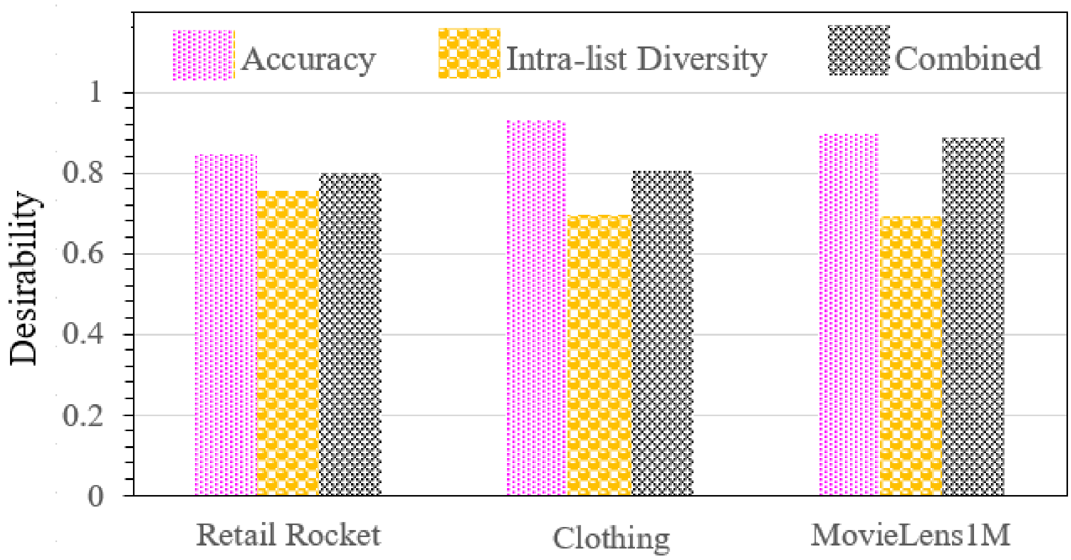

Figure 4.7: Comparative analysis of desirability scores for Retail Rocket, Clothing, and MovieLens1M datasets

Overall, the optimization results confirm that the proposed SFNN framework can effectively adapt its parameters to different datasets, achieving a context-aware balance between recommendation accuracy and diversity. This adaptability is crucial for practical deployment across heterogeneous domains with varying user behavior patterns and recommendation objectives.

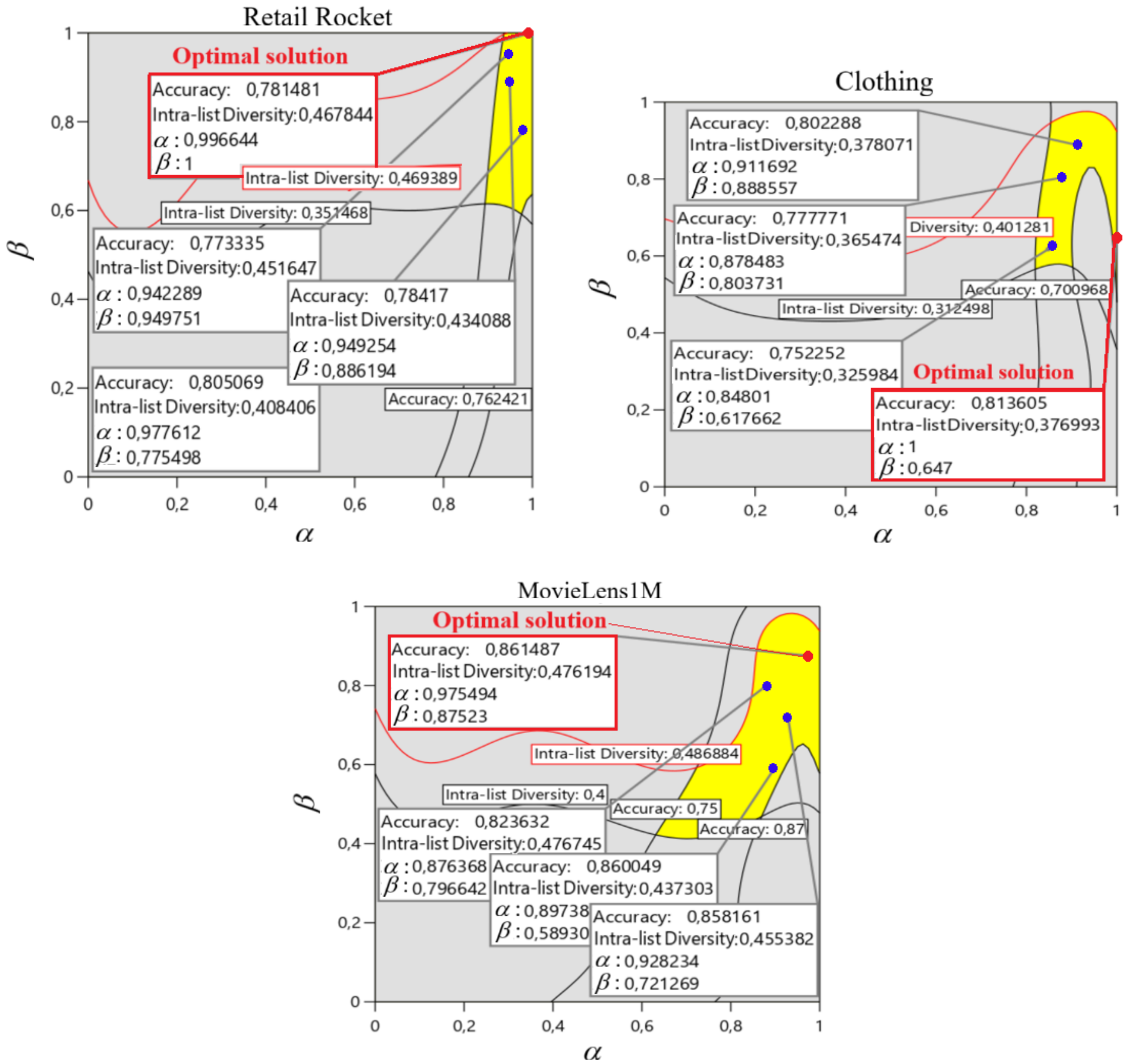

Figure 4.8: Comparative analysis of desirability scores for Retail Rocket, Clothing, and MovieLens1M datasets

4.5.4 Performance of SFNN at Optimal Parameters

Figure 4.9 presents a graphical visualization of the Normalized Discounted Cumulative Gain (NDCG) and Hit Ratio (HR) metrics evaluated at different recommendation cutoffs (Top-$N = 1, 5, 10$) for the three datasets: Retail Rocket, Clothing, and MovieLens1M. These results assess the ranking quality and retrieval effectiveness of the proposed SFNN-based recommender system using the optimal α and β parameters identified in the previous section.

The left panel of Figure 4.9 illustrates the evolution of NDCG scores as the recommendation list size increases. A consistent upward trend is observed across all datasets, indicating improved ranking quality for larger recommendation lists. Specifically, the NDCG@1 values reach 0.218, 0.312, and 0.403 for Retail Rocket, Clothing, and MovieLens1M, respectively, reflecting increasing ranking precision in datasets with more structured and dense user–item interactions. At $N = 10$, the NDCG values further increase to 0.421, 0.498, and 0.612, confirming that SFNN effectively prioritizes relevant items as the recommendation scope expands.

The right panel of Figure 4.9 reports the Hit Ratio results, which measure the system's ability to retrieve at least one relevant item within the top-N recommendations. Similar to NDCG, HR exhibits a monotonic increase with N across all datasets. The HR@1 values are 0.184 for Retail Rocket, 0.263 for Clothing, and 0.321 for MovieLens1M, highlighting the effectiveness of SFNN in accurately identifying the most relevant item for users. At $N = 10$, the Hit Ratio significantly improves, reaching 0.648, 0.722, and 0.801, respectively. These results demonstrate that SFNN maintains strong retrieval performance when generating larger recommendation lists.

Overall, the trends observed in Figure 4.9 confirm that the proposed SFNN model achieves robust ranking quality and retrieval effectiveness across different recommendation cutoffs. The consistent performance gains across all datasets further validate the adaptability of the model to diverse data distributions and user behavior patterns, reinforcing its suitability for real-world large-scale recommendation scenarios.

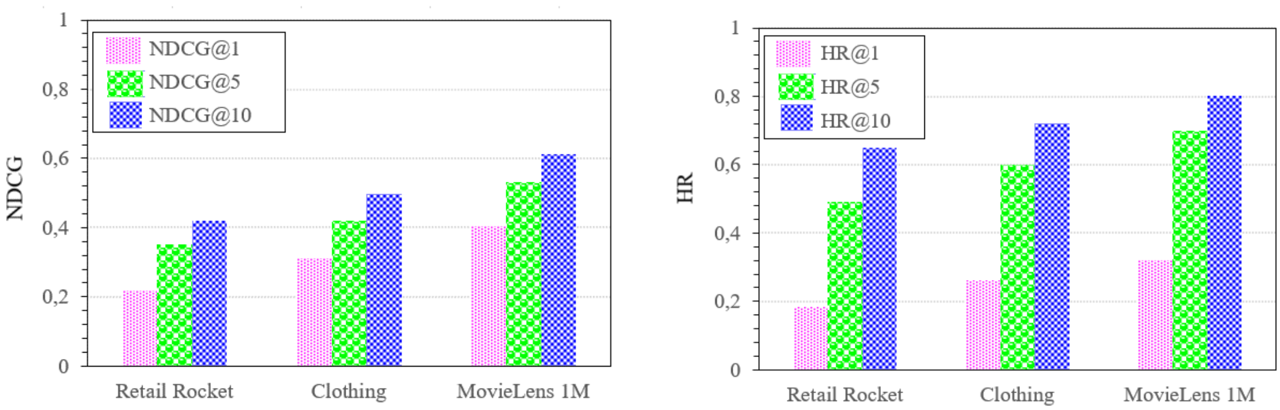

Figure 4.9: Performance comparison of NDCG and Hit Ratio at Top-N levels ($N = 1, 5, 10$) across Retail Rocket, Clothing, and MovieLens1M datasets

4.5.5 Comparative Evaluation with State-of-the-Art Models (RQ5)

Table 4.9 reports the comparative performance of the proposed SFNN model against several state-of-the-art recommendation approaches, including traditional collaborative filtering methods (BPR, NGCF), graph-based models (VGAE), and neural autoencoder-based techniques (MultiVAE and MultiDAE), across three benchmark datasets: Retail Rocket, Clothing, and MovieLens1M. For each dataset, 70% of the interactions were randomly sampled for training and the remaining 30% were used for testing. Each experiment was repeated five times, and the

Table 4.9: Performance comparisons of different recommendation models (The best result is bolded and the runner-up is underlined).

Dataset	Retail Rocket			Clothing			MovieLens-1M		
Method	NDCG@5	HR@5	ILD@5	NDCG@5	HR@5	ILD@5	NDCG@5	HR@5	ILD@5
BPR	0.2653	0.4821	0.3542	0.2541	0.4632	0.3725	0.3826	0.6248	0.3021
NGCF	0.2986	0.5169	0.3715	0.2814	0.4891	0.3948	0.4129	0.6547	0.3256
WGAE	0.3124	0.5298	0.4068	0.2938	0.5012	0.4213	0.4273	0.6681	0.3465
MultiVAE	0.3261	0.5411	0.4327	0.3075	0.5178	0.4399	0.4439	0.6824	0.3684
MultiDAE	0.3215	0.5386	0.4458	0.3027	0.5123	0.4475	0.4395	0.6781	0.3728
SFNN	**0.3852**	**0.5417**	**0.4728**	**0.3429**	**0.6421**	0.4172	**0.4510**	0.6652	**0.4725**
Impro. %	+18.12%	+0.11%	+6.05%	+11.51%	+24%	–6.77%	+1.59%	–2.52%	+26.74%
	NDCG@10	HR@10	ILD@10	NDCG@10	HR@10	ILD@10	NDCG@10	HR@10	ILD@10
BPR	0.2978	0.5447	0.3981	0.2836	0.5128	0.4014	0.4172	0.6813	0.3289
NGCF	0.3362	0.5864	0.4127	0.3146	0.5472	0.4285	0.4471	0.7129	0.3517
WGAE	0.3515	0.6023	0.4375	0.3281	0.5597	0.4498	0.4628	0.7285	0.3659
MultiVAE	0.3678	0.6154	0.4562	0.3446	0.5739	0.4632	0.4816	0.7438	0.3824
MultiDAE	0.3614	0.6112	0.4679	0.3398	0.5682	0.4716	0.4763	0.7392	0.3879
SFNN	**0.4219**	**0.6183**	**0.4768**	**0.3983**	**0.7221**	0.4211	**0.5127**	0.7312	**0.4760**
Impro. %	+14.7%	+0.47%	+1.9%	+15.58%	+25.82%	–10.7%	+6.45%	–1.69%	+22.71%

reported results correspond to averaged values.

As shown in Table 4.9, SFNN consistently achieves superior or competitive performance across both accuracy-oriented metrics (NDCG@K and HR@K) and diversity (intra-list diversity) at Top-5 and Top-10 recommendation cutoffs. Several important observations can be drawn from these results.

First, SFNN dominates most baseline models in terms of ranking quality and relevance. On the Retail Rocket and Clothing datasets, SFNN achieves the highest NDCG@5 and HR@5 scores, confirming its strong ability to correctly rank relevant items at the top of the recommendation list. These gains remain consistent at higher cutoffs (Top-10), demonstrating the robustness of the proposed approach.

Second, Table 4.9 highlights the advantage of SFNN over VAE-based baselines. On the Retail Rocket dataset, SFNN improves NDCG@5 by 18.12% compared to MultiVAE and increases intra-list diversity by 6.05% over MultiDAE. This indicates that the proposed fusion of graph-based learning and diversity-regularized latent modeling not only enhances relevance but also mitigates redundancy in recommended items.

Third, when compared with VGAE, SFNN shows consistent improvements across all datasets. In particular, on Retail Rocket, SFNN achieves a 23.3% improvement in NDCG@5 while simultaneously increasing diversity by 16.2%. These results suggest that SFNN captures richer structural and latent patterns than standard variational graph autoencoders.

Finally, relative to classical collaborative filtering methods such as BPR and NGCF, SFNN significantly outperforms these approaches across all evaluation metrics and datasets. For example, on the Retail Rocket dataset, SFNN surpasses NGCF by approximately 29% in terms of NDCG@5, illustrating the limitations of traditional methods in modeling complex and sparse user–item interactions.

Overall, the results summarized in Table 4.9 confirm that SFNN achieves an effective balance between recommendation accuracy and diversity, outperforming state-of-the-art baselines across heterogeneous datasets. This demonstrates the suitability of the proposed framework for real-world recommendation scenarios where both relevance and diversity are critical.

4.5.6 Limitations

Even though the SFNN model introduces a novel architecture combining GNN, Reg-VAE, and blockchain technology to improve recommendation accuracy, diversity, and data security, several limitations remain. First, while the secure framework ensures encryption and blockchain-based storage of user and product data, providing high confidentiality and integrity, this security mechanism may introduce additional computational overhead in large-scale, real-time environments. Second, the model's reliance on graph-based representations can limit its effectiveness in scenarios with extremely sparse or highly unbalanced datasets. In such cases, the GNN may struggle to capture meaningful user-item interactions, leading to degraded accuracy. Similarly, while the Reg-VAE promotes diversity through controlled randomization and latent space modeling, its performance is sensitive to hyperparameter settings and the quality of the input data. Third, like most deep learning-based recommender systems, SFNN suffers from interpretability challenges. The complexity of GNN embeddings and latent representations in the VAE makes it difficult to provide transparent explanations for individual recommendations, which may be critical in domains requiring explainability, such as healthcare or finance. While SFNN demonstrates strong performance in offline evaluation, its deployment in live production systems with high-frequency user interactions may require additional optimization or approximation techniques to maintain efficiency. Despite these limitations, SFNN provides a robust foundation for accurate, diverse, and secure recommendations, and future work can focus on enhancing interpretability, sparsity robustness, and real-time deployment capabilities.

4.6 Conclusion

This chapter presented the Secure Fusion Neural Networks (SFNN) recommender system, which integrates a Graph Neural Network (GNN) for accuracy and a Regularized Variational Autoencoder (Reg-VAE) with a diversity-promoting regularization term for recommendation diversity. An adapted weighted linear fusion function combines their outputs, with parameters optimized via a desirability function. Blockchain technology ensures strong data security and integrity without compromising efficiency. Experimental results on Retail Rocket, Clothing, and MovieLens1M datasets demonstrate that SFNN achieves a balanced improvement in both accuracy and diversity, outperforming state-of-the-art baselines. Ablation studies confirmed the importance of hybrid integration, showing that combining GNN and Reg-VAE effectively enhances recommendation quality. Future work will focus on improving robustness, interpretability, and performance in sparse, dynamic, and large-scale environments.

General Conclusion & Perspectives

Recommender Systems have undergone substantial evolution in response to the growing complexity of user needs across a wide range of application domains, including electronic commerce, multimedia streaming platforms, social networks, and intelligent assistant systems. While early research primarily focused on maximizing recommendation relevance and accuracy, contemporary recommender systems are now expected to satisfy additional and equally critical requirements such as diversity, novelty, robustness, and the preservation of security and privacy. This thesis fully embraces this multidimensional evolution by addressing two fundamental and complementary challenges: enhancing recommendation diversity through biologically inspired learning paradigms, and embedding security mechanisms directly within the recommendation process to ensure trustworthy and privacy-aware decision making.

The first major contribution of this thesis is the introduction of a novel graph-based recommendation model inspired by the adaptive growth and exploration behavior of natural mycelium networks. The proposed architecture leverages graph neural learning while incorporating two dynamic mechanisms that are applied exclusively during the recommendation generation phase. The first mechanism consists of a dynamic edge reweighting strategy that continuously updates the strength of item-to-item relationships based on observed user interactions, allowing the system to adapt its internal structure in real time. The second mechanism relies on a probabilistic exploration–exploitation controller that dynamically regulates the balance between popular and novel items in the recommendation list. Together, these mechanisms enable the model to overcome the classical trade-off between accuracy and diversity, producing recommendation lists that remain relevant while exhibiting significantly richer content variation.

Extensive experiments conducted on multiple real-world datasets from electronic commerce and media consumption domains demonstrate the effectiveness of this biologically inspired approach. The proposed model consistently achieves higher recall and improved harmonic performance compared to baseline graph-based and neural recommendation models, while showing a marked improvement in intra-list diversity. These results confirm that incorporating adaptive and bio-inspired exploration strategies into graph neural architectures constitutes a powerful paradigm for next-generation recommender systems.

The second major contribution of this thesis addresses a critical yet relatively underexplored dimension of recommender systems: security and data protection. A hybrid neural framework is proposed that integrates recommendation learning with dedicated security-oriented components designed to preserve user privacy and system integrity. By combining graph-based representations with latent-space regularization and adaptive fusion mechanisms, the proposed model ensures that diversity and accuracy objectives are met without exposing sensitive user information. Furthermore, the architecture is designed to remain robust against common threats such as profile injection attacks, inference-based privacy leakage, and malicious data manipulation, even when operating under privacy-sensitive or encrypted data settings.

Experimental evaluation confirms that the proposed secure framework achieves high recommendation accuracy while maintaining strong robustness and privacy-preserving properties. Across multiple datasets, the model demonstrates stable performance, competitive predictive accuracy, and substantial diversity gains, validating its suitability for deployment in real-world environments where trust and data protection are paramount.

Overall, this thesis makes several significant contributions to the recommender systems literature. It establishes diversity as a primary design objective rather than a secondary optimization criterion, introduces adaptive and biologically inspired architectural principles that promote personalized and exploratory recommendation behavior, and provides a concrete foundation for integrating security and privacy protection directly into the recommendation pipeline. Beyond technical advancements, the findings of this work also emphasize the broader societal implications of recommender systems, highlighting the importance of transparency, fairness, and user autonomy. By delivering recommendations that are both diverse and secure, recommender systems can foster fairer content exposure, enhance user trust, and support long-term engagement.

In conclusion, this thesis advocates a holistic rethinking of recommender system design, where accuracy, diversity, and security are not treated as competing objectives but as complementary pillars. Balancing these dimensions is essential for building the next generation of responsible, robust, and ethical recommender technologies capable of meeting the demands of modern digital ecosystems.

Future research may extend this work by exploring federated and decentralized learning settings to further strengthen privacy guarantees in large-scale environments. In addition, integrating explainability mechanisms into the proposed architectures could enhance transparency and user trust. Investigating fairness-aware diversification strategies and evaluating the models in real-time industrial deployment scenarios also represent promising directions. Finally, the incorporation of multimodal signals and large language models may further enrich contextual understanding and adaptive recommendation capabilities.

References

[1] F. Ricci, L. Rokach, B. Shapira, Introduction to Recommender Systems Handbook, Springer, 2011. doi:10.1007/978-0-387-85820-3.

[2] G. Adomavicius, A. Tuzhilin, 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) (2005) 734–749. doi:10.1109/TKDE.2005.99.

[3] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender Systems Survey, Knowledge-Based Systems 46 (2013) 109–132. doi:10.1016/j.knosys.2013.03.012.

[4] C. A. Gómez-Uribe, N. Hunt, The Netflix Recommender System: Algorithms, Business Value, and Innovation, ACM Transactions on Management Information Systems 6 (4) (2015) 13. doi:10.1145/2843948.

[5] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, R. Koper, Recommender Systems in Technology Enhanced Learning, Recommender Systems Handbook (2011) 387–415doi:10.1007/978-0-387-85820-3_12.

[6] M. Wiesner, D. Pfeifer, Health Recommender Systems: Concepts, Requirements, Technical Basics and Challenges, International Journal of Environmental Research and Public Health 11 (3) (2015) 2580–2607. doi:10.3390/ijerph110302580.

[7] M. J. Eppler, J. Mengis, The Concept of Information Overload: A Review of Literature, The Information Society 20 (5) (2004) 325–344. doi:10.1080/01972240490507974.

[8] S. S. Iyengar, M. R. Lepper, When Choice Is Demotivating: Can One Desire Too Much of a Good Thing?, Journal of Personality and Social Psychology 79 (6) (2000) 995–1006. doi:10.1037/0022-3514.79.6.995.

[9] F. Ricci, L. Rokach, B. Shapira, Recommender Systems Handbook, 2nd Edition, Springer, 2015. doi:10.1007/978-1-4899-7637-6.

[10] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2010. doi:10.1017/CBO9780511763113.

[11] C. Anderson, The Long Tail: Why the Future of Business Is Selling Less of More, Hyperion, 2006.

[12] N. Manouselis, C. Costopoulou, Recommender Systems in Learning, Recommender Systems Handbook (2013) 387–415doi:10.1007/978-1-4899-7637-6_12.

[13] J. Beel, B. Gipp, S. Langer, C. Breitinger, Research-paper Recommender Systems: A Literature Survey, International Journal on Digital Libraries 17 (4) (2016) 305–338. doi:10.1007/s00799-015-0156-0.

[14] Y. Zhang, X. Chen, Explainable Recommendation: A Survey and New Perspectives, Foundations and Trends in Information Retrieval 14 (1) (2019) 1–101. doi:10.1561/1500000066.

[15] M. D. Ekstrand, M. Tian, D. Kluver, Exploring Author Gender in Book Rating and Recommendation, Proceedings of the 12th ACM Conference on Recommender Systems (2018) 242–250doi:10.1145/3240323.3240373.

[16] P. Resnick, H. R. Varian, Recommender Systems, Communications of the ACM 40 (3) (1997) 56–58. doi:10.1145/245108.245121.

[17] G. Shani, A. Gunawardana, Evaluating Recommendation Systems, Recommender Systems Handbook (2011) 257–297doi:10.1007/978-0-387-85820-3_8.

[18] R. Burke, Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction 12 (4) (2002) 331–370. doi:10.1023/A:1021240730564.

[19] P. G. Campos, F. Díez, I. Cantador, Time-aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols, User Modeling and User-Adapted Interaction 24 (1) (2014) 67–119. doi:10.1007/s11257-012-9136-x.

[20] P. Cremonesi, Y. Koren, R. Turrin, Performance of Recommender Algorithms on Top-N Recommendation Tasks, Proceedings of the 4th ACM Conference on Recommender Systems (2010) 39–46doi:10.1145/1864708.1864721.

[21] P. Lops, M. de Gemmis, G. Semeraro, Content-based Recommender Systems: State of the Art and Trends, Recommender Systems Handbook (2011) 73–105doi:10.1007/978-0-387-85820-3_3.

[22] Y. Hu, Y. Koren, C. Volinsky, Collaborative Filtering for Implicit Feedback Datasets, Proceedings of the 8th IEEE International Conference on Data Mining (2008) 263–272doi:10.1109/ICDM.2008.22.

[23] Y. Koren, Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model, Proceedings of the 14th ACM SIGKDD Conference (2010) 426–434doi:10.1145/1835804.1835830.

[24] S. Rendle, Factorization Machines, Proceedings of the 10th IEEE International Conference on Data Mining (2009) 995–1000doi:10.1109/ICDM.2010.127.

[25] M. J. Pazzani, A Framework for Collaborative, Content-Based and Demographic Filtering, Artificial Intelligence Review 13 (5) (1999) 393–408. doi:10.1023/A:1006544522159.

[26] S. K. Lam, J. Riedl, Addressing Cold-Start Problems in Recommendation Systems, Proceedings of the 2nd ACM Conference on Recommender Systems (2008) 25–32doi:10.1145/1454008.1454012.

[27] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural Collaborative Filtering, Proceedings of the 26th International World Wide Web Conference (2017) 173–182doi:10.1145/3038912.3052569.

[28] G. Salton, C. Buckley, Term-weighting Approaches in Automatic Text Retrieval, Information Processing and Management 24 (5) (1988) 513–523. doi:10.1016/0306-4573(88)90021-0.

[29] G. Adomavicius, A. Tuzhilin, Context-Aware Recommender Systems, Recommender Systems Handbook (2011) 217–253doi:10.1007/978-0-387-85820-3_7.

[30] H. Ma, I. King, M. R. Lyu, Learning to Recommend with Social Trust Ensemble, Proceedings of the 32nd ACM SIGIR Conference (2011) 203–210doi:10.1145/2009916.2009943.

[31] L. Baltrunas, B. Ludwig, F. Ricci, Context Relevance Assessment and Exploitation in Mobile Recommender Systems, Personal and Ubiquitous Computing 15 (5) (2011) 507–526. doi:10.1007/s00779-010-0347-x.

[32] M. J. Pazzani, D. Billsus, Content-Based Recommendation Systems, The Adaptive Web (2007) 325–341doi:10.1007/978-3-540-72079-9_10.

[33] C.-N. Ziegler, S. M. McNee, J. A. Konstan, G. Lausen, Improving Recommendation Lists Through Topic Diversification, Proceedings of the 14th International World Wide Web Conference (2005) 22–32doi:10.1145/1060745.1060754.

[34] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-Based Collaborative Filtering Recommendation Algorithms, Proceedings of the 10th International World Wide Web Conference (2001) 285–295doi:10.1145/371920.372071.

[35] J. Herlocker, J. Konstan, L. Terveen, J. Riedl, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems 22 (1) (2004) 5–53. doi:10.1145/963770.963772.

[36] Y. Koren, R. Bell, C. Volinsky, Matrix Factorization Techniques for Recommender Systems, Computer 42 (8) (2009) 30–37. doi:10.1109/MC.2009.263.

[37] I. Cantador, I. Fernández-Tobías, S. Berkovsky, Relational Recommender Systems: A Survey, ACM Computing Surveys 49 (2) (2015) 1–41. doi:10.1145/2959104.

[38] D. Jannach, M. Jugovac, Measuring the Business Value of Recommender Systems, ACM Transactions on Management Information Systems 11 (4) (2020) 1–23. doi:10.1145/3386242.

[39] H. Steck, D. Liang, Balancing Accuracy, Diversity, and Novelty in Recommendations, ACM Transactions on Recommender Systems 1 (1) (2021) 1–25. doi:10.1145/3488561.

[40] X. Zhao, L. Zhang, L. Xia, Deep Reinforcement Learning for List-wise Recommendations, Proceedings of the 29th ACM International Conference on Information and Knowledge Management (2020) 249–258doi:10.1145/3340531.3411872.

[41] L. Chen, P. Pu, Trust Building in Recommender Systems, User Modeling and User-Adapted Interaction 30 (5) (2020) 907–947. doi:10.1007/s11257-020-09262-6.

[42] H. Steck, Calibrated Recommendations, Proceedings of the 12th ACM Conference on Recommender Systems (2018) 154–162.

[43] M. Ferrari Dacrema, D. Jannach, Beyond Accuracy: Evaluating Recommender Systems by Coverage, Novelty and Diversity, ACM Computing Surveys 54 (7) (2021) 1–38. doi:10.1145/3457607.

[44] M. Kunaver, T. Požrl, Diversity in Recommender Systems – A Survey, Knowledge-Based Systems 123 (2020) 1–15. doi:10.1016/j.knosys.2017.02.009.

[45] S. Vargas, P. Castells, Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems, Information Retrieval Journal 23 (4) (2020) 1–28. doi:10.1007/s10791-020-09372-5.

[46] Y. Ge, S. Zhao, J. Zhou, Understanding and Mitigating Popularity Bias in Recommendation, Proceedings of the 44th International ACM SIGIR Conference (2021) 137–146doi:10.1145/3404835.3462861.

[47] D. Kotkov, S. Wang, J. Veijalainen, A Survey of Serendipity in Recommender Systems, Knowledge-Based Systems 193 (2020) 105454. doi:10.1016/j.knosys.2020.105454.

[48] K. Christakopoulou, A. Banerjee, Adversarial Attacks on Recommender Systems, Proceedings of the 13th ACM International Conference on Web Search and Data Mining (2020) 112–120doi:10.1145/3336191.3371841.

[49] H. Steck, Evaluation of Recommendation Systems, ACM Transactions on Information Systems 39 (4) (2021) 1–38. doi:10.1145/3468507.

[50] C. A. Gómez-Uribe, N. Hunt, The Netflix Recommender System: Algorithms, Business Value, and Innovation, ACM Transactions on Management Information Systems 6 (4) (2015) 13. doi:10.1145/2843948.

[51] J. Li, Y. Zhang, L. Chen, E-commerce Recommender Systems: Review and Recent Advances, ACM Computing Surveys 54 (3) (2021) 1–40. doi:10.1145/3448082.

[52] P. Covington, J. Adams, E. Sargin, Deep Neural Networks for YouTube Recommendations, Proceedings of the 10th ACM Conference on Recommender Systems (2016) 191–198doi:10.1145/2959100.2959190.

[53] X. Amatriain, Recommender Systems in Industry: A Netflix Case Study, ACM Transactions on Interactive Intelligent Systems 10 (4) (2020) 1–25. doi:10.1145/3386342.

[54] M. Xia, X. Li, Z. He, A Survey on Recommender Systems for Education, IEEE Transactions on Learning Technologies 14 (3) (2021) 276–292. doi:10.1109/TLT.2020.3016310.

[55] X. Bai, X. He, Adaptive Learning Systems with Personalized Recommendations, Computers Education 148 (2020) 103797. doi:10.1016/j.compedu.2020.103797.

[56] R. Li, X. Zhu, X. Sun, Healthcare Recommender Systems: Review and Opportunities, Journal of Biomedical Informatics 112 (2020) 103611. doi:10.1016/j.jbi.2020.103611.

[57] N. Shah, A. Kaur, Applications of Recommender Systems in Healthcare, Artificial Intelligence in Medicine 116 (2021) 102078. doi:10.1016/j.artmed.2021.102078.

[58] N. Hariri, B. Mobasher, R. Burke, Context-Aware and Personalized News Recommender Systems, ACM Transactions on Information Systems 37 (3) (2019) 1–37. doi:10.1145/3323076.

[59] F. Ricci, I. Cantador, Applications of Recommender Systems: Beyond E-Commerce and Media, User Modeling and User-Adapted Interaction 30 (2) (2020) 287–324. doi:10.1007/s11257-020-09244-6.

[60] G. Shani, A. Gunawardana, Evaluating Recommendation Systems, Recommender Systems Handbook (2011) 257–297doi:10.1007/978-0-387-85820-3_8.

[61] M. Xia, X. He, W. Li, Metrics for Evaluating Recommender Systems: Accuracy, Diversity, and Novelty, ACM Computing Surveys 54 (6) (2021) 1–37. doi:10.1145/3468571.

[62] Y. Zhang, L. Chen, et al., A comprehensive review of recommender systems, arXiv preprint (2024).
URL https://arxiv.org/html/2407.13699v4

[63] O. A. S. Ibrahim, E. M. G. Younis, et al., Revisiting recommender systems: An investigative survey, International Journal of Recommender ResearchArXiv:2310.?? (2025).
URL https://d-nb.info/1363662252/

[64] Y. An, Y. Tan, X. Sun, G. Ferrari, Recommender system: A comprehensive overview of technical challenges and social implications, ICCK Transactions on Sensing, Communication, and Control 1 (1) (2024) 30–51. doi:10.62762/TSCC.2024.898503.

[65] Y. Zhao, et al., Fairness and diversity in recommender systems: A survey, ACM Computing Surveys (2025). doi:10.1145/3664928.

[66] F. Carnovalini, Popularity bias in recommender systems: The search for fairness in the long tail, Information 16 (2) (2025) 151. doi:10.3390/info16020151.

[67] Y. Zhang, X. Chen, Explainable recommendation: A survey and new perspectives, arXiv preprint (2018).
URL https://arxiv.org/abs/1804.11192

[68] M. Dong, F. Yuan, et al., Survey for trust-aware recommender systems: A deep learning perspective, arXiv preprint (2020).
URL https://arxiv.org/abs/2004.03774

[69] D. Kowald, Transparency, privacy, and fairness in recommender systems, arXiv preprint (2024).
URL https://arxiv.org/abs/2406.11323

[70] D. Jin, et al., A survey on fairness-aware recommender systems, Information Fusion 93 (2023) 101906. doi:10.1016/j.inffus.2023.101906.

[71] L.-W. Pan, W.-K. Pan, M.-Y. Wei, H.-Z. Yin, Z. Ming, A survey on sequential recommendation, Frontiers of Computer Science 20 (3) (2026) 2003606. doi:10.1007/s11704-025-41329-w.

[72] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, P. Jiang, Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, in: Proceedings of the 2019 Conference on Recommender Systems, 2019.
URL https://arxiv.org/abs/1904.06690

[73] V.-A. Tran, G. Salha-Galvan, B. Sguerra, R. Hennequin, Attention mixtures for time-aware sequential recommendation, arXiv preprint (2023).
URL https://arxiv.org/abs/2304.08158

[74] Z. Sun, Y. Xu, W. Liu, F. Wu, et al., A survey on federated recommendation systems, arXiv preprint (2022).
URL https://arxiv.org/abs/2301.00767

[75] Y. Mi, J. Shen, G. Zhao, J. Shen, X. Qian, A scenario-oriented survey of federated recommender systems: Techniques, challenges, and future directions, arXiv preprint (2025).
URL https://arxiv.org/abs/2508.19620

[76] S. Wang, H. Tao, J. Li, et al., Towards fair and personalized federated recommendation, Pattern Recognition 149 (2024) 110234. doi:10.1016/j.patcog.2023.110234.

[77] P. Wei, et al., Sequential recommendation system based on deep learning: Trends and methodologies, Electronics (2025). doi:10.3390/electronics14112134.

[78] F. P. Bianchi, et al., Ant colony optimization in recommender systems: A survey of algorithms and applications, Swarm and Evolutionary Computation 68 (2021) 101000. doi:10.1016/j.swevo.2021.101000.

[79] B. Patel, P. Desai, U. Panchal, Methods of recommender system: A review, in: 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE, 2017, pp. 1–4. doi:10.1109/ICIIECS.2017.8275856.

[80] G. Parthasarathy, S. Sathiya Devi, Hybrid recommendation system based on collaborative and content-based filtering, Cybernetics and Systems 54 (4) (2023) 432–453. doi:10.1080/01969722.2022.2062544.

[81] J. Lin, M. He, W. Pan, Z. Ming, Collaborative filtering with sequential implicit feedback via learning users' preferences over item-sets, Information Sciences 621 (2023) 136–155. doi:10.1016/j.ins.2022.11.064.

[82] N. Alharbe, M. A. Rakrouki, A. Aljohani, A collaborative filtering recommendation algorithm based on embedding representation, Expert Systems with Applications 215 (2023) 119380. doi:10.1016/j.eswa.2022.119380.

[83] F. Colace, D. Conte, M. De Santo, M. Lombardi, D. Santaniello, C. Valentino, A content-based recommendation approach based on singular value decomposition, Connection Science 34 (1) (2022) 2158–2176. doi:10.1080/09540091.2022.2106943.

[84] I. Gasmi, H. Seridi-Bouchelaghem, L. Hocine, B. Abdelkarim, Collaborative filtering recommendation based on dynamic changes of user interest, Intelligent Decision Technologies 9 (3) (2015) 271–281. doi:10.3233/IDT-140221.

[85] L. Chen, G. Chen, F. Wang, Recommender systems based on user reviews: the state of the art, User Modeling and User-Adapted Interaction 25 (2015) 99–154. doi:10.1007/s11257-015-9155-5.

[86] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, in: Recommender systems handbook, 2011, pp. 73–105. doi:10.1007/978-0-387-85820-3_3.

[87] A. L. Karn, R. K. Karna, B. R. Kondamudi, G. Bagale, D. A. Pustokhin, I. V. Pustokhina, S. Sengan, RETRACTED ARTICLE: Customer centric hybrid recommendation system for E-Commerce applications by integrating hybrid sentiment analysis, Electronic Commerce Research 23 (1) (2023) 279–314. doi:10.1007/s10660-022-09630-z.

[88] Z. Wang, Y. Qin, Y. Shi, M. Jiang, W. Wang, Research on e-commerce recommendation system based on matrix factorization algorithm, Concurrency and Computation: Practice and Experience 35 (22) (2023) e7739. doi:10.1002/cpe.7739.

[89] B. Alhijawi, S. Fraihat, A. Awajan, Multi-factor ranking method for trading-off accuracy, diversity, novelty, and coverage of recommender systems, International Journal of Information Technology 15 (3) (2023) 1427–1433. doi:10.1007/s41870-023-01158-1.

[90] K. V. Rodpysh, S. J. Mirabedini, T. Banirostam, Employing singular value decomposition and similarity criteria for alleviating cold start and sparse data in context-aware recommender systems, Electronic Commerce Research 23 (2) (2023) 681–707. doi:10.1007/s10660-021-09488-7.

[91] S. S. Lakshmi, T. AdiLakshmi, B. Abhinith, Timestamp Prediction Using Enhanced Adaptive Correlation Clustering-Based Recommender System for the Long Tail Items, in: International Conference on Communications and Cyber Physical Engineering 2018, Springer Nature Singapore, 2023, pp. 991–999. doi:10.1007/978-981-19-8086-2_93.

[92] M. Banerjee, J. Das, S. Majumder, Time Window Based Recommender System for Movies, in: Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences: PCCDS 2022, Springer Nature Singapore, 2023, pp. 381–393. doi:10.1007/978-981-19-8742-7_32.

[93] M. Jozani, C. Z. Liu, K.-K. R. Choo, An empirical study of content-based recommendation systems in mobile app markets, Decision Support Systems 169 (2023) 113954.

[94] L. Hou, Y. Huang, Optimizing the connectedness of recommendation networks for retrieval accuracy and visiting diversity of random walks, Physica A: Statistical Mechanics and its Applications 637 (2024) 129604.

[95] M. Umar, N. Mirza, S. Ribeiro-Navarrete, The impact of financial restatements on sell-side recommendation accuracy, Finance Research Letters 55 (2023) 103868.

[96] D. Pramod, Privacy-preserving techniques in recommender systems: state-of-the-art review and future research agenda, Data Technologies and Applications[Online]. Available: emerald.com (2023).

[97] S. Wang, P. Zhang, H. Wang, H. Yu, F. Zhang, Detecting shilling groups in online recommender systems based on graph convolutional network, Information Processing and Management 59 (2022) 103031. doi:10.1016/j.ipm.2022.103031.

[98] B. Alhijawi, M. Abo Alrub, M. Al-Fayoumi, Generalized Ethereum Blockchain-based recommender system framework, Information Systems 111 (2023) 102113. doi:10.1016/j.is.2022.102113.

[99] L. Mekouar, Y. Iraqi, I. Damaj, T. Naous, A survey on blockchain-based Recommender Systems: Integration architecture and taxonomy, Computer Communications 187 (2022) 1–19. doi:10.1016/j.comcom.2022.01.020.

[100] N. Hazrati, F. Ricci, Recommender systems effect on the evolution of users' choices distribution, Information Processing and Management 59 (2022) 102766. doi:10.1016/j.ipm.2021.102766.

[101] D. Peng, Y. Zhou, A long-tail alleviation post-processing framework based on personalized diversity of session recommendation, Expert Systems with Applications 249 (2024) 123769. doi:10.1016/j.eswa.2024.123769.

[102] Z. Li, Y. Xie, W. E. Zhang, P. Wang, L. Zou, F. Li, X. Luo, C. Li, Disentangle interest trend and diversity for sequential recommendation, Information Processing & Management 61 (3) (2024) 103619. doi:10.1016/j.ipm.2023.103619.

[103] H. Kou, J. Xu, L. Qi, Diversity-driven automated web API recommendation based on implicit requirements, Applied Soft Computing 136 (2023) 110137. doi:10.1016/j.asoc.2023.110137.

[104] A. De Biasio, M. Monaro, L. Oneto, L. Ballan, N. Navarin, On the problem of recommendation for sensitive users and influential items: Simultaneously maintaining interest and diversity, Knowledge-Based Systems 275 (2023) 110699.

[105] Y. Zhao, Y. Wang, Y. Liu, X. Cheng, C. C. Aggarwal, T. Derr, Fairness and diversity in recommender systems: a survey, ACM Transactions on Intelligent Systems and Technology (2023). doi:10.1145/3664928.

[106] A. Gyrard, A. Sheth, IAMHAPPY: Towards an IoT knowledge-based cross-domain well-being recommendation system for everyday happiness, Smart Health 15 (2020) 100083. doi:10.1016/j.smhl.2019.100083.

[107] L. Yang, S. Wang, Y. Tao, J. Sun, X. Liu, P. S. Yu, T. Wang, Dgrec: Graph neural network for recommendation with diversified embedding generation, in: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 2023, pp. 661–669. doi:10.1145/3539597.3570472.

[108] Y. Xu, H. Chen, Z. Wang, J. Yin, Q. Shen, D. Wang, X. Hu, Multi-factor sequential re-ranking with perception-aware diversification, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 5327–5337. doi:10.1145/3580305.3599869.

[109] N. Hazrati, F. Ricci, Recommender systems effect on the evolution of users' choices distribution, Information Processing & Management 59 (1) (2022) 102766. doi:10.1016/j.ipm.2021.102766.

[110] C. E. Berbague, N. E. I. Karabadji, H. Seridi, P. Symeonidis, Y. Manolopoulos, W. Dhifli, An overlapping clustering approach for precision, diversity and novelty-aware recommendations, Expert Systems with Applications 177 (2021) 114917. doi:10.1016/j.eswa.2021.114917.

[111] R. Sethi, M. Mehrotra, Cold start in recommender systems—a survey from domain perspective, in: Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020, Springer Singapore, 2021, pp. 223–232. doi:10.1007/978-981-15-9509-7_19.

[112] L. Peška, T. M. Tashu, T. Horváth, Swarm intelligence techniques in recommender systems-A review of recent research, Swarm and Evolutionary Computation 48 (2019) 201–219. doi:10.1016/j.swevo.2019.04.003.

[113] O. Stitini, S. Kaloun, O. Bencharef, An improved recommender system solution to mitigate the over-specialization problem using genetic algorithms, Electronics 11 (2) (2022) 242. doi:10.3390/electronics11020242.

[114] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Computational Intelligence Magazine 1 (4) (2006) 28–39. doi:10.1109/MCI.2006.329691.

[115] A. Jain, A. Singhal, Bio-inspired Approach for Early Diabetes Prediction and Diet Recommendation, SN Computer Science 5 (1) (2024) 182. doi:10.1007/s42979-023-02481-x.

[116] D. Kilitcioglu, N. Greenquist, A. Bari, Pyrorank: A Novel Nature-Inspired Algorithm to Promote Diversity in Recommender Systems, in: International Conference on Swarm Intelligence, Springer, Cham, 2023, pp. 139–155. doi:10.1007/978-3-031-36625-3_12.

[117] R. Logesh, V. Subramaniyaswamy, D. Malathi, N. Sivaramakrishnan, V. Vijayakumar, Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method, Neural Computing and Applications 32 (2020) 2141–2164. doi:10.1007/s00521-018-3891-5.

[118] C. R. da Costa Bento, E. C. G. Wille, Bio-inspired routing algorithm for MANETs based on fungi networks, Ad Hoc Networks 107 (2020) 102248. doi:10.1016/j.adhoc.2020.102248.

[119] K. K. Alaneme, J. U. Anaele, T. M. Oke, S. A. Kareem, M. Adediran, O. A. Ajibuwa, Y. O. Anabaranze, Mycelium based composites: A review of their bio-fabrication procedures, material properties and potential for green building and construction applications, Alexandria Engineering Journal 83 (2023) 234–250. doi:10.1016/j.aej.2023.10.012.

[120] O. Abdelhady, E. Spyridonos, H. Dahy, Bio-modules: Mycelium-based composites forming a modular interlocking system through a computational design towards sustainable architecture, Designs 7 (1) (2023) 20. doi:10.3390/designs7010020.

[121] A. K. Awasthi, A. K. Garov, M. Sharma, M. Sinha, GNN model based on node classification forecasting in social network, in: 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), IEEE, 2023, pp. 1039–1043. doi:10.1109/AISC56616.2023.10085118.

[122] S. Chen, A. Wulamu, Q. Zou, H. Zheng, L. Wen, X. Guo, Y. Zhang, MD-GNN: A mechanism-data-driven graph neural network for molecular properties prediction and new material discovery, Journal of Molecular Graphics and Modelling 123 (2023) 108506. doi:10.1016/j.jmgm.2023.108506.

[123] D. Jana, S. Malama, S. Narasimhan, E. Taciroglu, Edge Ranking of Graphs in Transportation Networks using a Graph Neural Network (GNN), arXiv preprint arXiv:2303.17485 (2023). doi:10.48550/arXiv.2303.17485.

[124] G. Xv, C. Lin, W. Guan, J. Gou, X. Li, H. Deng, B. Zheng, E-commerce Search via Content Collaborative Graph Neural Network, in: Proceedings of the 29th ACM

SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 2885–2897. doi:10.1145/3580305.3599320.

[125] D. Jin, L. Wang, Y. Zheng, G. Song, F. Jiang, X. Li, S. Pan, Dual intent enhanced graph neural network for session-based new item recommendation, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 684–693. doi:10.1145/3543507.3583526.

[126] H. Zhang, B. Wu, X. Yuan, S. Pan, H. Tong, J. Pei, Trustworthy graph neural networks: aspects, methods, and trends, Proceedings of the IEEE (2024). doi:10.1109/JPROC.2024.3369017.

[127] S. Wu, F. Sun, W. Zhang, X. Xie, B. Cui, Graph neural networks in recommender systems: a survey, ACM Computing Surveys 55 (5) (2022) 1–37. doi:10.1145/3535101.

[128] L. Xia, C. Huang, Y. Xu, J. Pei, Multi-behavior sequential recommendation with temporal graph transformer, IEEE Transactions on Knowledge and Data Engineering 35 (6) (2022) 6099–6112. doi:10.1109/TKDE.2022.3175094.

[129] W. Liu, Y. Zhang, J. Wang, Y. He, J. Caverlee, P. P. Chan, D. S. Yeung, P. A. Heng, Item relationship graph neural networks for e-commerce, IEEE Transactions on Neural Networks and Learning Systems 33 (9) (2021) 4785–4799. doi:10.1109/TNNLS.2021.3060872.

[130] H. Shi, J. Guo, Y. Ke, K. Wang, S. Yang, F. Qin, L. Chen, Personalized Image Aesthetics Assessment based on Graph Neural Network and Collaborative Filtering, Knowledge-Based Systems 294 (2024) 111749.

[131] K. Zhao, Y. Zheng, T. Zhuang, X. Li, X. Zeng, Joint Learning of E-commerce Search and Recommendation with a Unified Graph Neural Network, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 1461–1469. doi:10.1145/3488560.3498414.

[132] A. S. Rajasekaran, M. Azees, F. Al-Turjman, A comprehensive survey on blockchain technology, Sustainable Energy Technologies and Assessments 52 (2022) 102039. doi:10.1016/j.seta.2022.102039.

[133] S. Ying, Shared MF: A privacy-preserving recommendation system, arXiv preprint arXiv:2008.07759 (2020). doi:10.48550/arXiv.2008.07759.

[134] C. Villarán, M. Beltrán, User-centric privacy for identity federations based on a recommendation system, Electronics 11 (8) (2022) 1238. doi:10.3390/electronics11081238.

[135] Y. Himeur, A. Sayed, A. Alsalemi, F. Bensaali, A. Amira, I. Varlamis, G. Dimitrakopoulos, Blockchain-based recommender systems: Applications, challenges and future opportunities, Computer Science Review 43 (2022) 100439. doi:10.1016/j.cosrev.2021.100439.

[136] M. Javadi, M. Mazrooie, A. Bohlool, Improving the performance of recommender systems based on blockchain technology, Journal of Computer Science and Technology Studies 7 (7) (2025) 431–448.

[137] F. Sun, P. Wang, Y. Zhang, P. Kar, βfscm: An enhanced food supply chain management system using hybrid blockchain and recommender systems, Blockchain: Research and Applications 6 (1) (2025) 100245.

[138] L.-E. Wang, D.-C. Li, P. Liu, X.-X. Li, Bam_crs: Blockchain-based anonymous model for cross-domain recommendation systems, Journal of Computer Science and Technology 40 (2025) 340–358.

[139] S. Kumar, S. Pandey, U. Bhatt, On enhancing e-commerce shipping policies with blockchain and recommender systems, SN Computer Science 6 (2025) 145.

[140] A. Bahi, I. Gasmi, S. Bentrad, R. Khantouchi, MycGNN: enhancing recommendation diversity in e-commerce through mycelium-inspired graph neural network, Electronic Commerce Research 1 (2024) 1–31. doi:10.1007/s10660-024-09911-9.

[141] A. Bahi, I. Gasmi, S. Bentrad, M. W. Azizi, R. Khantouchi, M. Uzun-Per, Sfnn: A secure and diverse recommender system through graph neural network and regularized variational autoencoder, Knowledge-Based Systems 332 (2025) 114983. doi:10.1016/j.knosys.2025.114983.
URL https://doi.org/10.1016/j.knosys.2025.114983