

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**ENRICHING ITEM FEATURE REPRESENTATIONS IN SESSION-BASED
RECOMMENDATION WITH GLOBAL ITEM GRAPHS**



M.Sc. THESIS

Yunus KARATEPE

Department of Computer Engineering

Computer Engineering Programme

JULY 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**ENRICHING ITEM FEATURE REPRESENTATIONS IN SESSION-BASED
RECOMMENDATION WITH GLOBAL ITEM GRAPHS**

M.Sc. THESIS

**Yunus KARATEPE
(504221536)**

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

JULY 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**OTURUM TABANLI ÖNERİ SİSTEMLERİNDE EVRENSEL VARLIK ÇİZGESİ
İLE VARLIK ÖZELLİK VEKTÖRLERİNİN ZENGİNLEŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

**Yunus KARATEPE
(504221536)**

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ

TEMMUZ 2025

Yunus KARATEPE, a M.Sc. student of ITU Graduate School student ID 504221536 successfully defended the thesis entitled “ENRICHING ITEM FEATURE REPRESENTATIONS IN SESSION-BASED RECOMMENDATION WITH GLOBAL ITEM GRAPHS”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Şule GÜNDÜZ ÖĞÜDÜCÜ**
Istanbul Technical University

Jury Members : **Assoc. Prof. Dr. Yusuf YASLAN**
Istanbul Technical University

Assoc. Prof. Dr. Günce Keziban ORMAN
Galatasaray University

Date of Submission : **29 May 2025**

Date of Defense : **1 July 2025**





To my family and friends,



FOREWORD

Firstly, I would like to express my deepest gratitude to my advisor, Prof. Dr. Şule Gündüz Öğüdücü, for her invaluable guidance, unwavering support, and encouragement throughout my graduate studies. Her mentorship has played a crucial role in shaping both my academic development and professional outlook, and I will always take pride in having had the opportunity to work under her supervision.

I am especially thankful to my parents for their endless love, sacrifices, and belief in me. Their support has been the foundation of everything I have accomplished in my life. I also extend my heartfelt thanks to my brother, whose constant encouragement and presence have always motivated me to move forward.

My sincere thanks go to my friends who stood by me with their support and friendship throughout this journey. Their encouragement and shared moments helped me stay balanced and inspired. I would particularly like to express my appreciation to my dear friend Enes Tezcan, with whom I embarked on this graduate journey. From our undergraduate days through our master's studies, we have continuously supported and challenged each other, and his companionship has made this journey both easier and more meaningful. Finally, I am grateful to everyone who contributed to this process, directly or indirectly. This thesis would not have been possible without their presence in my life.

July 2025

Yunus KARATEPE

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xix
SUMMARY	xxi
ÖZET	xxv
1. INTRODUCTION	1
1.1 Purpose of Thesis	5
1.2 Contribution of Thesis	6
1.3 Structure of Thesis	7
2. RELATED WORKS	9
2.1 Traditional Recommender Systems	9
2.2 Session-Based Recommender Systems	11
2.3 Graph Related Methods in SBRS	12
3. ENRICHING ITEM REPRESENTATIONS IN SESSION-BASED RECOMMENDATION WITH GRAPH AGGREGATION	17
3.1 Preprocessing	17
3.1.1 Train-test split	17
3.1.2 Session slicing	18
3.1.3 Handling short sessions	18
3.2 Categorical Feature Representation	19
3.3 Global Graph Creation	19
3.4 Neighbor Aggregation On Item Features	21
3.5 Model Training	23
3.6 Session Embeddings	23
3.7 LSTM Model	24
3.8 SR-GNN Model	26
3.9 Implementation Details and Hardware	26
4. EXPERIMENTAL RESULTS	31
4.1 Dataset	31
4.2 Experiment Environment	32
4.3 Metrics and Results	34
5. CONCLUSION	41
REFERENCES	43
CURRICULUM VITAE	49



ABBREVIATIONS

CF	: Collaborative Filtering
CBF	: Content-Based Filtering
GNN	: Graph Neural Network
GCN	: Graph Convolutional Network
SBRs	: Session-Based Recommender Systems
SR-GNN	: Session-based Recommendation with Graph Neural Network
GCE-GNN	: Global Context Enhanced Graph Neural Network
LSTM	: Long-Short Term Memory
GRU	: Gated Recurrent Unit
IMDb	: Internet Movie Database
MDP	: Markov Decision Processes
RNN	: Recurrent Neural Networks
STAMP	: Short-Term Attention/Memory Priority
DAT-MDI	: Dual Attention Transfer Based on Multi-Dimensional Integration
ReGNN	: Repeat-aware Graph Neural Network
CM-GNN	: Contrastive Multi-level Graph Neural Network
BCS	: Broadly Connected Session
FGNN	: Full Graph Neural Network
CPU	: Central Processing Unit
GPU	: Graphics Processing Unit
RAM	: Random Access Memory
VRAM	: Video Random Access Memory
MRR	: Mean Reciprocal Ranking
ID	: Identification



SYMBOLS

$\mathcal{S}$	: Set of Sessions
s	: Session
s_i	: Item at index i in session s
L	: Predefined maximum session length
G	: Global item graph
V	: Nodes in the global item graph
E	: Edges in the global item graph
$E[i, j]$	: Edge between item i and item j
W	: Weights of the edges in the global item graph
W_{in}	: Incoming edge weights of the edges in the global item graph
$w_{i,j}$	: Edge weights between item i and j in the global item graph
$N(i)$	: Neighbors of item i
k	: Number of top-weighted neighbors used for item feature aggregation
$w_{i,j}$	: Weight of an edge from item i to item j
$ncoef$	: Neighbor Contribution Coefficient
n	: Minimum number of cold items in a cold session
K	: Minimum number of appearances for an item to be cold
$\mathbb{I}[\cdot]$	: Indicator function
$\forall(u, v)$	: For all (u, v) pairs



LIST OF TABLES

	<u>Page</u>
Table 4.1: Graph statistics over different training sets.	34
Table 4.2: Information about train, clipped train, validation, and test datasets. ..	34
Table 4.3: Overall results for LSTM and SR-GNN models.....	35
Table 4.4: Performance for $K = 5$, $n = 1$ and $n = 2$, model: SR-GNN.	36
Table 4.5: Performance for $K = 10$, $n = 1$ and $n = 2$, model: SR-GNN.	36
Table 4.6: Performance for $K = 50$, $n = 1$ and $n = 2$, model: SR-GNN.	36
Table 4.7: Number of cold items and cold sessions under different cold-start conditions depending on values of K and n	37



LIST OF FIGURES

	<u>Page</u>
Figure 3.1: An example scenario for creating bag-of-words representations from one-hot vectors.	20
Figure 3.2: Enriching item features with Normalized Weighted Neighbor Aggregation.	23
Figure 3.3: LSTM model architecture.	24
Figure 3.4: Creating session embeddings.	25
Figure 3.5: SR-GNN model architecture.	27
Figure 4.1: An example session and a purchase from the Dressipi dataset.	32
Figure 4.2: An example item with its defined features.	33
Figure 4.3: Overall MRR@20 and MRR@100 results of the SR-GNN model for <code>ncoef</code> values ranging from 0 to 4.	38
Figure 4.4: MRR@20 and MRR@100 results of the LSTM model for <code>ncoef</code> values between 0 and 1, with $K = 50$ and $n = 1$	38
Figure 4.5: MRR@20 and MRR@100 results of the LSTM model for <code>ncoef</code> values between 0 and 1, with $K = 50$ and $n = 2$	39



ENRICHING ITEM FEATURE REPRESENTATIONS IN SESSION-BASED RECOMMENDATION WITH GLOBAL ITEM GRAPHS

SUMMARY

With the advancements in artificial intelligence (AI), many industries that rely on human interaction have begun to utilize these powerful AI models. E-commerce stands out as one of the industries where modeling customer behavior is crucial. In e-commerce, it is very important to recommend items that will attract customers' attention and result in their purchase. A user's shopping experience is significantly influenced by how efficiently they can discover items that align with their preferences within a given time period.

There may be different types of shopping behaviors that can create the need for different types of recommendation models. Hence, there are several types of recommender systems: Collaborative Filtering, Content-Based Filtering, and Hybrid Systems that utilize both approaches. Collaborative filtering methods predict users' interests by analyzing preferences from multiple users. It operates under the assumption that users who have liked similar things will continue to like similar things in the future. Content-based filtering methods, however, analyze items and item features, then match them to the user's profile and make recommendations. It builds a profile for each user based on the features of items they have interacted with and suggests new items with similar characteristics. In addition, various types of recommender systems have been developed to accommodate different data characteristics and application requirements.

Session-Based Recommender Systems (SBRS) represent one of the most popular subtypes of recommender systems, focusing specifically on a short period of user interactions to generate recommendations. Unlike traditional methods such as collaborative and content-based filtering, which primarily rely on long-term historical user and item data, SBRS are designed to capture short-term, dynamic user preferences that emerge during a specific session. This approach proves especially beneficial in scenarios where historical user data is lacking or where user preferences are rapidly shifting. By focusing solely on the sequence and context of user actions in a session, SBRS can provide more relevant recommendations, enhancing the overall user experience.

SBRS are particularly prevalent in e-commerce platforms, where users often interact with items over short periods. In today's fast-paced digital environment, user interests can change rapidly, and the relevance of historical data may diminish quickly. For instance, consider a user who logs into an e-commerce application and purchases a mobile phone after viewing several related items. Post-purchase, the user's interest in mobile phones may wane, rendering past interaction data less effective for future

recommendations. Traditional recommendation methods, which rely heavily on long-term user profiles, might not adapt swiftly to such shifts in user intent. In contrast, SBRs focus on the user’s current session, capturing immediate preferences and providing timely, context-aware recommendations that align with the user’s interests.

Although SBRs appear to be the best system for modeling short-term user interests, they have their limitations, including the cold start problem and limited item diversity. The cold-start problem arises when a model lacks adequate historical data, limiting its ability to learn effectively and generate accurate recommendations. Additionally, limited item diversity often emerges as models tend to favor popular items, resulting in repetitive recommendations and a lack of variety. To deal with these problems, graph-based recommender systems or SBRs that leverage graph structures have been introduced.

In this study, we introduce a novel approach to enhance SBRs by enriching item representations through graph-based aggregation techniques and storing this information in the item’s feature vector. The core idea is to capture and embed item-item relationships directly into the feature vectors of individual items, thereby incorporating contextual information derived from their interactions within the item network. By constructing an item graph where nodes represent items and edges represent relationships, we apply graph aggregation methods to enrich items’ features for each item. This process enables each item to incorporate information coming from its neighborhood. Such enriched representations are particularly beneficial for items with limited interaction data, commonly referred to as "cold items.". Integrating these enriched item feature vectors into the SBRs aims to improve recommendation accuracy for tested models: Session-based Recommendation with Graph Neural Networks (SR-GNN) and Long-Short Term Memory (LSTM), especially in sessions involving cold items. By providing a more comprehensive understanding of each item’s context within the broader item network, the system can generate more relevant and personalized recommendations, enhancing the overall user experience.

Experiments are conducted on the Dressipi 2022 Dataset, which is mainly used in the past for the RecSys2022 Challenge. Experimental evaluations show that the proposed method is improving overall recommendation accuracy for both SR-GNN and LSTM models. Beyond overall recommendation accuracy, the proposed feature enrichment technique significantly improves the performance of SR-GNN on cold sessions (sessions that contain cold-unpopular items).

Looking ahead, there are several promising directions to further explore the impact of our feature enrichment approach. One area involves applying this method across various recommendation models, such as collaborative filtering, content-based filtering, and hybrid systems. Each model type may uniquely benefit from enriched item representations, potentially leading to improved recommendation accuracy. Additionally, evaluating the effectiveness of our approach on diverse datasets would help measure its generalizability and robustness. Another avenue worth investigating is the integration strategy of the enriched features. Rather than embedding them directly into the feature vector of the item, there may be other, perhaps better, ways to store them.

In conclusion, SBRS are crucial for e-commerce, offering the best possible methods to represent and model customer behavior in various ways. Several problems, such as cold start and limited item diversity, still exist, but there are different methods to tackle those challenges. Utilizing graphs is one of the most common approaches that addresses the cold start problem. One promising solution involves leveraging graph-based methods to enrich item representations. By incorporating contextual information from item-item relationships, these approaches can enhance the representation of less popular or new items, thereby improving recommendation quality in sessions involving such items. Continued advancements in both SBRS and graph-based techniques are expected to further enhance recommendation systems. This progress will contribute to more personalized and satisfying shopping experiences for users, ultimately benefiting both consumers and e-commerce platforms.





OTURUM TABANLI ÖNERİ SİSTEMLERİNDE EVRENSEL VARLIK ÇİZGESİ İLE VARLIK ÖZELLİK VEKTÖRLERİNİN ZENGİNLEŞTİRİLMESİ

ÖZET

Yapay Zeka alanındaki hızlı gelişmelerle birlikte, insanlarla doğrudan etkileşim içerisinde bulunan birçok sektör, bu güçlü teknolojilerden yoğun şekilde faydalanmaya başlamıştır. Bu sektörler arasında, kullanıcı davranışlarının doğru bir şekilde modellenmesinin büyük önem taşıdığı e-ticaret sektörü öne çıkmaktadır. E-ticaret uygulamalarında, müşterilerin ilgisini çekecek ve satın alma ile sonuçlanabilecek ürünlerin önerilmesi, kullanıcı memnuniyetini artırmanın ve satışları iyileştirmenin temel yollarından biridir. Bir kullanıcının alışveriş deneyimi, belirli bir zaman dilimi içerisinde ilgi alanlarına uygun ürünleri ne kadar hızlı ve etkili keşfedebildiğiyle doğrudan ilişkilidir.

Farklı kullanıcıların farklı alışveriş alışkanlıklarına sahip olması, öneri sistemlerinin de çeşitlenmesine neden olmuştur. Bu nedenle, öneri sistemleri veri yapısına ve kullanım senaryolarına bağlı olarak farklı gruplara ayrılmaktadır. Başlıca üç ana türden söz edilebilir: İşbirlikçi Filtreleme (Collaborative Filtering), İçerik Tabanlı Filtreleme (Content-Based Filtering) ve bu iki yaklaşımı birleştiren, her iki tarafın da kuvvetli yönlerini kullanan Hibrit Sistemler. İşbirlikçi filtreleme yöntemleri, bir kullanıcının tercihlerini, benzer geçmiş etkileşimlere sahip diğer kullanıcıların tercihlerine göre tahmin eder. Bu yöntem, kullanıcıların geçmişte benzer ürünleri beğendiği varsayımına dayanarak gelecekte de benzer ürünleri tercih edeceğini öngörür. Bir diğer taraftan, içerik tabanlı filtreleme yöntemleri, ürünlerin içerik özelliklerini analiz ederek kullanıcının geçmişte ilgilendiği ürünlerle benzer özelliklere sahip yeni ürünleri önermeye odaklanır. Bu yöntem, her kullanıcı için özelleştirilmiş bir profil oluşturarak, benzer nitelikteki ürünleri belirlemeye çalışır. Farklı veri yapıları ve uygulama ihtiyaçları doğrultusunda bu yöntemlere ek olarak daha birçok öneri sistemi çeşidi de geliştirilmiştir.

Bu bağlamda, Oturum Tabanlı Öneri Sistemleri (Session-Based Recommender Systems, SBRS), özellikle son yıllarda büyük ilgi gören ve kısa süreli kullanıcı etkileşimlerinden öneriler üretmeye odaklanan sistemlerdir. Geleneksel yöntemler çoğunlukla kullanıcıların uzun vadeli geçmiş verilerine ihtiyaç duymaktadır. Ancak SBRS, kullanıcının mevcut oturumundaki davranışları temel alarak anlık ilgi alanlarını tespit etmeyi hedefler. Bu özellikleri sayesinde, kullanıcı geçmiş verisinin bulunmadığı ya da kullanıcı tercihlerinin hızlıca değiştiği durumlarda önemli avantajlar sunar. SBRS, kullanıcının bir oturum içinde gerçekleştirdiği işlem sırası ve işlem bağlamı gibi unsurları dikkate alarak daha alakalı, zamana duyarlı ve kullanıcının o anlık isteklerine odaklı öneriler üretme potansiyeline sahiptir. Bu sayede, günümüzde

çevrimiçi alış-verişin epeyce yaygın olduğu ve kullanıcı isteklerinin çok hızlı değiştiği düşünüldüğünde, e-ticaret için muazzam bir öneme sahiptir.

Günümüzde, özellikle mobil uygulamalar ve web tabanlı alışveriş platformlarında, kullanıcıların ürünlerle etkileşimleri yoğunlukla kısa süreli oturumlar şeklinde gerçekleşmektedir. Bu kısa oturumlar sırasında, kullanıcının ilgi alanları hızla değişebilir ve bu da geleneksel sistemlerin yeterince esnek olamamasına, kullanıcının anlık isteklerine yönelik öneriler yapılamamasına neden olur. Örneğin, bir kullanıcı birkaç cep telefonu inceleyip bir tanesini satın aldıktan sonra, sonraki oturumlarında cep telefonlarına olan ilgisini kaybedebilir. Bu gibi durumlarda, geçmiş verilere dayanan öneri sistemleri güncel kullanıcı ihtiyacını doğru şekilde yansıtamayabilir. Buna karşın SBRS, kullanıcının o anki etkileşimlerine odaklandığı için güncel ihtiyaçlara daha uygun öneriler sunabilir.

Her ne kadar SBRS kısa vadeli kullanıcı tercihlerinin modellenmesinde oldukça başarılı olsa da, bazı sınırlamaları da beraberinde getirir. Bu sınırlamaların başında, soğuk başlangıç (cold-start) problemi ve sınırlı ürün çeşitliliği gelmektedir. Soğuk başlangıç problemi, hakkında yeterli geçmiş etkileşim verisi bulunmayan ürünlerin (veya kullanıcıların) doğru şekilde önerilememesi durumudur. Bu durum, özellikle yeni eklenen ürünlerin sistem tarafından göz ardı edilmesine neden olabilir. Sınırlı ürün çeşitliliği problemi ise, öneri sistemlerinin sıklıkla popüler ürünleri önermesi sonucunda oluşur; bu da kullanıcıya sunulan seçeneklerin çeşitliliğini azaltır. Bu tür zorluklarla başa çıkmak amacıyla, çizge tabanlı yapılar kullanılarak geliştirilen öneri sistemleri ön plana çıkmıştır. Bu sistemler, ürünler arası ilişkileri daha iyi modelleme kapasitesine sahiptir.

Bu çalışmada, SBRS performansını artırmak amacıyla, ürün temsillerinin çizge tabanlı bilgi toplama (aggregation) yöntemleri ile zenginleştirilmesi önerilmektedir. Önerilen yöntemde, ürünler arası ilişkileri içeren bir ürün grafiği oluşturularak, her ürün düğümünün komşularından bağlamsal bilgi edinmesi sağlanır. Elde edilen bu bilgilerin her ürünün özellik vektörüne entegre edilmesiyle, zenginleştirilmiş temsil vektörleri elde edilir. Bu vektörler, özellikle geçmiş etkileşim verisi az olan "soğuk ürünler" için anlamlı katkılar sağlar. Bu yöntem, sadece öneri doğruluğunu değil, aynı zamanda öneri çeşitliliğini de artırma potansiyeline sahiptir.

Çalışmada geliştirilen bu zenginleştirilmiş ürün vektörleri, oturum tabanlı öneri sistemlerinde sıklıkla kullanılan iki farklı mimariye entegre edilmiştir: Grafik Sinir Ağlarıyla Oturum Tabanlı Öneri (Session-based Recommendation with Graph Neural Networks, SR-GNN) ve Uzun Kısa Süreli Bellek (Long Short-Term Memory, LSTM). Özellikle soğuk ürün içeren oturumlar gibi zorlu senaryolarda, öneri doğruluğunu artırmak hedeflenmiştir. Ürünlerin daha geniş bir bağlamda temsil edilmesiyle, sistemin hem genel doğruluğu artmakta hem de kişiselleştirme kapasitesi gelişmektedir.

Deneyler, RecSys2022 Challenge kapsamında da kullanılan gerçek dünya e-ticaret verilerinden oluşan Dressipi 2022 veri seti üzerinde gerçekleştirilmiştir. Bu veri seti 1.1 milyon adet kullanıcı oturumlarından oluşmaktadır. Veri seti kendi içerisinde etiketlenmiştir: 1 milyon adet oturum eğitim verisi, 100 bin adet oturum ise 2 farklı test verisi olarak işaretlenmiştir. Test verilerin kendi içerisinde ikiye bölünmüş, bunlardan bir tanesi yarışma kapsamında katılımcılara sağlanmış, diğer test veri seti ise

gizli tutulmuştur. Yarışma sonlandığında ise tüm veriler paylaşılmıştır. Bu çalışmada eğitim verisini tamamının kullanımının, yalnızca en güncel belirli bir bölümünü kullanımına kıyasla daha başarısız sonuç verdiği gözlemlenmiştir. Son 2, 3 ve 4 aylık periyotlar kullanılarak yapılan testler sonucunda en güncel son 3 aylık eğitim verisinin kullanımına karar verilmiştir.

Çalışmada başarı değerlendirmesi için yaygın olarak kullanılan ölçütlerden biri olan Mean Reciprocal Ranking (MRR) metriği tercih edilmiştir. Değerlendirme sürecinde, kullanıcıya sunulan öneri listesinin ilk kaç ögesi içerisinde doğru ürünün yer aldığını analiz edebilmek amacıyla, MRR hesaplamalarında kesim noktaları 20 ve 100 olarak belirlenmiştir (MRR@20, MRR@100). Bu sayede, hem kısa liste önerilerinin (top-20) hem de daha geniş öneri listelerinin (top-100) başarımı ayrı ayrı analiz edilmiştir. SR-GNN modelinin MRR@20 başarısı, önerilen varlık özellik vektörü zenginleştirilmesi ile, 0.16968'dan 0.17144'e kadar artırılabilirken, MRR@100 başarısı 0.17360'tan 0.175356'a kadar artırılabilmiştir. Aynı şekilde LSTM mimarisi için de incelendiğinde başarılar MRR@20 için, 0.16421'den 0.16506'ya kadar, MRR@100 için ise 0.16816'dan 0.16903'e kadar artırılabilmiştir. Soğuk oturumlar üzerinde SR-GNN mimarisinde elde edilen başarı artışının LSTM mimarisine kıyasla çok daha net ve fazla olduğu gözlemlenmiştir.

Özet olarak, yapılan kapsamlı deneysel analizler, önerilen yöntemin hem SR-GNN hem de LSTM modellerinde genel öneri doğruluğunu artırdığını göstermektedir. Bununla birlikte, özellikle soğuk ürünler içeren oturumlarda SR-GNN mimarisinin başarımında daha net bir artış gözlemlenmiştir. LSTM mimarisinde ise genel başarıda anlamlı bir miktarda artış gözlenirken soğuk oturumlar üzerinde gözlemlenen artış SR-GNN mimarisine kıyasla daha az olarak bulunmuştur. Bu sonuçlar, çizge tabanlı zenginleştirmenin özellikle çizge mimarilerle bütünleştiğinde daha da etkili olduğunu göstermektedir.

Gelecekteki çalışmalar için önerilen yöntemin farklı öneri sistemlerine entegre edilmesi değerlendirilebilir. Özellikle içerik tabanlı veya hibrit sistemlerle birlikte kullanıldığında, zenginleştirilmiş ürün temsillerinin katkısı daha da belirgin hâle gelebilir. Ayrıca, farklı veri setlerinde yapılan deneyler sayesinde yöntemin genellenebilirliği ve dayanıklılığı test edilebilir. Bu bağlamda, özellik zenginleştirme sürecinde kullanılan entegrasyon stratejilerinin çeşitlendirilmesi ve alternatif yöntemlerin araştırılması da gelecek çalışmalar için anlamlı bir araştırma alanı sunmaktadır.

Sonuç olarak, oturum tabanlı öneri sistemleri, e-ticaret uygulamalarında kullanıcı davranışlarını doğru modelleyebilmek adına son derece önemli bir rol oynamaktadır. Bu sistemlerin sahip olduğu sınırlamaları aşmak adına çizge tabanlı zenginleştirme yaklaşımları etkili ve umut vadeden bir çözüm sunmaktadır. Ürünler arası ilişkilerden elde edilen bağlamsal bilginin ürün temsillerine entegre edilmesiyle, öneri kalitesi özellikle az etkileşimli ürünler için önemli ölçüde artırılabilir. SBRS ve çizge tabanlı tekniklerdeki ilerlemeler, hem kullanıcı memnuniyetini artırmakta hem de e-ticaret platformları için daha yüksek etkileşim ve dönüşüm oranları sağlamaktadır.



1. INTRODUCTION

With the increasing volume of user data on e-commerce applications, recommender systems have emerged as a helpful tool to assist people in making informed decisions. Without such systems, a user may have difficulty deciding what to view or, more importantly, purchase. Thus, making recommendations to guide users where to click, depending on their instant need, is essential for delivering timely and relevant content that aligns with their intent.

There are different types of recommender systems, each employing a different strategy to make recommendations for users. Collaborative Filtering (CF) relies on the idea that users who have interacted with similar items in the past are more likely to enjoy similar items in the future. It makes recommendations based on patterns in user behavior. Content-Based Filtering (CBF) focuses on the characteristics of items rather than users. Based on its features, CBF recommends items that are similar to what a user has previously liked. Additionally, hybrid methods combine both CF and CBF approaches according to need. Hybrid methods leverage the respective strengths of both models, aiming to deliver more accurate and robust recommendations.

As the world shifts toward more dynamic user preferences, there is a growing demand for recommender systems that can adapt to these changes in real-time. In many cases, a user's history does not exist, or the history at a specific moment may not relate to the current intent of the user. Session-Based Recommender Systems (SBRS) can provide more accurate recommendations without requiring a user's complete history. SBRS differ from traditional models (e.g., CF, CBF) in that they can operate on sequences of data with little to no history available, and provide accurate and context-aware recommendations for users' immediate preferences [1]–[3]. These sequences typically consist of items the user has recently viewed, enabling the system to infer short-term user intent rather than relying solely on long-term preferences. These sequences that contain short-term intents are called **sessions**. This focus on

immediate behavior enables session-based recommenders to adapt quickly to what the user is currently interested in, making them highly effective in dynamic environments such as e-commerce, media streaming, and news platforms.

Although SBRs were drawing much attention and performing fairly well in capturing short-term user preferences, they still have certain limitations. One of the main challenges lies in their limited ability to incorporate broader contextual signals, such as long-term trends, global item popularity, or relationships between items across different sessions. This often results in suboptimal performance when dealing with cold-start items, sparse sessions, or items with limited interaction history [4].

To address challenges such as cold start and limited item diversity, graph-based methods have been introduced into the session-based recommendation field. With recent developments in graph-based algorithms for the session-based recommendation field, they have become widely used to improve the quality of recommendations further [5,6]. Graph-based methods aim to better capture relationships between items, both within the session and globally. The use of graph-based methods demonstrated that these methods can be one of the solutions to challenging problems, such as cold start and limited item diversity. More specifically, evidence shows that utilizing complex link prediction algorithms in graph-based recommender systems increases the accuracy of predictions compared to more traditional methods such as CF and item-based Pearson correlations [7].

Most graph-based methods in the session-based recommendation field leverage Graph Neural Networks (GNNs) or Graph Convolutional Networks (GCNs), which are capable of learning rich representations of items. They also introduce extra model training to learn representations for either global or local item-item relationships. GNNs operate on structured data, such as item transition graphs, and iteratively update node embeddings by exchanging information with connected items. This mechanism allows them to capture both local and global patterns within sessions. GCN is a specific type of GNN that applies a convolution operation over graph structures, enabling the model to incorporate higher-order connectivity in a scalable way. In the context of session-based recommendation, GCNs are often used to encode

session graphs where items are nodes and co-occurrences of items define edges. This mechanism helps models to understand not only the transitions but also the indirect dependencies among items, enhancing the model’s ability to recommend relevant content even in noisy or sparse sessions. For instance, models like Session-based Recommendation with Graph Neural Network (SR-GNN) construct session graphs to model item transitions and utilize GNNs to learn session embeddings from both inter-session and intra-session information [8]. Similarly, Global Context Enhanced Graph Neural Network (GCE-GNN) enhances this approach by incorporating global context, learning item embeddings from both session-level and global-level graphs to better capture user intentions [9].

Although both GNN and GCN models appear to be among the most powerful approaches for representing item-item relationships while preserving the sequential nature of sessions, they have certain drawbacks. These models often require complex training procedures, high computational resources, and large-scale graph construction, which may not always be feasible, especially in real-time or resource-constrained environments. Moreover, their performance tends to degrade when dealing with extremely large graphs or items that have very few interactions, limiting their scalability and applicability in certain practical scenarios. Additionally, in scenarios with constrained resources, such models become infeasible, as their reliance on large-scale training makes them unsuitable for lightweight or real-time applications.

Graph aggregation techniques can be used as an alternative to GNN and GCN methods. Graph aggregation is incorporated in SBRS to capture complex relationships between items based on user interactions. Typically, user sessions—sequences of interactions with items are modeled as graphs where nodes represent items and edges denote transitions or co-occurrences between items. This process allows the recommender system to understand not just direct item-to-item transitions but also higher-order item-item relationships or similarities, enhancing its ability to predict the next item a user might interact with in a session. The graph aggregation steps can be defined as follows:

- **Message Passing:** Each node (item) collects information from its neighboring nodes. This involves transmitting features or embeddings through the edges connecting them.
- **Aggregation:** The collected information from neighboring nodes is combined using aggregation functions such as mean, sum, or attention mechanisms. This step synthesizes the local neighborhood information for each node.
- **Update:** The aggregated information is then used to update the node’s representation, often through neural network layers, enabling the model to learn richer item embeddings that encapsulate both local and global context.

In this work, we used graph aggregation on the global graph, which is created from sessions to enrich feature representations of items. In our work, GNN, GCN, or any neural networks related to graph structures are not used on the global graph. The first step is to create a global graph that contains item-item relationship information. The global graph is a directed, weighted structure where each unique item is represented as a node, and the connections between items, based on their transitions, are depicted as edges with associated weights. There is an outgoing and incoming edge between two items if they appear consecutively in any session. Edge weights are dependent on the frequency of these consecutive appearances. The primary reason for constructing the graph as directed rather than undirected is to enable the normalization of edge weights based solely on the incoming edges of a given node before the neighbor aggregation process. After normalizing the weights of incoming edges for every node, this global graph is used to receive item neighborhood information. The highest edge-weighted (incoming edges) k items are selected from the neighborhood of an item. Then, with a simple aggregation function, these highest weighted k neighbors are used to enrich an item’s feature vector. A tunable contribution coefficient controls the influence of neighboring items during this enrichment. This parameter allows the method to adapt to different levels of neighbor reliance. Higher contribution coefficients indicate that the neighbors of an item will have a greater impact compared to the item’s own feature vector; lower values mean that the item’s own feature vector will be more dominant in the calculation of the enriched feature vector. Finally, these enriched features are

fed into Long Short-Term Memory (LSTM) and SR-GNN models, allowing them to benefit from global contextual signals while maintaining training efficiency. In doing so, the proposed method addresses the cold-start problem by propagating useful signals across item neighborhoods, without introducing significant computational overhead.

Evaluations are carried out using the Dressipi Dataset, which was featured in the RecSys 2022 Challenge. This dataset offers a real-world e-commerce environment containing 1.1 million session-based interactions that resulted in purchases. It includes both temporal session data and detailed item-level feature information. These characteristics make it a suitable and challenging benchmark for assessing the effectiveness of SBRS.

1.1 Purpose of Thesis

This thesis aims to enhance the accuracy of recommendations for sequential recommendation models while maintaining a lightweight and straightforward training process. The proposed approach primarily focuses on enriching item features using neighborhood information to achieve better item representations. With the enriched item feature vectors, it focuses on improving the recommendation accuracy in cold-start scenarios, where items have limited interaction history. This design not only enhances the expressiveness of item features but also enables the use of simpler sequential model architectures, without sacrificing the item-item relationship information, such as LSTM and Gated Recurrent Unit (GRU). Additionally, this method proposes a novel approach to storing neighborhood information. It enriches the features of items to have additional information received from their neighbors. By doing so, it aims to eliminate the need for an additional vector or an embedding layer to capture global item-item relationship information.

In summary, the main goals of this thesis are twofold: to enhance the quality of item representations by embedding global neighborhood information directly into item features, and to maintain an efficient, low-complexity training pipeline that is compatible with standard sequential models. By eliminating the dependency on heavy graph-based learning during training and inference, the proposed method provides a scalable and adaptable solution for real-world recommendation systems.

Furthermore, the approach addresses some of the key challenges faced by sequential recommenders, such as cold-start items and limited session contexts, without the need for additional model components or complex training routines. The enriched item features serve as a lightweight but powerful bridge between local session information and global item relationships.

Through extensive experiments conducted on the Dressipi Dataset, this thesis demonstrates that incorporating graph-aggregated features into item representations can lead to significant improvements in recommendation accuracy, particularly in cold-start scenarios, while maintaining a lightweight model design, as demonstrated with both LSTM and SR-GNN models.

1.2 Contribution of Thesis

This thesis contributes to the field of session-based recommendation by introducing a novel approach for encoding item-item relationship information. Instead of relying on explicit graph structures during training, the proposed method enriches item feature vectors through an item neighbor aggregation process. Additionally, a contribution coefficient is introduced to balance the influence between an item’s original features and the aggregated information from its neighbors in the final enriched feature representation. To make it clearer, the contributions of this thesis are given below:

- **Feature Enrichment through Neighbor Aggregation:** A method is proposed to enrich item feature vectors by aggregating information from neighboring items, effectively capturing item-item relationships without relying on complex graph-based training.
- **Introduction of a Contribution Coefficient:** A tunable coefficient is introduced to control the balance between an item’s original features and the influence of its neighbors during feature enrichment.
- **Lightweight and Scalable Training Framework:** The proposed feature enrichment enables the use of standard sequential models such as LSTM and GRU, eliminating the need for specialized graph-based layers or large-scale graph construction, and resulting in faster, more scalable training.

- **Improved Performance in Cold-Start Scenarios:** The enriched features significantly enhance recommendation accuracy for items with limited historical interaction, addressing one of the core challenges in session-based recommendation.

These contributions collectively offer a simple yet effective framework for enhancing session-based recommendation systems, with a particular focus on balancing recommendation quality, computational efficiency, and model simplicity.

1.3 Structure of Thesis

This thesis is organized into four chapters: Introduction, Related Works, Enriching Item Representations with Graph Aggregation, and finally Conclusion. Chapter 1, *Introduction*, provides an overview of the study, outlining the motivation, problem definition, and the main contributions. Chapter 2, *Related Works*, reviews the existing literature on session-based recommendation systems, feature enrichment methods, and graph-based approaches. Chapter 3, *Enriching Item Representations in Session-Based Recommendation with Graph Aggregation*, provides a comprehensive description of the proposed methodology, covering the preprocessing procedures, graph construction process, feature aggregation technique, and the implementation details and hardware information. Chapter 4, *Experimental Results*, presents the evaluation conducted on the Dressipi dataset, along with performance comparisons and additional commentary. *Conclusion* summarizes the findings, discusses the limitations of the study, and suggests possible directions for future research.



2. RELATED WORKS

In this chapter, previous studies relevant to this thesis are reviewed with brief explanations to position the proposed approach within the broader literature. The chapter is organized into three parts: traditional recommender systems, SBRS, and the use of graph-based models and GNNs in recommendation. Each section outlines the strengths of the respective algorithms and highlights their key limitations, particularly in handling sparse data, dynamic user behavior, or scalability challenges. This literature review aims to provide a comprehensive understanding of the evolution of recommendation models, with a focus on how recent advancements have shaped session-based recommendation strategies and how this work seeks to address remaining gaps.

2.1 Traditional Recommender Systems

Recommender Systems have evolved and improved significantly over the years. Traditional recommender systems represent the earliest approaches that contributed to the development of the recommendation field. Traditional recommender systems primarily rely on users' historical data to understand their preferences and predict future actions [10,11]. These systems are generally classified into three main categories: CF, CBF, and hybrid methods.

Collaborative Filtering. Early work in CF laid the foundational concepts and demonstrated practical implementations for large-scale recommendation systems. One of the earliest implementations highlights the limitations of small community systems and emphasizes the need for scalable algorithms [12]. Likewise, an early comprehensive overview of CF recommender systems is presented in [13], which has informed subsequent research by addressing methodological challenges and real-world applications. In refining CF for diverse data types and complex user behaviors, [14] proposes a method tailored for implicit feedback datasets. Their work

addresses the inherent challenge of the absence of negative feedback by modeling implicit interactions, thereby extending CF techniques beyond explicit rating systems. Finally, these methods are further enhanced by the introduction of matrix factorization techniques that efficiently capture latent relationships between users and items [15]. However, while these methods have proven valuable in long-term user modeling, their reliance on persistent user histories limits their direct applicability to session-based recommendation scenarios, where user identifiers and long-term preferences may not be available.

Content-Based Filtering. While collaborative filtering has been effective in many recommendation scenarios, it relies heavily on user-item interaction histories, which may be unavailable or sparse, particularly in cold-start scenarios. In such cases, CBF becomes a viable alternative, as it leverages item metadata to make recommendations without requiring extensive historical interaction data. Several studies have addressed the challenges and methodologies associated with content-based recommendation systems. For instance, a systematic review of movie recommender systems discusses content analysis techniques applied to large datasets such as The Internet Movie Database (IMDb), highlighting the importance of rich item metadata in enhancing recommendation quality [16]. Additionally, content-based approaches have been extended to academic social networks by incorporating social signals and feature-based representations to improve recommendation accuracy, illustrating the adaptability of CBF across domains [17]. However, in session-based recommendation settings, where only a short sequence of item interactions is available, CBF methods may struggle to capture dynamic and sequential user preferences as they primarily rely on static item features rather than interaction patterns.

Hybrid Methods. Hybrid methodologies are designed to combine the community-driven insights from collaborative filtering (CF) with the rich item-specific attributes utilized in content-based filtering (CBF), aiming to overcome the limitations of each standalone approach. One such method proposes a hybrid algorithm that leverages both paradigms to address the cold-start problem by recommending items with limited collaborative data using their content features [18]. These systems enhance the accuracy and diversity of recommendations and have been successfully

applied across domains such as multimedia streaming [19,20], movie recommendation [21], tourism applications [22], and music platforms [23]. However, in session-based recommendation settings, where only short-term interaction data is available, hybrid methods may fall short due to their dependence on either persistent user profiles or static item metadata.

2.2 Session-Based Recommender Systems

While traditional methods can provide accurate recommendations based on historical user data, there are many scenarios where such data is unavailable or irrelevant, such as for anonymous users, cold-start users, or when users exhibit rapidly changing interests during short interaction periods. SBRS have emerged as an alternative to traditional recommendation approaches that leverage long-term user preferences. SBRS aim to capture the dynamic and context-dependent nature of user behavior, providing recommendations that are more relevant in the latest user decisions [1]. Markov Decision Processes (MDP) have been explored in earlier works as a framework for making sequential recommendations [24]. Markov Chains and Matrix Factorization are employed to capture different aspects of user-item interactions, where Markov Chains model short-term sequential patterns in user behavior and Matrix Factorization focuses on uncovering long-term user preferences based on historical data [25]. Various deep learning techniques have been explored for session-based recommendation tasks. A hierarchical recurrent neural network framework is introduced, which personalizes recommendations by effectively modeling sequential patterns and short-term interests in session data [26]. Evaluation of session-based recommendation approaches is essential for understanding their practical implications and limitations in real-world environments. A comprehensive evaluation framework benchmarks various session-based recommendation algorithms, highlighting the advantages of sequence learning techniques in capturing dynamic user behavior [27]. Additionally, [9,28,29] illustrated that it is possible to build effective SBRS that are context-aware. They enriched our understanding of how contextual information can be effectively integrated to enhance recommendation performance in dynamic, session-based environments. Schifferer et al. [30] employed

a context-aware ensemble approach in the RecSys 2022 Challenge, leveraging item and session-level metadata from the Dressipi dataset to enhance predictive accuracy in fashion recommendations.

Following the rise of attention mechanisms in deep learning, their application has also gained significant traction in the session-based recommendation domain. One of the earliest works in this direction is the Neural Attentive Recommendation Machine [31], which combines a recurrent neural network with an attention mechanism to model both the sequential behavior and the user’s main intent during a session. An LSTM model, enhanced with an attention mechanism, is proposed to address common challenges in session-based recommendation, such as short session lengths and item sparsity. Through empirical comparisons with standard Recurrent Neural Network (RNN) and LSTM models, the attention-augmented LSTM demonstrated superior performance, highlighting the benefit of selectively focusing on relevant parts of the session sequence during prediction. An LSTM model enhanced with an attention mechanism is introduced in [32]. Similarly, STAMP (Short-Term Attention/Memory Priority) is proposed in [33], employing a non-recurrent framework that dynamically attends to the most relevant part of a session, effectively balancing short-term preferences with global item popularity. Beyond sequential models, attention has been applied to graph-based representations, exemplified by the Target Attentive Graph Neural Network (TAGNN) for session-based recommendations in [34]. More recently, transformer architectures leveraging self-attention have also been explored for session-based recommendation [35], evaluating a transformer-based model against graph neural networks on the Dressipi RecSys2022 dataset, demonstrating the competitive potential of attention mechanisms even beyond recurrent and graph-based baselines.

2.3 Graph Related Methods in SBRS

Even though the SBRS have demonstrated strong outcomes in the field, they still have several challenges that are addressed by graph-related methods. One of the main drawbacks of SBRS is that they often struggle to capture information between different sessions. Before the widespread use of Graph Neural Networks, several studies employed graph-based techniques to model user behavior in session-based

recommendation tasks. One foundational approach is Factorizing Personalized Markov Chains (FPMC) [36], which models user navigation through items as a Markov chain, where transitions are learned through matrix factorization. While not a graph algorithm in implementation, the model implicitly operates on a user-item transition graph, capturing personalized next-item predictions without relying on deep learning. Another influential line of research is the session-based k-Nearest Neighbor (session-kNN) approach, where sessions are modeled as nodes connected by shared items, forming a session-item bipartite graph. A graph-based method for session-based recommendation is introduced using Neo4j Graph Database¹ [37]. In this method, items and sessions are represented as nodes, and their interactions are modeled as edges, forming a graph structure that captures the co-occurrence relationships between them. The system continuously and incrementally collects data from an e-commerce website, enabling efficient data management and real-time recommendations without the need for extensive training phases. Another notable study [38] employs a session similarity-based approach to address the cold-start problem commonly found in e-commerce. Their method clusters sessions to identify patterns, thereby facilitating more informed recommendations based on session similarities rather than user profiles. The performance of their framework suggests that clustering techniques can significantly bolster the recommendation process, particularly when user data is scarce. Moreover, a novel approach incorporating structural knowledge via message-passing mechanisms within the CF framework is presented [39]. Although this framework does not explicitly employ graph neural networks, it discusses the potential of integrating various forms of structural information, such as knowledge graphs, to enhance user behavior comprehension and drive more personalized recommendations. This highlights the versatility of graph-based approaches in modeling user interactions without relying solely on neural networks. Kang and McAuley contribute to this discourse with their self-attentive sequential recommendation model [40]. This model effectively employs attention mechanisms to discern and respond to patterns in user activity sequences without relying on graph learning strategies. The efficiency of their model in processing both sparse and dense datasets is noteworthy, making it a valuable

¹<https://neo4j.com/product/neo4j-graph-database/>

resource for exploring session-based techniques. A notable contribution that utilizes graph-based representations is presented, evaluating their performance on the Dressipi RecSys2022 dataset. [41]. Their work constructs a session-item interaction graph and performs recommendations via similarity-based and structural analysis methods, rather than through learned representations. The proposed framework demonstrates the effectiveness of traditional graph modeling in capturing item co-occurrence and session dynamics in real-world recommendation tasks.

These studies collectively illustrate that meaningful item transitions and session dynamics can be effectively captured using graph-based structures alone, offering practical and scalable alternatives to GNN-based methods. Such approaches are particularly valuable in scenarios with limited computational resources or where real-time recommendations are essential.

In recent years, with the growing interest in learning from structured data, graph learning techniques, particularly GNNs, have emerged as powerful tools in session-based recommendation. Unlike earlier approaches that relied on static graph structures or heuristic similarity measures, GNN-based methods learn item and session representations directly from the graph topology in a data-driven, end-to-end manner. These models are capable of capturing higher-order dependencies and long-range item transitions, which are difficult to model with traditional graph-based techniques.

The SR-GNN model is introduced to represent sessions as graphs and apply GNNs to capture item transitions within sessions [8]. Additionally, they incorporate inter-session information by representing each session as a combination of intra-session and inter-session signals. This fusion is achieved through an attention mechanism that allows the model to weigh relevant information from both levels effectively. An adaptive weighting mechanism is integrated into GNN architectures to dynamically emphasize important session signals. In their study, they evaluated this novel method on the Dressipi RecSys2022 dataset and benchmarked it against two widely recognized graph-based baselines: SR-GNN and GCE-GNN [42]. Dual Attention Transfer Based on Multi-Dimensional Integration (DAT-MDI) is proposed to combine session and global graphs with a GNN, thereby learning

better item representations in each session [43]. Then they combine multi-level session representations with a soft-attention mechanism. On another example, a Repeat-aware Graph Neural Network (ReGNN) is proposed to specifically model the tendency of users to revisit items during a session [44]. By incorporating a mechanism that accounts for repeat behaviors, their work demonstrates that explicitly modeling reoccurrence can significantly boost recommendation performance in session-based settings. The Contrastive Multi-level Graph Neural Network (CM-GNN) [45] is proposed, applying local, global, and hyper-level GCN to capture both fine-grained and aggregated item relationships effectively. The limitation that most existing SBRS focus solely on individual sessions, restricting efficient modeling of item-item relationships, is addressed in [46]. Moreover, session data often lacks the user’s historical data due to privacy or anonymization, making it difficult to utilize cross-session information. To overcome these challenges, it introduces the Full Graph Neural Network (FGNN) to better capture global item-item relationships. Furthermore, to broadly capture inter-session information, it proposes the Broadly Connected Session (BCS) graph, which links different sessions to each other based on their shared items. Authors evaluated their work on two widely used e-commerce datasets (Yoochoose and Diginetica) and showed that their method outperformed the state-of-the-art methods.

Although GNN methods have proven effective in the field of session-based recommendation, they also introduce large models that are time-consuming to train in many scenarios. Classical DNN and matrix factorization approaches remain favored in real-world applications due to their superior scalability, whereas GNNs struggle to scale efficiently with growing user–item interaction graphs [47]. In support of this, an environmental analysis is conducted that reveals the substantial energy consumption and carbon emissions associated with training large GNN models for recommendation tasks, further underlining the need for efficiency-focused architectures [48]. Additionally, many GNN methods still avoid modeling item-item relationships beyond the session scope because the item sets are too large, requiring a large number of parameters and time-consuming training procedures. In this study, we focus on enriching item representations by using global item-item relationships derived from the global item-item graph. Since there are far too many items to train and learn

item-item relationships in this scenario, we employed graph aggregation. We proposed a novel method for storing item-item information by embedding it within the feature vectors of the items themselves. This approach complements existing session modeling techniques, such as SR-GNN, by providing globally informed item representations without increasing the complexity of model training.



3. ENRICHING ITEM REPRESENTATIONS IN SESSION-BASED RECOMMENDATION WITH GRAPH AGGREGATION

In this chapter, we present a novel method that enriches item representations in session-based recommendation by incorporating global graph aggregation, which leverages item-item relationship information while maintaining sequential training. First, we describe the preprocessing of the dataset. Next, we present the item feature representation method, followed by the steps involved in constructing the global graph and explain how these features are enriched through graph-based aggregation. We then detail the model architectures used in our experiments. Finally, we present the implementation details and hardware information for our work.

3.1 Preprocessing

In this section, we describe the preprocessing strategies used to prepare session data for training.

3.1.1 Train-test split

The dataset spans a total of 18 months. The final month is reserved exclusively for validation and testing purposes. Consequently, the training set contains data from the preceding 17 months. Among these, only the most recent 3 months are used for training. This decision was informed by preliminary experiments comparing different training windows; using the latest 2 or 4 months yielded inferior performance compared to using the latest 3 months.

Fashion trends evolve rapidly over time, making temporal relevance a critical factor in recommendation performance. Since both validation and final testing are performed on the most recent month of the dataset, training on more recent data leads to better alignment with current item preferences and user behavior. However, a trade-off exists between using a large volume of data to capture diverse session patterns and focusing on temporally proximate data to maintain relevance. Empirical results

indicated that using data from the latest three months provided an effective balance, offering sufficient training samples while retaining recency. In contrast, including older sessions (e.g., a four-month window) or using too few sessions (e.g., two months) led to reduced accuracy.

As a result, the final experimental setup utilizes 3 months of training data, with the last month split equally into validation and test sets.

3.1.2 Session slicing

To maintain consistency and manage computational complexity, we retain only the most recent L items from each session, where L is a predefined maximum session length. This truncation ensures the model focuses on the most recent user behavior, which is typically more indicative of near-future interactions.

In addition to emphasizing recent user interactions, this truncation strategy reduces the amount of data processed by the model, thereby improving training efficiency. By limiting the session length, models are relieved from learning long-range dependencies that may be less relevant for predicting imminent action. In summary, this simplification accelerates the training process and enables the model to focus on the most influential items, typically those closer to the point of purchase, which are more predictive of the user’s final choice. Also, since some sessions will be shortened in length, the learning process becomes easier for each model. Although the overall model complexity remains unchanged, training becomes faster because the models are learning from shorter sequences.

3.1.3 Handling short sessions

Depending on the underlying model architecture, short sessions (i.e., sessions with fewer than L items) are treated differently:

- The LSTM model requires a fixed-length input. To meet this constraint, we repeat the first item of the session to the left until the session reaches length L . This ensures that all sessions in a training batch have the same dimensionality.

- SR-GNN-model, input sessions can naturally vary in length. Therefore, we apply no padding. The graph structure inherently accommodates sequences of different lengths, preserving the original session structure as long as it is within the allowed maximum.

This preprocessing strategy allows us to adapt the same dataset for multiple model architectures while ensuring compatibility with their respective input requirements.

3.2 Categorical Feature Representation

The dataset includes a large number of categorical features, each with its own unique set of possible values. For instance, one feature might represent the item's color with values like "red," "blue," or "green," while another might indicate the material type, such as "cotton," "leather," or "polyester." To numerically represent these categorical features, each value within a feature is encoded as a one-hot vector. This means that for a given feature, only the dimension corresponding to the observed value is set to 1, while all others are set to 0.

In this study, item feature representations are formed by concatenating the one-hot vectors of all categorical features, resulting in a single high-dimensional sparse vector for each item. This final structure corresponds to a bag-of-words-style encoding, where each dimension indicates the presence of a specific feature-value pair. An example of how these vectors are formed is illustrated in Figure 3.1. In this figure, three categorical features with different numbers of possible values are each shown as separate one-hot vectors. The final bag-of-words vector is obtained by concatenating these three vectors. Each segment in the final vector corresponds to one feature, and the position of the 1 within that segment indicates which value the item holds for that particular feature.

3.3 Global Graph Creation

To construct the global item graph, each item is treated as a node, and edges are added between consecutive items that appear within the same user session. Specifically, for every adjacent item pair (s_i, s_{i+1}) in a session, two directed edges are created: one from s_i to s_{i+1} and one from s_{i+1} to s_i . This bidirectional edge construction enables

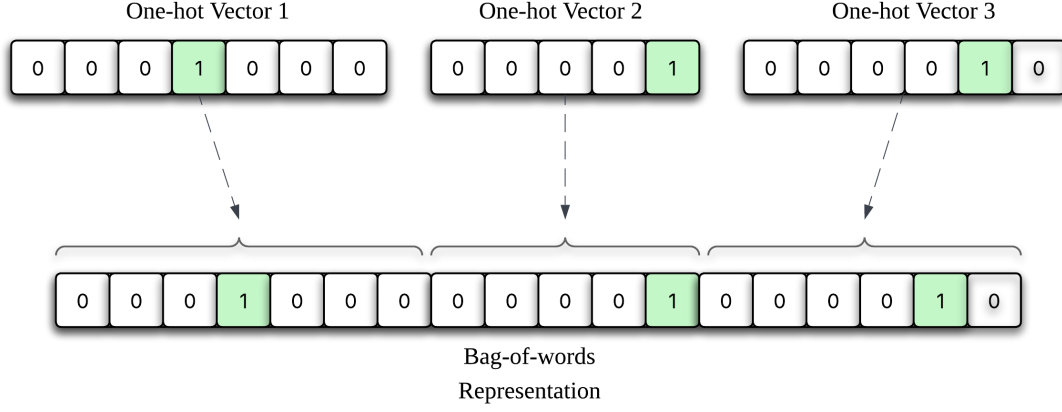


Figure 3.1: An example scenario for creating bag-of-words representations from one-hot vectors.

the graph to capture both forward and backward transitions, which is valuable for session-based recommendation, where both directions of influence can contribute to user intent modeling. However, the primary reason the graph is directed rather than undirected and created this way is to allow normalization only on the incoming edges for each node in the graph, without altering the outgoing edges, as those outgoing edges serve as incoming edges for other items in the graph.

For each directed edge between consecutive items, if the edge does not already exist in the graph, it is added with an initial weight of 1. If it already exists, its weight is incremented by 1. This process is repeated across all training sessions, so the final edge weights reflect the number of times each directed item transition occurs in the data. At this stage, the weights represent raw co-occurrence counts and are not yet normalized.

After all sessions have been processed, an **incoming edge normalization** step is applied. For each node v , the weights of all incoming edges (u, v) are normalized by the total weight of all such incoming edges to that node; this normalization process is given in equation 3.1.

$$\forall (u, v) \in E, \quad E[u, v] \leftarrow \frac{E[u, v]}{\sum_{(j, v) \in E} E[j, v]} \quad (3.1)$$

This normalization ensures that the sum of incoming edge weights for each node is 1, allowing the edge weights to be interpreted as contribution scores during the feature aggregation step. The full graph construction process is described in Algorithm 1.

The result is a weighted, directed graph where edges represent normalized transition likelihoods, and this graph is used to enrich item features through graph-based aggregation, as explained in Section 3.4.

Algorithm 1 Session Graph Construction and Incoming Edge Normalization

INPUT: Set of sessions S , where each session $s = [s_1, s_2, \dots, s_n]$

OUTPUT: Weighted directed graph $G = (V, E)$

```

1: Initialize node set  $V \leftarrow \emptyset$ , edge map  $E \leftarrow \emptyset$ 
2: for each session  $s \in S$  do
3:   for each adjacent pair  $(s_i, s_{i+1})$  in  $s$  do
4:      $u \leftarrow s_i, v \leftarrow s_{i+1}$ 
5:     Add  $u$  and  $v$  to  $V$ 
6:     if  $(u, v) \notin E$  then
7:        $E[u, v] \leftarrow 1$ 
8:        $E[v, u] \leftarrow 1$ 
9:     else
10:       $E[u, v] \leftarrow E[u, v] + 1$ 
11:       $E[v, u] \leftarrow E[v, u] + 1$ 
12:    end if
13:  end for
14: end for
15: for each node  $v \in V$  do
16:    $W_{\text{in}} \leftarrow \sum_{(u,v) \in E} E[u, v]$ 
17:   if  $W_{\text{in}} > 0$  then
18:     for each incoming edge  $(u, v) \in E$  do
19:        $E[u, v] \leftarrow \frac{E[u, v]}{W_{\text{in}}}$  {Normalize incoming edges}
20:     end for
21:   end if
22: end for
23: return  $G = (V, E)$ 

```

To avoid any data leakage, the graph is constructed solely from the training set, with no information from validation or test sessions. The graph is directed and weighted, representing item-to-item transitions observed in user sessions. Importantly, the graph captures both forward and backward relationships between items.

3.4 Neighbor Aggregation On Item Features

First, the **top- k highest-weighted neighbors** are retrieved from the global graph via sorting neighbors of an item from highest weight to lowest weight and taking the first

k of them. Then, normalized weighted neighbor aggregation is applied to enrich their feature vectors.

Normalized Weighted Neighbor Aggregation is used to enrich item features. For each item, we retrieve its top- k most strongly connected neighbors based on incoming edge weights in the graph. Once the neighbors are selected, their feature vectors are aggregated using a normalized, weighted sum. Neighbors with higher edge weights contribute more to the enriched representation. Additionally, a tunable parameter, `ncoef`, is introduced to control the overall influence of the neighbors during this aggregation, allowing the strength of the enrichment process to be adjusted based on the performance. The formula used to enrich item features via normalized weighted neighbor aggregation is given in equation 3.2.

Figure 3.2 illustrates the neighbor aggregation process applied to item A. First, the top- k neighbors of item A, items B, C, D, and E, are selected based on the highest edge weights. These are referred to as the highest-weighted k neighbors. Next, the edge weights connecting each neighbor to item A are normalized to sum to 1; these are called normalized weights. Each neighbor's feature vector is then multiplied by its corresponding normalized weight, and the resulting vectors are summed to obtain the contribution vector. Finally, this contribution vector is added to item A's original feature vector to produce its enriched feature vector.

$$\text{Feature}_j = \text{Feature}_j + \text{ncoef} \cdot \sum_{i \in N(j)} \frac{\text{Feature}_i \cdot w(i, j)}{\sum_{i' \in N(j)} w(i', j)} \quad (3.2)$$

Where,

- Feature_j represents the feature vector for item j .
- Feature_i represents the feature vector for item i .
- `ncoef` is the neighbor contribution coefficient. Higher values increase the influence of neighbor features relative to the item's original features during the enrichment process.
- $N(j)$ represents the highest weighted k neighbors of item j .

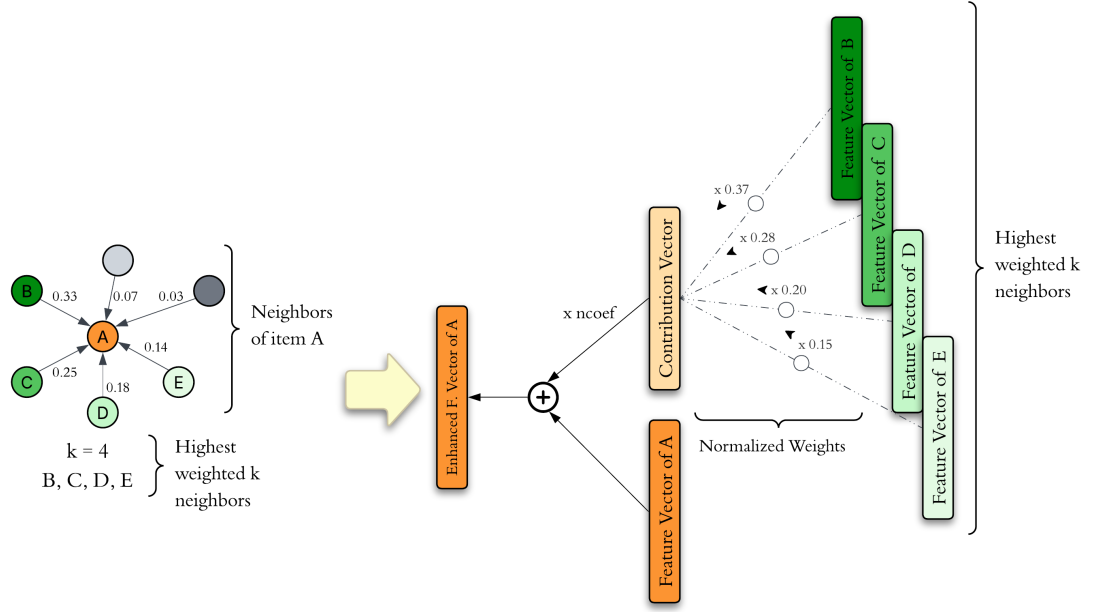


Figure 3.2: Enriching item features with Normalized Weighted Neighbor Aggregation.

- $w(j, i)$ represents the weight between item j and item i .

3.5 Model Training

In this study, both LSTM and SR-GNN models are employed. When selecting a model for training, it's crucial to ensure that it can accept both sequential item IDs and their corresponding sequential item features as inputs to the embedding layer. This dual-input capability allows the model to leverage both the structural and contextual information of items, enhancing its ability to learn meaningful representations and improve recommendation accuracy. **Cross Entropy Loss** is used to train both LSTM and SR-GNN models. Using the Bayesian Personalized Ranking (BPR) loss had an adverse effect on performance, slowing down the training process.

3.6 Session Embeddings

The global graph is designed to capture inter-session information, which is then integrated into the enriched item features through graph aggregation. The process of creating session embeddings, which are used during training, is outlined in Figure 3.4. The item index vector represents the item ID information, while the enriched item

features result from applying graph aggregation to the original item features. Item embeddings are generated using an embedding layer, and feature embeddings are produced by passing the feature vector (with a length of 908) through a linear layer, which projects it into a 256-dimensional vector. The item and feature embeddings are then concatenated to form session embeddings. These session embeddings serve as the input for training both the LSTM and SR-GNN models, enabling them to leverage both item-specific and feature-specific information effectively.

3.7 LSTM Model

For comparative analysis, we implemented an LSTM model in this study. Experiments were conducted with varying numbers of LSTM layers and found that a two-layer architecture with a hidden size of 512 yielded the best performance. A final linear projection layer was added to map the LSTM output to a vector matching the total number of items, enabling next-item prediction within a session. The LSTM architecture used in this study is shown in Figure 3.3.

While the LSTM model demonstrated reasonable accuracy with relatively simple training, it was slightly outperformed by the SR-GNN in evaluations.

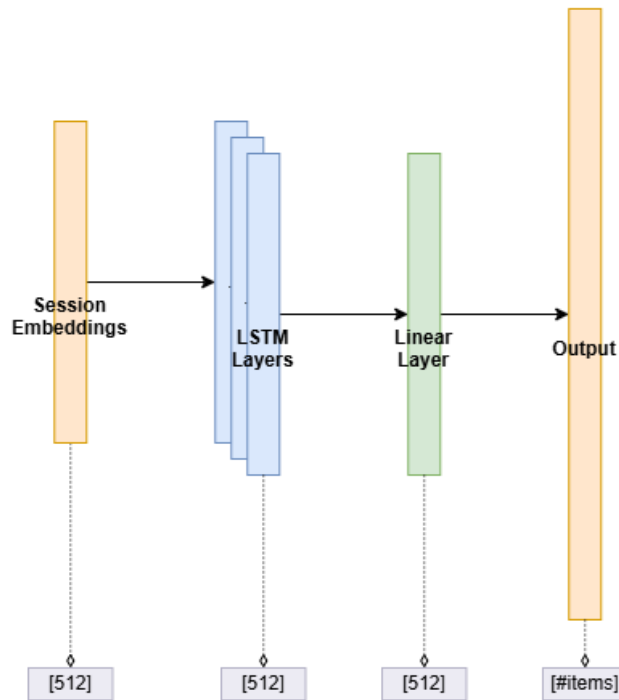


Figure 3.3: LSTM model architecture.

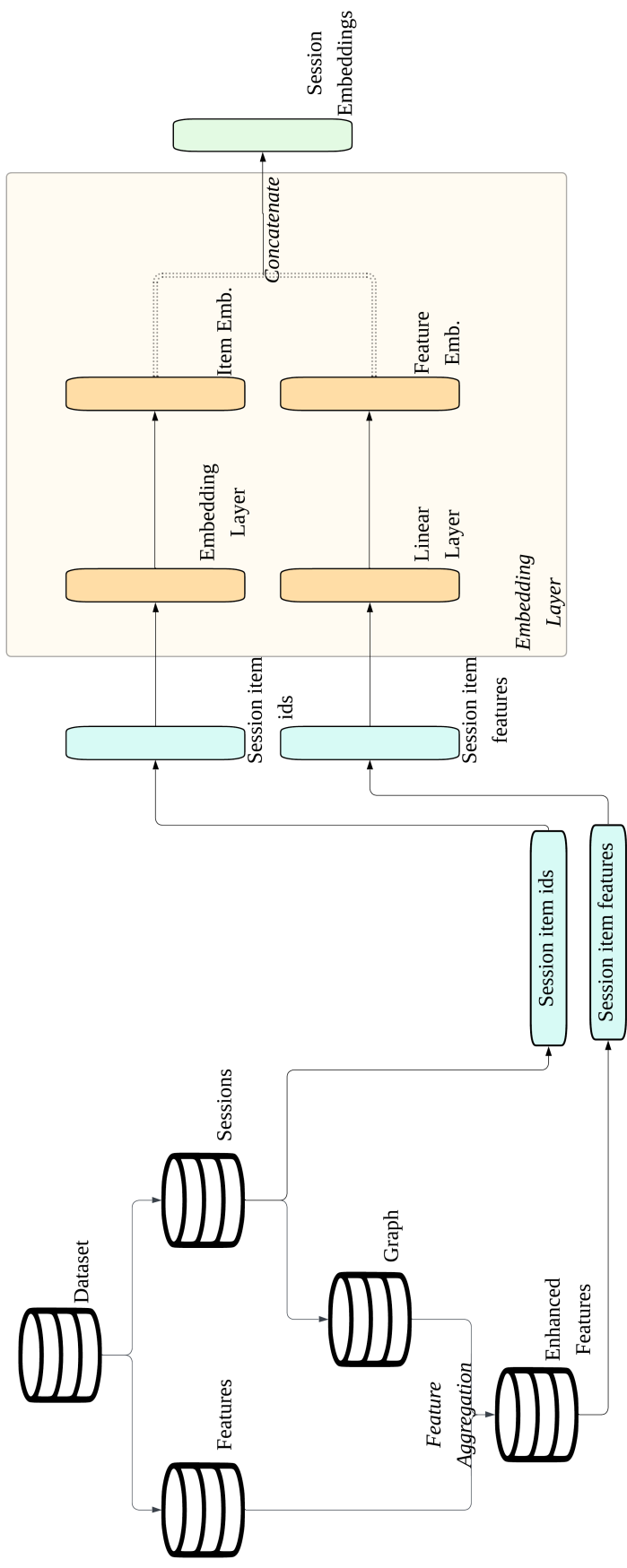


Figure 3.4: Creating session embeddings.

3.8 SR-GNN Model

The SR-GNN model, which uses item features, slightly outperformed the LSTM model in terms of overall recommendation accuracy, so the majority of the training was conducted using the SR-GNN architecture. In its original form, the SR-GNN model did not include item features and relied solely on item embeddings. Figure 3.5 illustrates the SR-GNN model architecture used in this study. The key difference from the original setup is the incorporation of feature embeddings. Once local session embeddings are created, they are passed through a GNN layer. An attention mechanism is then applied to the output of this GNN layer to calculate global session embeddings, refining the session representations by weighting the importance of different item embeddings. A session fusion layer is subsequently used to combine the local and global session embeddings. Finally, the item embeddings are dot-producted with the session fusion output to generate the final item scores. This approach enhances the model's ability to capture both local and global session dynamics while incorporating item feature information.

3.9 Implementation Details and Hardware

The presented models were implemented using the PyTorch deep learning framework. All experiments were conducted using a modular pipeline that supports both LSTM and SR-GNN models.

A comprehensive grid search was performed over a range of hyperparameters to identify the optimal configuration for each model. The parameters explored include the number of nearest neighbors used for feature enrichment (`num_neighbors`), the sequence length of input sessions (`sequence_length`), learning rate (`learning_rate`), batch size (`batch_size`), embedding dimension (`embedding_dim`), hidden state size (`hidden_size`), minimum session length (`min_session_length`), the base contribution coefficient for neighboring items (`ncoef`), the number of negative samples for BPR loss (`num_negatives`).

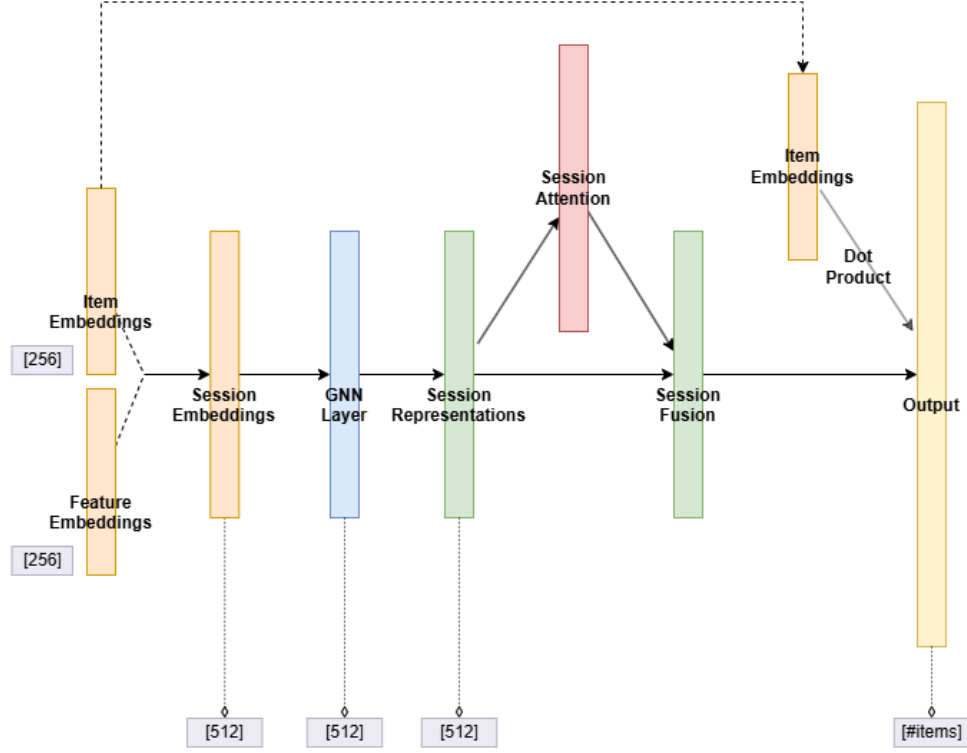


Figure 3.5: SR-GNN model architecture.

Item feature vectors were enriched prior to training using a graph-based aggregation procedure constructed on a global item graph. These enriched features were saved to disk, with file names containing related parameters, and reused during evaluation to ensure consistency. All models were trained using the Adam optimizer with Cross-Entropy and BPR loss. After extensive experimentation with both loss functions, Cross-Entropy loss was the one that gave better results by a large margin. Therefore, the remaining experiments were conducted using only Cross-Entropy Loss, and the formula for the Cross-Entropy Loss is given in equation 3.3.

$$\mathcal{L}_{CE} = - \sum_{i=1}^C y_i \log(\hat{y}_i) \quad (3.3)$$

Where,

- C is the total number of classes.
- y_i is the ground-truth label (one-hot encoded: 1 for the correct class, 0 for others).
- $\hat{y}_i$ is the predicted probability for class i (output of softmax).

Early stopping is applied based on validation Mean Reciprocal Ranking (MRR) values. A patience parameter is defined to control early stopping. If the model's validation MRR does not improve beyond a small threshold for a consecutive number of epochs specified by this parameter, the training process is terminated early to prevent overfitting and reduce computational cost.

Experiments were conducted simultaneously on two different local computers to reduce overall training time. The technical specifications of these machines are listed below:

- **First Computer:**

- **CPU:** Intel Core i7-11700KF
- **GPU:** NVIDIA RTX 3060 (12 GB VRAM)
- **RAM:** 32 GB
- **Operating System:** Windows 11 Home
- **Python:** 3.12.4
- **PyTorch:** 2.4.1
- **Cuda:** 12.1

- **Second Computer:**

- **CPU:** Intel Core i7-12700H
- **GPU:** NVIDIA RTX 4060 (8GB VRAM)
- **RAM:** 32 GB
- **Operating System:** Windows 11 Home
- **Python:** 3.12.4
- **PyTorch:** 2.3.1
- **Cuda:** 12.1

Models were trained and evaluated concurrently across the two machines to accelerate the grid search process. While both systems provided sufficient memory and

GPU resources for training, slight variations in training time were observed due to differences in GPU performance.





4. EXPERIMENTAL RESULTS

This chapter presents information about the dataset, describes the experimental setup, and discusses the results of evaluating the proposed method.

4.1 Dataset

The dataset used in this study is the Dressipi Dataset from the RecSys2022 Challenge. It is publicly available on Dressipi’s official website¹, where additional information about the dataset and the RecSys 2022 Challenge can also be found. It includes 1.1 million online retail sessions that resulted in a purchase. The dataset has two main components: session data and item feature data. The session data includes session IDs, item IDs, and the dates when users interacted with specific items. On the other hand, the item feature data consists of feature labels and their corresponding values. It’s important to note that the dataset is anonymized, meaning it doesn’t include any user-specific information. Additionally, the item features are also anonymized, with feature labels represented as integer numbers and their categorical values encoded numerically.

Figure 4.1 illustrates an example session along with its corresponding purchase from the Dressipi dataset. As depicted, session data and purchase information are typically stored separately within the dataset. It is important to note that this figure is provided only for illustrative purposes; in the actual dataset, both items and their associated features are anonymized.

Figure 4.2 illustrates an example item from the Dressipi dataset along with its associated features. In the actual dataset, both feature names and their values are anonymized: feature names are represented by integer keys, and their corresponding categorical values are also encoded as integers. This figure has been constructed to

¹<https://dressipi.com/downloads/recsys-datasets/>



Figure 4.1: An example session and a purchase from the Dressipi dataset.

visualize the internal structure of item representations within the dataset, and it is published on the original website of the Dressipi dataset.

Each item can have up to 73 features, and the number of possible values for each feature varies. These features may include product-related attributes such as color, style, material, length, and sleeve length, as it is shown in Figure 4.2, though the exact semantics are hidden. In this study, the features are treated as categorical variables and are one-hot encoded before being concatenated into a single bag-of-words-style sparse vector representing each item. After applying one-hot encoding and concatenating all individual feature vectors, each item is represented by a 904-dimensional sparse vector. This vector serves as the bag-of-words-style item representation used throughout the experiments.

4.2 Experiment Environment

Dressipi Dataset spans 18 months of session data. Out of these, 17 months were designated as the training set, while the remaining 1 month was reserved for testing. However, since fashion trends tend to evolve over time, training on the entire 17 months of data did not yield better results compared to using only the most recent portion of the data. As a result, only the last 3 months of data were used for training. In the Dressipi Dataset, there are 2 different test sets, and they both contain sessions from the



Figure 4.2: An example item with its defined features.

latest (18th) month. These files are named as “test_leaderboard” and “test_final” in the dataset. In this study, the dataset named “test_leaderboard” is used for validation, and the dataset named “test_final” is used for testing.

From Table 4.1, several key patterns are observed regarding the structure of the graphs built from different subsets of the training data. The number of unique edges and the average degree vary noticeably. As expected, the full graph built from all sessions (17 months) contains the highest number of edges and the highest average degree, since it has way more sessions than only the last 2, 3, and 4 months of data, resulting in more item transitions. Additionally, it can be seen that the average edge weight decreases as the time span increases. This suggests that within shorter time frames, item pairs tend to form more frequent and stronger relationships, while longer spans introduce many weaker, less frequent connections. This trend may also indicate that fashion trends and user interests change rapidly over time.

Despite these differences in edge-related statistics, the number of nodes remains fixed at 23691 across all graphs. This is because the global graph is initialized to include all unique items found across the full 17-month dataset, regardless of whether they appear in the subset being used. Items that are not present in the shorter time spans

Table 4.1: Graph statistics over different training sets.

Graph	Nodes	Edges	Avg. Degree	Min Wt.	Max Wt.	Avg. Wt.
All Sessions	23618	2158529	182.78677	3e-05	1.0	0.00471
Last 4 Months	9980	553596	110.94108	7e-05	1.0	0.00777
Last 3 Months	8813	474630	107.71134	8e-05	1.0	0.00829
Last 2 Months	7422	343972	92.68984	0.0001	1.0	0.00939

remain in the graph as isolated nodes, i.e., they have no incoming or outgoing edges. This consistent node set ensures a uniform graph structure across different training configurations.

Table 4.2: Information about train, clipped train, validation, and test datasets.

Dataset	Items	Sessions	Avg. Session Length
Train	23618	1000000	5.7438
Train (Last 3 Months)	8733	217072	10.2511
Validation	5780	50000	5.5228
Test	5777	50000	6.9599

When analyzing the average session lengths from Table 4.2, the last 3 months of data show the longest session lengths compared to other portions of the dataset by a large margin. All session length values include the purchased item in their count. For LSTM training, session length is selected as 6 for LSTM, which means all of the sequences used for training have a length of 5 (excluding the purchased item). Using values larger or smaller negatively impacted the performance of LSTM. In contrast, SR-GNN training accommodates variable-length input sequences, so the maximum session length was set to 6 for SR-GNN, which means that sessions that are initially shorter or have equal length to 6 are left unchanged. Only the sessions with more than 6 items are sliced to retain only the most recent 6 items.

4.3 Metrics and Results

Unlike studies that focus on proposing new model architectures, our goal is to evaluate the effectiveness of the proposed feature enrichment technique based on graph aggregation. The technique is model-agnostic and can be applied to any sequential recommendation model. Therefore, rather than comparing results against other state-of-the-art approaches, we examine how the performance of two baseline architectures, LSTM and SR-GNN, varies with respect to the `ncoef` parameter. This

allows us to isolate the impact of our proposed method and understand how different levels of neighbor contribution affect both overall recommendation performance and the performance in cold-start scenarios.

To evaluate model performance, **Mean Reciprocal Ranking** (MRR) at cutoff values 20 and 100 are used, denoted as MRR@20 and MRR@100, respectively. MRR measures the average of the reciprocal ranks of the true target item in the predicted ranking list, and the equation for MRR@ k calculation is given in equation 4.1.

$$\text{MRR}@k = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{1}{\text{rank}_i} \cdot \mathbb{I}[\text{rank}_i \leq k] \quad (4.1)$$

Here, S denotes the set of sessions contained in the test dataset, rank_i is the rank of the true item for session i , and $\mathbb{I}[\cdot]$ is the indicator function, which returns 1 if the true item appears within the top k predictions, and 0 otherwise.

Table 4.3: Overall results for LSTM and SR-GNN models.

ncoef	LSTM		SR-GNN	
	MRR@20	MRR@100	MRR@20	MRR@100
0	0.16421	0.16816	0.16968	0.17360
0.5	0.16488	0.16878	0.17090	0.17484
0.8	0.16489	0.16881	0.17104	0.17496
1	0.16506	0.16903	0.17144	0.17536
1.2	0.16485	0.16876	0.17131	0.17522

Table 4.3 presents the impact of the `ncoef` parameter on the overall performance of both the LSTM and SR-GNN model architectures. While the two models respond differently to changes in `ncoef`, a consistent upward trend can be observed as `ncoef` increases, especially in the SR-GNN setting. This suggests that incorporating neighbor feature enrichment contributes positively to model performance. While SR-GNN outperforms LSTM even without the proposed item representation enrichment, the performance gains become more pronounced when the enriched features are incorporated. This suggests that SR-GNN, with its graph-based structure and session-level attention mechanism, is more capable of leveraging globally informed item representations. As a result, the model not only achieves higher baseline performance but also exhibits more consistent and substantial improvements with the proposed method. In summary, the overall trend supports the effectiveness of

our proposed approach, indicating that the study successfully achieves its intended purpose. Additionally, when the `ncoef` value exceeds 1, its positive impact on performance starts to decrease. Therefore, it is not included in the remaining result tables. A value greater than 1 indicates that the aggregated influence of neighboring items outweighs the contribution of the current item’s own features, which can degrade the performance.

It is important to emphasize that the reported results are preliminary. Further improvements are likely achievable through more sophisticated feature extraction techniques and additional model fine-tuning.

Table 4.4: Performance for $K = 5$, $n = 1$ and $n = 2$, model: SR-GNN.

ncoef	$n = 1$		$n = 2$	
	MRR@20	MRR@100	MRR@20	MRR@100
0	0.04869	0.05100	0.02036	0.02193
0.5	0.05261	0.05449	0.03776	0.03840
0.8	0.05410	0.05588	0.04142	0.04203
1	0.05321	0.05500	0.0442	0.04498

Table 4.5: Performance for $K = 10$, $n = 1$ and $n = 2$, model: SR-GNN.

ncoef	$n = 1$		$n = 2$	
	MRR@20	MRR@100	MRR@20	MRR@100
0.0	0.08321	0.08670	0.03825	0.04072
0.5	0.08644	0.08993	0.05163	0.05389
0.8	0.08707	0.09054	0.05892	0.06107
1.0	0.08719	0.09066	0.06043	0.06268

Table 4.6: Performance for $K = 50$, $n = 1$ and $n = 2$, model: SR-GNN.

ncoef	$n = 1$		$n = 2$	
	MRR@20	MRR@100	MRR@20	MRR@100
0	0.11224	0.11545	0.05702	0.06090
0.5	0.11684	0.12016	0.06842	0.07104
0.8	0.11886	0.12199	0.08027	0.08281
1	0.11962	0.12281	0.08136	0.08405

The proposed method is evaluated under varying cold-item constraints, where an item is labeled *cold* if it appears in at most K sessions in the training set. Then, sessions in the test set with at least n cold items are labeled as cold sessions. Results are averaged over 3 random seeds. Experiments are conducted for n values of 1, 2, and K values of

5, 10, and 50. The values shown in the Table 4.4, 4.5, and 4.6 are obtained with the SR-GNN model.

- Higher `ncoef` generally improves MRR, validating its utility for cold items.
- When analyzing models on sessions containing 2 or more cold items, steeper performance gains are observed from `ncoef`, suggesting more improvement while dealing with sessions containing multiple cold items.

Table 4.7: Number of cold items and cold sessions under different cold-start conditions depending on values of K and n .

K	n	# Cold Items	# Cold Sessions
5	1	2257	540
	2	2257	60
10	1	2819	955
	2	2819	129
50	1	4458	6924
	2	4458	1877

Table 4.7 summarizes the experimental configurations and the corresponding sizes of the cold subsets used in our test scenarios. The column “# Cold Items” indicates how many unique items in the test set are treated as cold under the respective K threshold, while “# Cold Sessions” shows the number of sessions in which these cold items appear.

Figure 4.3 presents a scatter plot illustrating the MRR@20 and MRR@100 results across varying values of `ncoef`. The blue dots indicate MRR@20 scores, while the orange squares correspond to MRR@100 scores. The solid blue and orange lines represent the average performance over three different random seeds. As observed in the figure, there is a consistent improvement in performance as `ncoef` increases up to the value of 1. However, beyond this point, the impact of increasing `ncoef` becomes inconsistent and less dependable, with performance exhibiting fluctuations rather than a stable upward trend. Although the highest results are observed when `ncoef` equals 4 in the figure, this outcome is not consistent across seeds and thus cannot be considered reliable.

Figure 4.4 presents the MRR@20 and MRR@100 results for the LSTM model across three different random seeds and four different values of `ncoef`, where cold sessions

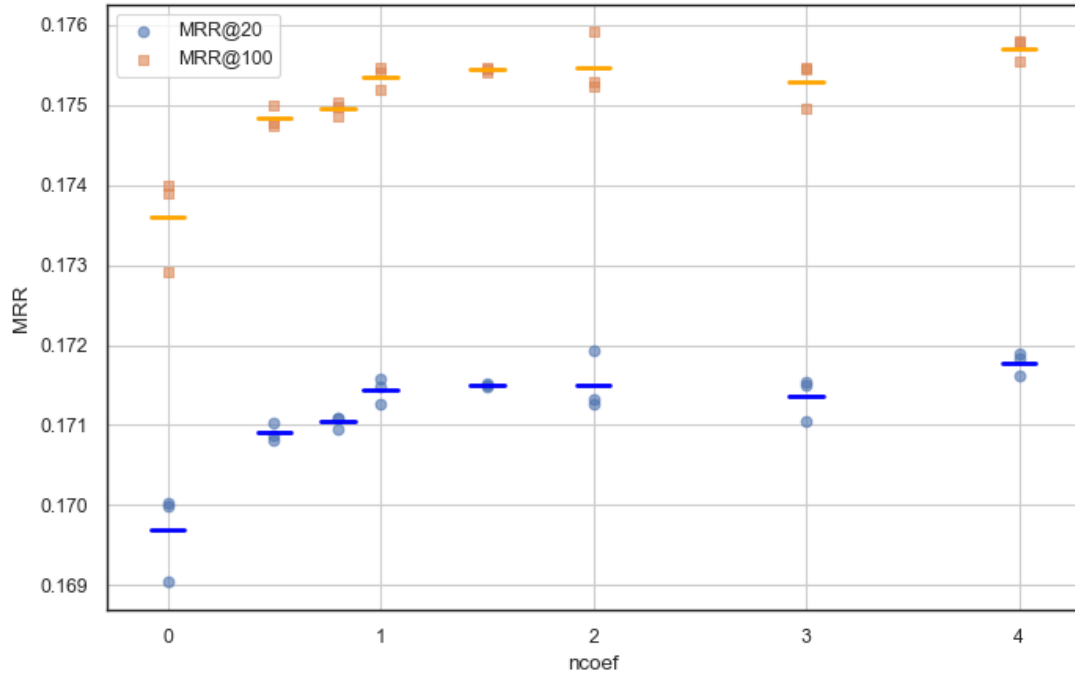


Figure 4.3: Overall MRR@20 and MRR@100 results of the SR-GNN model for `ncoef` values ranging from 0 to 4.

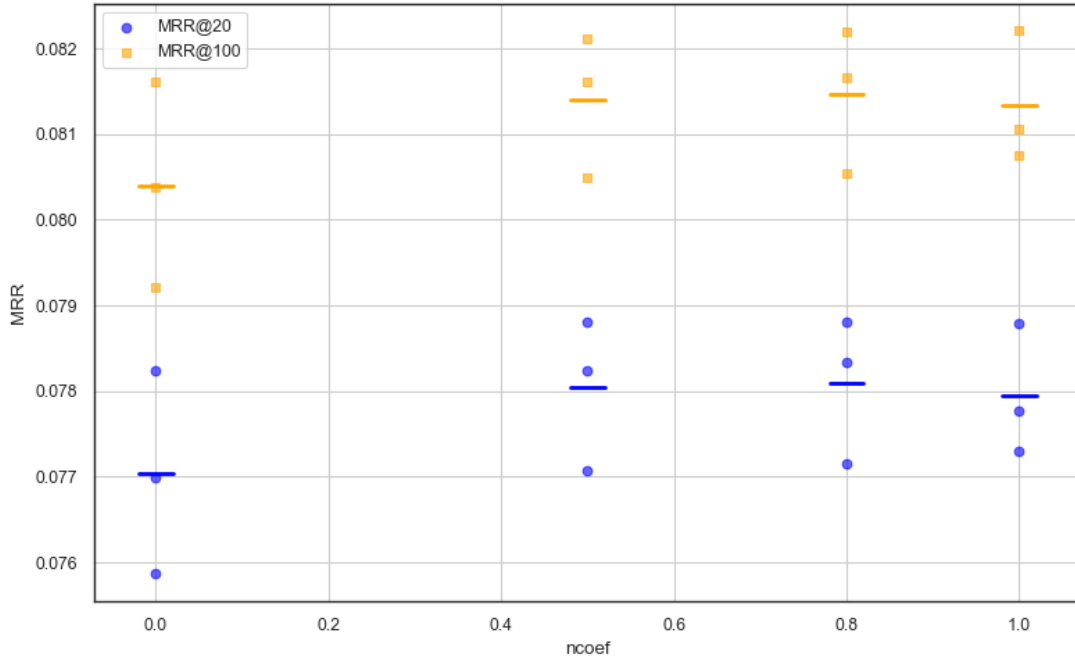


Figure 4.4: MRR@20 and MRR@100 results of the LSTM model for `ncoef` values between 0 and 1, with $K = 50$ and $n = 1$.

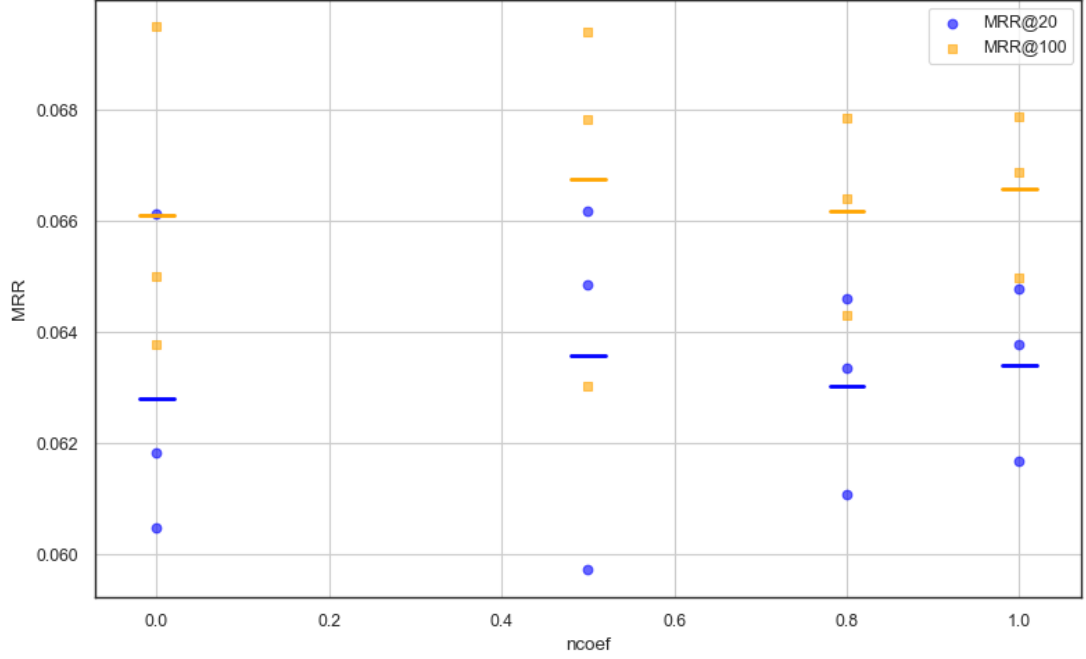


Figure 4.5: MRR@20 and MRR@100 results of the LSTM model for `ncoef` values between 0 and 1, with $K = 50$ and $n = 2$.

are defined with $K = 50$ and $n = 1$. Each result is plotted individually, with blue dots representing MRR@20 and orange squares indicating MRR@100 values. Similarly, Figure 4.5 shows the corresponding results for another cold session setting defined with $K = 50$ and $n = 2$. As can be observed from the figures, increasing `ncoef` continues to improve performance on cold sessions. However, the gains in MRR are less significant compared to those observed in the SR-GNN architecture.

In summary, the proposed feature enrichment method improves the overall performance of both the LSTM and SR-GNN models. The enhancements are particularly evident in cold session scenarios, with more pronounced gains as the number of cold items in a session increases, especially with higher values of the `ncoef` parameter. Notably, the impact of the proposed method is more substantial in the SR-GNN architecture, likely due to its ability to better leverage the enriched item representations through its graph-based structure.



5. CONCLUSION

In this work, we presented a feature enrichment strategy for SBRS, where item features are enhanced through graph-based aggregation. Our approach focuses on improving input representations via graph aggregation to boost overall performance and the performance in cold-start scenarios. It also provides a lightweight alternative to models that use GNNs to learn item-item relationships to represent item-item information.

Experimental results demonstrate that increasing `ncoef` improves the overall performance until the value of 1, across both LSTM and SR-GNN architectures. In addition to overall improvements, our method significantly boosts performance in sessions containing cold items, especially those with multiple cold items, showing its effectiveness in addressing cold-start challenges. These results validate the core idea of the study: that enriched features via the graph aggregation process can enhance item representations and lead to better recommendations, particularly under cold-start conditions. Furthermore, the findings indicate that lightweight graph-based enrichment of item features without using any graph-learning techniques can meaningfully improve session-based recommendation performance.

While the reported results are promising, there remains room for improvement through more advanced feature extraction techniques and further model-level optimizations. Additionally, in this study, the proposed method was evaluated only on the LSTM and SR-GNN architectures. Intuitively, it should positively impact the accuracy of any sequential recommendation model. Therefore, future work should explore integrating and testing the proposed approach within more sophisticated and diverse model architectures. Beyond architectural variations, it would also be valuable to investigate the effect of combining the proposed feature enrichment with self-supervised or contrastive learning techniques, which have shown promising results in related domains. Moreover, applying the method to larger and more diverse datasets, beyond the Dressipi dataset, could provide further insights into its generalizability.

and scalability. Finally, examining the interplay between enriched item features and user-level or temporal dynamics might unlock additional performance gains and open new research directions in session-based recommendation.



REFERENCES

- [1] **Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A. and Lian, D.** (2021). A survey on session-based recommender systems, *ACM Computing Surveys*, 54, 1–38.
- [2] **Twardowski, B.** (2016). Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks, *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16*, Association for Computing Machinery, New York, NY, USA, p.273–276, <https://doi.org/10.1145/2959100.2959162>.
- [3] **Sundermann, C.V., de Padua, R., Tonon, V.R., Marcacini, R.M., Domingues, M.A. and Rezende, S.O.** (2020). A context-aware recommender method based on text and opinion mining, *Expert Systems*, 37(6), e12618.
- [4] **Jannach, D., Lerche, L. and Jugovac, M.** (2015). Adaptation and Evaluation of Recommendations for Short-term Shopping Goals, *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15*, Association for Computing Machinery, New York, NY, USA, p.211–218, <https://doi.org/10.1145/2792838.2800176>.
- [5] **Sang, S., Yuan, W., Li, W., Yang, Z., Zhang, Z. and Liu, N.** (2023). Position-aware graph neural network for session-based recommendation, *Knowledge-Based Systems*, 262, 110201.
- [6] **Cao, W., Liu, Y., Cao, G. and He, Z.** (2022). Implicit user relationships across sessions enhanced graph for session-based recommendation, *Information Sciences*, 609, 1–14.
- [7] **Kurt, Z., Özkan, K., Bilge, A. and Gerek, O.** (2019). A similarity-inclusive link prediction based recommender system approach, *Elektronika IR Elektrotehnika*, 25(6).
- [8] **Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X. and Tan, T.** (2019). Session-Based Recommendation with Graph Neural Networks, 33, 346–353, <https://ojs.aaai.org/index.php/AAAI/article/view/3804>.
- [9] **Adomavicius, G., Mobasher, B., Ricci, F. and Tuzhilin, A.** (2011). Context-Aware Recommender Systems, *AI Magazine*, 32(3), 67–80, <https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v32i3.2364>, <https://onlinelibrary.wiley.com/doi/pdf/10.1609/aimag.v32i3.2364>.

- [10] **Herlocker, J.L., Konstan, J.A., Terveen, L.G. and Riedl, J.T.** (2004). Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems (TOIS)*, 22(1), 5–53.
- [11] **He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T.S.** (2017). Neural Collaborative Filtering, *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, p.173–182, <https://doi.org/10.1145/3038912.3052569>.
- [12] **Sarwar, B., Karypis, G., Konstan, J.A. and Riedl, J.** (2000). Application of dimensionality reduction in recommender system - a case study.
- [13] **Schafer, J., Frankowski, D., Herlocker, J. and Sen, S.** Collaborative filtering recommender systems, 291–324.
- [14] **Hu, Y., Koren, Y. and Volinsky, C.** (2008). Collaborative filtering for implicit feedback datasets, *2008 Eighth IEEE International Conference on Data Mining*.
- [15] **Koren, Y., Bell, R. and Volinsky, C.** (2009). Matrix factorization techniques for recommender systems, *Computer*, 42, 30–37.
- [16] **Widiyaningtyas, T. and Ariyanto, Y.** (2024). A systematic review of movie recommender systems, *ITEGAM- Journal of Engineering and Technology for Industrial Applications (ITEGAM-JETIA)*, 10.
- [17] **Rohani, V., Kasirun, Z. and Ratnavelu, K.** (2014). An enhanced content-based recommender system for academic social networks, 424–431.
- [18] **Li, L., Zhang, Z. and Zhang, S.** (2021). Hybrid algorithm based on content and collaborative filtering in recommendation system optimization and simulation, *Scientific Programming*, 2021, 1–11.
- [19] **Dwivedi, D.N. and Mahanty, G.** (2024). Leveraging personalized ai recommendations to enhance user experience in streaming services (ott platform), *Advances in Media, Entertainment, and the Arts*, 275–300.
- [20] **Yan, K.** (2023). A review of techniques used in e-commerce recommendation system, *Applied and Computational Engineering*, 4, 629–635.
- [21] **Singh, V.K., Mukherjee, M. and Mehta, G.K.** (2011). Combining collaborative filtering and sentiment classification for improved movie recommendations, *Lecture Notes in Computer Science*, 38–50.
- [22] **Pratama, D.E., Nurjanah, D. and Nurrahmi, H.** (2023). Tourism recommendation system using weighted hybrid method in bali island, *Jurnal Media Informatika Budidarma*, 7, 1189.
- [23] **Putra, K.R. and RACHMAN, M.A.** (2024). Perbandingan metode content-based, collaborative dan hybrid filtering pada sistem rekomendasi lagu, *MIND Journal*, 9, 179–193.

- [24] **Shani, G., Heckerman, D. and Brafman, R.I.** (2005). An MDP-based recommender system, *Journal of machine Learning research*, 6(Sep), 1265–1295.
- [25] **He, R. and McAuley, J.** (2016). Fusing similarity models with markov chains for sparse sequential recommendation, *2016 IEEE 16th international conference on data mining (ICDM)*, IEEE, pp.191–200.
- [26] **Quadrana, M., Karatzoglou, A., Hidasi, B. and Cremonesi, P.** (2017). Personalizing session-based recommendations with hierarchical recurrent neural networks, *Proceedings of the Eleventh ACM Conference on Recommender Systems*.
- [27] **Ludewig, M. and Jannach, D.** (2018). Evaluation of session-based recommendation algorithms, *User Modeling and User-Adapted Interaction*, 28, 331–390.
- [28] **Natarajan, N., Shin, D. and Dhillon, I.S.** (2013). Which app will you use next?, *Proceedings of the 7th ACM Conference on Recommender Systems*, 201–208.
- [29] **Tavakol, M. and Brefeld, U.** (2014). Factored mdps for detecting topics of user sessions, *Proceedings of the 8th ACM Conference on Recommender Systems*, 33–40.
- [30] **Schifferer, B., Liu, J., Rabhi, S., Titericz, G., Deotte, C., De Souza P. Moreira, G., Ak, R. and Onodera, K.** (2022). A Diverse Models Ensemble for Fashion Session-Based Recommendation, RecSysChallenge '22, Association for Computing Machinery, New York, NY, USA, p.10–17, <https://doi.org/10.1145/3556702.3556821>.
- [31] **Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T. and Ma, J.** (2017). Neural Attentive Session-based Recommendation, *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17*, Association for Computing Machinery, New York, NY, USA, p.1419–1428, <https://doi.org/10.1145/3132847.3132926>.
- [32] **Kumar, C. and Kumar, M.** (2024). Session-based recommendations with sequential context using attention-driven LSTM, *Computers and Electrical Engineering*, 115, 109138.
- [33] **Liu, Q., Zeng, Y., Mokhosi, R. and Zhang, H.** (2018). STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18*, Association for Computing Machinery, New York, NY, USA, p.1831–1839, <https://doi.org/10.1145/3219819.3219950>.
- [34] **Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L. and Tan, T.** (2020). TAGNN: Target Attentive Graph Neural Networks for Session-based Recommendation,

SIGIR '20, Association for Computing Machinery, New York, NY, USA, p.1921–1924, <https://doi.org/10.1145/3397271.3401319>.

- [35] **Lu, Y., Gao, Z., Cheng, Z., Sun, J., Brown, B., Yu, G., Wong, A., Perez, F. and Volkovs, M.** (2022). Session-based Recommendation with Transformers, *Proceedings of the Recommender Systems Challenge 2022*, RecSysChallenge '22, Association for Computing Machinery, New York, NY, USA, p.29–33, <https://doi.org/10.1145/3556702.3556844>.
- [36] **Steffen Rendle, Christoph Freudenthaler, Z.G.L.S.T.** (2010). Factorizing Personalized Markov Chains for Next-Basket Recommendation, *Proceedings of the 19th International Conference on World Wide Web (WWW)*, ACM, pp.811–820.
- [37] **Delianidi, M., Salampasis, M., Diamantaras, K., Siomos, T., Katsalis, A. and Karaveli, I.** (2021). A Graph-based Method for Session-based Recommendations, *Proceedings of the 24th Pan-Hellenic Conference on Informatics*, PCI '20, Association for Computing Machinery, New York, NY, USA, p.264–267, <https://doi.org/10.1145/3437120.3437321>.
- [38] **Esmeli, R., Bader-El-Den, M. and Abdullahi, H.** (2020). Session similarity based approach for alleviating cold-start session problem in e-commerce for top-n recommendations, *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*.
- [39] **Wang, X., He, X., Wang, M., Feng, F. and Chua, T.** (2019). Neural Graph Collaborative Filtering, *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 165–174.
- [40] **Kang, W. and McAuley, J.** (2018). Self-attentive sequential recommendation, *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206.
- [41] **Delianidi, M., Diamantaras, K., Tektonidis, D. and Salampasis, M.** (2023). Session-Based Recommendations for e-Commerce with Graph-Based Data Modeling, *Applied Sciences*, 13(1), <https://www.mdpi.com/2076-3417/13/1/394>.
- [42] **Ozbay, B., Tugay, R. and Oguducu, S.G.** (2024). A GNN Model with Adaptive Weights for Session-Based Recommendation Systems, *Proceedings of the 2024 9th International Conference on Machine Learning Technologies*, ICMLT '24, Association for Computing Machinery, New York, NY, USA, p.258–264, <https://doi.org/10.1145/3674029.3674070>.
- [43] **Chen, C., Guo, J. and Song, B.** (2021). Dual Attention Transfer in Session-based Recommendation with Multi-dimensional Integration, *Proceedings of the 44th International ACM SIGIR Conference on Research and Development*

in *Information Retrieval*, SIGIR '21, Association for Computing Machinery, New York, NY, USA, p.869–878, <https://doi.org/10.1145/3404835.3462866>.

- [44] **Xian, X., Fang, L. and Sun, S.** (2020). Regnn: a repeat aware graph neural network for session-based recommendations, *IEEE Access*, 8, 98518–98525.
- [45] **Wang, F., Gao, X., Chen, Z. and Lyu, L.** (2023). Contrastive multi-level graph neural networks for session-based recommendation, *IEEE Transactions on Multimedia*, 25, 9278–9289.
- [46] **Qiu, R., Huang, Z., Li, J. and Yin, H.** (2020). Exploiting Cross-session Information for Session-based Recommendation with Graph Neural Networks, *ACM Trans. Inf. Syst.*, 38(3), <https://doi.org/10.1145/3382764>.
- [47] **Zhang, J., Xue, R., Fan, W., Xu, X., Li, Q., Pei, J. and Liu, X.** (2024). *Linear-Time Graph Neural Networks for Scalable Recommendations*, <https://arxiv.org/abs/2402.13973>, 2402.13973.
- [48] **Purificato, A. and Silvestri, F.** (2024). *Eco-Aware Graph Neural Networks for Sustainable Recommendations*, <https://arxiv.org/abs/2410.09514>, 2410.09514.



CURRICULUM VITAE

Name SURNAME: Yunus KARATEPE

EDUCATION:

- **B.Sc.:** 2021, Yildiz Technical University, Faculty of Electrical and Electronics, Department of Computer Engineering
- **M.Sc.:** 2025, Istanbul Technical University, Faculty of Computer and Informatics Engineering, Department of Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2021-2022 Assistant Software Test Specialist at Testinium.
- 2022-2022 Software Engineer at Vestel.
- 2023-2025 Research Assistant at Yildiz Technical University, Computer Engineering.
- 2025-Current Research Assistant at Istanbul Technical University, Computer Engineering.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Karatepe, Y., Ögüdücü, Ş., (2025).** Enriching Item Representations in Session-Based Recommendation with Global Item Graphs. *International Conference on Machine Learning and Machine Intelligence*, July 25-27, 2025 Kyoto, Japan.