
GAZİ ÜNİVERSİTESİ
Gazi
1926

DEVELOPING AN E-COMMERCE RECOMMENDATION SYSTEM USING DEEP LEARNING AND OPTIMIZATION ALGORITHMS

Rand JAWAD KADHIM ALMAHMOOD

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER ENGINEERING DEPARTMENT

GAZI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MAY 2025

ETHICAL STATEMENT

I, at this moment, declare that in this thesis study, I prepared the thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information, and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct.
- All information, documents, assessments, and results have been presented in accordance with scientific ethical conduct and moral rules.
- All material used in this thesis that is not original to this work has been exhaustively cited and referenced,
- No change has been made in the data used,
- The work presented in this thesis is original,

Or else, I admit all loss of rights to be incurred against me.

Rand JAWAD KADHIM ALMAHMOOD

20/05/2025

DEVELOPING AN E-COMMERCE RECOMMENDATION SYSTEM USING DEEP LEARNING AND OPTIMIZATION ALGORITHMS

(Ph. D. Thesis)

Rand JAWAD KADHIM ALMAHMOOD

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

May 2025

ABSTRACT

The exponential growth of mobile applications and online e-commerce platforms has facilitated the accumulation of extensive user-generated data, incorporating product reviews and ratings. The user reviews heavily impact how people decide what to buy, brand reputation, and overall user satisfaction. However, existing rating prediction models often overlook individual user tendencies and the semantic richness embedded within review texts. To address these limitations, this study introduces two complementary deep learning- based frameworks to enhance the identification and recommendation of helpful reviews. The first proposed model, Balanced Helpful Review Quantification Using User Tendencies (BHRQUT), incorporates personalised rating behaviours and a novel auxiliary feature derived from actual and predicted ratings. This feature is further integrated with text-based inputs processed through a balanced CNN-BiLSTM architecture, leveraging word embeddings and natural language processing techniques. Experimental evaluations on Amazon 5-score domain datasets reveal that BHRQUT achieves an accuracy of up to 97%, significantly outperforming traditional models in capturing user preferences and identifying helpful content. The second model, Pre-trained Deep Helpful Review Recommendation Framework (PRE-DHRRF), applies transfer learning on large-scale review datasets (Amazon and Yelp) to better understand domain-specific language patterns, informal expressions, and contextual semantics. This framework combines a tendency-based collaborative filtering algorithm with data augmentation strategies to address data sparsity and class imbalance issues. Results show that PRE-DHRRF attains an accuracy of 98% with a minimal generalisation gap of 0.3%, demonstrating its robustness in detecting helpful reviews across diverse datasets. Together, these models contribute to advancing the state of review helpfulness prediction by addressing user behavioural tendencies and linguistic nuances in review texts. The results highlight the practical significance of the proposed systems in improving e-commerce recommendation platforms, enabling better decision-making, and enhancing user experience through smart filtering of valuable information.

Science Code	:	92432
Key Words:	:	E-commerce, balanced deep recommender systems, differences in users' opinions, pre-trained model, review analysis.
Page Number	:	121
Supervisor	:	Assoc. Prof. Dr. Adem TEKEREK

DERİN ÖRENME VE OPTİMİZASYON ALGORİTMALARI KULLANILARAK E-TİCARET ÖNERİ SİSTEMİ GELİŞTİRİLMESİ

(Doktora Tezi)

Rand JAWAD KADHIM ALMAHMOOD

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Mayıs 2025

ÖZET

Mobil uygulamaların ve çevrimiçi e-ticaret platformlarının hızlı büyümesi, kullanıcı tarafından oluşturulan ürün yorumları ve puanlamalar gibi büyük miktarda verinin toplanmasını mümkün kılmıştır. Bu yorumlar, tüketici kararlarının şekillenmesinde, marka itibarında ve genel kullanıcı memnuniyetinde kritik bir rol oynamaktadır. Ancak mevcut puanlama tahmin modelleri, genellikle bireysel kullanıcı eğilimlerini ve yorum metinlerinde yer alan anlamsal zenginliği göz ardı etmektedir. Bu sınırlamaları aşmak amacıyla, bu çalışma, faydalı yorumların tanımlanması ve önerilmesini geliştirmeyi hedefleyen iki tamamlayıcı derin öğrenme tabanlı çerçeve sunmaktadır. İlk önerilen model olan Kişisel Eğilimlere Dayalı Dengeli Faydalı Yorum Ölçümleme (BHRQUT) modeli, kullanıcıların kişiselleştirilmiş puanlama davranışlarını ve gerçek ile tahmin edilen puanlamalardan türetilmiş yeni bir yardımcı özelliği bünyesinde barındırmaktadır. Bu özellik, kelime gömme (embedding) ve doğal dil işleme teknikleri kullanılarak dengeli bir CNN-BiLSTM mimarisi ile işlenen metin tabanlı girdilerle bütünleştirilmiştir. Amazon 5 dereceli puanlama verisetleri üzerinde yapılan deneysel değerlendirmeler, BHRQUT modelinin %97'ye varan doğruluk oranı ile geleneksel modellere kıyasla kullanıcı tercihlerini yakalama ve faydalı içeriği tanımlama konusunda önemli başarılar elde ettiğini göstermektedir. İkinci model olan Ön Eğitilmiş Derin Faydalı Yorum Öneri Çerçevesi (PRE-DHRRF), Amazon ve Yelp gibi büyük ölçekli yorum verisetleri üzerinde transfer öğrenme uygulayarak, alanına özgü dil kalıplarını, gayriresmî ifadeleri ve bağlamsal anlamları daha iyi kavramayı amaçlamaktadır. Bu çerçeve, eğilim tabanlı işbirlikçi filtreleme algoritmasını veri artırma stratejileriyle birleştirerek veri seyrekliği ve sınıf dengesizliği gibi sorunları ele almaktadır. Deneysel sonuçlar, PRE-DHRRF modelinin %98 doğruluk oranı ve yalnızca %0.3'lük çok düşük bir genelleme farkı ile, farklı verisetlerinde faydalı yorumları tespit etme konusunda yüksek bir dayanıklılık sergilediğini ortaya koymaktadır. Bu iki model birlikte, hem kullanıcı davranış eğilimlerini hem de yorum metinlerindeki dilsel nüansları dikkate alarak faydalı yorum tahmini alanında önemli bir ilerleme sağlamaktadır. Bulgular, önerilen sistemlerin e-ticaret öneri platformlarını geliştirme, bilinçli karar alma süreçlerini destekleme ve değerli içeriklerin akıllı filtrelenmesi yoluyla kullanıcı deneyimini artırma konusundaki pratik değerini vurgulamaktadır.

1. Giriş

E-ticaretin yükselişi ve çevrimiçi platformların çoğalması, tüketiciler tarafından paylaşılan yanıltıcı ve güvenilmez ürün incelemeleri ile puanlamaların yayılmasını kolaylaştırmış ve bu durum deepfake içeriklerin ortaya çıkmasına zemin hazırlamıştır. Bu giderek büyüyen sorun, bireylerin bilinçli satın alma kararları almakta zorlanmaları nedeniyle kullanıcılar arasında kaygı ve güvensizlik yaratabilmektedir. Gelişmiş teknolojilerin hâkim olduğu günümüzde, gerçek ve yanıltıcı bilgiyi ayırt edebilen sağlam sistemlerin uygulanması büyük önem taşımaktadır. Derin öğrenmeye dayalı inceleme analiz algoritmaları, güçlü ayrıştırma yetenekleri ve değerli incelemeleri tespit etme konusundaki etkinlikleri nedeniyle popülerlik kazanmıştır. Ancak, mevcut birçok derin öğrenme yaklaşımı hâlâ yararlı incelemeleri doğru bir şekilde tanımlamakta zorluk çekmektedir. E-ticaret platformları, öneri hizmetlerini geliştirmek için faydalı içeriği filtreleme ve önceliklendirme konusunda önemli zorluklarla karşı karşıyadır. Bu platformlar, tüketici davranışlarını şekillendirmek amacıyla büyük ölçüde kullanıcı kaynaklı incelemelere dayansa da, tüm incelemeler karar verme sürecine eşit derecede katkı sağlamaz. Kullanıcı geri bildirimlerinde bulunan büyük veri hacmi, dilsel çeşitlilik ve önyargılar gibi faktörler nedeniyle yararlı incelemeleri belirlemek ve vurgulamak karmaşık bir görev olmaya devam etmektedir.

E-ticaret, elektronik cihazlar kullanılarak ürünlerin çevrimiçi pazarlanması ve satılmasını ifade eder. Üç ana eğilime ayrılabilir: İşletmeden İşletmeye (B2B), İşletmeden Tüketiciye (B2C) ve Tüketiciden Tüketiciye (C2C) [1]. E-ticaret pazarlamasının başarısı büyük ölçüde alışveriş deneyimini bireysel müşterilere göre uyarlayan öneri sistemlerinden kaynaklanmaktadır. E-ticaret için bir öneri sistemi geliştirmek, platformların müşterilere doğru zamanda en uygun bilgileri sunmasını sağlayarak müşteri sadakatini artırmakta ve daha akıcı bir alışveriş deneyimi oluşturmaktadır. Simgeler, tercihlerini, gezinme geçmişlerini veya satın alma davranışlarını girdi verisi olarak sunan çok sayıda kullanıcıyı sembolize eder. Katman 2 olan Öneri Sistemi ve Öneriler, giriş verilerini işlemek için algoritmalar (örneğin işbirlikçi veya içerik tabanlı filtreleme) kullanan öneri sistemini içerir. İçerik tabanlı öneri sistemleri, öğelerin özelliklerine dayanır. Kullanıcının daha önce seçtiği ögeyle benzer özellikler taşıyan içerikleri önerir. İşbirlikçi filtreleme (Collaborative Filtering - CF) olarak bilinen öneri algoritmaları ise kullanıcılar ve nesneler arasındaki benzerliklere dayanır. CF'ye göre, daha önce aynı öğeyi seçip birbirine öneren iki kullanıcı benzer kabul edilir [2].

Öneri çıktısı, her kullanıcıya özel olarak hazırlanmış ürün önerilerini temsil eden bir listedir. Katman 3 olan Çıktı Katmanı, kullanıcının kişiselleştirilmiş önerileri almasını temsil eder. Tek kullanıcı simgesi, sistemin önerilerine dayanan benzersiz ve kişiselleştirilmiş bir alışveriş deneyimini simgeler.

Öneri sistemleri, bilgi türlerine göre genellikle işbirlikçi filtreleme ve içerik tabanlı yaklaşımlar olarak sınıflandırılmaktadır [3]. İşbirlikçi filtreleme, kullanıcıların geçmiş etkileşimlerini ve benzerliklerini analiz eder [4]; ancak "soğuk başlangıç" problemi gibi sınırlamaları vardır. Bu çalışmada, kullanıcı ve ürün puanlama önyargılarını dikkate alan sadeleştirilmiş eğilim tabanlı bir model ile daha doğru tahminler yapılması amaçlanmıştır [5].

Yararlı inceleme sınıflandırmasında dengesiz veri dağılımını ele almak için aşırı örnekleme tabanlı bir veri artırma yöntemi uygulanmıştır. Özellikle az temsil edilen "yararsız" incelemeler, eşanlamlı sözcüklerle zenginleştirilerek modelin genelleme gücü artırılmıştır. Çalışma, veri yetersizliği, soğuk başlangıç ve aşırı öğrenme gibi temel zorlukları ele alarak gerçek dünya e-ticaret verilerine dayalı bir öneri sistemi geliştirmektedir.

Amazon'dan Video Oyunları, Müzik Aletleri, Kitaplar-5, Giyim-Ayakkabı-Takı; Yelp'ten ise Restoranlar, Alışveriş ve Güzellik & Spa kategorileri seçilmiştir. Bu seçim, dilsel çeşitlilik ve tüketici davranışlarının geniş bir yelpazesini kapsayarak sonuçların genellenebilirliğini artırmaktadır.

Araştırma kapsamında, BHRQUT ve PRE-DHRRF olmak üzere iki derin öğrenme modeli önerilmiştir. NVIDIA TITAN XP grafik kartları desteğiyle, doğal dil işleme ve metin sınıflandırma uygulamaları için modeller Amazon ve Yelp verileri üzerinde eğitilmiştir. BHRQUT modeli dengelenmiş CNN-BiLSTM yapısını; PRE-DHRRF modeli ise bu mimariyi önceden eğitilmiş katmanlarla genişletmektedir. Modellerin etkinliği karşılaştırmalı analizlerle değerlendirilmiş; veri dengesi, ön eğitim ve eğilim modellemesinin katkıları detaylı biçimde incelenmiştir.

Bu araştırmanın temel katkıları aşağıda özetlenmiştir:

- Yelp ve Amazon gibi popüler e-ticaret sitelerinden alınan inceleme veri kümelerinin özelliklerine dayalı olarak derin öğrenme modelleri aracılığıyla bilgi içeren geri bildirimlerin sınıflandırılmasına yönelik bir çözüm sunulmuştur. Dört farklı Amazon inceleme veri kümesi ve bir Yelp Akademik İnceleme veri kümesi kullanılarak değerlendirmeler yapılmıştır. Önerilen yöntem; doğruluk, kesinlik, duyarlılık, F1 skoru, yanlış sınıflandırma oranı (MAE, RMSE), karışıklık matrisi ve genelleme farkı açısından mevcut yöntemlerden daha iyi sonuçlar elde etmiştir.
- Amazon inceleme veri setinde, faydalılık oranının yıldız puanlaması (3,4) üzerindeki etkisi TCF ile ve TCF olmadan olmak üzere dört farklı senaryoda incelenmiştir.
- İlk model (BHRQUT) için giriş cümle uzunluğunun performansa etkisi analiz edilmiş; ardından, en yüksek doğruluğa sahip Amazon veri kümesi kategorisi seçilmiş ve Yelp veri kümesiyle birlikte ikinci modelde (PRE-DHRRF) test edilmiştir.
- İşbirlikçi filtreleme algoritmalarına dayalı birçok öneri modelinin temel zorluklarından biri olan seyrek veri (sparsity) problemi ele alınmıştır. Bu sorunu aşmak için eğilim tabanlı işbirlikçi filtreleme yöntemleri ile transdüser tabanlı önceden eğitilmiş modeller birleştirilmiş; bu sayede sınırlı veriye aşırı öğrenme engellenmiş ve bilinmeyen örnekler üzerindeki performans iyileştirilmiştir.
- Bu çalışma, derinlemesine ve uygulanabilir inceleme önerileri sunan önceden eğitilmiş bir çerçeve ortaya koymaktadır. Bu çerçeve, sınıflandırma sonuçlarını geliştirmek amacıyla iki yönlü bir mimari kullanır. Birinci yol, veri kümesinin özelliklerine ve sınıflandırma hedeflerine uyarlanmış özel bir CNN-BiLSTM modeli içerir. İkinci yol ise, metin dizilerini işleme konusunda üstün performans gösteren önceden eğitilmiş bir transdüser modeli (Roberta-base) içermektedir. Bu model önce belirli bir veri kümesinde eğitilmekte, ardından ağırlıkları diğer veri kümelerinde yeniden kullanılmaktadır. Bu iki yönlü yapı, özel modelin yüksek performansı ile genel modelin genelleme yeteneğini birleştirerek çerçevenin uyarlanabilirliğini ve doğruluğunu artırmaktadır.

Bu genişletilmiş özetin geri kalan bölümleri şu şekilde yapılandırılmıştır:

Bölüm 2, kullanılan yöntemleri, veri kaynaklarını, deneysel düzeni ve araştırma sorularını yanıtlamada kullanılan analiz tekniklerini ayrıntılı olarak açıklamaktadır. Bölüm 3, elde edilen temel bulguları ilgili şekil ve tablolarla destekleyerek sunmakta ve mevcut literatürle ilişkili olarak kritik bir tartışma sunmaktadır. Son olarak, Bölüm 4 çalışmanın temel katkılarını özetleyerek gelecek araştırmalara yönelik önerilerde bulunmaktadır. Bu yapısal

yaklaşım, araştırmanın kavramsallaştırmadan sonuçlara kadar olan sürecinin açık ve tutarlı bir şekilde takip edilmesini sağlamayı amaçlamaktadır.

2. Metodoloji

2.1. Veri Kümesi Açıklaması

Son yıllarda büyük ölçekli veri kümeleri, farklı alanlardaki araştırmaların ilerlemesinde önemli bir rol oynamaktadır. Bu bağlamda Amazon ve Yelp inceleme veri kümeleri, derin öğrenme modelleri ile öneri sistemlerinin geliştirilmesinde yaygın olarak kullanılmaktadır. Kullanıcı yorumları, ürün metaverileri ve satın alma davranışları gibi bilgiler sayesinde bu veri kümeleri; duygu analizi, öneri sistemleri ve tüketici davranışı modellemesi gibi birçok alanda uygulanabilirlik sunmaktadır. Bu çalışmada önerilen model, iki vaka çalışması kapsamında eğitilmiştir.

Birinci vaka çalışmasında, Amazon İnceleme Verisi (2018) veri kümesinden dört kategori seçilmiştir. "5-core" olarak bilinen bu alt küme, kullanıcıların en az beş ürünü değerlendirdiği ve her ürünün en az beş kullanıcı tarafından değerlendirildiği örneklerden oluşmaktadır. İki küçük örneklemli kategori (Video Oyunları – C1, Müzik Aletleri – C2) tüm verileriyle kullanılırken; iki büyük örneklemli kategori (Kitaplar-5 – C3, Giyim-Ayakkabı-Takı – C4) için her birinden 1.000.000 inceleme alınmıştır. Amazon veri kümesi 29 kategori, 75 milyonun üzerinde yorum ve yaklaşık 346 milyon puan içermektedir.

İkinci vaka çalışmasında iki veri türü kullanılmıştır. İlk türde, Amazon'dan seçilen iki kategori (DS1: Müzik Aletleri, DS2: Giyim-Ayakkabı-Takı) yer almaktadır. İkinci tür ise Yelp Akademik İnceleme Verisi'dir (DS3). Yelp verisi 6.990.280 inceleme içermekte olup; işletme bilgileri, kullanıcı yorumları, kullanıcı profilleri, müşteri giriş kayıtları, kısa ipuçları ve görsel metaveriler gibi çeşitli bileşenler barındırmaktadır. Bu çalışmada, DS1'in tamamı (231.392 yorum), DS2'den 1.000.000 ve DS3'ten 200.000 yorum analiz edilmiştir. Tablo 2.1, her kategoriye ait toplam 5 yıldızlı yorum sayısını ve çalışmada kullanılan alt kümeleri göstermektedir:

Çizelge 2.1. Amazon 5 yıldız puanlı ve Yelp inceleme veri kümeleri, yorum sayıları

Kategori	5 Yıldızlı Yorum Sayısı	Önerilen Modelde Kullanılan 5 Yıldızlı Yorum Sayısı
Kitaplar-5 (Books-5)	27,164,983	1000000
Giyim, Ayakkabı ve Takı (Clothing...)	11,285,464	1000000
Müzik Aletleri (Musical Instruments-5)	231,392	231,392
Video Oyunları (Video Games-5)	497,577	497,577
Yelp İnceleme Verisi	6,990,280	200,000

2.2. Veri Ön İşleme Teknikleri

Bu çalışmada kullanılan veriler, Amazon ve Yelp inceleme veri kümelerinden elde edilmiştir. Her iki veri kümesi sekiz temel ön işleme adımına tabi tutulmuştur:

1. Hedef özelliklerin filtrelenmesi: Yelp veri kümesinden user_id, item_id, rating ve yorum metni; Amazon veri kümesinden ise user_id, item_id, rating ve yorum özeti alanları seçilmiştir.
2. Eksik değerlerin temizlenmesi: Seçilen alanlarda eksik değerlere sahip kayıtlar veri kümesinden çıkarılmıştır.
3. Tahmin edilen puanların hesaplanması: Veri seyrekliği problemini azaltmak için eğilim tabanlı işbirlikçi filtreleme (CF) yaklaşımı kullanılmıştır. Kullanıcı (Ut) ve ürün eğilimleri (It), kullanıcı-ürün puanları üzerinden türetilmiş; KNNBasic algoritması ile benzer kullanıcılar veya ürünler arasındaki ilişkilerden tahmini puanlar (PR1–PR4) hesaplanmıştır. Bu tahminler, aşağıdaki formüller ile belirlenmiştir:

Kullanıcı eğilimi:

$$Ut = (\sum(Uui - Viu)) / |I.u| \quad (2.1)$$

Ürün eğilimi:

$$It = (\sum(Uui - Vui)) / |U.i| \quad (2.2)$$

Tahmin puanları (koşullara göre):

$$PR1 = \max(Vui + It, Viu + Ut) \quad (2.3)$$

$$PR2 = \min(Vui + It, Viu + Ut) \quad (2.4)$$

$$PR3 = \min(\max(Vui, (Viu + Ut)\cdot\sigma) + (Vui + It)\cdot(1 - \sigma), Viu) \quad (2.5)$$

$$PR4 = Viu\cdot\sigma + Vui\cdot(1 - \sigma) \quad (2.6)$$

Bu yöntem, seyrek verilerde daha iyi temsil gücüne sahip öznitelikler üretmeyi amaçlamaktadır.

4. Gerçek puanların optimize edilmesi: Ortalama puanlar hesaplanarak derecelendirme değerleri optimize edilmiştir.
5. Yüksek değerli özelliklerin oluşturulması: Faydalı (1) ve faydasız (0) yorumlar, orijinal ya da tahmin edilen puanlara dayalı olarak etiketlenmiştir.
6. Sınıf dengesinin sağlanması: Faydalı (1) ve faydasız (0) örneklerin dengesini sağlamak amacıyla azınlık sınıfına ait örnekler, SbURA (Eşanlamlı Tabanlı Faydasız Yorum Artırımı) algoritması ile çoğaltılmıştır. SbURA, cümlelerdeki kelimeleri eşanlamlılarıyla değiştirerek veri çeşitliliğini artırır.
7. Önişlenmiş yorum metni ve etiketler: Amazon ve Yelp veri kümelerine ait yorumlar; stopword temizleme, kök bulma (stemming) ve tokenizasyon işlemlerinden geçirilmiştir. Bu adımlar geleneksel doğal dil işleme (NLP) süreçlerine özgü olup, özellikle Balanced CNN-BiLSTM modelinde uygulanmıştır.

- Geleneksel NLP Yöntemleri: nltk kütüphanesinden stopword'ler çıkarılmış, PorterStemmer() ile kök bulma yapılmış, keras.preprocessing.text.Tokenizer ile metinler tokenize edilmiştir.
- Dönüştürücü Tabanlı Yöntemler (Transformers): RoBERTa gibi modeller için özelleştirilmiş tokenizasyon işlemleri kullanılmıştır. RoBERTa, Byte-Pair Encoding (BPE) ile alt kelime düzeyinde ayrıştırma yapar. Girdi metni, modelin eğitildiği sözlük ile uyumlu şekilde özel sembollerle (örneğin <s>, </s>) işlenir. Bu işlem Hugging Face transformers kütüphanesi ile entegre bir şekilde yürütülmüştür.

2.3. Vaka Çalışmalarında Kullanılan Derin Öğrenme Modelleri

2.3.1. Vaka çalışması 1: BHRQUT modeli

Bu bölümde, kullanıcı eğilimlerinin nicelenmesine dayalı dengeli yararlı öneri modeli olan BHRQUT modelinin genel yapısı gösterilmektedir. Modelin değerlendirilmesi iki senaryo üzerinden gerçekleştirilmiştir.

Birinci senaryoda (Vaka 1), TCF algoritması, veri artırımı, CNN ve BiLSTM modellerinin avantajları birleştirilmiştir. Bu süreçte, Surprise kütüphanesindeki Tendencies sınıfı kullanılarak kullanıcı-ürün eğilim farkları üzerinden yeniden derecelendirme yapılmış ve veri iki alt senaryoya ayrılmıştır:

Vaka 1_1: Ortalama puan (Ort = (Gerçek Puan + Tahmin Puanı)/2) ≥ 3 olan yorumlar, Ort ≥ 0.3 ise “yararlı”, Ort ≤ 0.2 ise “yararsız” olarak etiketlenir.

Vaka 1_2: Daha sıkı bir eşik uygulanır; Ort ≥ 4 için, Ort ≥ 0.4 “yararlı”, Ort ≤ 0.3 “yararsız” olarak sınıflandırılır [83].

İkinci senaryoda (Vaka 2) ise TCF algoritması kullanılmaz; yalnızca gerçek puanlara dayalı filtreleme yapılır ve yine iki alt senaryoya ayrılır:

Vaka 2_1: ACR ≥ 0.3 “yararlı”, ACR ≤ 0.2 “yararsız” [84].

Vaka 2_2: Daha katı eşiklerle ACR ≥ 0.4 “yararlı”, ACR ≤ 0.3 “yararsız” olarak değerlendirilir.

Model, tahmin edilen puanlar ve gerçek puanlara dayalı olarak her bir yoruma ikili etiket atar. Ardından, SbURA (Eş Anlamlı Tabanlı Yararsız Yorum Artırımı) gibi veri artırma yöntemleri ile veri dengelenir. Veri ön işleme süreci geleneksel NLP teknikleriyle gerçekleştirilir. Bu temsiller kullanılarak CNN-BiLSTM modeli, 5 katlı çapraz doğrulama ile 20 epoch boyunca eğitilir. Model yalnızca kullanıcı puanları ve yazılı yorumlara odaklanır. Başarı değerlendirmesi; RMSE, MSE, MAE, doğruluk, geri çağırma, F1-skoru, kesinlik ve karmaşıklık matrisi gibi metriklerle yapılır. Ön işleme adımları ayrıntılı şekilde açıklanmış olup, ardından ağ mimarisi ve değerlendirme süreçleri sunulmuştur.

2.3.2. Vaka çalışması 2: PRE-DHRRF modeli

Bu vaka çalışması, müşteri yorumlarının analizinde başarıyı artırmak amacıyla veri ön işleme, örneklem dengesi sağlama ve transfer öğrenmesini bir araya getiren PRE-DHRRF modelinin katkılarını ortaya koymaktadır. Modelin ilk adımı, eğilim temelli işbirlikçi filtreleme algoritması, veri artırımı yoluyla dengeleme ve doğal dil işleme (NLP) tekniklerini içeren kapsamlı bir ön işleme sürecini kapsamaktadır.

İkinci aşamada, Amazon ve Yelp veri kümeleri üzerinde, daha önce eğitilmiş ve ağırlıkları kaydedilmiş iki model (Balanced CNN-BiLSTM ve RoBERTa-base) kullanılarak transfer öğrenme gerçekleştirilmiştir. Eğitim süreci, 20 epoch boyunca 5 katlı çapraz doğrulama ile yürütülmüştür.

Üçüncü aşamada ise model performansı doğrulanmış ve hiperparametreler optimize edilmiştir. Değerlendirme sürecinde; genelleme farkı, doğruluk, kesinlik, geri çağırma, F1 skoru, RMSE ve MAE gibi metrikler kullanılmıştır. Bu süreçteki tüm adımlar detaylı olarak açıklanmıştır.

Üst bileşen, giriş veri kümesinin ön işleme adımlarını; alt bileşen ise her biri 5 katlı çapraz doğrulama ile doğrulanan iki farklı önceden eğitilmiş modelden oluşmaktadır. Sol taraftaki model, ince ayar yapılarak Önceden Eğitilmiş Model I (Balanced CNN-BiLSTM) ile yorum sınıflandırması için; sağ taraftaki model ise Önceden Eğitilmiş Model II (RoBERTa-base) ile yorum sınıflandırması için kullanılmıştır.

2.4. Vaka Çalışması 1'in Tekrarlanabilir Sonuçları (BHRQUT Modeli)

Bu çalışmada, Amazon Review-5 veri setinden dört kategori (Video Games, Musical Instruments, Books-5 ve Clothing, Shoes, and Jewelry) seçilerek Case Study 1 uygulanmıştır. Veri seti; kullanıcı ve ürün kimlikleri, değerlendirme puanları, inceleme metinleri, ürün özetleri ve tarih gibi çeşitli öznitelikler içermektedir. Yardımcılık etiketleri (0/1), kullanıcı eğilimlerine dayalı işbirlikçi filtreleme algoritması ile hesaplanan ortalama puanlara göre belirlenmiştir.

Önişleme adımlarında doğal dil işleme teknikleri kullanılmış ve kelimeler 50 boyutlu gömme vektörlerine dönüştürülmüştür. En sık geçen 50.000 kelime içeren sözlükte her

kelimeye özgü bir indeks atanmış ve bu gömme vektörleri modelin giriş katmanında kullanılmıştır. Metin dizileri, kategoriye göre farklı uzunluklarda olup maksimum 44 token'a kadar uzanmaktadır.

Model, 2×2 çekirdek boyutuna sahip dört konvolüsyon katmanı (Conv1D) içermektedir. İlk katmanda 100 filtre, sonraki katmanlarda ise 64 filtre kullanılarak hem düşük hem yüksek düzeyli öznitelikler çıkarılmıştır. 1×1 çekirdekli max-pooling katmanı ile boyut indirgeme sağlanmıştır. Ardından, 200 gizli birimli çift yönlü LSTM (BiLSTM) katmanı kullanılarak dizisel bağlam bilgisi modellenmiştir. Modelin genelleme kapasitesini artırmak amacıyla her katmandan sonra %50 oranında dropout uygulanmış ve aktivasyon fonksiyonu olarak ReLU tercih edilmiştir.

Model eğitimi, 20 epoch boyunca 5 katlı çapraz doğrulama yöntemi (KFold) ile gerçekleştirilmiştir. Her bir katlamada verinin %80'i eğitim, %10'u doğrulama ve %10'u test için ayrılmıştır. Optimizasyon için Adam algoritması, hata fonksiyonu olarak ise Ortalama Kare Hata (MSE) kullanılmıştır. Çıkış katmanında sigmoid aktivasyon fonksiyonuna sahip tam bağlantılı (dense) bir yapı yer almakta olup, incelemelerin "faydalı" veya "faydasız" olarak ikili sınıflandırmasını sağlamaktadır.

2.5. Vaka Çalışması 2'nin (PRE-DHRRF) Tekrarlanabilir Sonuçları

Bu vaka çalışması, önerilen modeli Amazon İnceleme Veri Kümesi'nin iki alt kümesi, Müzik Aletleri (DS1) ve Giyim, Ayakkabı ve Takı (DS2),ile birlikte, e-ticaret araştırmalarında yaygın olarak kullanılan Yelp Akademik İnceleme Veri Kümesi üzerinde değerlendirir. Değerlendirme sürecinde, ürün puanı, ürün kimliği (ASIN), kullanıcı kimliği ve inceleme özeti gibi temel öznitelikler filtrelenmiştir. Önerilen yapı, hem faydalı hem de faydasız kullanıcı yorumlarını analiz edebilen iki ayrı önceden eğitilmiş model üretmeyi amaçlamaktadır.

Ayrıca çalışma, Yelp veri kümesinden İnceleme Verisi Dosyasını (DS3) da içermektedir. Bu veri kümesinde inceleme kimliği, kullanıcı kimliği, işletme kimliği, yıldız puanı (1–5), inceleme metni ve tarih gibi öznitelikler yer alır. Analizde kullanılan öznitelikler inceleme kimliği, işletme kimliği, yıldız puanı ve inceleme metnidir. Tablo 2.2, Amazon ve Yelp veri

kümelerinden alınan öznitelik bilgilerini örneklerle sunarak yapılandırılmış bir karşılaştırmalı analiz sağlar.

Çizelge 2.2. Amazon inceleme veri kümesi ve Yelp akademik inceleme veri kümesinden örnekler

Amazon review dataset		Yelp academic review dataset	
Features	Values	Features	Values
overall	4	review_id	BiTunyQ73aT9WBnpR9DZGw
vote	2.0	user_id	OyoGAe7OKpv6SyGZT5g77Q
verified	True	business_id	7ATYjTIgM3jUlt4UM3IypQ
Review time	11 14, 2011	stars	5
Reviewer_id	A18ZXVF5AQFK35	useful	1
asin	0151004714	funny	0
style	{'style:': ' Straight to Straight'}	cool	1
Reviewer name	meep	text	I've taken many spin classes over the year...
Review text	Meep bought several for the home studio. They are nicel...	date	2012-01-03 15:28:18
Summary	very nice cables		
Unix review time	1321228800		
image	NAN		

Önceden eğitilmiş modeller, metin sınıflandırması ve diğer doğal dil işleme (NLP) görevlerinde yüksek doğruluk ve sağlamlıklarıyla bilinmektedir. Büyük ölçekli veri kümeleriyle eğitildikleri için genelleme yetenekleri yüksektir ve fazla öğrenmeyi azaltırlar. Bu da onları öneri sistemleri araştırmalarında değerli kılar. Etkinliklerinin temelinde, görev özelinde ince ayar yapılması (fine-tuning) yer alır.

Bu çalışmada, iki önceden eğitilmiş modelin ağırlıkları kullanılmıştır: Vaka Çalışması 1'de geliştirilen özel Balanced CNN-BiLSTM modeli ve standart RoBERTa-base modeli. CNN-BiLSTM modeli, sıralı ve mekânsal özellikleri çıkarmak için CNN ve çift yönlü LSTM birimlerini birleştirir. Amazon inceleme verileriyle eğitilmiş bu model, Vaka Çalışması 2'de inceleme faydalılığını sınıflandırmak için yeniden kullanılmıştır (Bakı Şekil 2.9).

Değerlendirme tutarlılığı için Vaka Çalışması 1'deki veri bölme stratejisi burada da uygulanmış; 5 katlı çapraz doğrulama ile 20 epoch boyunca eğitim yapılmıştır. Model, 256 batch size, 0.01 öğrenme oranı, 0.5 dropout oranı ve Adam optimizasyonuyla yapılandırılmıştır. Girdi uzunluğu Amazon veri kümeleri için sırasıyla 21 ve 24, Yelp için 690 olarak belirlenmiştir. Kelime dağarcığı 50.000 ve gömme boyutu 50'dir. Çıkış

katmanında sigmoid aktivasyon ve kayıp fonksiyonu olarak ortalama karesel hata (MSE) kullanılmıştır. Modelin toplam parametre sayısı 2.966.903'tür.

RoBERTa-base modeli ise TensorFlow üzerinde uygulanmış ve 160 GB'lık büyük bir metin koleksiyonuyla önceden eğitilmiş bir BERT türevidir. Veriler, RoBERTa tokenizer ile 50.000 kelimelik bir kelime dağarcığına göre dönüştürülmüş ve her giriş 768 boyutlu vektörlerle temsil edilmiştir. Model, 12 dönüştürücü katmandan oluşmakta ve çoklu başlıklı dikkat mekanizmaları içermektedir. Eğitimde 32 batch size, 10 epoch, 7e-5 öğrenme oranı, 0.1 dropout ve Adam optimizasyonu kullanılmıştır. Toplamda 124.647.170 öğrenilebilir parametre içermektedir.

PRE-DHRRF yapısında, her iki model de inceleme faydalılığı sınıflandırmasında tam performans sağlamak üzere görev verilerine göre yeniden eğitilmiştir. İkili sınıflandırma için Binary Cross-Entropy kaybı ve Adam optimizasyonu tercih edilmiştir. Tüm hiperparametreler ve ayarlar Tablo 2.3'te özetlenmiştir.

Çizelge 2.3. Modellerimizde (BHRQUT ve PRE-DHRRF) kullanılan hiperparametre ayarları

Hyperparameter	BHRQUT model	PRE-DHRRF model	
		pre-trained model I. Balanced CNN-BiLSTM	pre-trained model.II (Roberta-base)
Batch Size	256	256	256
Number of Epochs	20	20	20
K-Fold	5	5	5
Activation Function	Relu	Relu	Gelu
Dense Activation	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adam	Adam	Adam
Loss Function	MSE	MSE	Binary Categorical Crossentropy
Dropout	0.5	0.5	0.1
Learning Rate	0.01	0.01	5.00E-07
MAX_NB_WORDS	50,000	50,000	50,000
Embedding Dimension	50	50	768
maxLen (Input Lengths)	21,22,23,24,25,30,38,44	21, 24, 690	21, 24, 690

2.6. İnce Ayar (Fine-Tuning) ve Performans Değerlendirmesi

Model değerlendirmesi, derin öğrenme veya makine öğrenimi modellerinin başarımını, avantajlarını ve sınırlılıklarını ölçmek için çeşitli metriklerle gerçekleştirilir. Modelin etkinliği, araştırmanın erken aşamalarında test edilmeli ve ilerleyen süreçte de izlenmelidir.

Kullanılan performans ölçütleri arasında karışıklık matrisi, genelleme farkı, doğruluk, hassasiyet, geri çağırma, F1 skoru, RMSE ve MAE yer almaktadır.

2.7. Deneysel Yapılandırma

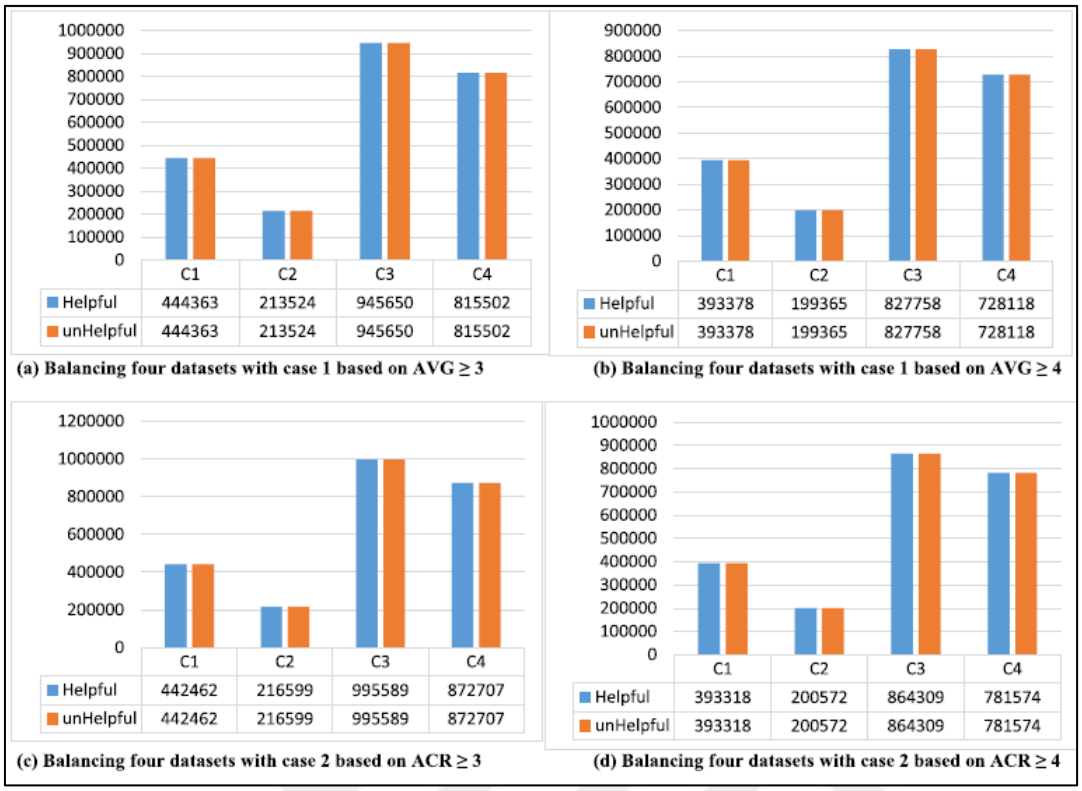
Deneylerimiz, Python programlama dili kullanılarak Intel(R) Core(TM) i7-5500U CPU (2.40 GHz), 64-bit işletim sistemi ve NVIDIA [6] tarafından sağlanan TITAN XP grafik kartlarıyla (40 GB GPU RAM) donatılmış bir sistemde gerçekleştirilmiştir. Sistem RAM'i 83.5 GB, disk alanı ise 235.7 GB olup, günlük yaklaşık 11 saatlik hesaplama kullanımıyla çalışmıştır.

Derin öğrenme algoritmalarının uygulanması ve analizler için birçok kütüphane kullanılmaktadır. Günümüzde kütüphaneler sürekli güncellenmekte ve yenileri geliştirilmektedir; ancak hâlen tek bir kütüphane baskın değildir. Kullanıcılar deneyimlerine göre farklı kütüphaneler tercih etmektedir. Çalışmamızda scikit-surprise, NLTK, TensorFlow ve Keras kütüphaneleri kullanılmıştır.

3. Deneysel Sonuçlar

3.1. Vaka Çalışması 1: Sonuçlar ve Deneysel Doğrulama

Vaka Çalışması 1'de önerilen modelin etkinliği, dört veri kümesi kategorisi (C1, C2, C3, C4) üzerinde değerlendirildi. Her kategori, iki yol ve her yolda iki vaka olmak üzere toplam dört vakadan oluşmaktaydı: (1_1), (1_2), (2_1) ve (2_2). Tüm vakalarda amaç, yorumları faydalı ve faydasız olarak sınıflandırmaktı. Veri kalitesini artırmak için TCF algoritması kullanıldı ve veri dengesini sağlamak üzere Eşanlamlı Bazlı Faydasız Yorum Artırma (SbURA) algoritması uygulandı. Ön işleme sonrası veri örnekleme yapısı Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Yol 1 ve Yol 2'deki veri kümelerinin dört Amazon-5 puan kategorisi için veri dengeleme

Performans değerlendirmesinde en yaygın kullanılan metrik doğruluktur. Önerilen öneri çerçevesinin tahmin gücünü değerlendirmek için eğitim doğruluğu analiz edilmiş ve Tablo 3.1'de sunulmuştur. Modelin tahmin performasını göstermek için çeşitli grafiksel ölçütler kullanılabilir. Önerilen model, C2 veri setinde (1_1) vakasında %97 yüksek doğruluk ve 0,03 düşük kayıp değeri elde etmiştir. Buna karşılık, C3 veri setinde (2_2) vakasında %88 en düşük doğruluk ve 0,09 daha yüksek kayıp değeri gözlemlenmiştir. Analizler ayrıca, model performansının (1_2) vakasında ortalama puan (avg) 4 ve üzerinde olduğunda, (2_2) vakasında ise puan 4'ü aştığında azaldığını göstermiştir. Bu nedenle, öneri performansını artırmak için faydalılık oranının ortalama ≥ 3 seviyesinde tutulması önerilmektedir.

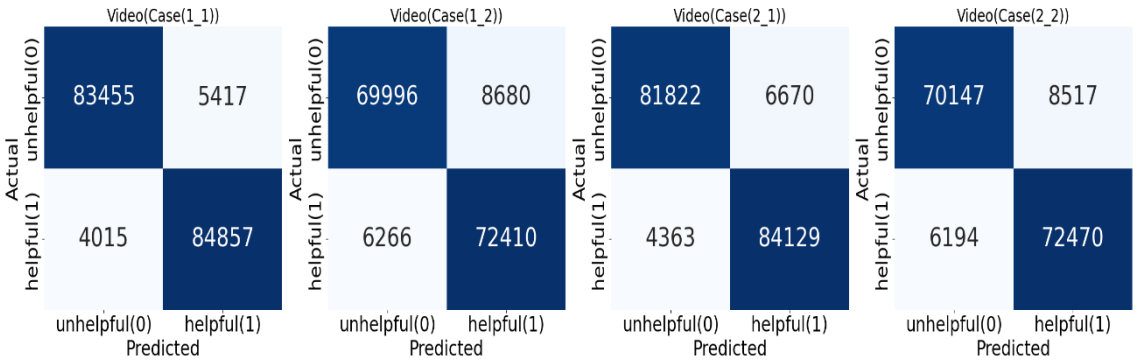
Çizelge 3.1. Modelimizin her kategori (C1, C2, C3 ve C4) için doğrulama ve eğitim sonuçları (Kesinlik, duyarlılık, F1-skora ve doğruluk) vaka isimlerine göre

Category of Amazon Score-5	Case Name	Precision		Recall		F1-Score		Accuracy	
		Val	Train	Val	Train.	Val	Train.	Val	Train
Cl	case (1-1)	94%	96%	95.4%	98%	95%	97%	95%	97%
	case (1-2)	89.4%	93%	92%	95%	90.6%	94%	90.4%	94%
	case (2-1)	92.6%	94%	95%	98%	94%	96%	94%	96%
	case (2-2)	89.8%	93%	92.2%	96%	91%	95%	90.8%	94%
Average								92.55%	95.25%

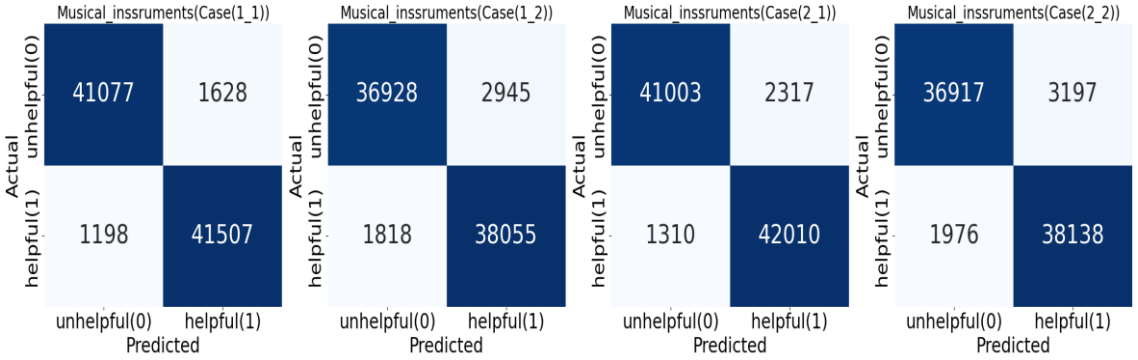
Çizelge 3.1. (devam) Modelimizin her kategori (C1, C2, C3 ve C4) için doğrulama ve eğitim sonuçları (Kesinlik, duyarlılık, F1-skora ve doğruluk) vaka isimlerine göre

Category of Amazon Score-5	Case Name	Precision		Recall		F1-Score		Accuracy	
		Val	Train	Val	Train.	Val	Train.	Val	Train
C2	case (1-1)	96.2%	97%	96.8%	99%	97%	98%	97%	98%
	case (1-2)	92.8%	96%	95.4%	98%	94%	97%	94%	97%
	case (2-1)	94.8%	97%	97%	99%	96%	96%	96%	98%
	case (2-2)	92.4%	95%	95%	98%	94%	97%	94%.	97%
Average								95.25%	97.5%
C3	case (1-1)	93.4%	95%	94.4%	96%	94%.	96%	94%	96%
	case (1-2)	86.6%	90%	91.6%	95%	89%	93%	88.8%	92%
	case (2-1)	92%	95%	93.8%	96%	93%	95%	93%	95%
	case (2_2)	85.6%	89%	91%	96%	88%	92%	88%	92%
Average								90.95%	93.75%
C4	case (1-1)	92.2%	93%	94.2%	97%	93%	95%	93%	95%
	case (1-2)	89.2%	91%	92.6%	95%	91%	93%	90.8%	93%
	case (2-1)	91.8%	94%	94.2%	96%	93%	95%	93%	95%
	case (2-2)	88%	91%	93%	95%	90.8%	93%	90%	93%
Average								91.7%	94%

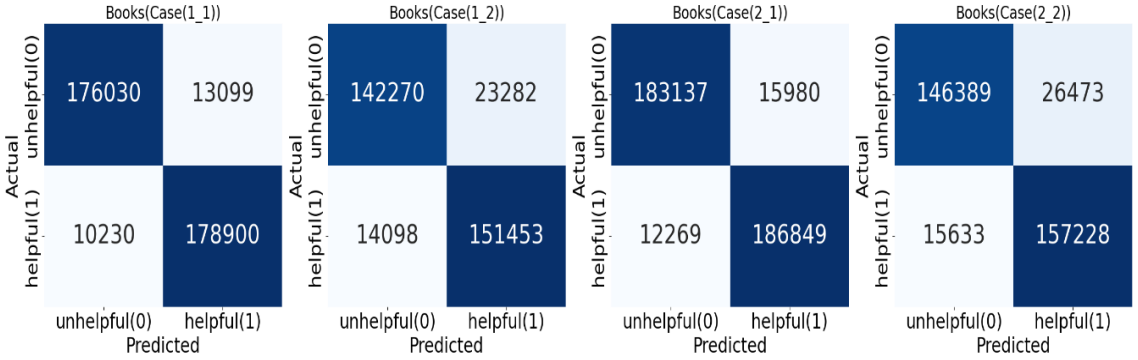
Karışıklık matrisi, sınıflandırma modellerinin performansını değerlendirmede yaygın olarak kullanılan bir araçtır. Önerilen modelimizde, karışıklık matrisi yatay eksende tahmin edilen sınıfları, dikey eksende gerçek sınıfları gösterir ve doğru ile yanlış sınıflandırmaların ayrımını sağlar. Matrisin çapraz elemanları doğru pozitif (TP) ve doğru negatif (TN) sayısını temsil ederek sınıflandırma doğruluğunu gösterir. Bu yapı, farklı eğitim veri kategorilerindeki sınıflandırma raporu bileşenlerinin çıkarılmasında temel rol oynar. Karışıklık matrislerinden elde edilen sonuçlar, tüm vakalarda ve sınıflarda doğru pozitif tahminlerin arttığını ve böylece model doğruluğu ve kalitesinin iyileştiğini göstermiştir. Şekil 3.2, önerilen modelin test aşamasında oluşturulan karışıklık matrislerini sunmaktadır. Alt şekiller (a), (b), (c) ve (d) sırasıyla C1, C2, C3 ve C4 veri kümeleri için uygulanan dört vakayı göstermektedir.



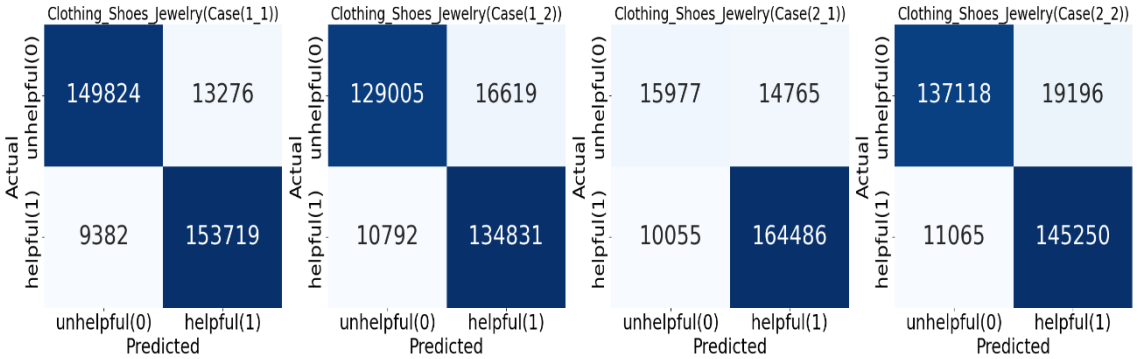
(a) C1 için modelin (1_1), (1_2), (2_1) ve (2_2) vakalarındaki karışıklık matrisleri.



(b) C2 için modelin (1_1), (1_2), (2_1) ve (2_2) vakalarındaki karışıklık matrisleri.



(c) C3 için modelin (1_1), (1_2), (2_1) ve (2_2) vakalarındaki karışıklık matrisleri.



(d) C4 için modelin (1_1), (1_2), (2_1) ve (2_2) vakalarındaki karışıklık matrisleri

Şekil 3.2. Yol 1 ve Yol 2'de vaka çalışması 1 modeline ait karışıklık matrisleri (a-d), C1, C2, C3 ve C4 veri setleri için iki vakayı göstermektedir

Tablo 3.2, önerilen yöntemle birlikte çeşitli mevcut yaklaşımların karşılaştırmalı analizini sunmaktadır. RMSE, MSE ve MAE gibi değerlendirme metrikleri kullanılarak farklı kullanıcı görüşlerinin etkisi ölçülmüştür. Literatürde, kişiselleştirilmiş öneri hizmetlerini geliştirmeye yönelik çeşitli modeller yer almaktadır. RHRN modeli [7], inceleme anlamsal çıkarıcı, kullanıcı profili oluşturucu ve kullanıcı/ürün öneri jeneratörü olmak üzere üç aşamalı bir mimari sunar. Bu çerçevede, oy değerine göre yorumların faydalılığını sınıflandırmak için hibrit CNN–BiLSTM modeli kullanılmıştır. AFRAM modeli [8], müşteri yorumları ve puanlarından gizli özellikleri yakalamaya odaklanır ve dikkat mekanizmasıyla geliştirilmiş CNN ve LSTM tabanlı çift yol mimarisi ile moda öneri sistemi geliştirir. Machine-Review Generation Model [9], insan ve makine tarafından yazılan yorumlarda bağlam duyarlı dikkat mekanizmalarını kullanarak faydalı yorumlar üretip açıklamaya çalışır. DUPIA modeli [10], kullanıcı yorumlarını analiz ederek kullanıcı tercihleri ve ürün çekiciliği arasındaki farkı ayırt eden dikkat mekanizmaları içeren CNN tabanlı bir yaklaşım sunar. Son olarak, [11] matriks çarpanlarına ayırma (MF) yöntemiyle tahmini puanlama ve faydalı önerileri değerlendirirken, müşteri yorumlarından Latent Dirichlet Allocation (LDA) kullanarak ürün özellik matrislerini çıkarmıştır.

Çizelge 3.2. Faydalılık incelemesi analiz modellerinin RMSE, MSE ve MAE'ye göre performans karşılaştırması

Models	RMSE				MSE				MAE			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
RHRM [7]			0.592								0.468	
AFRAM [8]								1.098				0.801
Machine-generated reviews[9]	0.881											
DUPIA [10]		0.833										
Deep Feature Extractor [11]					1.581	1.475	1.221	1.211				
BHRQUT. On case (1_1) with avg>=3	0.204	0.161	0.218	0.223	0.041	0.026	0.047	0.050	0.071	0.038	0.089	0.091
BHRQUT. On case (1_2) with avg>=4	0.265	0.219	0.290	0.264	0.070	0.048	0.084	0.069	0.133	0.073	0.154	0.139
BHRQUT. On case (2_1) with avg>=3	0.217	0.184	0.233	0.225	0.047	0.034	0.054	0.051	0.076	0.047	0.102	0.101
BHRQUT. On case (2_2) with avg>=4	0.263	0.226	0.301	0.268	0.069	0.051	0.089	0.072	0.126	0.083	0.179	0.145

Başka bir çalışmada [12], duygu analizini geliştirmek için CNN ve dikkat tabanlı BiGRU modeli kullanan derin öğrenme yaklaşımı önerilmiştir. Bu yöntem, derecelere dayalı olarak pozitif ve negatif yorumları etkili şekilde ayırt eden derin öğrenme algoritmaları kullanmaktadır. Kitap kategorisi (C3) kapsamında, maksimum yorum metni uzunluğunun etkisini inceleyerek, RHRM [7] ve SLCABG [12] modellerinin sınıflandırma performanslarını karşılaştırdık. Değerlendirme Kesinlik, Duyarlılık, F1-Skor ve Doğruluk metrikleriyle yapıldı. Sonuçlar, modelimiz için (1_1) vakasında optimum maksimum dizi uzunluğunun 24 olduğunu gösterirken, SLCABG modeli (maksimum metin uzunluğu 648) RHRM modelini (metin uzunluğu 2817) geride bırakmaktadır. Bu bulgular Tablo 3.3'te ayrıntılı olarak sunulmaktadır.

Çizelge 3.3. Kitap kategorisi (C3) kapsamında maksimum girdi cümle uzunluğunun BHRQUT modeli ve diğer modeller üzerindeki etkisinin karşılaştırılması

Models	measured score for helpfulness	Max•length	Precision	Recall	F1 *_score*	Accuracy
RHRM [7]	vote	2817	85.54%	88.73%	87.10%	86.14%
SLCABG [12]	rating	648	92.9%	93.8%	93.3%	93.4%
Our BHRQUT. case (1_1)	Average rating	24	93.4%	94.4%	94%	94%

3.2. Vaka Çalışması 2: Sonuçlar ve Deneysel Doğrulama

Önerilen modelin etkinliği, iki Amazon veri kümesi (DS1 ve DS2) ile bir Yelp Akademik İnceleme veri kümesi kategorisi kullanılarak değerlendirilmiştir. Değerlendirme, iki önceden eğitilmiş model üzerinde gerçekleştirilmiştir: Model I (Dengeli CNN-BiLSTM) ve Model II (RoBERTa-base). Her veri kümesinde yorumlar ikili olarak sınıflandırılmıştır: yararlı (1) veya yararsız (0). Veriler, ön işleme adımlarıyla hazırlanmıştır. Modelin maksimum dizi uzunluğu üzerindeki etkisini incelemek için hem kısa hem de uzun dizi uzunluklarına sahip veri setleri kullanılmıştır. Bu vaka çalışması, kullanıcıların ürünlere dair yararlı yorumlarını analiz ederek çevrim içi alışverişte karar doğruluğunu artırmayı ve eğitim verisindeki gürültü ve dengesizlik nedeniyle aşırı öğrenmeyi (overfitting) önlemeyi amaçlamaktadır.

Kullanışlı ve kullanışsız yorumları daha iyi sınıflandırmak için, yalnızca değerli (ortalama puan ≥ 3) ve değersiz (ortalama puan < 3) yorumlar eğitim verisi olarak seçilmiştir. Ortalama

puan, dört eğilim algoritmasına dayalı olarak gerçek ve tahmin edilen puanların ortalamasıyla hesaplanmıştır [5].

Modelimiz, sınıflandırma performansında en iyi metrikleri elde etmiştir. Sonuçlarımız, aşağıdaki çalışmalarla karşılaştırılabilir niteliktedir:

Differentiated Recommendation Framework [13]: Özellik çıkarımı, bilgi grafı, veri artırma ve FM tabanlı bilgi transferiyle öneri üretir, soğuk başlangıç problemini çözer.

CF(NRCF) + CF(DCCF) [14]: Kullanıcıların değerli yorumlarına göre etkili komşular seçilir.

Derin öğrenme tabanlı metin incelemesi [15]: Derin ağlar ve transformer modelleriyle öneri doğruluğu artırılır.

Duygu Analiz Sistemi [16]: Yorumlar toplanır, duygu analizi yapılır ve filtreleme ile öneri sunulur.

Vaka çalışması 2'de, sekiz farklı değerlendirme metriği (doğruluk, kesinlik, geri çağırma, F1 skoru, MAE, RMSE, karışıklık matrisi, genelleme farkı) kullanılmıştır. Tüm deneyler, 5 katlı çapraz doğrulama ile 20 epoch boyunca 100 iterasyonda gerçekleştirilmiştir. Eğitim verisinin %80'i eğitim, %20'si test için kullanılmıştır.

Karışıklık matrisi, sınıflandırma modellerinin başarımını analiz etmek için kullanılır. N-Actual, TN, FP, TP, FN değerleri ile model performansı ölçülmüştür (Tablo 3.4). Elde edilen sonuçlar, Model II'nin (RoBERTa-base) tüm veri kümelerinde Model I'e (Balanced CNN-BiLSTM) göre daha yüksek TP ve TN, daha düşük FP ve FN değerleri sunduğunu göstermektedir. Bu, transformer tabanlı önceden eğitilmiş modellerin metin sınıflandırma görevlerinde daha etkili olduğunu ortaya koymaktadır.

Çizelge 3.4. Üç farklı veri kümesiyle her bir önceden eğitilmiş model için önerilen çerçevenin karışıklık matrisi

Pre-trained Model Name	Dataset	N-Actual	TN	FP	TP	FN
Model. I (BalancedCNN-BiLSTM)	DS1	85409	41171	1534	41455	1249
	DS2	326201	149004	14096	154676	8425
	DS3	66394	31359	1838	31370	1827
Model.II Roberta-base	DS1	85409	41793	912	42192	513
	DS2	326201	154197	8903	155755	7346
	DS3	66394	31456	1741	31991	1206

Derin öğrenme modellerinin performansını değerlendirirken, özellikle inceleme sınıflandırması ve tavsiye sistemleri bağlamında, doğruluk ve modelin yeni verilere genelleme yeteneği kritik öneme sahiptir. Bu bağlamda genelleme farkı, eğitim ve doğrulama doğrulukları arasındaki fark olarak tanımlanır ve modelin aşırı öğrenmeye karşı dayanıklılığını gösterir. Küçük bir genelleme farkı, modelin eğitim ve doğrulama verileri üzerinde tutarlı performans sergilediğini, yani veriye aşırı uyum sağlamadan anlamlı örüntüler öğrendiğini gösterir. Büyük bir fark ise modelin veri setine özgü gürültüleri öğrendiğini ve genelleme kabiliyetinin düşük olduğunu gösterebilir. Bu nedenle, genelleme farkının değerlendirme metriklerine dahil edilmesi, modelin güvenilirliği ve gerçek dünya uygulamalarına uygunluğu açısından derinlemesine içgörüler sunar. Tablo 3.5, üç farklı veri kümesi (DS1, DS2, DS3) üzerinde iki önceden eğitilmiş modelin (Model I: Dengelemeli CNN-BiLSTM, Model II: RoBERTa-base) doğruluk, kayıp ve genelleme farkı karşılaştırmasını sunmaktadır. Genelleme farkı, doğrulama doğruluğu ile eğitim doğruluğu arasındaki fark olarak hesaplanmıştır.

Çizelge 3.5. Önerilen modelin üç farklı veri kümesi üzerindeki genelleme farkı değerlendirmesi

Pre-trained Model Name	Dataset	Accuracy		Generalization Gap	Loss	
		Val	Train		Val	Train
Model.I (BalancedCNN-BiLSTM)	DS1	0.97	0.98	0.01	0.027	0.022
	DS2	0.93	0.95	0.02	0.050	0.046
	DS3	0.945	0.988	0.043	0.060	0.023
Model.II (Roberta-base)	DS1	0.983	0.986	0.003	0.065	0.062
	DS2	0.95	0.954	0.004	0.114	0.110
	DS3	0.956	0.962	0.006	0.116	0.110

Tablo 3.6 ve Şekil 3.3 (a-c)'de performans sonuçları sunulmuştur. Üç veri kümesinde de, inceleme analiziyle geliştirilen derin öğrenme modelleri, RMSE ve MAE açısından önerilen modele kıyasla daha başarılı sonuçlar vermektedir. Ancak, önerilen modelin Pretrained Model II kullanılarak geliştirilen versiyonu en düşük hata oranlarını elde etmiş ve en başarılı performansı göstermiştir.

DS1 (Müzik Aletleri Yorumları) veri kümesinde, Pretrained Model II ile geliştirilen model MAE: 0.016, RMSE: 0.129 ile en iyi sonucu verirken; Pretrained Model I kullanılan versiyon MAE: 0.032, RMSE: 0.178 elde etmiştir. BHRQUT modeli ise MAE: 0.038, RMSE: 0.161 değerleriyle daha ılımlı sonuçlar sunmuştur.

DS2 (Giyim, Ayakkabı ve Takı Yorumları) veri kümesinde, Pretrained Model II ile geliştirilen model yine en düşük hata oranlarını (MAE: 0.049, RMSE: 0.223) elde etmiş, Pretrained Model I ise MAE: 0.069, RMSE: 0.262 ile onu takip etmiştir. BHRQUT modeli ise MAE: 0.091, RMSE: 0.225 değerlerine ulaşmıştır.

DS3 (Yelp Yorumları) veri kümesinde, Pretrained Model II ile geliştirilen model MAE: 0.044, RMSE: 0.210 ile en başarılı performansı göstermiştir. Pretrained Model I kullanılan versiyon MAE: 0.052, RMSE: 0.228 değerlerine ulaşırken, karşılaştırılan taban modeller önemli ölçüde daha yüksek hata oranlarına sahiptir: Derin öğrenme tabanlı model [15] MAE: 0.755, RMSE: 1.025 ve CF(NRCF) + CF(DCCF) modeli [14] MAE: 0.909, RMSE: 1.220 ile en düşük başarıyı göstermiştir.

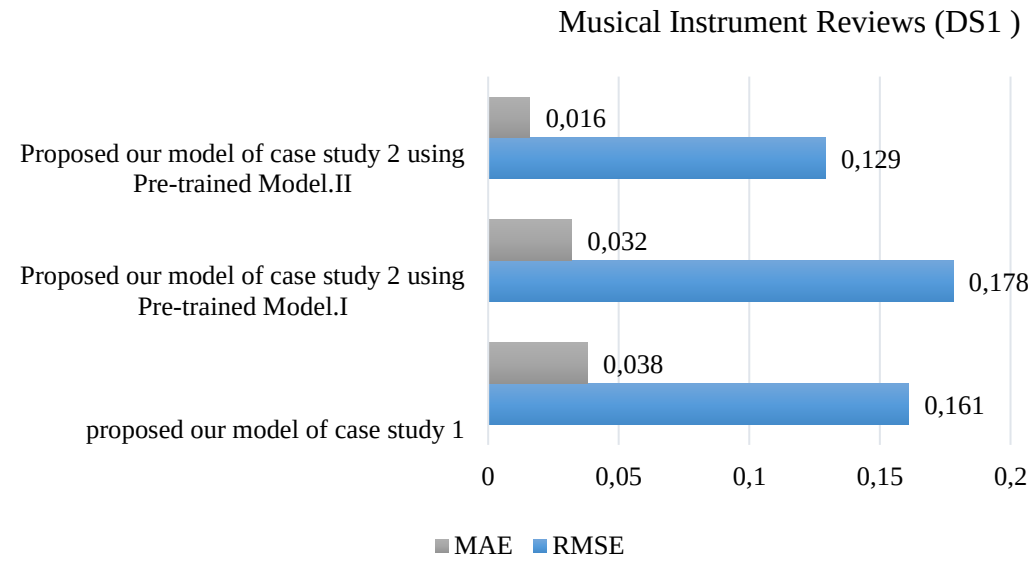
Bu sonuçlar, özellikle Pretrained Model II tabanlı önerilen modelin, tahmin hatalarını en aza indirerek üstün doğruluk sağladığını ve klasik/yöntemsel modellere göre daha güvenilir olduğunu ortaya koymaktadır.

Çizelge 3.6. Modellerin performansına ilişkin kapsamlı analiz

Study	Dataset	RMSE	MAE	Accuracy	Precision	Recall	F1score
Differentiated recommendation framework [13]	DS2			0.8125			
CF(NRCF)+ CF(DCCF) [14]	DS3	1.22	0.909				
Derin öğrenme tabanlı metin incelemesi [15]	DS3	1.025	0.755				

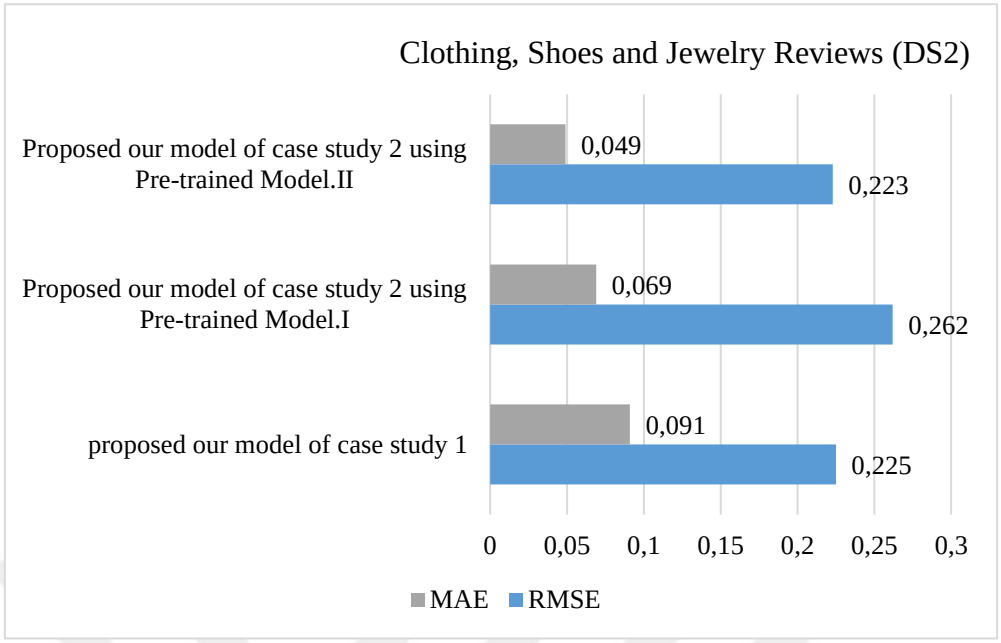
Çizelge 3.6. (devam) Modellerin performansına ilişkin kapsamlı analiz

Study	Dataset	RMSE	MAE	Accuracy	Precision	Recall	F1score
Duygu Analiz Sistemi[16]	DS3			0.73	0.88	0.73	
Proposed BHRQUT in case study 1	DS1	0.161	0.038	0.97	0.962	0.968	0.97
	DS2	0.225	0.091	0.93	0.92	0.94	0.93
Proposed model in case study 2 using a Pre-trained Model. I	DS1	0.178	0.032	0.97	0.962	0.97	0.97
	DS2	0.262	0.069	0.93	0.92	0.95	0.93
	DS3	0.228	0.052	0.945	0.946	0.944	0.95
Proposed model in case study 2 using a Pre-trained Model.II	DS1	0.129	0.016	0.983	0.978	0.988	0.983
	DS2	0.223	0.049	0.95	0.945	0.955	0.95
	DS3	0.210	0.044	0.956	0.948	0.963	0.956

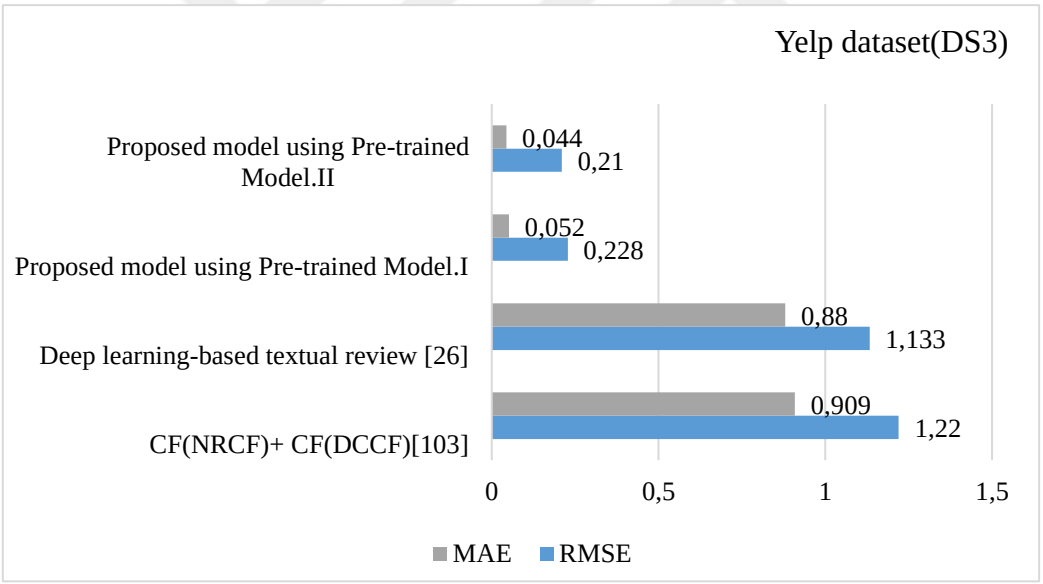


(a) Müzik aleti yorumları (DS1) için hata metriklerinin karşılaştırması

Şekil 3.3. (a-c) Üç veri kümesi (DS1, DS2 ve DS3) üzerinde RMSE ve MAE hata metriklerinin karşılaştırmalı analizi



(b) Giyim, ayakkabı ve takı yorumları (DS2) veri kümesi için hata metriklerinin karşılaştırması

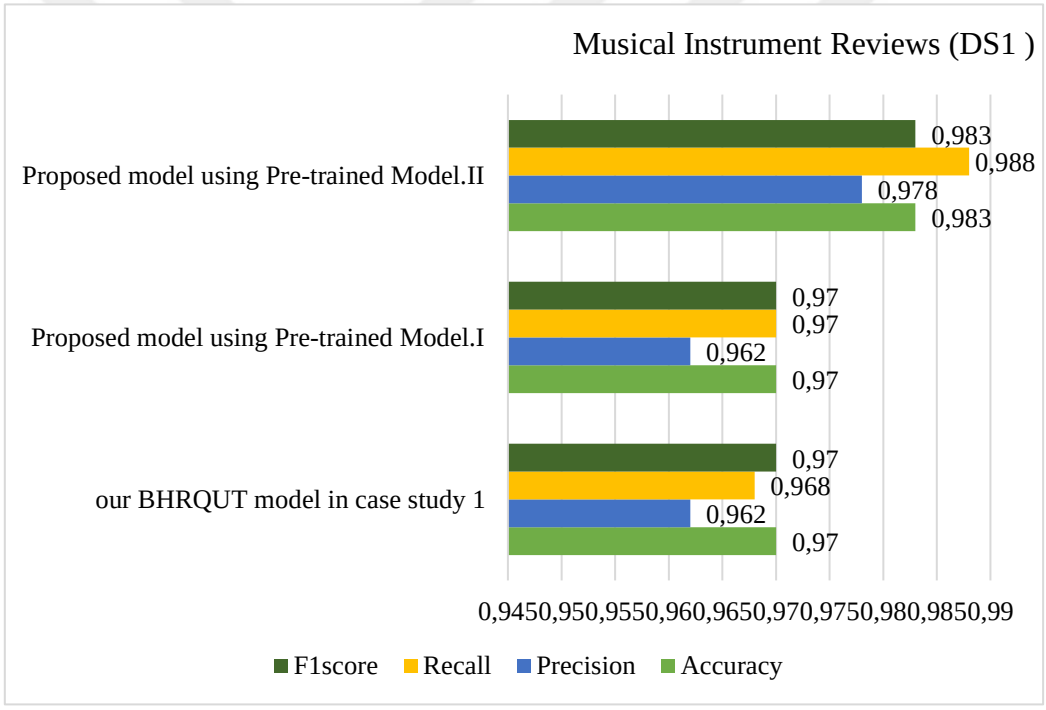


(c). Yelp veri kümesi (DS3) için hata metriklerinin karşılaştırması

Şekil 3.3. (devam) (a-c) Üç veri kümesi (DS1, DS2 ve DS3) üzerinde RMSE ve MAE hata metriklerinin karşılaştırmalı analizi

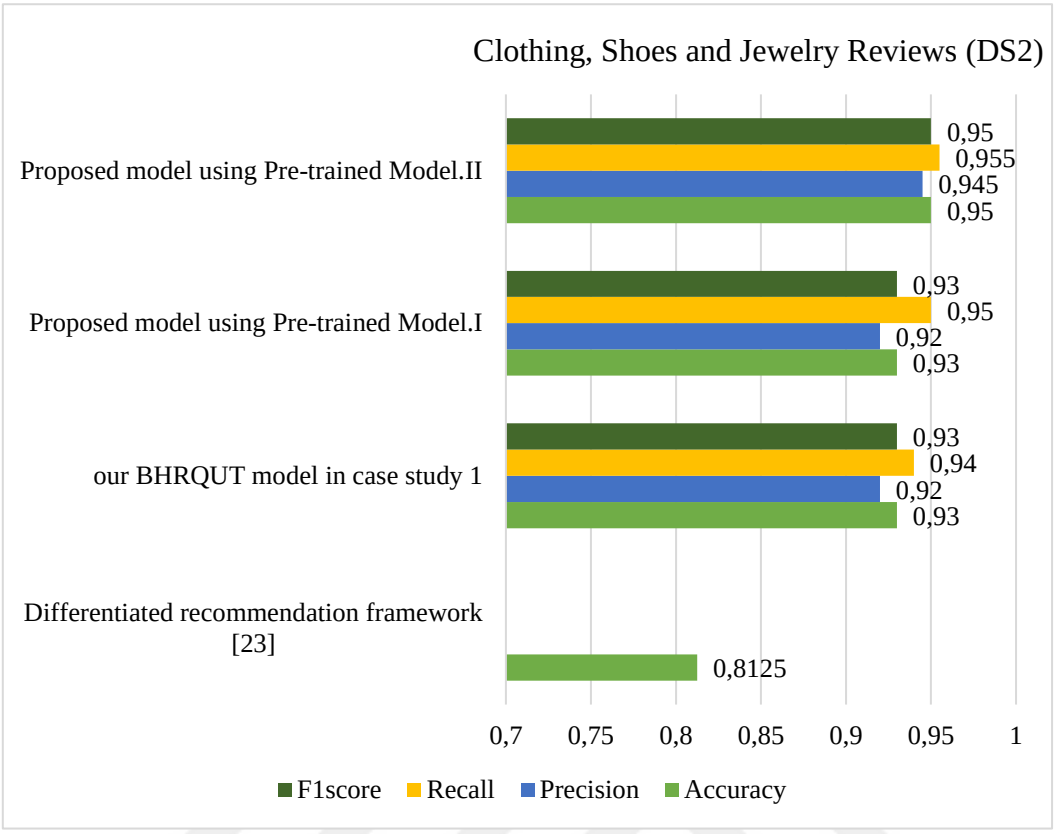
Tablo 3.6 ve Şekil 3.4 (a-c)'de görüldüğü gibi, önerilen modelde kullanılan Önceden Eğitilmiş Model II, diğer modelleri geride bırakmıştır. Müzik Aletleri İncelemeleri (DS1) veri setinde doğruluk, hassasiyet, duyarlılık ve F1 skoru sırasıyla %98.3, %97.8, %98.8 ve %98.3 olarak ölçülmüştür. Önceden Eğitilmiş Model I kullanan önerilen model, BHRQUT modelini duyarlılıkta (%97 ve %95) geçerken, doğruluk, hassasiyet ve F1 skorunda

BHRQUT ile benzer performans göstermiştir. Yelp veri setinde (DS3), Sentiment Analysis System [16] doğruluk, hassasiyet ve duyarlılık değerleri sırasıyla 0.73, 0.88 ve 0.73 iken, Önceden Eğitilmiş Model II kullanan önerilen modelde bu değerler 0.956, 0.948 ve 0.963 olarak belirlenmiştir. Giyim, Ayakkabı ve Takı İncelemeleri (DS2) veri setinde, Differentiated Recommendation Framework [13] doğruluk oranı 0.8125 iken, önerilen modelde Önceden Eğitilmiş Model I için 0.93 ve Model II için 0.95 olarak bulunmuştur. Beş modelin doğruluk, hassasiyet, duyarlılık ve F1 skorlarına göre sıralaması şu şekildedir: Önceden Eğitilmiş Model II kullanan önerilen model > Önceden Eğitilmiş Model I kullanan önerilen model > BHRQUT > Differentiated Recommendation Framework > Sentiment Analysis System.

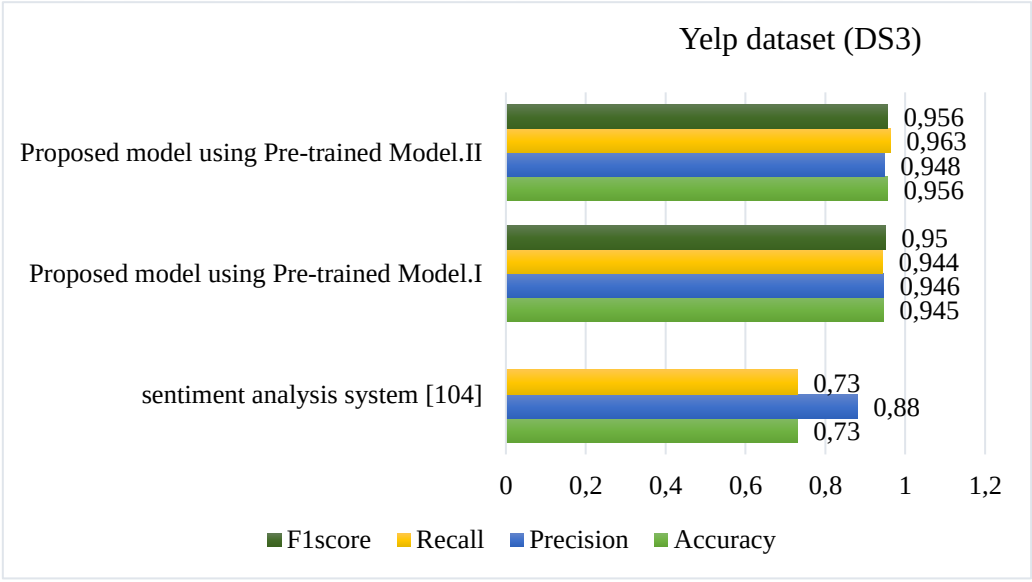


(a) DS1 veri setinde model performansının Doğruluk, Hassasiyet, Duyarlılık ve F1-Skoru ile değerlendirilmesi

Şekil 3.4. (a-c) Üç veri seti DS1, DS2 ve DS3 üzerinde modelin doğruluk, hassasiyet, duyarlılık ve F1-Skoru metrikleri ile performans değerlendirmesi



(b) DS2 veri setinde model performansının Doğruluk, Hassasiyet, Duyarlılık ve F1-Skoru ile değerlendirilmesi



(c) DS3 veri setinde model performansının Doğruluk, Hassasiyet, Duyarlılık ve F1-Skoru metrikleri ile değerlendirilmesi

Şekil 3.4. (devam) (a-c) Üç veri seti DS1, DS2 ve DS3 üzerinde modelin doğruluk, hassasiyet, duyarlılık ve F1-Skoru metrikleri ile performans değerlendirmesi

4. Tartışma ve Sonuç

Önerilen BHRQUT modeli, dört farklı Amazon inceleme kategorisi veri setinde RMSE, MSE, MAE, Doğruluk, Geri Çağırma (Recall), Hassasiyet (Precision) ve F1-skora göre değerlendirildi. Sonuçlar, modelimizin tüm bu metriklerde temel modelleri geride bıraktığını göstermektedir (Bkz. Tablo 3.2 ve 3.3). BHRQUT modeli, Video Oyunları (C1), Müzik Aletleri (C2) ve Kitaplar-5 (C3) kategorilerinde RMSE'de %76,84'e varan iyileşme sağlamıştır. Ayrıca MSE ve MAE metriklerinde de sırasıyla %98,35 ve %95,26 oranında önemli gelişmeler kaydedilmiştir.

Veri setlerinde özet inceleme metinlerinin maksimum kelime uzunluğu sabitlenerek model performansı incelenmiş ve çoğu ürün kategorisinde 21 token uzunluğunun en uygun olduğu belirlenmiştir. Kısa cümle uzunluklarının doğruluğu artırdığı ve aşırı öğrenmeyi (overfitting) önlediği gözlemlenmiştir. Özellikle kitap kategorisinde, 24 token uzunluğundaki girişler ile BHRQUT modeli, çok daha uzun metin kullanan modellerden (RHRM ve SLCABG) %6-9 arasında performans artışı göstermiştir.

Çalışmada hem küçük hem büyük veri setleri kullanılarak, modelimizin diğer yararlılık analizi modellerine göre üstün olduğu kanıtlanmıştır. Özellikle (1_1) vakasında, Balanced CNN-BiLSTM tabanlı derin öğrenme modeli diğer tüm konfigürasyonlardan daha yüksek performans sergilemiştir. Müzik Aletleri (C2) veri seti, tüm metriklerde en iyi sonuçları vermiştir. Ayrıca, ortalama değerlendirme (avg ≥ 3) ya da ACR ≥ 3 kullanımı, daha yüksek eşik değerlerine kıyasla daha başarılı bulunmuştur.

BHRQUT modeli, TCF ve DA yöntemleri ile desteklenmiş ve %97 doğrulukla diğer derin öğrenme modellerini geride bırakmıştır. Karışıklık matrisi analizleri, modelin faydalı yorumları sınıflandırmada üstün olduğunu ve kullanıcı eğilimlerinin daha çok yardımcı oy sayısına bağlı olduğunu göstermiştir. Sonuçlar, önerilen modelin gerçek dünyada Amazon, Alibaba ve Netflix gibi platformlarda kullanılabilirliğini ve kullanıcı tercihleriyle satın alma davranışı arasındaki ilişkiyi başarıyla ortaya koymaktadır.

İkinci model PRE-DHRRF'de, iki önceden eğitilmiş model, Balanced CNN–BiLSTM (Model I) ve Roberta-base (Model II), üç farklı veri seti (DS1, DS2, DS3) üzerinde sınıflandırma performansını değerlendirmek için kullanılmıştır.

Tablo 3.4'da, iki modelin (Balanced CNN-BiLSTM ve Roberta-base) üç veri setindeki (DS1, DS2, DS3) karışıklık matrisi bileşenleri (TN, FP, TP, FN) verilmiştir. Bu bileşenler doğruluk ve geri çağırma gibi temel değerlendirme metriklerini doğrudan etkiler. Yüksek TP değerleri doğruluğu artırırken, düşük FN değerleri geri çağırmayı iyileştirir.

Tablo 3.5 'te görüldüğü üzere, Model I (Balanced CNN-BiLSTM), üç veri setinde güçlü genelleme yeteneği ve aşırı öğrenme (overfitting) ile etkili mücadele göstermiştir. DS1'de doğrulama doğruluğu %97, eğitim doğruluğu %98 olup genelleme farkı %1'dir. DS2 için doğrulama ve eğitim doğruluğu sırasıyla %93 ve %95, genelleme farkı %2'dir. En büyük fark %4.3 ile DS3 veri setinde gözlemlenmiştir. Tüm veri setlerinde kayıp (loss) değerleri düşük seviyelerde kalmıştır. Bu sonuçlar, Model I'in eğitim verilerinden aşırı öğrenmeden güvenilir şekilde öğrendiğini göstermektedir.

Model II (Roberta-base) ise özellikle uzun ve karmaşık metin içeren DS3 veri setinde güçlü performans sergilemiş, tüm veri setlerinde çok düşük genelleme farklarıyla başarılı sonuçlar elde etmiştir. DS1'de doğrulama ve eğitim doğrulukları sırasıyla %98.3 ve %98.6'dır, genelleme farkı yalnızca %0.3'tür. DS2 ve DS3 için genelleme farkları %0.4 ve %0.6 olarak raporlanmıştır. Kaybı değerleri Model I'e kıyasla biraz daha yüksek olmakla beraber, küçük genelleme farkları modelin karmaşık kalıpları aşırı öğrenmeden öğrendiğini ortaya koymaktadır.

Tablo 3.6, Şekil 3.3 ve 3.4'de sunulan sonuçlar, önerilen yaklaşımın diğer yöntemlere göre önemli performans iyileştirmeleri sağladığını göstermektedir. Balanced CNN-BiLSTM modeli, üç veri setinde Roberta-base modelini genel olarak geride bırakmıştır. Önerilen modelin doğruluk oranları DS1 için %98.3, DS2 için %95 ve DS3 için %95.6'dır. Hassasiyet (Precision), duyarlılık (Recall) ve F1 skorları da yüksek seviyelerdedir. Ayrıca, önerilen model RMSE ve MAE değerlerinde de diğer yöntemlere kıyasla daha düşük hata oranları göstermiştir.

Özellikle PRE-DHRRF modeli, önceki Balanced CNN-BiLSTM tabanlı BHRQUT modeline göre DS1 ve DS2 veri setlerinde daha iyi sonuçlar sunmuştur. Bu iyileşmeler, özellikle Recall ve MAE metriklerinde belirgin olup, diğer metriklerde de benzer gelişmeler gözlemlenmiştir. Derin öğrenme temelli yaklaşımların, iyi hazırlanmış veri setleri ile

birlikte, inceleme metni sınıflandırmasında istatistiksel metriklerle uyumlu olduğu ortaya konmuştur.

PRE-DHRRF modeli, kullanıcıların ürünleri değerlendirme eğilimlerindeki farklılıkları ve dengesiz veriyi dengelemek için veri artırma (data augmentation) tekniklerini kullanmaktadır. Doğal dil işleme yöntemleriyle ön işlenen verilerle, 20 epoch ve 5 kat çapraz doğrulama ile eğitilen modeller, ortalama %98.3 doğruluk, %97.8 hassasiyet, %98.8 duyarlılık ve %98.3 F1 skoruna ulaşmıştır. Her iki model de ürün öneri sistemlerini geliştirmek amacıyla kullanıcı tarafından oluşturulan içerikleri analiz etmek için geliştirilmiştir. Ancak, özellik entegrasyonu, model bileşimi ve ince ayar stratejileri açısından farklı yaklaşımlar benimsemektedir. Tablo 4.1, bu iki modelin mimari farklarını öne çıkarmaktadır.

Elde edilen sonuçlar, önerilen çerçevenin e-ticaret platformlarında inceleme ve değerlendirme sistemlerini önemli ölçüde iyileştirme potansiyelini ortaya koymaktadır. Düşük genelleme farkı ve yüksek performans, modelin gerçek dünya uygulamalarında da güvenilir olduğunu göstermektedir. Bu yaklaşım, kullanıcıların faydalı yorumları otomatik olarak belirlemesini sağlayarak, daha bilinçli satın alma kararlarını destekler ve kullanıcı memnuniyetini artırır. Ayrıca, içerik denetleyiciler için inceleme filtreleme süreçlerini kolaylaştırarak iş yükünü azaltır ve operasyonel verimliliği yükseltir. Böylece dijital platformlarda kaynakların etkin kullanımı ve kullanıcı deneyiminin geliştirilmesi sağlanır.

Çizelge 4.1. BHRQUT ve PRE-DHRRF modellerinin karşılaştırmalı genel görünümü

Aspect	BHRQUT	PRE-DHRRF
Model Mimarisi	Özelleştirilmiş Dengeli CNN-BiLSTM Modeli	Önceden Eğitilmiş ve Dengelenmiş CNN-BiLSTM + RoBERTa backbone
Veri Seti	Amazon Reviews data (C1, C2, C3, and C4) with textual reviews (summary), user-id, item-id, and overall ratings features	Amazon Reviews (DS1, DS2) with textual reviews (summary), user-id, item-id, and overall ratings features, and Yelp Dataset (DS3) with user-id, item-id, full review text, and star ratings features
Girdiler	Embedding tokenized sequences + helpful label (0/1)	Embedding tokenized sequences + helpful label (0/1) in pre-trained Balanced CNN-BiLSTM, and Positional encoding tokens+helpful label (0/1) with using RoBERTa model

Çizelge 4.1. (devam) BHRQUT ve PRE-DHRRF modellerinin karşılaştırmalı genel görünümü

Aspect	BHRQUT	PRE-DHRRF
Maksimum Girdi Uzunlukları	Tested on 21, 22, 23, 24, 25, 30, 38, 44 token sequence lengths	Tested on 21, 24, 690 token sequence lengths
Performans	Hesaplama maliyeti düşük, rekabetçi doğruluk	Özellikle kısa ve uzun incelemelerde daha yüksek doğruluk ve daha iyi genelleme
Overfitting Handling	Dengelenmiş eğitim verisi ve %0.5 dropout kullanımı	Dengelenmiş eğitim verisi, önceden eğitilmiş bilgi ve düşük dropout oranı (%0,1) kullanımı
Avantajlar	• Hafif yapı • Daha hızlı eğitim süreci • Göreve özel uyarlanabilirlik • Kısa incelemelerde başarılı performans	• Üstün genelleme yeteneği • Güçlü anlamsal anlayış • Ölçeklenebilirlik • Hem kısa hem uzun incelemelerde başarılı performans
Yenilikler	TCF algoritması, veri artırma ve NLP tekniklerinin CNN ile yerel özellik çıkarımı ve BiLSTM ile sıralı modellemenin birleşimi	TCF algoritması, veri artırma yöntemleri ve büyük bir NLP korpusundan transfer öğrenmenin birleştirilmesi

5. Sınırlamalar ve Gelecek Çalışmalar

Bu çalışmanın temel sınırlamalarından biri, inceleme sınıflandırmasında yalnızca puanlama temelli göstergelere dayanılmasıdır; bu yaklaşım, daha karmaşık bağlamsal analiz gerektiren durumlarda yetersiz kalabilir. Ayrıca, uzun inceleme metinlerinin işlenmesi sırasında model performansı ve bellek kullanımı kısıtlamaları gözlemlenmiştir. Tendency tabanlı işbirlikçi filtreleme yönteminin büyük veri setlerinde ölçeklenebilirlik sorunları yaşaması da önemli bir diğer kısıtlamadır.

Gelecek çalışmalarda, modellerin uzun metinleri daha iyi işleyebilmesi için dikkat mekanizmaları ve hiyerarşik yapılar gibi gelişmiş mimariler entegre edilebilir. Ayrıca, ürün görselleri, stok durumu ve üretim tarihi gibi ek bağlamsal verilerin modele dahil edilmesi faydalı olacaktır. İşbirlikçi filtrelemede hesaplama yükünü azaltacak daha verimli yöntemlerin geliştirilmesi, ön işleme ve eğitim süreçlerinin bellek optimizasyonu da araştırılması gereken alanlardır.

Bunun yanı sıra, incelemelerin zaman dinamiklerinin modellenmesi ve NLTK gibi doğal dil işleme araçlarının derin öğrenme kütüphaneleriyle entegrasyonu, sistemlerin gerçek dünya uygulamalarında daha etkili çalışmasını sağlayabilir. Son olarak, farklı derin öğrenme

algoritmalarının çeşitli e-ticaret senaryolarında karşılaştırmalı değerlendirilmesi, modellerin genel uygulanabilirliğini artıracaktır.

Kaynaklar

1. Papazoglou, M. P., & Tsalgatidou, A. (2000). Business to business electronic commerce issues and solutions. *Decision Support Systems*, *29*(4), 301-304.

2. Ricci, F., Rokach, L. and Shapira, B. (2010). Introduction to recommender systems handbook. in Recommender systems handbook, Springer, pp. 1–35.

3. Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2022). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4425-4445.

4. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).

5. Sreepada, R. S., Patra, B. K., Chakrabarty, A., & Chandak, S. (2018, January). Revisiting tendency based collaborative filtering for personalized recommendations. In *Proceedings of the ACM India joint international conference on data science and management of data* (pp. 230-239).

6. Niero, B. NVIDIA's AI Revolution: A Comprehensive Strategic Analysis.

7. Li, Q., Li, X., Lee, B., & Kim, J. (2021). A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation Service. *Applied Sciences*, *11*(18), 8613.

8. Li, W., & Xu, B. (2020). Aspect-based fashion recommendation with attention mechanism. *IEEE Access*, *8*, 141814-141823.

9. Ouyang, S., & Lawlor, A. (2021). Improving explainable recommendations by deep review-based explanations. *IEEE Access*, *9*, 67444-67455.

10. Zhang, X., Liu, H., Chen, X., Zhong, J., & Wang, D. (2020). A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness. *Information Sciences*, *519*, 306-316.

11. Shoja, B. M., & Tabrizi, N. (2019). Customer reviews analysis with deep neural networks for e-commerce recommender systems. *IEEE access*, *7*, 119121-119130.

12. Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE access*, *8*, 23522-23530.

13. Yan, C., Chen, Y., & Zhou, L. (2019). Differentiated fashion recommendation using knowledge graph and data augmentation. *Ieee Access*, *7*, 102239-102248.

14. Lee, S., Shin, H., Choi, I., & Kim, J. (2022). Analyzing the impact of components of yelp. com on recommender system performance: case of Austin. *IEEE Access*, *10*, 128066-128076.

15. Gheewala, S., Xu, S., Yeom, S., & Maqsood, S. (2024). Exploiting deep transformer models in textual review based recommender systems. *Expert Systems with Applications*, *235*, 121120.

16. Hassan, A. K. A., & Abdulwahhab, A. B. A. (2017). Reviews Sentiment analysis for collaborative recommender system. *Kurdistan journal of applied research*, 87-91.

Bilim Kodu : 92432
Anahtar Kelimeler : E-ticaret, dengeli derin öneri sistemleri, kullanıcı görüşlerindeki farklılıklar, önceden eğitilmiş model, yorum analizi.
Sayfa Adedi : 121
Danışman : Doç. Dr. Adem TEKEREK

ACKNOWLEDGEMENTS

This thesis was conducted as part of a challenging and rewarding journey in artificial intelligence and recommender systems. Throughout the research process, I encountered technical challenges and intellectual growth opportunities that significantly impacted my academic and personal development.

First and foremost, all praise is due to Allah, the Most Merciful, the Most Compassionate, for His countless blessings during this educational journey. Without His grace and kindness, the successful completion of this project would not have been possible.

I want to express my gratitude to my advisors, "Assoc. Prof. Dr. Adem TEKEREK" and "Asst. Prof. Dr. M. Mutlu YAPICI", from “Gazi University”, whose guidance and support were instrumental at every stage of this thesis.

I would also like to extend my deepest thanks to the thesis jury members for their valuable comments and suggestions, which improved this work from its inception to its final version.

Special, I owe a great deal of appreciation in particular to my deceased brother in his grave, “Dr. Surgeon, Kadhim Al-Abadi ”. He always urged me to complete my studies and provided me with his affection and tenderness. I owe a great deal of appreciation to my family, my father, my mother, my children, my husband, and my sisters. Thank you for your unwavering patience, encouragement, and support during the entire duration of my graduate studies. Their trust in me has been an ongoing source of encouragement.

This endeavour would not have been possible without the generous support of the Republic of Iraq, particularly from the Faculty of Basic Education at the University of Thi-Qar.

Lastly, thanks go to my colleagues and friends in my beloved country, Iraq, for their technical discussions and moral support throughout the research. Completing this thesis would not have been possible without the contributions of all these individuals and institutions.

CONTENTS

Page

ABSTRACT iv

ÖZET v

ACKNOWLEDGEMENTS xxxvi

CONTENTS xxxvii

LIST OF TABLES xl

LIST OF FIGURES xli

1. INTRODUCTION 1

2. LITERATURE REVIEW 19

2.1. Types of Deep Learning Techniques for E-Commerce Recommendation Systems 19

2.1.1. Supervised learning 20

2.1.2. Unsupervised learning 21

2.1.3. Semi-supervised learning 21

2.1.4. Self-supervised learning 22

2.1.5. Multi-task learning 22

2.1.6. Hybrid models 23

2.1.7. Transfer learning 23

2.2. Customers' Review Analysis in E-commerce with Recommendation System 24

2.2.1. Sentiment analysis 24

2.2.2. Sentiment analysis focused on specific aspects (ABSA) 25

2.2.3. Topic modelling 26

2.2.4. Text classification 26

2.2.5. Review rating prediction 27

2.2.6. Review trend analysis 27

2.2.7. Personalised review summarisation 28

Page

2.2.8. User-generated content analysis 28
2.2.9. Trustworthiness analysis 29
2.3. User Behaviour Analysis in Recommender Systems 30
2.3.1. Studies on the application of CNN 30
2.3.2. Studies on the application of RNN 31
2.3.3. Studies on the application of LSTM 32
2.3.4. Studies on the application of autoencoders 33
2.3.5. Studies on the application of NLP 34
3. METHODOLOGY AND MATERIAL 41
3.1. Dataset Description 41
3.2. Data Preprocessing Techniques 42
3.3. Deep Learning Models Used in Case Studies 48
3.3.1. Case study 1: BHRQUT model 48
3.3.2. Case study 2: PRE-DHRRF model 53
3.4. Model Training and Hyperparameter Selection 59
3.4.1. Reproducible results of case study 1 (BHRQUT) 64
3.4.2. Reproducible results of case study 2 (PRE-DHRRF) 67
3.5. Fine-Tuning and Performance Evaluation 70
3.6. Experiment Configuration 73
4. EXPERIMENTAL RESULTS 77
4.1. Case Study 1 (BHRQUT): Results and Experimental Validation 77
4.2. Case Study 2 (PRE-DHRRF: Results and Experimental Validation 83
5. DISCUSSION AND CONCLUSION 99
5.1. Analysis of the BHRQUT Model 99
5.2. Analysis of the PRE-DHRRF Model 103
5.3. Comparative Overview of BHRQUT and PRE-DHRRF Models 106

Page

5.4. Implications and Practical Significance 107

6. LIMITATION AND FUTURE WORK 109

REFERENCES 113

CURRICULUM VITAE 121

LIST OF TABLES

Table		Page
Table 2.1.	Emphasises the priority limitations and illustrates how our proposed model addresses these gaps	37
Table 3.1.	Amazon score-5 and Yelp review datasets, number of reviews	42
Table 3.2.	Examples of the Amazon review dataset and the Yelp academic review dataset	68
Table 3.3.	Hyperparameters are set up in our models (BHRQUT and PRE-DHRRF).	70
Table 4.1.	Validation and training loss results of our model for each category (C1, C2, C3, and C4) in each case name	78
Table 4.2.	Results of our model for each category (C1, C2, C3, and C4) in terms of Validation and Training (Precision, Recall, F1-score, and Accuracy) for each case name	79
Table 4.3.	Performance comparison of helpfulness review analysis models based on RMSE, MSE, and MAE	81
Table 4.4.	Effect of the max sequence length of the input sentences on the performance of the (BHRQUT) model	82
Table 4.5.	Comparison of the Impact of Maximum Input Sentence Length on the BHRQUT Model and Other Models within the Book Category (C3)	83
Table 4.6.	Confusion matrix of our framework for each pre-trained model with three datasets	85
Table 4.7.	Evaluation of the generalization gap for the proposed model across three datasets	86
Table 4.8.	Comprehensive analysis of model performance	94
Table 5.1.	Comparative Overview of BHRQUT vs. PRE-DHRRF	106

LIST OF FIGURES

Figure		Page
Figure 1.1.	Overview of the recommendation system	2
Figure 1.2.	LSTM layers architecture	13
Figure 1.3.	Representation of the BiLSTM	14
Figure 2.1.	İllustratites the categorization of sentiment analysis techniques	25
Figure 2.2.	Attention-based LSTM model	33
Figure 2.3.	NLP-based architecture	36
Figure 3.1.	Optimization ratings with adapting steps of the tendencies collaborative filtering algorithm	44
Figure 3.2.	Over-sampling based on SbURA for balancing binary classes	48
Figure 3.3.	Overview of the BHRQUT model	50
Figure 3.4.	Formulation of BHRQUT architecture	52
Figure 3.5.	An architecture of our proposed PRE-DHRRF framework	54
Figure 3.6.	Flowchart illustrating the steps methodology of our case study 2	55
Figure 3.7.	Construction of pre-trained (Roberta-base) model	58
Figure 3.8.	Effect of dropout on a neural network	60
Figure 3.9.	Formulation of activation functions	62
Figure 3.10.	Construction of the proposed case study 1	67
Figure 4.1.	Balancing data for the four Amazon-5 score categories of datasets in path1 and path2	77
Figure 4.2.	Confusion matrices (a-d) for the model of case study 1 in path1 and path2, featuring two cases for the C1, C2, C3, and C4 datasets	80
Figure 4.3.	Balancing three datasets (DS1, DS2, and DS3) on the SbURA algorithm	84
Figure 4.4.	(a-f) Accuracy and loss during training and validation in five-fold cross-validation	87
Figure 4.5.	(g-1) Accuracy and loss during training and validation in five-fold cross-validation	90

Figure **Page**

Figure 4.6. (a-c) Comparative analysis of error metrics RMSE and MAE across three datasets DS1, DS2, and DS3 94

Figure 4.7. (a-c) Performance evaluation of the model on three datasets DS1, DS2, and DS3 using Accuracy, Precision, Recall, and F1-Score metrics..... 96

1. INTRODUCTION

The rise of e-commerce and the proliferation of online platforms have facilitated the spread of misleading and unreliable product reviews and ratings posted by consumers, paving the way for the emergence of deepfakes. This growing challenge can generate anxiety and distrust among users, as individuals struggle to make informed purchasing decisions. In an era of advanced technology, it is essential to implement robust systems capable of distinguishing between genuine and deceptive information. Deep learning-based review analysis algorithms have gained popularity due to their powerful discriminative capabilities and their effectiveness in detecting valuable reviews. However, many existing deep learning approaches still struggle to identify helpful reviews accurately. E-commerce platforms face considerable challenges in filtering and prioritizing useful content to enhance their recommendation services. Although these platforms rely heavily on user-generated reviews to shape consumer behavior, not all reviews contribute equally to the decision-making process. The task of identifying and highlighting helpful reviews remains complex due to the vast volume of data, diverse linguistic patterns, and inherent biases present in user feedback.

E-Commerce involves marketing and selling products online using electronic devices. It can be divided into three trends: Business to Business (B2B), Business to Consumer (B2C), and Consumer to Consumer (C2C) [1].

Online purchasing has been increasingly popular in recent years, particularly during the coronavirus outbreak, and this has led to an explosion of data and information on the World Wide Web [2]. Due to concealed items that are just impossible to find, people may find it challenging to select a product that suits their needs and preferences, as well as their expectations and behaviours. Because of this, there needs to be a bridge or connecting link that structures the relationships of inclinations between users, items, or users and items. In response to this challenge, the multi-directional recommendation system emerged, expanded, and became significant in various fields, including information systems, social media, economics, and learning. The main goal of these systems is to create a predictive model capable of estimating the prediction function [3].

The success of e-commerce marketing largely stems from recommendation systems, which tailor the shopping experience to individual customers. Creating a recommendation system for e-commerce helps platforms deliver the most relevant information to customers at the right time, fostering customer loyalty and creating a more seamless shopping experience. The overall perspective of the recommendation system, illustrated in Figure 1.1, is a three-layer process. Layer 1 is the Input Layer that represents customers entering the platform. The icons symbolise multiple users providing their preferences, browsing history, or purchase behaviour as input data. Layer 2 is the Recommender System and Recommendation. The central layer contains the recommendation system, which processes the input data using algorithms (e.g., collaborative or content-based filtering). The item features are what the content-based recommender systems rely on. It suggests things that share features with the one the user previously selected. The recommendation algorithms known as collaborative filtering (CF) depend on the similarity between users and objects. According to CF, two users are similar if they have previously selected and suggested the same item to one another [4].

The recommendation output is a list, representing personalised product suggestions tailored to each user. Layer 3 is the Output Layer, representing the customer receiving personalised recommendations. The single-user icon signifies a unique, tailored shopping experience based on the system's suggestions.

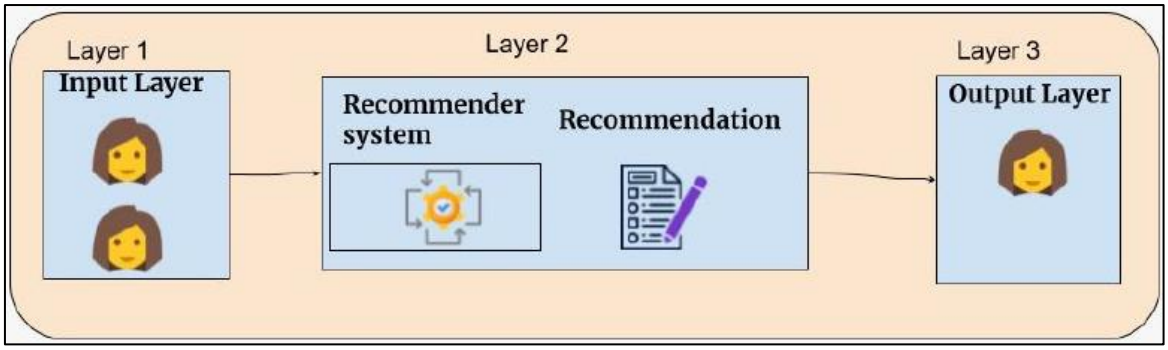


Figure 1.1. Overview of the recommendation system

Product customers provide reviews in text format [5]. The main way these reviews function as an essential information source for customers making judgments about what to buy is by rating things on a scale of 1 to 5 (often referred to as stars) [6]. In the era of sophisticated technology, having a reliable system that can discriminate between accurate and fraudulent

information is essential. The rise of e-commerce and the development of online platforms have facilitated the spread of misleading and unreliable product reviews and ratings posted by consumers, paving the way for deepfakes. This can create anxiety and distrust as individuals cannot easily make purchasing decisions. In the age of advanced technology, it is crucial to have a robust system that can distinguish between real and fake information. Preferences for unrated items were predicted using collaborative filtering and K-Nearest Neighbours (KNN) algorithms based on the similarity between consumers' ratings. However, consumers may assign extreme ratings, the lowest or highest possible scores, without sufficient justification. This raises concerns regarding the reliability of star ratings, as they often lack explanatory context. To enhance the credibility of these ratings, they should be supported by detailed customer reviews. Such reviews enable other users to assess the fairness and validity of the ratings, thereby improving the transparency and reliability of the evaluation process [7].

To improve recommendation systems, user reviews are essential, especially when overcoming the drawbacks of conventional collaborative filtering techniques. When users submit few or no ratings, collaborative filtering suffers from the well-known rating sparsity problem, even if it works effectively with enough rating information. This problem makes it harder for the algorithm to understand user preferences and narrows the coverage of the suggestion space. Instead of scalar ratings, user reviews offer more detailed, fuller feedback that provides essential information about item attributes and user preferences [8]. Incorporating user reviews into recommender systems, though, presents several challenges. Processing and analysing massive amounts of unstructured text data requires a lot of computer power and sophisticated natural language processing methods. Extracting consistent and trustworthy features is further complicated by the subjective nature of evaluations, which includes a range of writing styles and possible biases. The authors emphasise the necessity for reliable techniques to efficiently understand and apply this information to increase suggestion accuracy without adding additional biases or complexity.

Deep learning-based review analysis algorithms are very popular due to their strong discriminative capabilities and ability to detect valuable reviews. However, many deep learning methods based on review analysis must identify valuable reviews more effectively.

E-commerce platforms face significant challenges in filtering helpful reviews to improve recommendation services. While these platforms rely heavily on user-generated reviews to influence purchasing decisions, not all reviews contribute equally to the decision-making process. Identifying and prioritising helpful reviews is complex due to the sheer volume of data, diverse linguistic expressions, and potential biases in user feedback. Existing methods need to more effectively utilise the rich knowledge underlying the features learned by deep neural networks. Li et al highlighted the limitations of traditional approaches and proposed a hybrid CNN-based model [9]. The embedded Convolutional Neural Network (CNN) model was utilised to enhance and transform additional features into feature vectors. These vectors were subsequently used for sentiment classification through a Bidirectional Long Short-Term Memory (BiLSTM) network [10].

Review sites have become an essential source of information for consumers when deciding what to buy. However, with the increasing number of reviews online, consumers face the problem of information overload. To alleviate this problem, many e-commerce platforms, such as Amazon and Yelp, have developed a review-rating scoring system allowing users to rate their reviews. Xinzhe Li and colleagues proposed a mechanism for e-commerce platforms to incentivise relevant and valuable reviews by incorporating ratings and votes as part of their multichannel entry design [11]. The usefulness of online reviews was estimated by integrating review content, history, and product details using a context-sensitive coding mechanism and CNN (Convolutional Neural Network) [12]. To gain a deeper understanding of user preferences and product features, Suyu and colleagues developed a neural recommendation framework that incorporates the usefulness of reviews into the recommendation process. This method aims to improve recommendation quality [13]. Moreover, the Dual Attention Recommendation Model using Reviews and their Helpfulness (DARMH) model provides a multi-task learning approach for rating prediction and review usefulness [14].

Recommendation models are mainly classified as collaborative filtering and content-based recommendation frameworks based on the types of information available [15]. Collaborative filtering models analyse user preferences using historical interaction data and user behavioural similarities [16]. A user similarity matrix is used to infer user preferences and tendencies. However, since this approach relies heavily on users' historical data, the problem of “cold start” for new users arises. Propensity modelling has improved personalised

recommendations, incorporating complex factors such as temporal dynamics[17]. İn this thesis, our work adopts a simplified propensity-based model incorporating user and item rating biases for improved rating predictions.

The primary function of recommender systems is to provide tailored information and recommendations that improve user experience and increase user satisfaction and engagement. However, data scarcity is a significant problem for these systems. Since real-world datasets often need more complete user-item interaction data, this leads to sub-optimal performance. The dependence of recommender systems on sparse user-item interaction data further reduces their effectiveness and makes them vulnerable to problems such as overfitting or undergeneralisation. Data augmentation techniques play a critical role in overcoming such data gaps and improving the accuracy of recommender systems [18]. Data augmentation extends data sets by producing additional or modified versions of existing data. For example, work in [19]. aimed to increase the amount and diversity of training data by using adversarial examples in collaborative filtering and content-based filtering algorithms through feature injection, synthetic data generation, and data augmentation techniques.

Data Augmentation is a technique used to artificially increase the size and diversity of a dataset by creating modified versions of existing data. It is widely used in machine learning, especially in review classification, natural language processing, and speech recognition. The CorpusAugmentation algorithm is a data augmentation method that uses LDA and n-grams to generate positive and negative trigrams of each review text. Researchers have studied the effects of text data augmentation on sentiment classification quality by combining direct data augmentation (DDA), back translation (BT), pre-trained data augmenter (PREDATOR), and BART augmentation methods with various deep learning and machine learning classification methods [20]. Another study fine-tuned the T5 model to create new questions from user reviews, using reviews as documents and review summaries as answers [21]. A study focused on the impact of data augmentation on dialogue intention classification [22]. To better capture relationships between users and items, a user-item knowledge graph was constructed, allowing for enriched interaction modeling. To address challenges related to limited data, a data augmentation method incorporating knowledge transferwas employed. This approach refines item representations by assigning more informative labels and reducing redundancy[23].

Traditional recommendation systems often fall short in accurately inferring user preferences, particularly in data-scarce scenarios. To mitigate this, sequential recommendation models have been developed to predict user behavior, such as viewing or purchasing, based on prior activity. These models are particularly effective in domains with sparse interactions by leveraging pretrained knowledge from related tasks[24]. A notable example is the Deep Transfer Learning with Multimodal Embedding (DTLME) model, which integrates textual, visual, and metadata features to enhance predictive performance under limited interaction history conditions [25]. Recommender systems often rely on numeric ratings or purchase histories, which may miss nuanced information in user reviews. Therefore, the textual reviews improve recommendations by leveraging deep transformer models to extract latent features [26]. Developing language model-based recommendation systems (LMRS) that effectively balance user privacy with high-performance recommendation algorithms remains a significant challenge. While recommendation systems leveraging customer review analysis have reached a certain level of maturity, their integration and optimisation within the e-commerce domain are yet to be fully realised.

To address class imbalance in helpful review analysis, this study applies an oversampling data augmentation technique. Specifically, unhelpful reviews, typically underrepresented, are syntactically varied by substituting words with their synonyms, thereby preserving semantic meaning while generating diverse input samples. This approach enhances the model's ability to generalize by increasing exposure to varied training data. Rather than reiterating the broader importance of helpful reviews, the focus here is on how data augmentation improves classification accuracy and model robustness in real-world recommendation scenarios. This study addresses challenges such as data insufficiency, cold start, and overfitting by constructing a recommender system utilising a real-world e-commerce dataset.

Specific categories from the Amazon and Yelp datasets were selected to ensure a representative and meaningful evaluation of our proposed models in the context of real-world e-commerce scenarios. For Amazon, categories such as Video Games, Musical Instruments, Books-5, and Clothing, Shoes, and Jewelry were chosen due to their high volume of user reviews and the diversity of consumer opinions they attract. These categories are central to everyday online shopping experiences and involve complex decision-making processes where helpful reviews are especially valuable. Similarly, for Yelp, categories like

Restaurants, Shopping, and Beauty & Spas were selected because they reflect common consumer services that rely heavily on user-generated feedback. These domains often involve subjective experiences and varied language patterns, making them ideal for testing the robustness and generalization capabilities of review analysis models. Together, the selected categories capture a broad spectrum of consumer behavior and linguistic diversity, thereby enhancing the validity and applicability of our findings to the broader e-commerce landscape.

For this research, various study items were investigated using two analysis frameworks (BHRQUT, PRE-DHRRF). The increase in computer capacity and NVIDIA's TITAN XP graphics cards have advanced Deep Learning (DL) for various applications, including NLP and text classification, which is the focus of this study. This research will use a DL model to train on Amazon and Yelp datasets to develop an e-commerce recommendation system. To enhance model performance, this research optimises user rating values using the Tendency Collaborative Filtering (TCF) algorithm, balances review samples through data augmentation, and explores strategies to reduce overfitting while improving accuracy. Two novel deep learning frameworks are introduced: BHRQUT, which utilises a balanced CNN-BiLSTM architecture, and PRE-DHRRF, which extends this structure by incorporating pretrained layers. Both models are trained in review-based text sequences, and their effectiveness is evaluated through comparative analysis. This approach aims to investigate the role of data balancing and pretrained components in improving review analysis, an area not yet fully explored in related work.

The key contributions of this research are as follows:

- We offer a solution for classifying informative feedback by leveraging deep models based on the characteristics of review datasets from popular e-commerce sites like Yelp and Amazon, along with the latest advancements in deep learning. We utilized four sets of Amazon review datasets and one Yelp Academic Review dataset to evaluate our studies. Our proposed method outperforms the baseline investigations in terms of accuracy, precision, recall, F1 score, misclassification rate (MAE, RMSE), confusion matrix, and generalization gap.
- We examined how the size of the helpfulness measured ratio affected the star scores (3,4) in the Amazon review dataset, both with and without TCF, in four different scenarios.

- We also analysed how the input sentence length influenced the first model's performance (BHRQUT). Subsequently, we selected the Amazon review dataset category with the highest case accuracy and tested it along with the Yelp dataset in the second model (PRE-DHRRF).
- The sparsity problem presents a fundamental challenge for many recommendation models based on collaborative filtering algorithms. To address this issue, we propose a combination of propensity collaborative filtering methods and transducer-based pre-trained models to prevent the model from overfitting to limited data and to enhance its performance on unseen instances.
- This paper presents a pre-trained framework for deep, practical review proposals. This framework utilises a two-path architecture to enhance classification results. The first path employs a custom model, CNN-BiLSTM, tailored to fit the nuances specific to the dataset and classification objectives. The second path incorporates a pre-trained transducer model (Roberta-base), recognised for its superior ability to process text sequences. This model is initially trained on a specific dataset, after which its weights are reused on other datasets. This two-way configuration boosts the framework's adaptability and accuracy by leveraging the generalisation capabilities of a generic model combined with the high performance of a custom model.

Deep learning

Deep learning, a subfield of machine learning, has become an increasingly prevalent application area for deep neural networks. Rather than relying on specialised algorithms tailored to specific tasks, deep learning emphasises data-driven learning to develop more generalised solutions across diverse datasets. This approach holds significant promise for addressing artificial intelligence challenges within machine learning. Deep neural networks are multilayer neural networks that contain two or more hidden layers [27]. In deep learning, there is a structure based on learning multiple levels of features or representations of data. High-level features are derived from lower-level features to form a hierarchical representation [28]. Some of these features represent the data better. In deep learning methods, practical algorithms are used for hierarchical feature extraction that best describe the data instead of manually extracted features.

In recommendation systems, deep learning is crucial by leveraging neural networks to capture complex user-item interactions, extract meaningful features from textual reviews, and enhance personalisation. These models are widely applied in domains such as monitoring user activity, digital libraries, movies to watch, text and news to read on web pages, product recommendations, personalised advertising, content filtering, and user preference prediction, significantly improving recommendation accuracy and user engagement [29], [30].

The factors that make deep learning architectures a fascinating field of study include:

- Deep learning, in contrast to conventional machine learning, can automatically learn and extract pertinent features from preprocessed data without human engineering.
- Deep neural networks are characterised by their ability to represent the creation of accurate and reliable relationships and patterns in data due to their multiple layers of learning.
- Models based on deep learning are appropriate for real-world applications because they can manage high-dimensional and large-scale datasets.
- Deep learning models can retrieve implicit data by using multiple hidden layers.
- The continuous advancements in activation functions (e.g., ReLU, GELU), data training strategies, and data augmentation methods improve model performance and generalisation. Additionally, transfer learning allows models to leverage pre-trained knowledge from large datasets, reducing the need for extensive training data and accelerating learning in new tasks. This is particularly valuable in applications with limited labelled data, such as recommendation systems, where pre-trained models can be fine-tuned for personalised recommendations.

In deep neural networks, multiple hidden layers facilitate the hierarchical extraction of complex patterns from data. These networks progressively learn representations, transitioning from simple to intricate features across successive layers. Each layer establishes relationships with its predecessor, enabling a more detailed analysis of input data. This structured learning process enhances the model's capability to offer more precise and informed decisions [31]. Deep learning specialisations known as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have various types of architectures designed. Specifically, Bidirectional Long Short-Term Memory (BiLSTM) handles

sequential data in both forward and backwards directions. These algorithms are used in many subject areas, including the classification of helpful reviews in recommendation systems, and these strategies are useful for various problems because they provide superior results comparable to or even better than human reasoning.

Convolutional neural networks (CNN)

The Convolutional Neural Networks (CNNs) algorithm is similar to standard supervised learning techniques. It takes images as input, extracts essential features from each image, and then trains a classifier based on these features. One of the main advantages of CNNs is their capacity to learn feature representations automatically. Unlike traditional supervised learning methods that necessitate manually created feature extraction algorithms, CNNs independently carry out feature extraction and representation learning. During the training phase, the network optimises the classifier parameters and the extracted features by minimising the classification error.

Furthermore, the hierarchical structure of CNNs enables the extraction of features at varying levels of complexity, ranging from simple patterns to more abstract representations. This automated feature extraction and prioritisation process allows the network to adapt to the specific problem at hand, eliminating the need for manual feature engineering. The ability of CNNs to learn and refine features directly from the input data constitutes a significant advantage over conventional learning methods, as the network independently handles the entire feature extraction process.

The fundamental component of this type of neural network is its structure, which serves as a feature extractor. This is achieved through template matching, facilitated by applying convolutional filtering operations. In the initial layer, the network filters the input image with multiple convolutional kernels, producing "feature maps." These feature maps are then normalised using an activation function and may be resized. This process can be repeated multiple times, where the resulting feature maps from one layer are further processed with new convolutional kernels. This iterative filtering generates new feature maps, which are again normalised and resized before undergoing additional convolutional operations. Ultimately, the values from the final set of feature maps are concatenated into a feature

vector, which serves as the output of the first processing block and acts as the input for the next stage of the network [32].

The selection of hyperparameters in the design of CNN networks is a critical challenge that significantly impacts model performance. The choice of these hyperparameters and their respective values is influenced by various factors, including the nature of the dataset, the expertise and intuition of the researcher, advancements in the field, and empirical results obtained from different applications. Proper hyperparameter tuning is essential to optimise the network's learning process and generalisation capabilities. The main parameters that can be used when designing a model using CNN are as follows:

- Number of convolutional Layers.
- Number of kernel layers.
- Filter Size.
- the step size for the convolution operation (stride).
- Padding: Same (zero-padding) or valid (no padding).
- Pooling layers(type and number).
- Batch size.
- Number of epochs.
- Learning rate.
- Optimiser algorithm.
- Loss function.
- Dropout rate.
- L1/L2 Regularisation (weight decay).
- Batch Normalisation.

In numerous studies, hyperparameters are optimised through iterative experimentation using a test-and-evaluate approach. The most optimal configuration for the model, yielding the best performance, is identified based on empirical results [33].

Bidirectional long short-term memory (BiLSTM)

In recommendation systems that use sequential inputs (such as user-item or user-user relationships), prior input data is required to predict a new, helpful recommender for the

user. A group of deep neural networks called recurrent neural networks (RNNs) is made to process sequential input in a loop that uses prior knowledge to classify the data by inferring meaning from frames, because of their capacity to build connections with the past. RNNs are adept at time-based tasks, but with this approach, the model keeps all the data. Activities require specific types of information, but not all of it. Therefore, not all of the past has to be preserved to categorise particular behaviours. RNNs, in contrast to conventional feedforward networks, have feedback loops, which enable them to capture temporal dependencies in input and maintain information from previous time steps. However, the vanishing and ballooning gradient issues plaguing typical RNNs make it challenging to recognise long-term relationships in sequences [34].

In predicting long events in the past, RNNs require a distinct architectural design. Long Short-Term Memory (LSTM) networks, a more effective type of RNN, have been proposed to solve these issues. Since RNNs' internal architecture repeats itself, it is theoretically possible for them to remember long-term activity. However, the criteria for recall must be carefully chosen because, in practice, such parameter selection is often infeasible. Despite claims that this architecture accounts for information from earlier time points, it is said that the gradient loss problem prevents this from occurring. By incorporating a customised gating system that controls the information flow and consists of forget, input, and output gates, LSTMs improve upon conventional RNNs, as illustrated in Figure 1.2. This process is achieved using an activation function. We can use various activation functions, such as Tanh, Sigmoid, or Relu. LSTMs are particularly useful for problems with long-range dependencies due to their architecture, enabling them to remember or forget information over extended periods selectively. Consequently, LSTMs have been effectively applied in numerous fields, including time-series forecasting, speech recognition, and natural language processing in review analysis to suggest products. In a Long Short-Term Memory (LSTM) unit, the current input vector xt and the previous hidden state $ht-1$ are fed into three key gating mechanisms: the forget gate, the input gate, and the output gate. The forget gate determines which parts of the previous cell state $Ct-1$ should be discarded, allowing the network to eliminate irrelevant or outdated information. Simultaneously, the input gate regulates the amount of new information from the current input and hidden state that should be added to the memory, facilitating the incorporation of relevant new data. The candidate cell state, generated using a tanh activation, provides potential values to be added to the cell state. These two components work together to update the cell state Ct, which serves as the unit’s internal

memory and is passed on to the next time step. Lastly, the output gate determines which parts of the cell state should be exposed as the output hidden state ht by applying a filtered version of the memory through a tanh activation followed by element-wise multiplication. This updated hidden state ht is then propagated to the next LSTM unit in the sequence or layer, enabling the model to capture long-term dependencies in sequential data.[35].

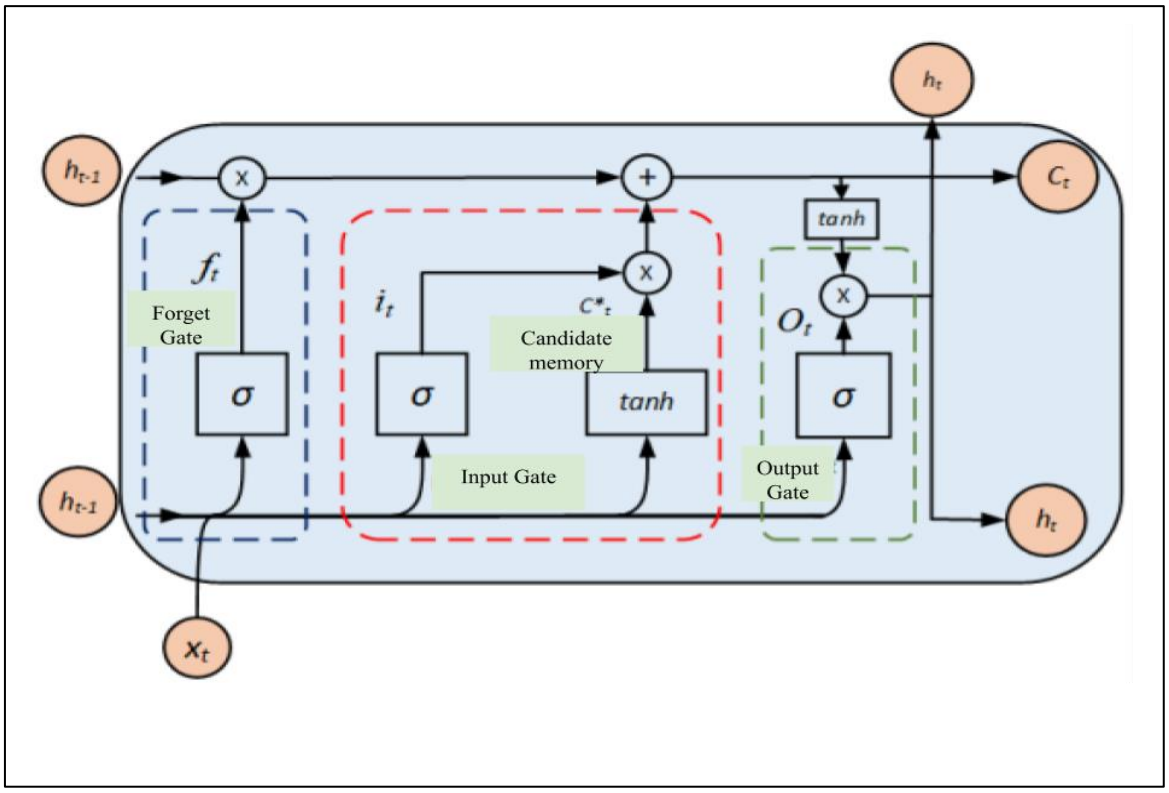


Figure 1.2. LSTM layers architecture

By processing input data in both forward and backwards directions, Bidirectional Long Short-Term Memory (BiLSTM) networks, which build upon LSTMs, further enhance sequential modelling by resolving ambiguity by considering both preceding and succeeding context. This is helpful in text-to-speech synthesis and sentiment classification to comprehend sequential data relationships and attain greater accuracy. Two LSTM layers are used; one reads the sequence from the future to the past, and the other from the past to the future. BiLSTMs capture a more thorough depiction of context. This bidirectional processing improves efficiency in tasks like machine translation, named entity identification, and sentiment analysis that call for a deeper comprehension of dependencies [36]. Figure 1.3. The BiLSTM improves sequence modelling by processing input data in forward and backwards directions. The Forward LSTM propagates input data from past to future

($x_{t-1} \rightarrow x_t \rightarrow x_{t+1}$)The Backwards LSTM processes the sequence in the reverse direction ($x_{t+1} \rightarrow x_t \rightarrow x_{t-1}$). Where t represents the time of the sequence data, the hidden states are generated by both binary directions, forward and backwards LSTMs. The output state incorporates information from both the past and the present. ($y_{t-1} \rightarrow x_t \rightarrow y_{t+1}$)and future ($y_{t+1} \rightarrow y_t \rightarrow y_{t-1}$)Contexts. The BiLSTM network is beneficial for sequential data tasks and natural language processing (NLP), where contextual awareness is essential.

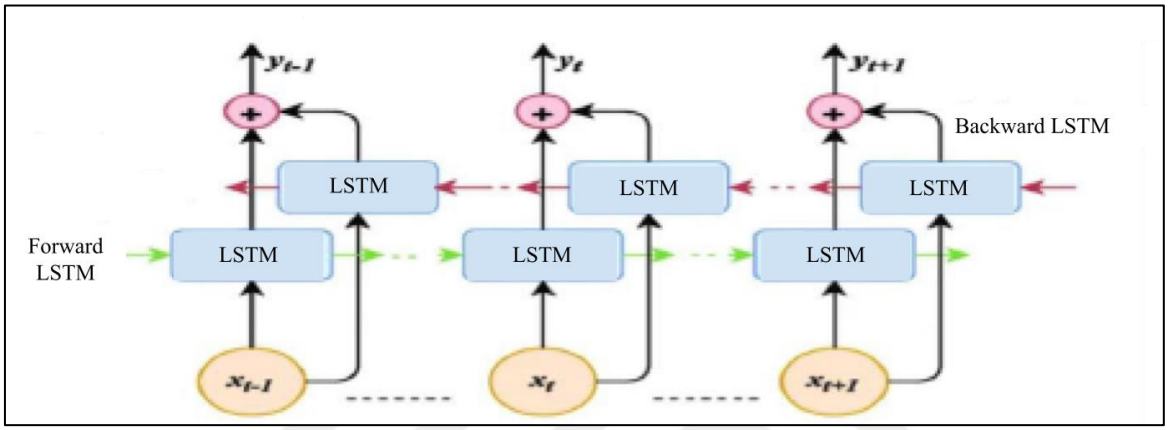


Figure 1.3. Representation of the BiLSTM

A common problem in machine learning, particularly in its application within e-commerce, is that models are prone to overfitting. Overfitting occurs when a model captures randomness or irregularities in the training data, resulting in excellent performance on the training set but poor generalisation to unseen data, such as the validation set. This issue is especially critical for e-commerce platforms, where recommendation systems depend on models to analyse diverse and noisy user-generated data, including reviews and ratings. Models that overfit may struggle to accurately predict the helpfulness of reviews, undermining their ability to enhance user experience and inform purchasing decisions. Rice et al. [37] emphasised that overfitting affects model robustness and limits its capacity to generalise to adversarial or complex real-world scenarios. Thus, addressing overfitting is essential for ensuring the reliability and scalability of machine learning models in e-commerce applications. The accuracy of the recommendation model plays a crucial role in determining the model's success or failure. If the product the system recommends aligns with the user's preferences, the user may develop trust in the system. Consequently, achieving high accuracy and robust performance in text classification is of utmost importance. However, similarity-based algorithms encounter serious challenges, such as overfitting and observation sparsity, which hinder the development of recommender systems [38].

Recommender systems often struggle with limited user-item interaction data and diverse user preferences, which can reduce the accuracy of traditional models. To address these challenges, transfer learning leverages insights from related tasks or domains, enhancing recommendation quality despite sparse data. This approach is particularly valuable in e-commerce, where personalized, relevant suggestions are essential for user satisfaction. Zhang et al. [39]highlighted its potential to improve recommender systems by utilising pre-trained models and fine-tuning them for domain-specific applications. However, despite its promise, effectively transferring knowledge while addressing domain discrepancies and ensuring scalability remains a significant challenge for e-commerce platforms. This paper proposes a new model to improve accuracy, eliminate the issues, and simultaneously address both challenges.

The Yelp Academic Review Dataset and Amazon Review Dataset serve as well-established benchmarks for research in e-commerce recommendation systems, particularly in analysing customer reviews and ratings. A comprehensive collection of customer reviews from the Amazon Review Dataset was developed to support sentiment analysis, recommendation systems, and natural language processing (NLP) research. Product and user IDs, review content, star ratings, review summaries, helpfulness votes, and timestamps are among the structured data it contains. the study has made substantial contributions to the dataset, which has been made available in several editions. This data set was created primarily to assess customer feedback, refine text classification algorithms, detect fraudulent reviews, and improve personalised recommendation models. Researchers frequently use it to train deep learning models like XLNet, GPT-2, and TFRobertaForSequenceClassification, which allow for more precise sentiment categorisation and functional review prediction. Additionally, the dataset is a valuable tool for temporal analysis, enabling researchers to examine customer satisfaction patterns across time. The Amazon Review Dataset remains one of the most significant benchmarks for AI-driven review analysis and recommendation system development due to its vast size and diversity across product categories [40]. Likewise, the Yelp Academic Review Dataset, available from Yelp since 2013, encompasses structured information on businesses, including customer-written reviews, star ratings, business attributes, and user interactions. This dataset is invaluable for studying consumer behaviour, sentiment analysis, and personalised recommendation models within local businesses, such as restaurants and retail stores [41]. In our research, we selected these datasets to develop a rating-based recommendation system that employs natural language processing (NLP) and

deep learning models to extract meaningful insights from textual reviews and numerical ratings. By leveraging these datasets, we aim to enhance personalised product and service recommendations, ensuring that users receive relevant suggestions based on prior customer feedback. Furthermore, analysing sentiment trends and review helpfulness enables more accurate predictions of user preferences, contributing to developing more effective and customer-centric e-commerce platforms.

Problem statement

Recent studies on review-based personalised recommendation systems have concentrated on the fact that they frequently include irrelevant or deceptive information that could influence consumers' buying decisions, such as ads or phoney reviews. Therefore, it is indisputable that offering suggestion services without considering review quality could decrease recommendation performance. Furthermore, it would significantly enhance purchasing selections to be able to suggest the finest favourable evaluations for a specific product. Additionally, overfitting is a prevalent issue with recommendation models, including their use in e-commerce. Overfitting is the process by which a model performs exceptionally well on the training set but poorly on unseen data, like the validation set, because it captures unpredictability or inconsistencies in the training data. In e-commerce platforms, where recommendation systems use models to examine noisy and varied user-generated data, such as reviews and ratings, this problem is significant. The capacity of reviews to improve user experience and guide purchase decisions may be compromised by overfitting models, which may not be able to forecast reviews' helpfulness. Scientific/artistic problems to be solved should be defined in detail. The literature is scanned for this by revealing previous studies' relationships, similarities, and differences. Conceptual frameworks, methods, techniques, and paradigms should also be utilised.

Objectives

Our research employs an optimal deep model to analyse customers' reviews based on their comments and differences in preferences. To learn the content of reviews (helpful or unhelpful) with high accuracy without noise, enhance product review recommendations to improve buying decisions, eliminate unhelpful reviews for purchasing decisions, like advertising reviews, false, fake, or not helpful reviews, and help the buyer to make the right

decision in a short time. The following challenges will be addressed in our research: 1) How can we predict ratings based on the differences in the customers' tendencies? 2) Which rating value is most useful for showcasing the helpful review? 3) Performance of which deep learning techniques are best suited to analyse the text of a review? 4) Which transfer learning and data augmentation algorithms are used to improve model performance while saving both time and resources?

This document's remaining sections are arranged as follows: Related research on various methods based on rating and review deep recommendation systems is covered in Sect. 2. The proposed strategy and model were explained in the third section. In the fourth part, the description of the implementation of this study included details of the classification process of the customers' reviews. Experimental results were explained in the fifth section. The discussion and conclusion were included in the sixth section of the thesis. Finally, the last section shows limitations and how we can mitigate these challenges in further work.

2. LITERATURE REVIEW

As e-commerce continues to expand in the post-COVID era, online retailers are increasingly seeking innovative and efficient strategies to deliver a dynamic and personalised shopping experience that adapts to consumers' evolving needs and objectives. The development of advanced methodologies for generating recommendations has significantly enhanced the ability of e-commerce platforms to increase shopper profitability while simultaneously saving users' time. These approaches enable users to access the products they seek with minimal effort, efficiently, and promptly. Data mining, a critical technique, involves analysing large datasets to extract meaningful insights or identify relationships among items. Recommender systems, widely recognised as prominent data mining applications, leverage these insights to provide users with personalised and relevant product suggestions [42].

Recommendation systems have been fundamental to early e-commerce platforms, including Amazon, Yelp, and CDNOW. These systems were initially developed to address the critical need to assist users in discovering suitable products through recommendations and customer feedback. Over time, Amazon has pioneered implementing various recommendation techniques, ranging from algorithmic methods, such as collaborative filtering, to non-algorithmic approaches, including non-personalised recommendations. The overarching goal of recommendation systems is to achieve large-scale personalisation by offering services adapted to the specific needs of individual users. To this end, Amazon meticulously records and analyses customer actions to enhance its recommendation algorithms [43]. Moreover, personalisation is seen as a way to build long-term customer relationships. Recommendation systems are still critical in e-commerce today and continue to grow.

2.1. Types of Deep Learning Techniques for E-Commerce Recommendation Systems

In the context of recommendation systems for e-commerce, deep learning techniques are tailored to provide personalised and efficient recommendations. Deep learning is a subset of machine learning, and the different kinds of learning in deep learning are based on how models are trained with data. These techniques use other learning approaches, depending on the data availability and the problem's nature. By selecting the proper learning method (or a combination), e-commerce platforms can provide personalised recommendations that boost user engagement and sales.

2.1.1. Supervised learning

Supervised learning is essential in recommendation systems, especially when labelled data is available, like user-item interactions with explicit feedback (likes, purchases, or numerical evaluations). The main goal is learning a mapping function that anticipates user preferences for unseen things to provide tailored suggestions. The Neural Collaborative Filtering (NCF) framework is grounded in supervised learning and leverages labelled data, such as user-item interaction data (e.g., ratings and clicks). In this context, Positive Interaction (1) denotes instances where a user has engaged with or expressed a preference for an item (e.g., by viewing, purchasing, or providing a positive rating), while Negative Interaction (0) refers to situations where the user has not interacted with or has shown disinterest in the item (e.g., by not viewing, purchasing, or rating it). These labelled interactions train neural networks for modelling the underlying relationships between users and items [16]. On the other hand, [44] introduced the concept of interaction-based neighbourhood modelling. Instead of treating all interactions equally, this model groups users or items into local neighbourhoods based on their interaction patterns. Aimed to enhance recommendation quality by accounting for local relationships, which may be more beneficial in datasets with localised user behaviour or preferences. [45] The ONCF model was further built on NCF by introducing a more expressive interaction model based on outer products. This model is explored using outer product embeddings to capture more complex interactions between user-item pairs in the collaborative filtering task. The outer product allows for richer interactions by extending the interaction space, allowing for better feature representations in the learned embeddings. This method improves the modelling of user-item interactions by using a higher-order interaction model compared to previous models. While ONCF offered a more expressive approach to modelling user-item interactions through outer product embeddings, the model may struggle to adapt to dynamic user-item interactions where the relationships between users and items evolve. ONCF assumes static embeddings, which might not capture temporal changes effectively. By integrating reliability estimates into the recommendation process, the [46] model enhanced the interpretability and trustworthiness of collaborative filtering systems, addressing a significant gap in traditional NCF approaches. Sreepada et al. (2018) employed a heuristic, tendency-based collaborative filtering method closer to unsupervised learning or statistical modelling. The focus is leveraging deviations in user-item tendencies to make recommendations without requiring labelled data or explicitly optimising a supervised loss function [17].

2.1.2. Unsupervised learning

Considering user behaviour and product attributes, unsupervised learning in recommendation systems finds patterns, structures, or clusters without needing explicitly labelled data. Rather than being trained using preset user-item interactions or ratings, unsupervised learning algorithms group comparable individuals, products, or behaviours by analysing underlying relationships in the data. For this reason, it is beneficial for exploratory analysis and recommendation-making when labelled data is limited or inaccessible. İn reference [47] A behaviour-based recommender system that leverages customers' natural behaviours, such as browsing and clicking, on e-commerce platforms. The methodology involves clustering active customers, determining neighbourhoods, and calculating product reputation based on similar users. An unsupervised clustering approach based on product categories is developed to enhance recommendation accuracy. The study presented a unified matrix factorisation method based on dynamic multi-view clustering (MFDMC) and employed an end-to-end training paradigm [48]. The approach enhanced the interpretability of user/item representations and improved recommendation performance in real-world datasets.

2.1.3. Semi-supervised learning

Semi-supervised learning is helpful in e-commerce applications where user interaction data is frequently sparse or partial, since it uses labelled and unlabeled data to enhance model performance when recommending things. In many real-world situations, only a tiny percentage of user-item interactions have clear labels, like reviews or purchase confirmations, while most data, including browser history, clicks, or time spent on product sites, are left unlabeled. To overcome this difficulty, semi-supervised learning uses the patterns and structures found in the unlabeled data to improve the prediction accuracy of recommendation models. The approach that combined item side information and labelled complementary item pairs to generate practical recommendations for cold items. The technique used semi-supervised adversarial learning to enhance recommendation quality in e-commerce settings [49]. In [50], a probabilistic chain graph model was proposed that integrated semi-supervised learning with latent factor models. The approach addressed rating sparsity in recommendation systems by combining Bayesian networks and Markov random fields, improving performance in sparse data scenarios.

2.1.4. Self-supervised learning

One method for improving recommendation algorithms in e-commerce platforms is self-supervised learning. This learning technique allows models to learn strong user and item representations without needing explicit labels by immediately producing supervisory signals from the data. Sequence recommendation is studied using the dual-tower retrieval architecture. Self-supervised multi-modal pretraining aligns different item feature combinations, enhancing generalisation across a range of e-commerce datasets [51]. By automating data augmentation procedures, the study [52] presented a unified framework that uses self-supervised learning (SSL) to improve recommendation systems. This improves the resilience and quality of learned user-item representations. This method produced strong embeddings for recommendation using pretext tasks, including applying contrastive learning or predicting missing interactions. The approach has drawbacks, such as restricted generalizability across various e-commerce datasets, possible overfitting to supplemented data, and scalability problems brought on by the processing demands of SSL and automated augmentation techniques. Furthermore, the system relies mainly on creating successful pretext tasks, which may not always produce the best outcomes, and struggles to handle cold-start issues for new users or objects. To improve recommendation accuracy in e-commerce platforms, the study [53] developed a social recommender system to maintain beneficial social interactions and permit tailored cross-view knowledge transfer through adaptive semantic alignment. The reliance on high-quality social data, scalability problems, domain specificity, cold-start difficulties for users with little social interaction, evaluation bias, and possible overfitting in self-augmented learning, even though it provided a novel framework for incorporating social relations into recommendation systems. These restrictions must be addressed for wider usage in e-commerce platforms.

2.1.5. Multi-task learning

Multi-task learning is a sophisticated deep-learning technique that trains a single model to do several tasks simultaneously. With this method, the model may use common representations across functions, improving generalisation and learning efficiency. Multi-task learning lowers the chance of overfitting and improves overall model performance by finding and exploiting task similarities. To provide a balanced learning process that takes into consideration the goals. İn the J-NCF study a multi-task learning approach is used to

combine explicit feedback (rating prediction) and implicit feedback (ranking) into a single neural architecture [54].

2.1.6. Hybrid models

Hybrid recommendation systems integrate multiple learning approaches to enhance recommendation quality by leveraging diverse data sources and methodologies. In e-commerce, these systems combine collaborative filtering, content-based filtering, and sequential modelling to address challenges such as the cold-start problem and dynamic user preferences. A hybrid recommendation system that combined collaborative filtering with content-based filtering using a weighted stochastic block model, addressing the cold-start problem and improving accuracy in e-commerce recommendations [55] Stochastic block models, especially when weighted and hybridised, can be complex, making it difficult for practitioners to interpret the recommendations or understand the rationale behind predictions. .A hybrid neural collaborative filtering (HNCF) model addressed key challenges in recommendation systems by integrating content-based and collaborative filtering techniques with deep learning. Including external features and using a multi-layer perceptron makes it a robust and scalable approach capable of generating accurate and diverse recommendations. The framework is indeed hybrid learning due to its combination of multiple strategies and methodologies [56]. The study [57] proposed a hybrid graph neural network model that captured user multi-behaviour information for improved recommendations. Despite integrating various user interactions to enhance recommendation accuracy in e-commerce platforms, Graph neural networks are computationally expensive, especially when applied to large-scale e-commerce datasets with millions of users and items.

2.1.7. Transfer learning

Representation transfer, which applies pre-trained models to domain-specific recommendation tasks, has been shown to improve recommendation accuracy by fine-tuning transformer-based models such as BERT and GPT to encode textual information like user reviews and item descriptions[58]. This approach addresses key challenges of traditional deep learning recommenders, including heavy reliance on large training datasets and computational resources, especially under sparse user-item interactions.

However, its effectiveness in e-commerce depends on overcoming issues such as data dependency, domain mismatch, overfitting, and the high computational cost involved. Transfer learning also helps mitigate biases in collaborative filtering for domains with limited interaction data [59], but scalability remains a challenge for large, heterogeneous e-commerce systems, particularly when high-quality source data is scarce.

To tackle these limitations, integrated frameworks combining pre-trained model adaptation with embedding techniques have been proposed to enhance scalability and address cold-start and sparsity problems [60]. Moreover, hybrid methods that unify fine-tuning strategy with active learning show improvements in data efficiency and accuracy in cross-system recommendations, though challenges around domain alignment, scalability, and computational complexity still require further investigation [61].

our research differs from the existing works as it only recommends informative and useful items to improve customer satisfaction when choosing products and reduce the amount of time to research any product using a combination of transfer learning, hybrid deep learning with multi-task Learning, supervised learning as statistical methods from using tendencies collaborative filtering algorithm(TCF), and data augmentation while handling multiple objectives (review classification and recommendation).

2.2. Customers' Review Analysis in E-commerce with Recommendation System

Customer review analysis in e-commerce makes it possible to understand customers' opinions, refine items, and improve the whole buying experience. Review analysis can result in more precise and individualized product recommendations when paired with a recommendation system. Several forms of customer review analysis are frequently employed in e-commerce with recommendation systems, including the following:

2.2.1. Sentiment analysis

SA (Sentiment Analysis) determines the overall sentiment (positive, negative, neutral) of a review. it is used in the recommendation System to help the system understand the general mood of reviews for a product, which can influence product recommendations. For example, products with positive sentiments may be recommended more frequently. This is the

overarching task focused on understanding and classifying the sentiment of text (e.g., positive, negative, neutral) in e-commerce or other domains. It involves analysing user reviews, feedback, or other textual data to gauge sentiments. İt is an activity of Natural Language Processing (NLP), which is the process of extracting and evaluating people's opinions.

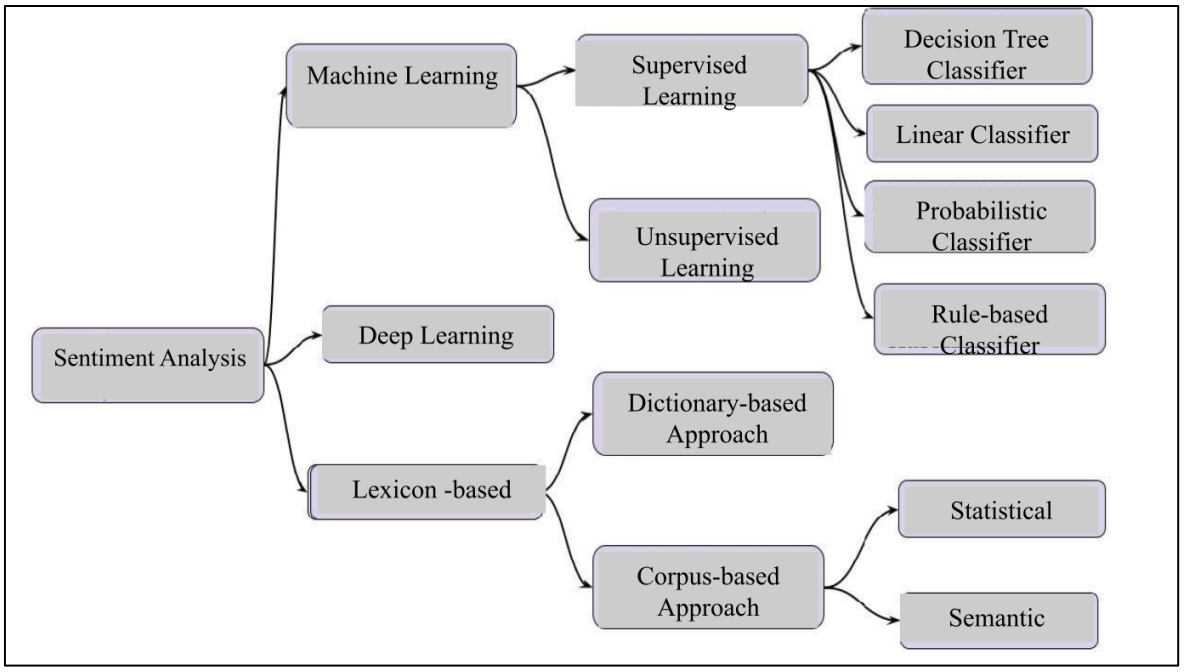


Figure 2.1. İllustratites the categorization of sentiment analysis techniques

Sentiment analysis is classified into three primary methods for text analysis, as illustrated in Figure 2.1. above. Most methods within the machine learning approach fall under supervised learning, while deep learning techniques are used to recognise patterns and establish connections during text analysis. Lastly, the semantic approach is often applied as a subset of the lexicon corpus-based method.[62].

2.2.2. Sentiment analysis focused on specific aspects (ABSA)

This approach is focused on extracting specific aspects or features mentioned in reviews (e.g., product quality, customer service, delivery speed) and evaluating sentiment on each. The recommendation system can make product recommendations based on consumer preferences for particular characteristics by determining whether product components are liked or disliked. Weiqianli and Bugaoxu (20202) used a specific application of the ABSA approach, tailored for the fashion domain. It leverages reviews to extract and evaluate

sentiments to particular aspects, using these insights to improve recommendation quality. Attention mechanisms further enhance the model’s ability to focus on the most meaningful content in reviews [25].

2.2.3. Topic modelling

The topic modelling identifies recurring topics, such as "battery life," "size," or "colour," from textual data. These themes can be used to recommend products that fit a user's preferences in the context of recommendation systems. For example, by using the identified subjects, the system can suggest related products to a user who has previously bought items related to particular themes, increasing the suggestions' relevancy and personalization. Shafqat and Byun (2019) used topic modelling and sentiment analysis to suggest a recommendation system for promoting underutilised tourist destinations. They used Latent Dirichlet Allocation (LDA) to discover important features of tourist sites by extracting themes from reviews and social media posts. Opinions on these factors were assessed using sentiment analysis, which enabled the system to identify and rate the attractiveness of lesser-known attractions. This strategy promotes sustainable tourism by balancing the distribution of visitors, raising awareness of underappreciated places, and showing how natural language processing (NLP) can be used to improve recommendation systems. The combination of topic modelling and sentiment analysis shows promise for identifying under-emphasised tourist spots but is limited by data quality, model constraints, and computational demands. Advanced techniques, such as transformers or multimodal analysis, could improve system robustness and effectiveness [64].

2.2.4. Text classification

The text classification classifies reviews into predetermined groups (e.g., verified purchase/non-verified, helpful/unhelpful). Reviews categorised as "helpful" or originating from "verified purchases" may be given higher weight in the recommendation system to boost their credibility. The paper [65] introduced a model based on Amazon and Yelp datasets aimed at automatically predicting the helpfulness of online reviews. A customer's expectation of a review is significantly influenced by the review's sentiment and the customer's awareness of relevant product information. An attention-based model is adapted to evaluate review helpfulness by considering these factors to address this challenge. The

model's effectiveness is demonstrated through experiments, highlighting its potential to enhance the identification of helpful reviews in e-commerce platforms. This model for review helpfulness evaluation is effective but does not account for external factors, such as user-specific preferences or cultural differences, which may influence perceptions of helpfulness.

2.2.5. Review rating prediction

Predicting a customer's potential rating for a product based on their past behaviour and reviews is integrating review rating prediction into recommendation systems. The effectiveness and personalisation of recommendations can be increased by combining this predictive ability with collaborative filtering techniques to make product recommendations that a user will likely find appealing. Collaborative filtering methods, like matrix factorisation, for example, examine user-item interaction patterns to find commonalities and forecast user preferences The system may recommend products that closely match the user's expected ratings by fusing these predictions with collaborative filtering, which will increase user engagement and happiness [66] Although review rating prediction offers valuable information for recommendation systems, these drawbacks might be addressed to improve the method's resilience and adaptability, for example, by implementing hybrid techniques, explainable models, or real-time learning.

2.2.6. Review trend analysis

Recommendation systems that use review trend analysis must track changes in consumer sentiment and product reviews over time. Products with improved feedback are promoted, while those with dropping sentiment are demoted, allowing the system to adapt its suggestions based on temporal trends in reviews, such as changes in sentiment from positive to negative. This dynamic modification aligns recommendations with current customer attitudes, improving their relevance and reliability. For example, using temporal sentiment analysis, product reviews have been clustered over time, which helps to understand how customer impressions change over time [67]. Reviewing trend analysis is valuable for tailoring suggestions to change consumer preferences. Still, its efficacy and dependability in practical applications depend on resolving issues with data quality, model accuracy, and

computing scalability. Advanced contextual models and hybrid strategies can lessen these restrictions.

2.2.7. Personalised review summarisation

Personalised review summarisation in recommendation systems involves condensing user reviews into concise formats, highlighting key points, such as pros and cons, tailored to individual preferences. This approach lets users quickly grasp product strengths and weaknesses, facilitating informed decision-making. Recommendation systems can leverage these personalised summaries to suggest products that align with users' interests and needs. To enhance tailored review summarisation in recommender systems, the creation of a Transformer Reasoning Network has also been investigated [68]. Also, [69] An investigation proposed a pre-trained personalised review summarisation method that effectively integrates personalised information of users and products into the salience estimation of input reviews. Still, challenges like scalability, bias, and loss of context must be addressed to ensure reliable and effective summaries. We should adopt explainable AI and hybrid summarisation models to guarantee accurate and useful summaries.

2.2.8. User-generated content analysis

User-generated content (UGC) such as videos, photos, and Q&A sections is used in recommendation systems to improve content analysis and product authenticity assessments. According to research, recommendation algorithms can adapt suggestions to each customer's unique profile by analysing user-generated content (UGC) in conjunction with browsing history and purchase information. This improves the accuracy of recommendations by retrieving preference information [70]. A promising combination of user-generated content (UGC) and augmented reality (AR) in recommender systems was highlighted by González et al. (2021); nevertheless, its applicability is limited by some issues. First, the method uses much computing power, especially when processing multimedia-rich and varied user-generated content (UGC) in real time, like text, photos, and videos. Scalability is another issue since the system might not be able to manage the constant flow of big datasets produced by large user bases. The system's success is further complicated by its reliance on the calibre and applicability of user-generated content (UGC), since biased, noisy, or incomplete content can decrease the accuracy of recommendations. The gathering and analysing of

sensitive user data raises further privacy and security concerns, necessitating strict compliance with data protection regulations. In addition, UGC frequently incorporates context-dependent and subjective information, which makes it challenging to generalise suggestions across a range of users. The AR integration adds more limitations, requiring technological know-how and suitable devices, which may restrict acceptance and accessibility. The framework's usability and resilience could be significantly improved by addressing these constraints with sophisticated multimodal learning approaches, privacy-preserving mechanisms, and scalable architectures.

2.2.9. Trustworthiness analysis

In recommendation systems, trustworthiness analysis quantifies the dependability of reviews by considering user reputation, review length, and consistency. By giving more weight to deemed reliable evaluations, these systems hope to improve users' decision-making ability. According to the study [71], collaborative filtering recommendations can be enhanced by examining how variations in review attributes, such as length and rating, impact perceived helpfulness. The research also investigated combining trust indicators, such as user reputation, to improve recommendation performance. Although functional, trustworthiness analysis in recommendation systems has some drawbacks. The subjective nature of elements like review length and consistency, which may not necessarily correlate with trustworthiness, makes it difficult to evaluate the reliability of reviews accurately. Metrics measuring user reputation frequently depend on past data, which might not fully represent the reviewer's present actions or intentions, particularly regarding recently enrolled users or intentional manipulation. Users may be discouraged from engaging due to privacy concerns over the system's dependence on metadata, including user profiles and activity patterns. Furthermore, even sophisticated detection algorithms can be circumvented by fraudulent or misleading reviews designed to look genuine, which diminishes the effectiveness of trustworthiness assessments. The method also has trouble with cultural and contextual variations since generalisation is made more difficult by the possibility that different locations or domains have different standards for reliability. The scalability of the system in settings with large volumes of review data may be impacted by the increased computational complexity of implementing strong trustworthiness checks. Integrating cutting-edge machine learning methods, such as deep learning and multimodal analysis, is necessary to

overcome these constraints and improve the precision and resilience of trustworthiness assessments.

2.3. User Behaviour Analysis in Recommender Systems

Recommender systems have become essential to modern digital platforms, providing personalised content suggestions based on user preferences. These systems analyse user behaviour, including interactions, ratings, and browsing history, to generate accurate and relevant recommendations. Understanding user behaviour is critical in enhancing the performance of recommender systems, as it helps address challenges such as accuracy, data sparsity, and the cold-start problem. Deep learning has significantly advanced the capabilities of recommender systems by leveraging sophisticated algorithms to extract meaningful patterns from large-scale user data. Techniques such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRUs), Deep Matrix Factorisation (DMF), and Deep Neural Networks (DNNs) have been widely adopted to improve recommendation accuracy. Furthermore, feature extraction techniques like autoencoders and attention mechanisms enhance the system's ability to analyse user interactions effectively. By integrating these deep learning approaches, recommender systems can better understand user preferences, refine ranking algorithms, and improve the overall user experience. Various methodologies are used to analyse user behaviour in recommender systems. It examines the role of deep learning models in processing user-generated data, addressing key challenges, and optimising recommendation performance. Additionally, it discusses how behavioural insights contribute to refining personalisation strategies, ensuring a more accurate and dynamic recommendation process.

2.3.1. Studies on the application of CNN

Convolutional Neural Networks (CNNs) have emerged as powerful tools for analysing user behavioural trends in e-commerce recommender systems, especially when visual and contextual information is crucial. Traditional collaborative filtering methods often struggle with the cold-start problem and fail to capture complex relationships between users and products. By leveraging CNNs, recommender systems can extract deep feature representations from product images, user interactions, and behavioural data, enhancing

recommendation accuracy. The visual appearance of clothing products conveys valuable information, which has led to an increased focus on visual recommendation systems in recommendation system development. The Style-aware Bayesian Personalised Ranking (SBPR) model was introduced to capture the relationship between users and items, enabling the representation of user preferences at the style level. Each dataset was processed by extracting implicit feedback, and the relationships between specific channel responses in each layer of the network were analysed using a convolutional neural network (CNN) based on VGG architectures. The VGG-19 model, consisting of 16 convolutional layers and five pooling layers, was employed, with the max-pooling layers replaced by average-pooling layers. The model was trained using data from Amazon.com, Tradesy.com, and a second-hand clothing exchange platform, focusing on clothing, footwear, and accessories categories. Empirical evaluations on challenging public datasets demonstrate the effectiveness of this recommendation framework in capturing user style preferences and addressing the cold-start problem [72].

To infer latent user preferences, CNN-based models have been used to analyse browsing patterns, purchase histories, and engagement metrics [73]. Moreover, integrating CNNs with knowledge graphs and attention mechanisms enables personalised and context-aware recommendations, improving user satisfaction and engagement. The ability of CNNs to process high-dimensional visual data makes them particularly effective for fashion, electronics, and home decor recommendations, where aesthetics significantly influence purchasing decisions. Thus, CNN-based user behaviour analysis has become crucial in modern e-commerce recommendation systems, enhancing predictive accuracy and addressing key challenges such as sparsity and cold-start issues.

2.3.2. Studies on the application of RNN

Recurrent Neural Networks (RNNs) are a class of neural networks characterised by their ability to process sequential data through bidirectional information flow, allowing data to propagate from deeper layers to initial layers. Unlike traditional feedforward networks, RNNs incorporate recurrent connections, enabling them to retain information over time. This memory mechanism allows the network to consider a specific number of previous states at a given time step$t$ t, making RNNs particularly well-suited for tasks involving temporal dependencies. Due to their ability to model sequential patterns, RNNs more closely resemble

the functioning of the human nervous system, where information processing is not strictly unidirectional. As a result, RNNs are particularly well-suited for context-dependent applications, especially those involving the processing of temporal sequences where data points are interdependent rather than isolated. Examples include sequential learning and speech recognition, where the information order is crucial. However, for tasks requiring the retention of long-range dependencies, such as video sequence classification, the inherent "short-term memory" of traditional RNNs proves insufficient. Standard RNN architectures, often called vanilla RNNs, are limited in their ability to retain information over extended sequences, typically beginning to lose memory after approximately fifty iterations. Furthermore, the bidirectional flow of information in RNNs complicates the training process, making optimisation more challenging. More advanced architectures, such as Long Short-Term Memory (LSTM) networks, have been developed to address these limitations, offering improved mechanisms for capturing long-range dependencies in sequential data. These large-scale "transient memory" networks have significantly transformed machine-based speech recognition and the interpretation and generation of text in natural language processing (NLP). From a theoretical standpoint, RNNs possess substantially greater potential than traditional neural networks, as they can theoretically approximate any computational function. However, practical implementation remains a challenge despite this theoretical flexibility, as there is no straightforward methodology for structuring RNNs to perform such functions effectively [74].

2.3.3. Studies on the application of LSTM

The Long Short-Term Memory (LSTM) model consists of four primary layers: an embedding layer, an adaptive attention layer, an LSTM layer, and an output layer. The embedding layer constructs a memory matrix at sequence time by encoding the user’s basket information into two-dimensional levels. The adaptive attention layer comprises intra-basket and inter-basket attention mechanisms. The intra-basket mechanism is designed to capture the diversity of user interests within a single basket, while the inter-basket mechanism represents variations across multiple baskets over time. This mechanism allows the model to identify items closely related to the predicted one using a max-pooling function. The LSTM algorithm is utilised to dynamically adjust candidate items within inter-basket and intra-basket adaptive user representations. The model aggregates relevant elements in the

output layer that characterise a user's preferences. The softmax function is then applied to predict the following item to be purchased in a future basket [75].

To address these challenges, Liu et al. (2020) introduced the Next Basket Attribute-Aware Multi-Level Attention-Based Recommendation Model (NbRAM), incorporating a sequence layer using LSTM. This model was tested using the same dataset from a previous study (Ta-Feng and JingDo) and evaluated using standard performance metrics, adding precision and HitRate measures. Experimental results demonstrated that NbRAM outperformed the earlier IIAAN approach in terms of recall (0.227), F1-score (0.125), NDCG (0.167), precision (0.087), and HitRate (0.291), particularly for N-Top = 5 when applied to the JingDo dataset [76]. Figure 2.2. presents the general architecture of the NbRAM model.

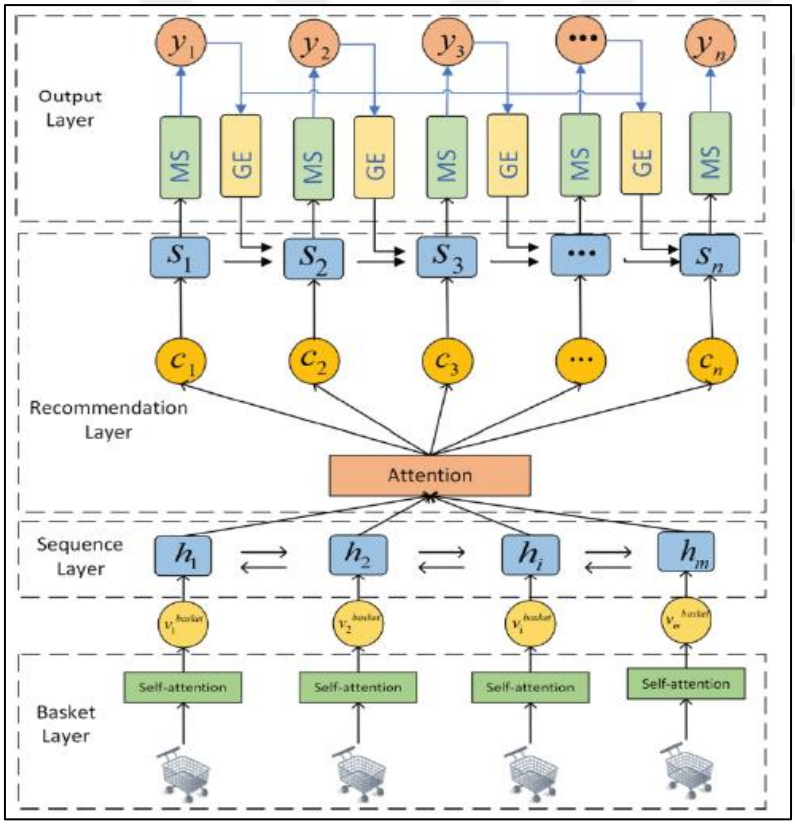


Figure 2.2. Attention-based LSTM model [73]

2.3.4. Studies on the application of autoencoders

A recent study employed a knowledge graph to tackle the cold-start problem and improve the accuracy of recommendation systems. Furthermore, implementing an Autoencoder Neural Network in the hidden layer effectively reduced the issue of lack of data by

leveraging the word2vec approach. The connections between the knowledge graph and Autoencoder nodes were labelled using a Semantics-Aware Autoencoder (SemAuto). The study thoroughly analyses the proposed method and demonstrates how the SemAuto model can generate content-based descriptions for recommended items. Both point-based and binary description styles were explored, utilising various types of information available in DBpedia. The study further investigates how the efficiency of the proposed descriptions varies based on the kind of information—categorical or factual—used to train the SemAuto model [77]. A denoising autoencoder (DAE) was utilised in a recent study to enhance recommendation accuracy. Specifically, the study employed user rating trend information to predict ratings for the top$N$ recommended items using the Rating-Trend Collaborative Denoising Autoencoder (UT-CDAE) framework. Encoding user rating trends is critical when addressing incomplete or noisy data. The model incorporates two additional node vectors within the DAE architecture to capture user preferences effectively: the High Tendency User Node (HTUN) and the Low Tendency User Node (LTUN). Low-value ratings are encoded using a single node vector, whereas high-value ratings are represented through the combined participation of both node vectors. The study evaluates the model using various performance metrics for top$N$ recommendation accuracy, including precision ($P@N$), recall ($R@N$), mean average precision (MAP), normalised discounted cumulative gain (NDCG), NDCG@N, and mean reciprocal rank (MRR). The UT-CDAE model was compared against the standard DAE and the Collaborative Denoising Autoencoder (CDAE) using the ML-100K and ML-1M datasets. Furthermore, its performance was benchmarked against three recent deep learning-based recommendation models—Caser, Neural Collaborative Filtering (NCF), and Enhanced Collaborative Autoencoder (ECAE)—on the ML-1M dataset. The results demonstrate that, compared to existing methodologies, the UT-CDAE framework improves performance and robustness in generating top$N$ recommendations.

2.3.5. Studies on the application of NLP

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language. NLP techniques integrate computational linguistics with machine learning and deep learning approaches to process textual and spoken data. Key applications of NLP include sentiment analysis, machine translation, named entity recognition, and text summarisation [77]. Traditional

methods, such as rule-based systems and statistical models, have been complemented by advanced deep learning architectures, including Transformer-based models like BERT and GPT, significantly enhancing the ability to capture contextual meaning. Additionally, NLP plays a crucial role in recommender systems, chatbots, and voice assistants by facilitating more natural and personalised interactions between users and AI-driven applications. As research advances, integrating NLP with knowledge graphs and multimodal learning is expected to improve the accuracy and efficiency of language-based AI systems. A recent study employed Term Frequency-Inverse Document Frequency (TF-IDF), a vectorisation technique commonly used in natural language processing (NLP), to develop a bidirectional personal review generation model. The model utilised LSTM, RNN, encoder-decoder architecture, and attention mechanisms to generate explanations for recommendations. The study used two datasets from the Amazon 5-star review (Video and Toys) and two additional datasets (Beer and Yelp). The experiment was evaluated using several performance metrics, including Root Mean Square Error (RMSE) for assessing recommendation rating accuracy, Normalised Discounted Cumulative Gain at rank K (NDCG@K) for evaluating predicted ranking performance, and Delta Root Mean Square Error (DeltaRMSE to measure improvements in machine-generated reviews. Figure 2.3. illustrates the different phases involved in text processing within the CNN-LSTM model [78].

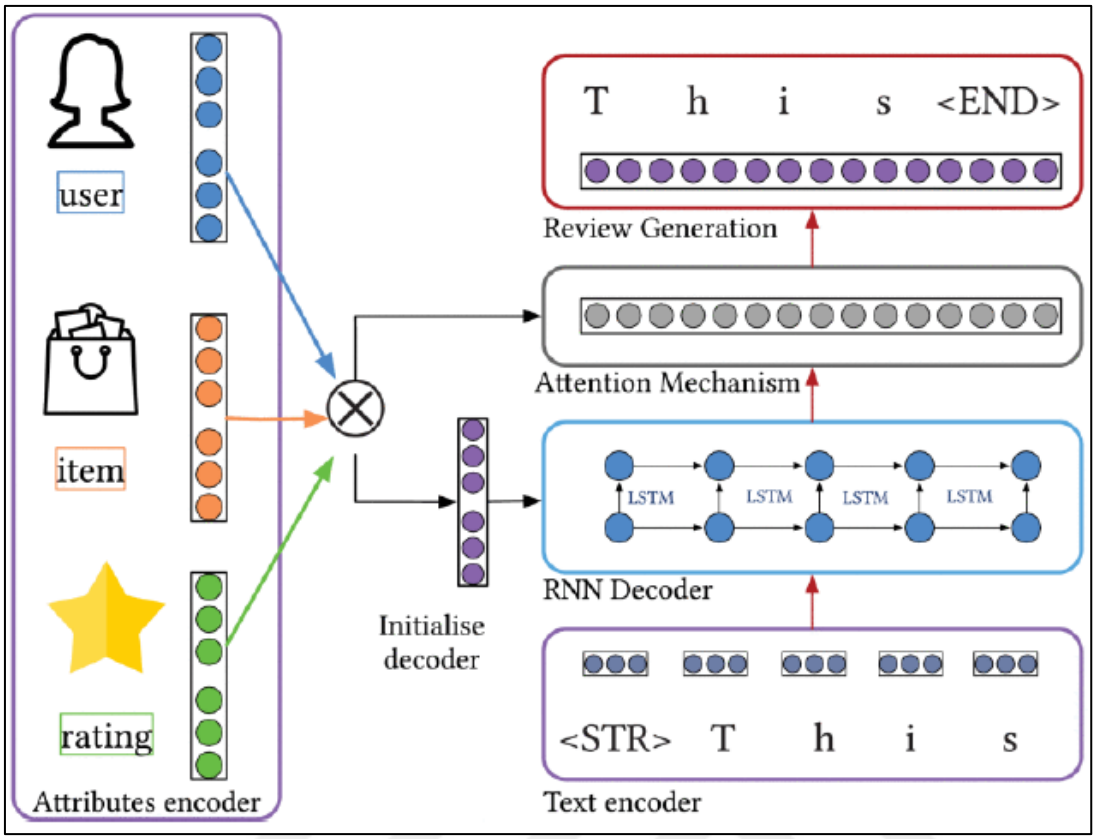


Figure 2.3. NLP-based architecture[78]

In [79], the authors proposed UltraGCN to enhance the existing Graph Convolutional Network (GCN) framework by eliminating specific messaging layers that contribute to computational overhead during the training process. While GCNs primarily function by aggregating and clustering similar information, this mechanism often results in significant delays during training. To address this issue, the authors optimised the model by removing redundant layers, thereby reducing the transmission of such messages and improving overall training efficiency.

Our research adopted a hybrid nature method to analyse reviews with a rule-based approach for labelling the data (related to lexicon-based methods). It transitioned into a text classification approach for training and predictions. While the initial labels rely on simple rules, the deep learning model has the potential to learn more complex patterns and provide more accurate predictions.

In Table 2.1, we present the limitations of existing methods and the corresponding solutions offered by our proposal. This provides a clear comparative framework demonstrating our model's novelty and effectiveness.

Table 2.1. Emphasises the priority limitations and illustrates how our proposed model addresses these gaps

Reference /Author	Methodology	Limitations	Proposed solution
[11] Xinzhe Li et al. (2023)	The prediction of helpful reviews is achieved through the interaction of a multi-channel (CNN), which extracts semantic features from review text, and a rating value encoder that integrates voting values exceeding 10.	Helpfulness-based recommendation systems exhibit bias, as reviews with higher helpfulness scores disproportionately attract more votes, while reviews with fewer but high customer ratings are often overlooked. This bias is particularly pronounced when recommendations are confined to a single category within 5-Amazon's domain.	A simplified propensity-based model integrating rating biases was used, and the analysis was conducted using two distinct datasets sourced from Amazon and one dataset obtained from Yelp.
[12] Olatunji, Li, and Lam (2019)	a neural deep learning model for predicting the helpfulness of online product reviews, emphasising a context-aware approach to capture semantic relationships within review texts.	The model primarily focuses on the textual content of reviews, potentially overlooking other influential factors such as reviewer credibility, temporal dynamics, or product-specific characteristics that may affect perceived helpfulness.	
[13] Ge et al. (2019)	Hierarchical layers from CNN to encoder, helpfulness-aware reviews	The model heavily depends on user-provided helpfulness ratings, which may be biased, as popular reviews might receive higher ratings regardless of their value. This can impact the model's learning process.	
[14] Liu et al (2022)	deep learning-based recommendation model that integrates ratings and review helpfulness through a multi-task learning approach.	Helpfulness bias, cold-start issues, scalability challenges, dependence on review quality, and limited generalizability.	
[16] He et al. (2017)	Neural Collaborative Filtering	Cold start problem; limited scalability to large datasets	Improved generalisation and scalability through optimised feature extraction (average rating) from actual rating and predicted rating that is achieved from four conditions, a tendency-based algorithm integration rating biases
[17] Sreepada et al. (2018)	Tendency-Based Collaborative Filtering algorithm	The model might overfit individual user tendencies if the data is not regularised correctly or there is not enough diversity in the training set. Another limit is that it is dependent on rating data only.	

Table 2.1. (continued) Emphasises the priority limitations and illustrates how our proposed model addresses these gaps

Reference /Author	Methodology	Limitations	Proposed solution
[19] Abulaish and Sah (2019)	Data augmentation techninques and CNN model	The study might not account for user-specific preferences or needs, which can be critical in real-world question-answering scenarios.	Efficient integration of deep learning with large-scale data through feature optimisation and diversification with adapting Synonym-Based Unhelpful Review Augmentation (SbURA) Algorithm on two popular domains of (Amazon review, Yelp review) datasets
[21] Abonizio et al (2021)	Data augmentation, deep learning, and machine learning algorithms.	Some augmentation methods may alter the text's original meaning, leading to mislabeled data and potentially degrading model performance. Another limitation is that it is dependent on just text data	
[22] Derek Chen and Claire Yin (2022)	Explore various data augmentation techniques to enhance intent classification models.	The quality of generated data varies significantly across different augmentation techniques. Some methods lead to substantial improvements, while others have minimal or even negative impacts on model performance, like random word insertion or KNN retrieval, which can produce noisy data that harms performance.	
[23] YAN et al (2019)	Knowledge graph integration and data augmentation via pretrained model	The framework is explicitly tailored for fashion recommendations, and its effectiveness in other domains or applications may not be guaranteed without significant modifications.	
[24] Li et al. (2016)	Factorising Personalised Markov Chains (FPMC) with Transfer by Pattern Recommendation (TBPR)	The transfer learning approach is practical, but it heavily depends on the quality and relevance of the auxiliary data. If the source and target domains are not closely related, the transferred knowledge may not be beneficial and could potentially degrade the model's performance.	Through comprehensive training on extensive datasets, domain adaptation techniques enhance the training process, effectively mitigating or preventing the issue of overfitting to improve performance with customer review analysis.

Table 2.1. (continued) Emphasises the priority limitations and illustrates how our proposed model addresses these gaps

Reference /Author	Methodology	Limitations	Proposed solution
[25] Jafri et al (2022)	integrating multimodal data and leveraging pre-trained deep learning models.	Integrating multimodal embeddings might pose challenges for scalability, especially with large datasets or systems requiring real-time recommendations.	
[26] Gheewala et al (2024)	Leveraging deep transformer models (e.g., BERT, GPT, Roberta) to improve the effectiveness of recommended systems that rely on textual reviews	While transformers help extract features from reviews, they face challenges when dealing with users or items without reviews (cold-start issues).	

3. METHODOLOGY AND MATERIAL

Initially, we present an overview of the selected dataset and a description of the data preprocessing method. Finally, we describe the theoretical foundations and architecture of the proposed balanced deep learning models.

3.1. Dataset Description

In recent years, the availability of large-scale datasets has played a critical role in advancing research across various domains. Among these, the Amazon Review and Yelp review datasets have emerged as a valuable resource for researchers and practitioners, enabling the development and evaluation of deep learning models and recommendation algorithms. This dataset provides comprehensive information, including user reviews, product metadata, and purchasing behaviours, facilitating a wide range of applications in sentiment analysis, recommender systems, and consumer behaviour modelling. Our proposed model is trained on two case studies. In case study 1, we utilized four distinct categories from the Amazon Review Data (2018) dataset. Amazon Review Data is a subset of 5-core reviews, where each user reviews at least five products, and at least five users review each product. This dataset contains 29 categories, 75,257,650 reviews, and 345,769,901 ratings. Specifically, we selected two small-sample categories, Video Games (C1) and Musical Instruments (C2), where all available reviews were included. Additionally, for two large-sample categories, Books-5 (C3) and Clothing, Shoes, and Jewelry (C4), we extracted a subset of 1,000,000 reviews from each category.

The full dataset contains over 142.8 million reviews collected between May 1996 and July 2018 on an open-access platform(http://jmcauley.ucsd.edu/data/amazon/, accessed June 1, 2024). These reviews include various attributes such as ratings, textual feedback, and helpfulness votes. Furthermore, the dataset encompasses product metadata, including descriptions, category details, price, brand, and image features, as well as user interaction graphs (e.g., also-viewed and also-purchased relationships), making it an extensive and well-structured resource for large-scale analysis in recommender systems and user behavior modeling.

Two different types of datasets were used in our second case study. For the first type of dataset, two categories were selected from Amazon Review Data (2018): Musical Instrument Reviews (DS1) and review Clothing, Shoes and Jewelry (DS2). The second dataset type is the Yelp Academic Review Dataset(DS3) (https://www.yelp.com/dataset, accessed June 1, 2024). This dataset contains 6,990,280 reviews and ratings. The data is organised in multiple JSON files and includes the following details: Business information aimed at making recommendations based on user's previous interactions and interests (Business Data), user-generated reviews about companies (Review Data), details of users who posted reviews (User Data), check-in information of business customers (Check-in Data), short recommendations for businesses from users (Tip Data), and metadata for pictures posted on Yelp (Photo Data). This study selected and used the entire DS1 dataset (231,392 reviews), 1,000,000 reviews from the DS2 dataset, and 200,000 reviews from the DS3 dataset. Table 3.1. represents the total count of reviews in the dataset that received a 5-star rating for each category and the subset of 5-star reviews that were utilized in the study using the proposed model.

Table 3.1. Amazon score-5 and Yelp review datasets, number of reviews

Item Category	Number of Reviews(5-star)	Number of Reviews(5-star) with our model
Books-5	27,164,983	1000000
Reviews-Clothing-Shoes and Jewelry	11,285,464	1000000
Musical-Instruments-5	231,392	231,392
Video-Games-5	497,577	497,577
Yelp Review Data Fil	6,990,280	200,000

3.2. Data Preprocessing Techniques

The input content data are the Amazon reviews dataset and the Yelp academic review dataset, which are preprocessed with eight steps as follows:

1. Filter the target features: user_id, item_id, rating, and review text from the Yelp dataset, and user_id, item_id, rating, and summary of review text from the Amazon review dataset.
2. Drop the missing values: drop missing rows from each filtered feature in this step.

3. Predicting rating: The implicit difference of customers' tendencies using the tendency-based CF algorithm is a data preprocessing technique used to eliminate the sparsity problem in the CF (collaborative filtering) algorithm [17] . The KNNBasic algorithm estimates predicted ratings by computing similarities between user and item pairs through collaborative filtering techniques. It considers four distinct cases to evaluate these similarities and identifies the k-nearest neighbors accordingly. Based on these neighbors, the algorithm generates predicted ratings for user-item interactions. Ultimately, the resulting output consists of enhanced feature representations, including the predicted rating feature. We explain these steps in Figure 3.1. and algorithm 3.1 where:

Ut refers to the user tendency, $U_{t.} = \frac{\sum_{i=I.u} Uui - Viu}{I.u}$ (3.1)

It is the item tendency. It $= \frac{\sum_{i=I.u} Uui - Vui}{U.i}$ (3.2)

Uui is the rating given by user u on item I, Viu is the average rating for item I, I.u is the absolute of group items evaluated by user u, Vui is the average rating given by user u across all items, U.i is the absolute of group users who have evaluated item i. the four formula of predectedRting(PR1, PR2, PR3, and PR4) are calculated upend of four conditions (a, b, c, and d), the PR1 is computed as:

PR1=max (Vui+It,Viu+Ut) (3.3)

When condition (a) (U_t, I_t) has a positive inclination, the formula of PR2 is that when condition (b), the inverse of condition (a) (Ut, It), has a negative inclination.

PR2=min (Vui+It,Viu+Ut) (3.4)

The predicted rating (PR3) is computed on condition © I_t is positive, and U_t has a negative inclination as:

PR3=min (max (Vui, (Viu+Ut) σ)+(Vui+It)(1- σ)),Viu) (3.5)

Where the parameter σ controls the item's and user's mean contributions. The last condition is the opposite of the third condition, which is a positive user tendency, U_t with a negative item tendency I_t is calculated as:

$$PR4= (Viu *\sigma) +Vui(1- \sigma) \quad (3.6)$$

Algorithm 1 shows the step-by-step pseudocode that captures the logic of user-item tendency calculations, rating predictions, and the four conditions used to compute the feature-predicted rating using the Tendencies-based collaborative filtering algorithm.

Algorithm: Tendency-based collaborative filtering algorithm

Input:

- dataModel: A data structure holding user-item-rating data
- σ (sigma): Tuning parameter ∈ [0, 1] (default: 0.5)

Output:

- Predicted ratings for all user-item pairs

Step 1: Initialize Recommender System

Procedure TendenciesBasedRecSys(dataModel, $\sigma \leftarrow 0.5$)

1. Store dataModel and σ
2. Call calculateTendencies()
3. Call generatePredictions()

Step 2: Calculate Tendencies

Procedure calculateTendencies()

1. Initialize userTendencies and itemTendencies as empty dictionaries
2. For each user in training set:
   - Compute user tendency as:

userTendency←$U_{t.} \leftarrow \frac{\sum_{i=I.u} Uui-Viu}{I.u}$

3. For each item in training set:
   - Compute item tendency as:

itemTendency←$I_t \leftarrow \frac{\sum_{i=I.u} Uui-Vui}{U.i}$

Step 3: Generate Predictions

Procedure generatePredictions()

1. For each user in training set:
   - For each item in training set:
     - Compute predicted rating using calculatePrediction(user, item)
     - Store in predictions[user][item]
     - Insert into predictedRatings[user][predictedRating] list

Step 4: Calculate Individual Prediction

Function calculatePrediction(user, item)

1. If rating exists in data model, return the actual rating
2. If either tendency is not available, return MIN_VALUE

Figure 3.1. Optimization ratings with adapting steps of the tendencies collaborative filtering algorithm

3. Retrieve:
   - $U_t \leftarrow$ userTendency[user], $I_t \leftarrow$ itemTendency[item]
   - Vui ← user average rating, Viu ← item average rating
4. Compute predicted rating based on 4 cases:
   - Case 1: $U_t > 0$ & $I_t > 0 \rightarrow$ predicted ← *max (Vui+I_t,Viu+U_t)*
   - Case 2: $U_t < 0$ & $I_t < 0 \rightarrow$ predicted ← *min (Vui+It,Viu+U_t)*
   - Case 3: $U_t < 0$ & $I_t > 0 \rightarrow$ predicted ← min (max (Vui, (Viu+U_t) σ)+(Vui+I_t)(1- σ)),Viu)
   - Case 4: Ut > 0 & It < 0 →predicted ← (Viu *σ) +Vui(1-σ)
5. Format and return rating (2 decimal precision)

Step 5: Predict Rating on Demand

Function predictRating(user, item)

Return predictions[user][item] if available, else MIN_VALUE

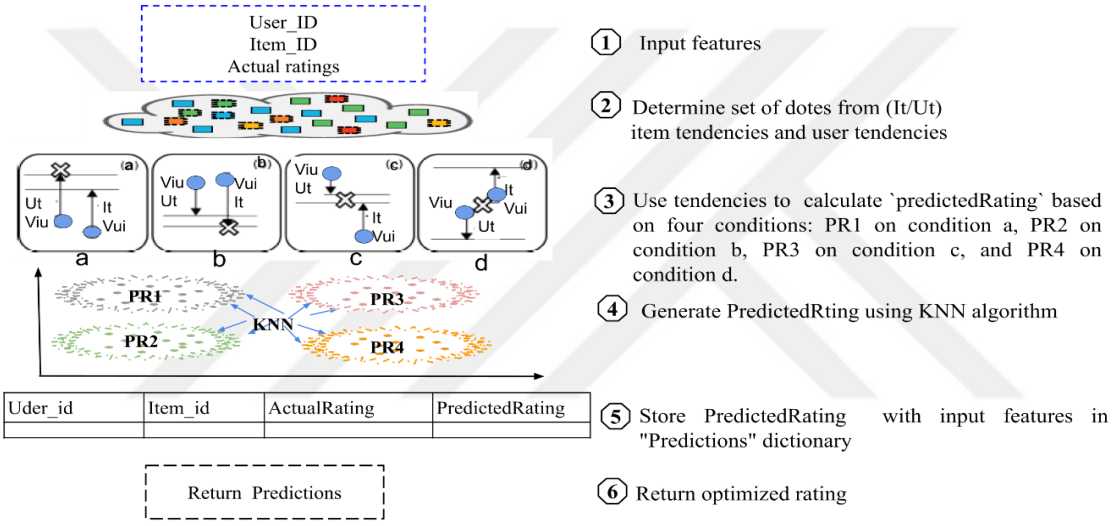


Figure 3.1. (continued) Optimization ratings with adapting steps of the tendencies collaborative filtering algorithm

As mentioned earlier, simple formulas were used for the four conditions and calculations. Also discussed is the CF algorithm, which is based on the similarity of associations between users on the same item. This method states that if two users exhibit a similar classification pattern, there is a high probability of matching in missing classifications. However, since identifying these links is relatively complex, most solutions require a lot of information. Therefore, similar algorithms have significant problems when dealing with sparse datasets.

4. Optimize the actual rating values by calculating the average rating.
5. Generate high-value features, such as helpful and unhelpful reviews, primarily based on either the optimal rating or the original rating.

6. Balance helpful (1) and unhelpful (0) review instances as in Figure 3.2. To balance the distribution of classes in the dataset, the over-sampling approach based on the Synonym-Based Unhelpful Review Augmentation (SbURA) algorithm [80] requires copying instances from the minority class, unhelpful (0). This helps to ensure that the deep model gets enough samples from all classes, preventing it from being biased against the majority class (1). SbURA is a data augmentation method that generates synonyms for words in sentences to balance text datasets. This produces examples that are syntactically diverse yet semantically comparable. Algorithm 3.2 displays the procedure of the SbURA algorithm. The input is the original dataset with text reviews and helpful labels. The output is the dataset containing the balanced augmented review text.
7. Pre-tokenized review text and labels: We should tokenize the review texts of groups (Amazon review and Yelp review) datasets that we used in the two case studies by stopword removal, stemming, and tokenizing. These processes represent fundamental steps in a standard natural language processing (NLP) pipeline. In our approach, they were applied alongside case study 1 and case study 2 with (Balanced CNN-BiLSTM). However, their application is not exclusive to transformer-based models, such as case study 2 with Pre-Trained Model II (RoBERTa-base), and may not always align with model-specific preprocessing requirements. The following section outlines how these steps integrate into traditional NLP pipelines compared to transformer-based architectures.

- Traditional Pipelines: These steps are used in traditional NLP pipelines for our task, such as review classification as input pre-tokenizing data for training (Balanced CNN-BiLSTM) to break the texts of reviews into smaller, easier-to-understand chunks. These adjustments to the data before training the deep model are called preprocessing. In text classification, we need to use nltk.corpus.stopwords to extract meaningful information from the review text according to our task. Reducing words to their essential forms is called stemming, with the function PorterStemmer(). The Porter Stemmer is often used because of its balance of simplicity, speed, and effectiveness, especially in general-purpose NLP. Although the root serves as a basic form for related words in terms of meaning, it may not be a word in the language on its own. İt is necessary to improve text representation by grouping similar words. Tokenization is used in NLP on keras.preprocessing.text. Tokenizer to break down phrases and paragraphs into more straightforward and processable elements. Collecting data (a sentence) and breaking it

into understandable components (words) are the first steps in the NLP process. The vocabulary introduced to word segmentation profoundly influences its precision. Striking a balance between efficiency and exactness is paramount when selecting words for these algorithms. An NLP would reach heightened precision if equipped with a comprehensive dictionary's vocabulary. This assertion resonates with algorithms tailored for niche purposes. The lexicon introduced to word segmentation dictates its precision level. Choosing the right words for these algorithms demands a harmony between efficiency and exactness [81]. The tokens are embedded using a word embedding approach by mapping various word indices to embedding vectors during training as an input layer.

- Transformer Pipelines: For transformer pipelines, like the case study with Pre-Trained Model II. (Roberta-base), These steps are replaced by advanced tokenization methods and model-specific preprocessing tailored to handle textual data more effectively. First, isolating specific columns from a structured dataset (e.g., a DataFrame) and converting them into a format-compatible input in list format with the RoBERTa tokenizer. Roberta uses a special tokenizer based on Byte-Pair Encoding (BPE) to split text into subword units. The tokenizer ensures consistency with the vocabulary used during Roberta's pretraining. Roberta, in addition to special tokens, expects special tokens such as <s> (start of sentence) and </s> (end of sentence) around the input text. Sequence-pair tasks also include a separator token </s></s> between the sequences. Each word in the input text is broken into subwords or tokens (Convert Text to Tokens) using the BPE vocabulary. As the Output compatibility, the pipeline directly prepares the dataset for frameworks like Hugging Face’s transformers library, ensuring smooth integration for training and evaluation [82].

Algorithm: Synonym-Based Unhelpful Review Augmentation (SbURA) Algorithm

Input: rawDS // Dataset with text reviews and helpfulness labels
Output: augDS // Augmented unhelpful reviews

```
augDS ← {}
unhelpful_reviews ← filter rawDS where helpful = 0 and drop NaN values
num_helpful_reviews ← total number of helpful reviews in rawDS
num_unhelpful_reviews ← size of unhelpful_reviews
target_augment_count ← num_helpful_reviews - num_unhelpful_reviews
augmented_count ← 0
while augmented_count < target_augment_count do
review ← randomly choose a review from unhelpful_reviews
tokens ← tokenise review into words using nltk.word_tokenize
augmented_tokens ← []
for each token in tokens do
synonyms ← get synonyms for token using wordnet.synsets
if synonyms is not empty then
 augmented_token ← randomly choose a synonym from synonyms
Else
 augmented_token ← token
append augmented_token to augmented_tokens
End
augmented_review ← join augmented_tokens into a sentence
add augmented_review to augDS
augmented_count ← augmented_count + 1
End
return augDS
```

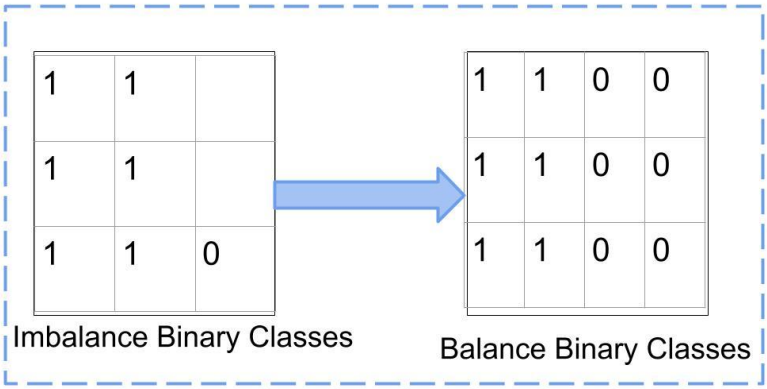


Figure 3.2. Over-sampling based on SbURA for balancing binary classes

3.3. Deep Learning Models Used in Case Studies

3.3.1. Case study 1: BHRQUT model

In this part, we detail the overview of the proposed balanced helpful recommendation model, with quantifying users' tendencies, BHRQUT model, which is illustrated in Figure 3.3. To test the performance of review analysis on item reviews, we follow two cases: one Case1

combines the advantages of the TCF algorithm, CNN, data augmentation, and the BiLSTM model, in this case, re-rating of items is performed by analyzing differences between users and items using the Tendencies class from the Surprise package. The dataset is then filtered into two paths Case 1 and Case 2, her path is divided into two subpaths: Case 1 is partitiond into Case 1_1 and Case 1-2, where Case1_1 a review is considered highly helpful if the average rating (avg), computed as avg = (Actual Rating (ACR) + Predicted Rating (PRR)) / 2, is ≥ 3. In this scenario, reviews are labeled helpful if avg ≥ 0.3, while those with avg ≤ 0.2 are categorized as unhelpful. A stricter threshold is applied in Case 1_2, where avg ≥ 4 is required, classifying reviews as helpful if avg ≥ 0.4 and unhelpful if avg ≤ 0.3, following the methodology proposed in [83], while the Case 2 is the same but does not use the TCF algorithm, in this case, filtering is performed based solely on the actual rating of the dataset without utilizing the Tendencies class. The dataset is again divided into two subcases: Case 2_1, where a review is labeled helpful if ACR ≥ 0.3 and unhelpful if ACR ≤ 0.2, following the approach in [84], and Case 2_2, where a more stringent threshold is applied, categorizing reviews as helpful if ACR ≥ 0.4 and unhelpful if ACR ≤ 0.3. We utilize the TCF algorithm to enhance the review feature to predict optimal rating and assign a binary label (helpful, unhelpful) to each row based on the average rating values computed from the 'Actual Rating' and 'Predicted Rating' columns. Then we balance the dataset by employing one of several data augmentation techniques, such as the Synonym-Based Unhelpful Review Augmentation (SbURA) algorithm, resulting in an equal number of rows for each label. Following data preprocessing, the review representation is structured using a traditional Natural Language Processing (NLP) pipeline. Subsequently, a CNN-BiLSTM model is trained over 20 epochs using a 5-fold cross-validation technique. The model is designed to analyse the textual summary content of user reviews, considering only rating interactions and written reviews by the user. The evaluation process incorporates various performance metrics, including Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE), accuracy, recall, F1-score, precision, and the confusion matrix. The preprocessing steps applied to the dataset are detailed in (See Section 3.2), followed by an in-depth discussion of the CNN-BiLSTM network architecture, its mathematical formulation, and the evaluation process.

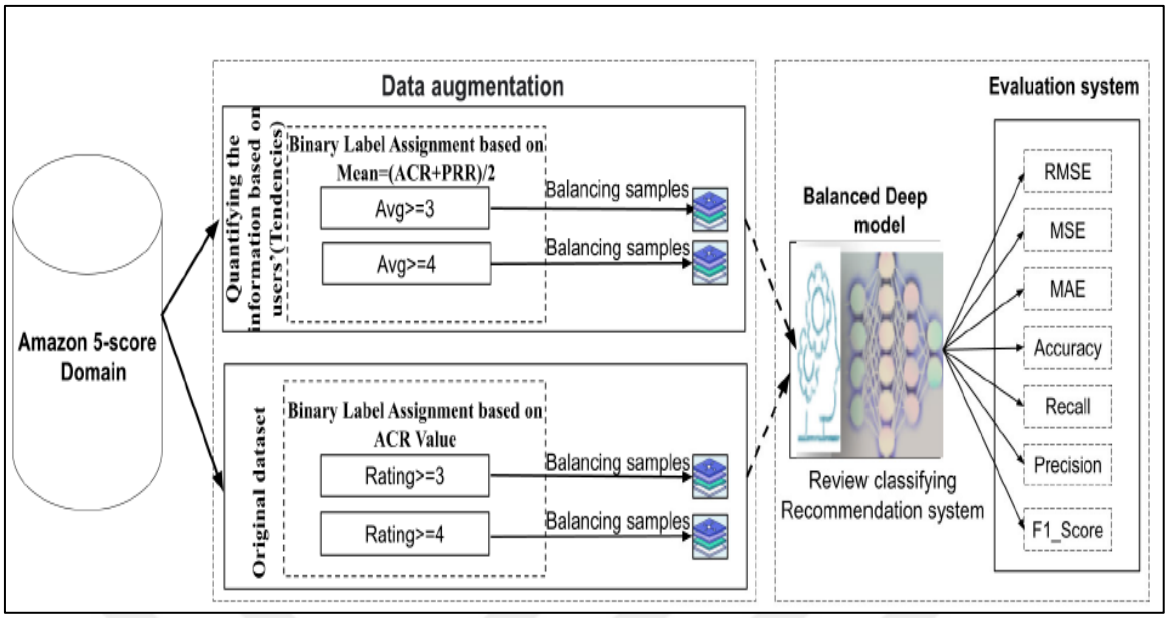


Figure 3.3. Overview of the BHRQUT model

Hypothetical background of the balanced CNN-BiLSTM network

Feature extraction begins with structural attributes such as review length, sentence count, and word count. The organisation of content significantly influences review quality. For instance, excessively long reviews may include vague or ambiguous details, which can diminish the effectiveness of product recommendations. Additionally, metadata features such as star ratings and review summaries play a crucial role. Customers provide star ratings alongside their written reviews, and the discrepancy between an item's actual average rating and individual user ratings can be a key factor in assessing review significance.

To improve the accuracy of helpful review analysis in product reviews, we leverage the strengths of TCF and the balanced CNN-BiLSTM model, introducing the BHRQUT framework. The process begins with dataset collection based on user ratings and behavioural tendencies. After applying NLP-based preprocessing and optimising user profile features through binary label assignment and dataset balancing techniques, the embedding layer captures the semantic meaning of words and sentences. This layer transforms categorical data into continuous vectors (0,1), enhancing efficiency in analysis, classification, and prediction tasks.

Subsequently, the extracted feature vectors and attributes are fed into a Convolutional Neural Network (CNN) for recommendation. The CNN layers perform feature extraction on the

input-balanced dataset, capturing intricate patterns within the data. In our proposed model, four Conv1D layers are incorporated, where each layer's output is determined based on its respective inputs, word embedding dimensions, maximum sequence length, kernel filter size, and neuron biases. The mathematical formulation of the convolutional layer output is presented in "Eq. 3.7".

$$Y_i^{(n)} = \sum_{j=0}^{h-1} \sum_{f=0}^{D-1} X_{(i.s+j-p)} K_{j,f}^{(n)} + b^{(n)} \tag{3.7}$$

In our model, two matrices represent the input text sequence and the convolutional filter. The input text sequence is denoted as a matrix X with dimensions $L \times D$, where L represents the sequence length in tokens, and D denotes the dimensionality of the word embeddings. Similarly, the convolutional kernel is defined as a matrix K with dimensions $h \times D$, where h corresponds to the filter height, and D aligns with the embedding dimension. The output vector at position n, denoted as Y, is calculated as the filter traverses the input matrix. The stride determines how many words the filter shifts at each step, with each element's position $(i+j, f)$ in the input matrix X corresponding to the position (j, f) in the filter matrix K. To maintain consistent output dimensions, padding (p) is applied by appending zero tokens at the beginning and end of the sequence. Additionally, the bias term b, associated with each layer, is regulated by "Eq. 3.8".

$$\Delta b_i^{(n)} - \frac{\beta}{n} . \frac{\theta.L}{Y_{i,j,f}} + m. \Delta b_i^{(n-1)} \tag{3.8}$$

In this framework, bi represents the bias associated with each neuron, while the regularisation parameter is denoted by β. The variables n and m correspond to the total number of samples used for training and testing. The loss function is represented as the $\frac{\theta.L}{Y_{i,j,f}}$, is defined as the Mean Squared Error (MSE) concerning the output at position (i, j, f) for the f-th filter. A max-pooling layer is applied to generate a more abstract and compact representation of the input data, enabling subsequent layers to focus on higher-level features. The max pooling operation for a given feature map is mathematically formulated in "Eq. 3.9".

$$MP_{i,n} = \max\left(Y_{i,s+f,n}\right) for\ f = 0,1, \dots, m-1 \tag{3.9}$$

In this formulation, MPi,n represents the element in the output feature map at the position (i,n), while Yi,n denotes the corresponding element in the input feature map. The maximum value is selected from a pooling window of size m along the sequence length dimension L. This approach enhances the network's ability to generalize to unseen data. To further improve sequence learning, these extracted features are integrated with BiLSTMs within the CNN-BiLSTM architecture, as illustrated in Figure 3.4. The Bidirectional Long Short-Term Memory (BiLSTM) model captures user preferences in both forward and backward directions, addressing challenges associated with cold-start problems in recommendation systems.

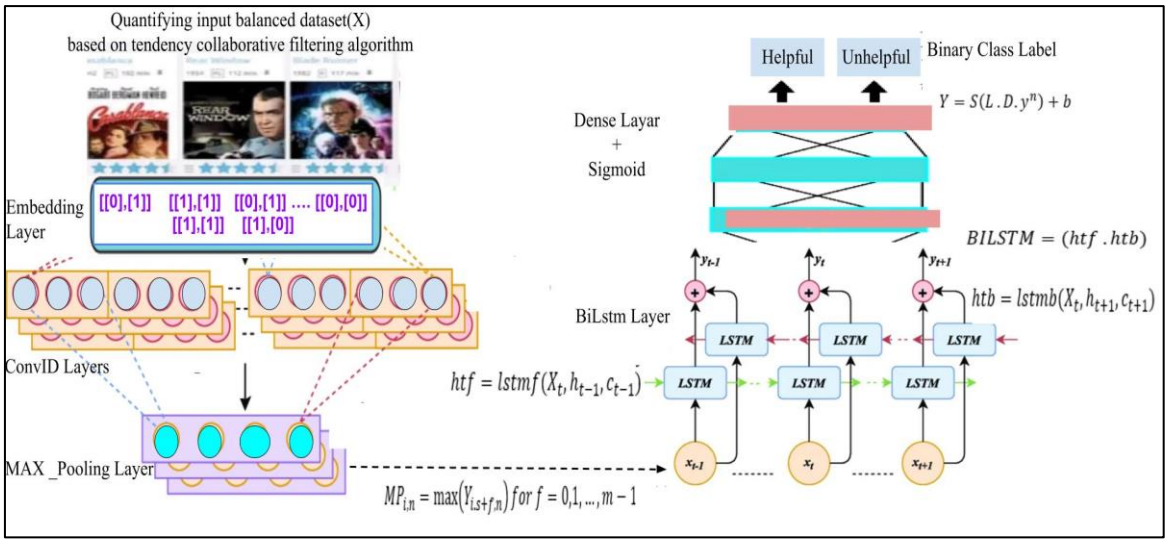


Figure 3.4. Formulation of BHRQUT architecture

The BiLSTM architecture employs two LSTM networks to process the input sequence instead of just one. One LSTM works on the original sequence, while the other analyzes a reversed or modified version [85]. This bidirectional structure enhances pattern recognition by capturing contextual dependencies in both forward and backward directions, thus improving the model's ability to comprehend sequential relationships within the data.

The mathematical formulation of a BiLSTM consists of both forward and backward LSTM layers. At each time step, the BiLSTM generates hidden states from both directions. The forward hidden state at time t, denoted as htf, is calculated based on the previous hidden state $ht-1$ and the cell state $ct-1$, as defined in "Eq. 3.10". Similarly, the backward hidden state at time t, represented as htb, is determined using the future hidden state $ht+1$ and the

cell state ct+1, as expressed in "Eq. 3.11". The input at time t, denoted as Xt, is included in both computations to capture contextual dependencies from both temporal directions.

$$h_{tf} = lstmf(X_t, h_{t-1}, c_{t-1}) \tag{3.10}$$

$$h_{tb} = lstmb(X_t, h_{t+1}, c_{t+1}) \tag{3.11}$$

The output of the BiLSTM at each time step t is obtained by concatenating the forward and backward hidden states. This operation integrates contextual information from both temporal directions, as expressed in "Eq. 3.12".

$$BiLSTM = (h_{tf}.h_{tb}) \tag{3.12}$$

Finally, the fully connected layer is constructed by merging three key components: the flattened, Dense, and output layers. This integration is based on the width of the input sequence (L), the height of the input sequence (D), the bias term (b), the number of output units (yn), and the activation function sigmoid (S), which classifies the output to binary values (0,1) [86], as formulated in "Eq. 3.13".

$$Y = S(L.D.y^n) + b \tag{3.13}$$

3.3.2. Case study 2: PRE-DHRRF model

Case Study 2 outlines the effectiveness of integrating dataset preprocessing techniques, sample size balancing, and transfer learning as essential components for improving performance in customer review analysis, as depicted in Figure 3.5. The process incorporates a tendencies-based collaborative filtering algorithm, data augmentation through oversampling, and a natural language processing (NLP) pipeline for preprocessing rating and review texts of two categories from the Amazon customer review dataset and one group from the Yelp Academic dataset, all of which contribute to more robust dataset preparation. In the subsequent phase, we trained the datasets using previously saved weights from two pre-trained models: Balanced CNN-BiLSTM and RoBERTa-base. The training procedure employed a 5-fold cross-validation strategy over 20 epochs. The final phase involved validating model performance and fine-tuning hyperparameters. Evaluation metrics included

generalization gap, accuracy, precision, recall, F1 score, root mean square error (RMSE), and mean absolute error (MAE). Each stage of the process is described in detail to ensure transparency and reproducibility.

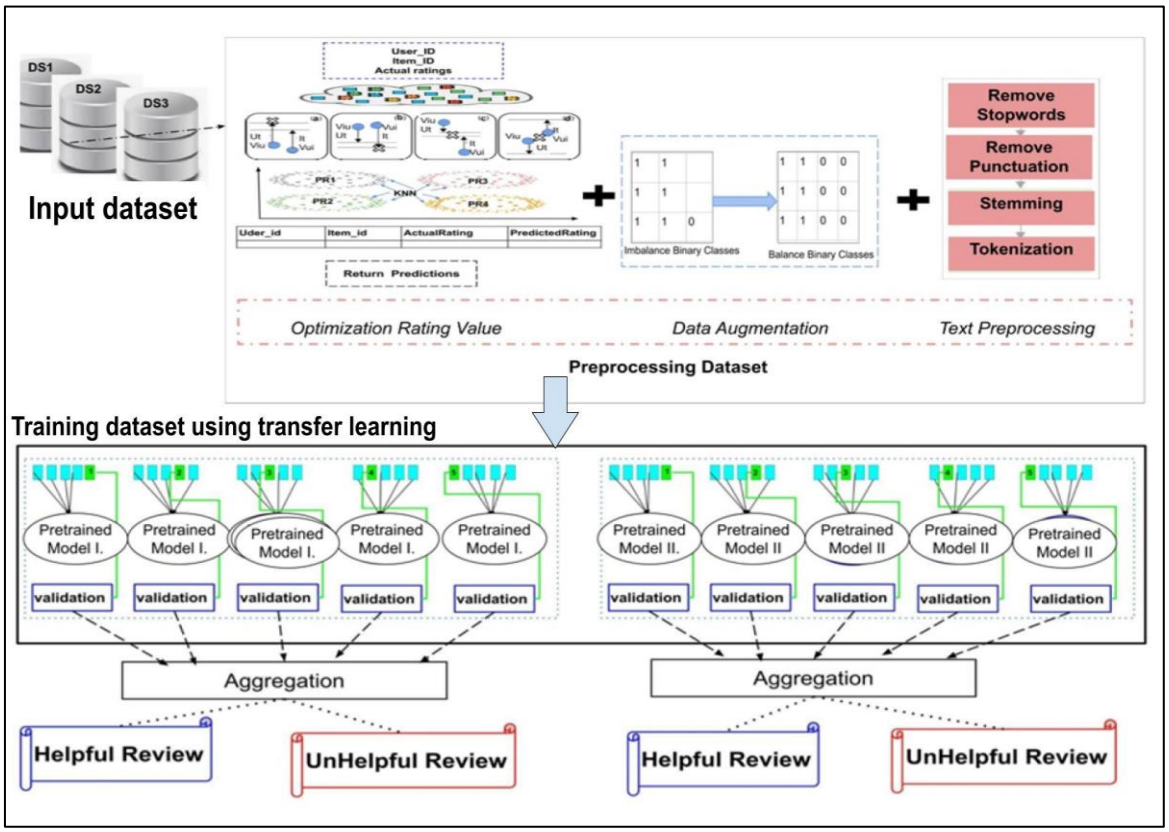


Figure 3.5. An architecture of our proposed PRE-DHRRF framework. Top component: Preprocessing the input dataset. The bottom component consists of two pre-trained transformer models, each validated on 5-fold cross-validation; the left one is fine-tuned for review classification using Pre-trained Model I (Balanced CNN-BiLSTM), and the right one is fine-tuned for review classification using Pre-trained Model II. (Roberta-base).

The system can provide useful recommendations to users based on our proposed case study 2, as shown in Figure 3.6.

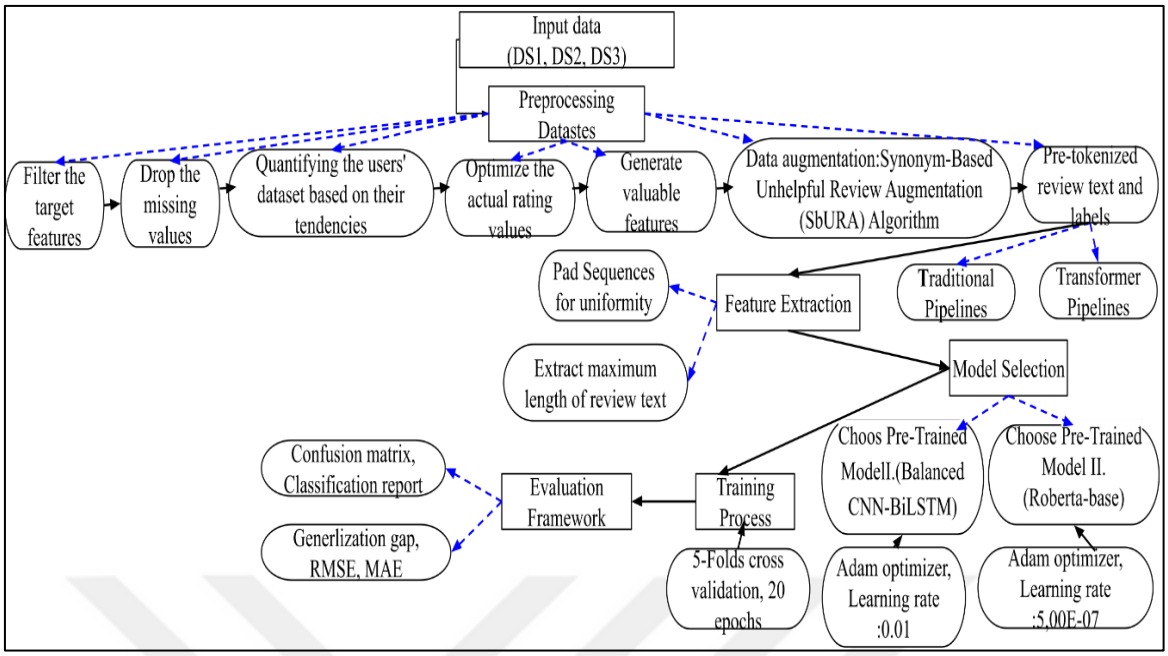


Figure 3.6. Flowchart illustrating the steps methodology of our case study 2

Hypothetical background of the pre-trained balanced CNN-BiLSTM model

Input features were extracted using tokenized text sequences and padded for consistency. Additionally, we included domain-specific features, such as review length to determine the maximum number of words for input to our model layer, and sentiment scores (helpful label), to enhance contextual understanding. Padding = ' same' in our previous model (Balanced CNN-BiLSTM) was crucial for ensuring dimensional consistency and compatibility with pre-trained weights. In convolutional operations, the output dimensions shrink without padding as the filter slides across the input, potentially leading to information loss at the edges. By applying padding='same', the output dimensions are maintained, matching the input size, essential for preserving architectural consistency throughout the model.

To enhance the formulation of the Custom Balanced CNN-BiLSTM as a pre-trained model, the proposed CNN-BiLSTM model integrates Convolutional Neural Networks (CNN) for feature extraction and Bidirectional Long Short-Term Memory (BiLSTM) for sequential modeling. The embedding layer maps tokenized words into dense vectors, while the Max Pooling layer reduces dimensionality while preserving dominant features. The BiLSTM layer captures long-range dependencies bidirectionally, as demonstrated in (See Section 3.3.1) with case study 1 and as (See Equations 3.4-3.9). Following this, A fully connected layer is employed to integrate features by combining the representations learned from the

CNN-BiLSTM architecture with those obtained from a pre-trained model. Specifically, the hidden states generated by the BiLSTM are concatenated with the pre-trained feature vector Pt, as defined in "Equation 3.14.", to enhance the overall feature extraction process.

$$H_t = BiLSTM \oplus P_t \tag{3.14}$$

The final classification layer outputs a binary helpfulness label is performed using a fully connected layer followed by a sigmoid activation, as in "Eq. 3.15".

$$Y = S(L.D.y^n.H_t) + b \tag{3.15}$$

This integration is based on the width of the input sequence (L), the height of the input sequence (D), the bias term (b), the number of output units (yn), and the activation function sigmoid (S), which classifies the output to binary values (0,1)

Hypothetical background of the pre-trained balanced Roberta-base model

The raw text input, denoted as X is tokenized using RoBERTa's tokenizer, which converts textual data into a structured format suitable for input into a neural network. This process produces tokenized inputs, attention masks, and segment identifiers, collectively referred to as X_{tokens}. To ensure consistency in input dimensions, padding and truncation are applied to maintain a fixed sequence length of L, yielding a tokenized representation as "Eq. 3.16".

$$X_{tokens} = \{x1, x2, \dots, x_L\} \tag{3.16}$$

Each tokenized input X_{tokens}.. is then embedded into a dense vector space using word embeddings and positional encodings, forming the input matrix, then processed using a pretrained RoBERTa model to extract contextual embeddings [82].

$$E = Embedding(X_{tokens}) = Positional\ Encoding(X_{tokens}) \tag{3.17}$$

Let $E \in R^{L \times d}$ denote the input embedding matrix, where L is the sequence length and d represents the hidden dimension of the RoBERTa model (typically d=768 for RoBERTa-base). The input embeddings incorporate both token embeddings and positional encodings. Positional encodings are crucial for preserving the sequential order of tokens, which is

otherwise not inherently captured by self-attention mechanisms. The embedding representation E is passed through a stack of Transformer encoder blocks. Each block consists of the following key components:

- Multi-Head Self-Attention (MHSA):

This mechanism enables the model to attend to information from different positions within the input sequence simultaneously. It does so by projecting the input into multiple sets of query, key, and value vectors (heads), applying scaled dot-product attention independently across each head, and then concatenating and projecting the results. This facilitates learning complex dependencies and contextual relationships between tokens.

- Layer Normalization:

Applied after the self-attention and feed-forward sublayers, layer normalization stabilizes and accelerates training by normalizing the summed input across the feature dimension. This helps mitigate internal covariate shift and improves convergence.

- Feed-Forward Network (FFN):

Each transformer block includes a position-wise feed-forward network, typically consisting of two linear layers separated by a non-linear activation function such as GELU (Gaussian Error Linear Unit). The FFN processes each token's representation independently, enabling non-linear transformations that enhance the model’s expressive capacity.

- Dropout:

Dropout is applied at various stages (e.g., after attention and FFN layers) to prevent overfitting by randomly zeroing out elements of the input tensor during training, thus improving the generalization ability of the model.

The output of the i-th Transformer block can be formally expressed as shown in“Eq. 3.18” and illustrated in Figure 3.7. These stacked layers collectively enable the model to capture both local and global dependencies in the input sequence, which is essential for effective downstream tasks such as classification.

$$H_i = LayerNorm(FFN\left(LayerNorm\left(MultiHeadAttention(H_{i-1})\right)\right)\right) \quad (3.18)$$

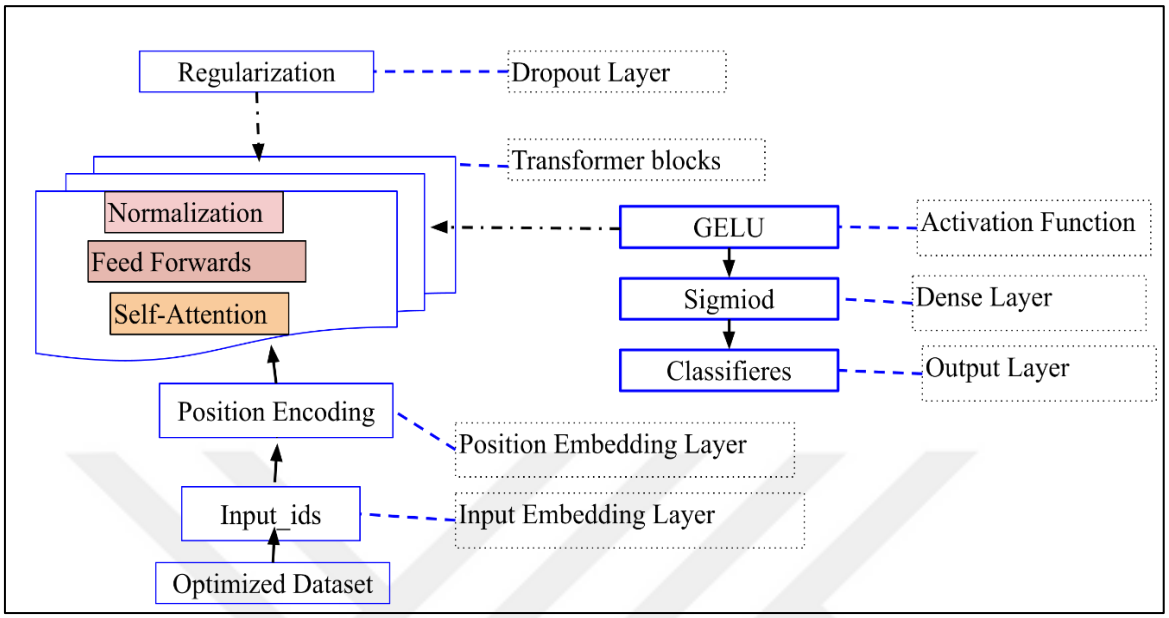


Figure 3.7. Construction of pre-trained (Roberta-base) model

The model generates a hidden state representation $H_{TL} \in R^{L\times d}$, where d denotes the hidden dimension of the RoBERTa model. The first hidden state, corresponding to the classification (CLS) token, is extracted as $h_{cls} = H_{TL}[0]$. The [0] index extracts the first row of this matrix, which corresponds to the [CLS] token (classification token).This CLS representation is then passed through a fully connected layer to obtain the logits:

$$z = Wh_{cls} + b \quad (3.19)$$

where, $W \in R^{d\times 2}$ is the weight matrix and $b \in R^2 is\ the\ bias\ vector$. They are trainable parameters.

To introduce non-linearity, an activation function such as GELU (Gaussian Error Linear Unit) is applied:

$$z' = GELU(z) \quad (3.20)$$

The error function is complex and slow for deep learning models. A practical alternative is using the sigmoid function $\sigma(x)$, which is easier to compute:

The final class probabilities are computed using the sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.21}$$

The Approximation: Instead of the erf-based GELU, we use:

$$\text{GELU}(x) \approx x \cdot \sigma(1.702x) \tag{3.22}$$

The scaling factor 1.702 makes the sigmoid function closely match the shape of the erf-based GELU.

$$Y = \sigma(z') \tag{3.23}$$

where Y represents the predicted probability distribution over the two classes [87].

3.4. Model Training and Hyperparameter Selection

This section introduces the key hyperparameters related to model training and architecture, discusses their importance, and methods for determining their value range, addressing their efficiency and accuracy, particularly for deep learning networks. Hyperparameters play a crucial role in deep learning because they dictate how a model learns and generalises from data. Unlike model parameters (such as weights and biases) learned during training, hyperparameters are set before training begins and significantly impact the model's performance, convergence speed, and ability to generalise to unseen data. Hyperparameters are crucial in defining the model architecture and optimising its training process. They influence the model's structure, such as the number of hidden layers, the choice of activation functions, and training efficiency and accuracy through parameters like the learning rate (LR) of the optimizer, batch size, and loss function [88].

Drouput regularazition technique

Dropout is a regularization technique that randomly deactivates a subset of neurons during training with a specified probability, thereby reducing the network’s reliance on specific weights. This stochastic process enhances generalization and mitigates overfitting by preventing the model from memorizing training data Figure 3.8. The recommended

probability for neuron deactivation typically ranges between 20% and 50%. If the dropout rate is excessively high, the model may become overly simplistic, compromising its learning capacity. Conversely, a very low dropout rate may have a negligible impact on regularization. Furthermore, it is advisable to use a higher learning rate with decay and increased momentum, as the reduced number of active neurons necessitates stronger weight updates per batch to maintain training stability and convergence efficiency.

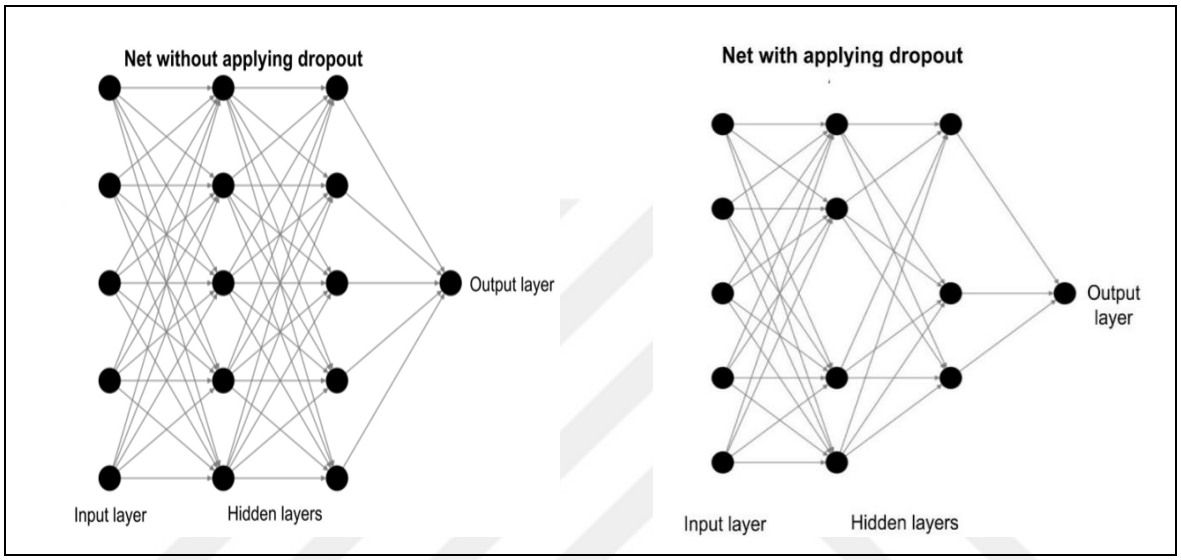


Figure 3.8. Effect of dropout on a neural network

Activation functions

Activation functions are crucial in deep learning for introducing nonlinear properties to the output of neurons. Without an activation function, a neural network would behave as a simple linear regression model, unable to capture complex patterns within data. Activation functions must be differentiable to compute weight gradients and perform backpropagation effectively. The choice of activation depends on the type, structure, and scale of the dataset, as it directly influences the network's learning efficiency and convergence behaviour. The main functions of the activation are Sigmoid, Hyperbolic Tangent (TanH), Rectified Linear Units (ReLU), and Gaussian Error Linear Unit (GELU). Figure 3.8 and its calculations are in “Eq. 3.24, 3.25, 3.26, and 3.27”, respectively.

One of the most used activation functions is the Sigmoid function, mathematically expressed

as: $f(x) = \frac{1}{e^{-x}}$ (3.24)

The Sigmoid function maps any real-valued input to a range between 0 and 1, making it particularly useful for binary classification tasks. By constraining activations within this range, it prevents excessively large values, ensuring stable learning. However, the Sigmoid function suffers from the vanishing gradient problem, which can slow down the training of deep networks.

A more advanced alternative is the Hyperbolic Tangent (TanH) function, given by:

$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$ (3.25)

Unlike the Sigmoid function, TanH is zero-centered, with outputs ranging from −1 to 1, allowing for a more balanced gradient flow during training. Its steeper derivative enables faster learning compared to the Sigmoid function. However, despite its advantages, TanH still suffers from the vanishing gradient problem, making it unsuitable for deep networks with many layers.

In recent years, the Rectified Linear Unit (ReLU) function has become the most widely used activation function due to its simplicity and effectiveness in deep learning models. Mathematically, ReLU is defined as:

$f(x) = \begin{cases} 0, x \leq 0 \\ x, x > 0 \end{cases}$ (3.26)

ReLU has gained popularity because it overcomes the vanishing gradient problem present in Sigmoid and TanH, making it particularly effective in multi-layer networks. Unlike Sigmoid and TanH, ReLU does not similarly activate all neurons, improving the network's ability to learn complex representations. However, ReLU has two main drawbacks. First, its range extends from zero to infinity, which can lead to unstable activations. Second, for negative input values, ReLU outputs zero, causing some neurons to become permanently inactive, a phenomenon known as the "dying ReLU" problem. To address this issue, several ReLU variants have been introduced, including Leaky ReLU, Parametric ReLU (PReLU), Exponential Linear Unit (ELU), and Scaled Exponential Linear Unit (SELU), Figure 3.9.

Leaky ReLU introduces a small slope (e.g., 0.01x) for negative inputs, allowing small gradient updates and preventing neurons from becoming permanently inactive. Despite these limitations, ReLU remains the default choice for deep learning models due to its computational efficiency and fast convergence. If vanishing gradients become an issue, Leaky ReLU is often suggested as an alternative.

For deep architectures like Transformers, the GELU function [89] has been widely adopted due to its smooth activation properties, improved gradient propagation, and enhanced training stability, contributing to superior performance in large-scale natural language processing (NLP) tasks. The GELU function is defined as:

$$f(x) = x.\varphi(x) \tag{3.27}$$

where $x.\varphi(x)$ is the standard normal cumulative distribution function (CDF). An alternative approximation is:

$$f(x)=x\cdot\sigma(1.702x) \tag{3.28}$$

Where x represents a range of input values for which the activation functions are calculated.

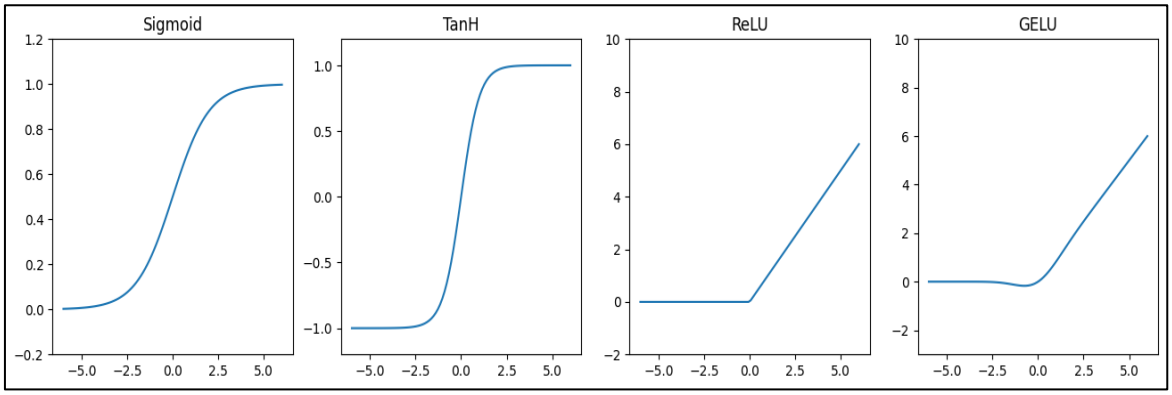


Figure 3.9. Formulation of activation functions

Adam optimizer algorithm

Optimization algorithms, such as Adaptive Moment Estimation (Adam), play a crucial role in training deep neural networks by efficiently minimizing the loss function and enhancing model performance. The choice of optimizer directly impacts the convergence speed,

stability, and overall effectiveness of the learning process. Adam is widely adopted due to its adaptive learning rate, momentum-based updates, and bias correction, which address some challenges associated with traditional optimization methods.

Adam is a widely used optimization algorithm in deep learning, recognized for its efficiency and effectiveness across various neural network architectures. Adam combines the advantages of stochastic gradient descent (SGD) with momentum and RMSprop, which incorporates bias correction and momentum into RMSprop to enhance performance, especially in the later stages of optimization. It calculates both the exponential moving average of past gradients and the squared gradients. Moreover, Adam maintains an exponentially decaying average of past gradients, ensuring balanced memory efficiency between GD (Gradient Descent) with momentum and RMSprop. Due to its robustness and low hyperparameter tuning requirements, aside from the learning rate (LR), the learning rate function controls the extent of parameter adjustments during each training iteration, influencing the convergence speed and stability of a model. While its values typically range between 0 and 1, the optimal selection depends on factors such as the optimizer, dataset characteristics, and model architecture. To improve training efficiency and stability, techniques such as learning rate schedules and adaptive optimization methods are commonly employed. Additionally, batch size, which determines the number of samples processed before updating model parameters, significantly impacts both computational efficiency and training accuracy, while the hyperparameter epoch represents the total number of iterations the learning algorithm performs over the entire training dataset. A single epoch ensures that each sample in the dataset has been processed once, allowing the model to update its internal parameters accordingly. Although increasing the batch size can reduce model performance, this effect can be mitigated by appropriately adjusting the learning rate in conjunction with batch size, ensuring balanced optimization and improved learning dynamics. Adam is frequently recommended as the default optimization algorithm for training deep neural networks [90].

Loss functions

Loss functions are fundamental in deep learning-based recommendation systems, as they quantify the discrepancy between predicted and actual outcomes, guiding model optimization. The selection of an appropriate loss function depends on the recommendation

task, such as rating prediction, implicit feedback modeling, review classification, or ranking optimization, with the primary objective of minimizing prediction errors to enhance recommendation accuracy. Loss functions are broadly classified into regression-based and classification-based types. Regression loss functions [90], such as Mean Squared Error (MSE), are employed for continuous value predictions and are particularly suitable when target outputs follow a normal distribution. MSE, computed as the average of squared differences between predicted and actual values, is differentiable, making it well-suited for gradient descent optimization. Mathematically, it is expressed in "Eq. 3.29".

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.29}$$

Conversely, classification loss functions, such as Binary Cross-Entropy (BCE), are used in binary classification tasks where models predict probabilities between 0 and 1 [91]. This loss function penalizes incorrect and low-confidence predictions by leveraging logarithmic computations, ensuring that the model learns to produce both accurate and confident predictions.

To maintain the convention where lower loss values signify better model performance, the computed result is multiplied by -1 as in "Eq. 3.30".

$$BCE = \frac{1}{n}\sum_{i=1}^{n}[y_i \log p(y_i) + (1 - y_i)\log(1 - p(y_i))] \tag{3.30}$$

where yi is the true label (either 0 or 1), and $p(yi)$ represents the predicted probability of the instance belonging to class 1. The average loss across an entire training dataset of size n.

The choice of an appropriate loss function directly impacts the effectiveness of a recommendation system by optimizing model performance, reducing error, and improving user satisfaction.

3.4.1. Reproducible results of case study 1 (BHRQUT)

To implement and evaluate our case study 1 approach by selecting four categories from the Amazon Review-5 dataset: Video Games (C1), Musical Instruments (C2), Books-5 (C3), and Clothing, Shoes, and Jewelry (C4). The dataset comprises multiple features, including

Reviewer ID, Product ID (ASIN), Reviewer Name, Overall Review Score, Review Text, Product Summary, Total Votes Cast by the Reviewer, Product Image, Review Accuracy (Boolean Value), Product Attributes (e.g., Style), and Review Date. Table 3.2 shows examples of all the features and their values.

To enhance the predictive capabilities of our model, we utilized a summary of the review text along with helpfulness features, represented as binary values (0 or 1), which were derived by filtering the average rating attribute computed using a tendencies-based collaborative filtering algorithm. The selected features were subsequently preprocessed using natural language processing techniques. As part of the model training process, we employed a word embedding approach, wherein words were learned and mapped to embedding vectors by associating various word indices during the training phase. This embedding layer served as the input to our balanced CNN-BiLSTM model, enabling effective representation learning for review classification.

The input dimension of 50,000 represents the most frequently occurring words, with each token assigned a unique integer ranging from 0 to 49,999. The output embedding dimension, set to 50, defines the size of the dense vector space in which words are embedded. This dimensionality reduction is intended to minimize model complexity and computational cost, particularly when working with large datasets. The generated word embeddings capture semantic relationships between words within the given data context. In this approach, the embedding layer functions as a lookup table, where each of the top 50,000 most frequent words in the vocabulary is mapped to a corresponding 50-dimensional vector. Given that the study encompasses four dataset categories, the maximum sequence length of text statements varies across categories. Specifically, we observed maximum sequence length values of 21, 22, 23, 24, 25, 30, 38, and 44 tokens for different categories.

The input text sequence data was processed using a convolutional kernel of size 2×2. This kernel configuration enables the model to capture and interpret local word patterns while preserving essential contextual information, especially given the relatively short nature of the text sequences (i.e., review summaries). In this context, predicting helpfulness may rely on identifying concise yet significant word pairs. Four Convolutional (ConvID) layers were employed to extract meaningful feature representations. The first ConvID layer utilized a filter size of 100 to capture a broad range of textual features while maintaining a balance

between model complexity and feature extraction. The subsequent three ConvID layers incorporated 64 filters, a design choice that facilitates dimensionality reduction in deeper layers. This reduction is crucial for focusing on higher-level, abstract representations of the input data, thus enhancing the model's ability to discern meaningful patterns.

Additionally, a max-pooling layer was applied to downscale the spatial dimensions of the extracted feature maps. This layer selects the highest value from each filter kernel's receptive field, effectively reducing the height and width of the feature maps while preserving their most salient characteristics. A kernel size of 1×1 was employed in this layer, ensuring that key features are retained while mitigating computational complexity and minimizing the risk of overfitting.

After the max-pooling layer, the outputs were passed to a single Bidirectional Long Short-Term Memory (BiLSTM) layer with 200 hidden units. The bidirectional aspect of the LSTM allows the model to effectively capture both forward and backwards contextual dependencies, thereby improving the overall sequence representation. The use of 200 hidden units gives the model a high capacity for learning and retaining patterns over time, which is especially beneficial in natural language processing (NLP) tasks where understanding contextual relationships across a sequence is crucial. To enhance the model's generalization and robustness against data variations, a dropout rate of 0.5 was applied after each layer. Additionally, a batch normalisation layer was added to reduce the risk of overfitting by normalising the activations. The Rectified Linear Unit (ReLU) activation function was employed across all layers due to its computational efficiency and ability to promote representational sparsity, which supports the learning process.

To ensure a robust evaluation of the model, we trained our model on 20 epochs and employed a 5-fold cross-validation strategy using the KFold method. In each fold, the dataset was partitioned such that 90% of the data was allocated for training and validation, while the remaining 10% was used exclusively for testing. Within the 90% training portion, we further split the data, using 80% for actual training and 10% for validation. As a result, the data distribution for each fold consisted of 80% for training, 10% for validation, and 10% for testing. This approach enabled thorough performance assessment while maintaining data integrity and preventing information leakage from the test set during the training and validation phases.

The model was initialised with a learning rate of 0.01 and optimised using the Adam algorithm. The Mean Squared Error (MSE) loss function was employed to enhance accuracy by minimising prediction errors. The final layers of the model consisted of a fully connected architecture comprising three key components: a flatten layer, a dense layer, and an output layer. The dense layer utilised a sigmoid activation function, which is particularly suitable for binary classification tasks where each input sample is categorised as either helpful or unhelpful. The overall structure of the proposed model, along with the parameter configurations for this case, is presented in Figure 3.10.

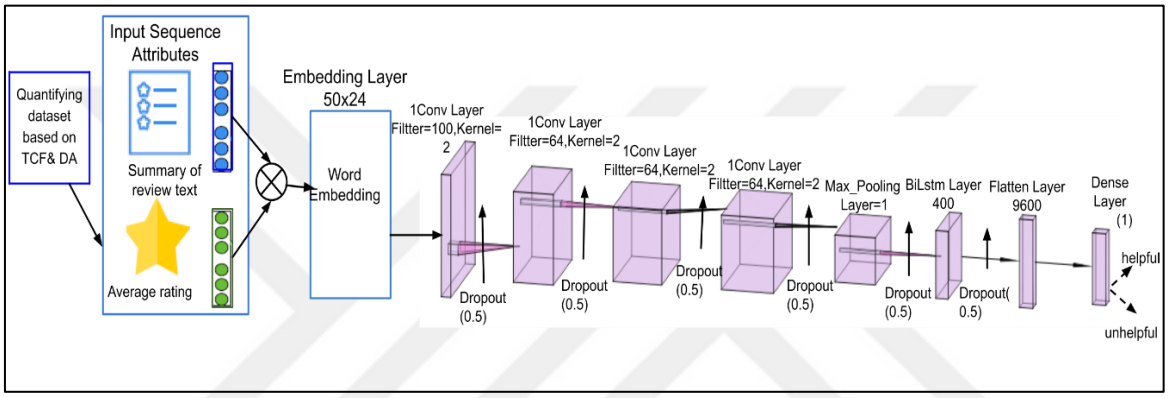


Figure 3.10. Construction of the proposed case study 1

3.4.2. Reproducible results of case study 2 (PRE-DHRRF)

This case study evaluates the proposed model using two subsets of the Amazon Review Dataset, Musical Instruments (DS1) and Clothing, Shoes, and Jewelry (DS2) in addition to the widely used Yelp Academic Review Dataset, which is commonly employed in e-commerce research. The evaluation process involves filtering essential attributes, including the overall item rating, unique item identifier (ASIN), reviewer ID, and the review summary text. The proposed framework is designed to generate two distinct pre-trained models capable of analyzing both helpful and unhelpful user reviews.

Furthermore, the study incorporates the Review Data File (DS3) from the Yelp dataset, which contains attributes such as review ID, user ID, business ID, star rating (ranging from 1 to 5), review text, and date. From this dataset, the selected features used in the analysis include review ID, business ID, star rating, and review text. Table 3.2 provides illustrative examples of attribute information, presenting one representative column from the Amazon

Review Dataset alongside one from the Yelp Academic Review Dataset. Each column is further broken down into feature–value pairs to enable a structured and comparative analysis.

Table 3.2. Examples of the Amazon review dataset and the Yelp academic review dataset

Amazon review dataset		Yelp academic review dataset	
Features	Values	Features	Values
overall	4	review_id	BiTunyQ73aT9WBnpR9DZGw
vote	2.0	user_id	OyoGAe7OKpv6SyGZT5g77Q
verified	True	business_id	7ATYjTIgM3jUlt4UM3IypQ
Review time	11 14, 2011	stars	5
Reviewer_id	A18ZXVF5AQFK35	useful	1
asin	0151004714	funny	0
style	{'style:': ' Straight to Straight'}	cool	1
Reviewer name	meep	text	I've taken many spin classes over the year...
Review text	Meep bought several for the home studio. They are nicel...	date	2012-01-03 15:28:18
Summary	very nice cables		
Unix review time	1321228800		
image	NAN		

Pre-trained models are widely recognized for their accuracy and robustness in text classification and other natural language processing (NLP) tasks. Due to their training on large-scale datasets, they offer improved generalization and help mitigate overfitting, making them particularly valuable in recommender system research. A key component of their effectiveness is the fine-tuning process, which ensures optimal adaptation to downstream tasks.

In this study, we utilize the saved weights of two pre-trained models: a custom Balanced CNN-BiLSTM model developed in Case Study 1, and the standard RoBERTa-base model. The Balanced CNN-BiLSTM combines convolutional neural networks (CNNs) with bidirectional long short-term memory (BiLSTM) units to extract sequential and spatial features. It was originally trained on Amazon review datasets and is reused in Case Study 2 for classifying review helpfulness. The structure of this model is illustrated in Figure 3.10.

To ensure consistency in evaluation, we adopted the same data partitioning strategy from Case Study 1 in Case Study 2, applying 5-fold cross-validation over 20 epochs. The

Balanced CNN-BiLSTM model was configured with the following hyperparameters: batch size of 256, learning rate of 0.01, dropout rate of 0.5, and the Adam optimizer. The maximum input lengths were set to 21 and 24 tokens for the Amazon Musical Instruments and Clothing datasets, respectively, and 690 tokens for the Yelp dataset. The vocabulary size was fixed at 50,000, and the embedding dimension was set to 50. The fully connected output layer employed a sigmoid activation function and mean squared error (MSE) as the loss function to classify reviews as helpful or unhelpful. The model contained 2,966,903 parameters, of which 2,965,391 were trainable.

The RoBERTa-base model, implemented in TensorFlow, is an enhanced version of BERT, pre-trained on 160 GB of text from sources including BooksCorpus, Wikipedia, and OpenWebText. To prepare our data for this model, we adapted each dataset into lists of review texts and corresponding helpfulness labels. These inputs were tokenized using the RoBERTa tokenizer, which maps text tokens to integer indices within a 50,000-token vocabulary. Each token sequence is then embedded into a 768-dimensional vector with positional encoding to preserve word order.

The RoBERTa model comprises 12 transformer layers, each containing multi-head self-attention mechanisms and feed-forward sub-networks. The model's training configuration included a batch size of 32, 10 epochs, a learning rate of 7e-5, and 3072 hidden units per feed-forward layer, with GELU (Gaussian Error Linear Units) as the activation function. The model was regularized using a dropout rate of 0.1 and optimized using Adam. In total, RoBERTa-base includes 124,647,170 trainable parameters.

Within the PRE-DHRRF framework, we fine-tuned all layers of both the Balanced CNN-BiLSTM and RoBERTa models on domain-specific datasets to maximize performance in classifying helpful versus unhelpful reviews. For binary classification, we compiled the models using the Binary Cross-Entropy loss function and the Adam optimizer. This fine-tuning ensures full adaptation of the pre-trained architectures to the target classification task. Table 3.3 summarizes all hyperparameters and configuration settings applied in the two case studies.

Table 3.3. Hyperparameters are set up in our models (BHRQUT and PRE-DHRRF)

Hyperparameter	BHRQUT model	PRE-DHRRF model	
		pre-trained model I. Balanced CNN-BiLSTM	pre-trained model.II (Roberta-base)
Batch Size	256	256	256
Number of Epochs	20	20	20
K-Fold	5	5	5
Activation Function	Relu	Relu	Gelu
Dense Activation	Sigmoid	Sigmoid	Sigmoid
Optimizer	Adam	Adam	Adam
Loss Function	MSE	MSE	Binary Categorical Crossentropy
Dropout	0.5	0.5	0.1
Learning Rate	0.01	0.01	5.00E-07
MAX_NB_WORDS	50,000	50,000	50,000
Embedding Dimension	50	50	768
maxLen (Input Lengths)	21,22,23,24,25,30,38,44	21, 24, 690	21, 24, 690

3.5. Fine-Tuning and Performance Evaluation

The final step is to analyze and calculate the quality score utilizing the values of each attribute. This is achieved through the weighted sum method. The three main factors establishing the quality criteria are structural elements, metadata, and readability. Reviews that attain higher quality scores indicate superior quality. This stage concludes with the calculation of a quality score and the ranking of reviews based on quality. Model evaluation involves using various metrics to assess the performance, advantages, and disadvantages of a deep learning or machine learning model. The model's effectiveness should be evaluated during the early stages of research. Additionally, model evaluation aids in monitoring its performance in the future. Performance measures include the confusion matrix, generalization gap, accuracy, precision, recall, F1 score, RMSE, and MAE.

To evaluate the classification performance of our proposed models, we employed several metrics, including confusion matrix, Accuracy, Precision, Recall, F1-score, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and generalisation gap. The mathematical formulations of these metrics are presented in “Eq. 3.31 -3.38”. The popular Python machine learning library, sklearn.library provides these measurement metrics.

In our study, the confusion matrix is structured with two rows: the first row comprises True Negatives (TN) and False Positives (FP), while the second row contains False Negatives (FN) and True Positives (TP). Specifically, True Positives (TP) represent the number of actual helpful reviews correctly classified as helpful, whereas True Negatives (TN) denote the number of actual unhelpful reviews accurately identified as unhelpful. Conversely, False Positives (FP) refer to the number of unhelpful reviews incorrectly predicted as helpful, and False Negatives (FN) indicate the number of helpful reviews misclassified as unhelpful.

Among the various evaluation metrics, accuracy is the most commonly utilised. It is defined as the ratio of correctly classified reviews, both helpful and unhelpful, to the total number of reviews. This metric provides an overall measure of the model's classification performance by assessing the proportion of correctly identified instances across all categories.

The accuracy ($Accuracy$) of the model is determined by the ratio of correctly classified instances, both helpful and unhelpful reviews, to the total number of reviews, as formulated in "Eq. 3.31".

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.31}$$

Where TP denotes true positives, TN represents true negatives, FP signifies false positives, and FN corresponds to false negatives.

Precision quantifies the proportion of correctly predicted helpful reviews out of all reviews classified as helpful by the model. It is mathematically defined in "Eq. 4.2".

$$Precision = \frac{TP}{TP+FP} \tag{3.32}$$

Similarly, Recall measures the percentage of correctly predicted helpful reviews relative to the total number of truly helpful reviews, as given in "Eq. 3.33".

$$Recall = \frac{TP}{TP+FN} \tag{3.33}$$

The F1-score provides a harmonic mean of precision and recall, ensuring a balanced evaluation between these two metrics. A higher F1-score indicates a stronger classification capability of the recommender system. The F1-score is formulated in “Eq. 3.34”.

$$F1 - score = 2.\frac{Precision.Recall}{Precision+Recall} \tag{3.34}$$

In addition to classification metrics, statistical evaluation measures such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used to assess prediction performance by quantifying the deviation between actual and predicted target values. The MAE, representing the average absolute difference between predicted and actual values, is defined in “Eq. 3.35.”

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y_i} - y_i\right| \tag{3.35}$$

Where $\hat{y}i$ represents the predicted value, yi is the actual value, and n denotes the total number of samples.

RMSE, a widely used metric for error measurement, assigns greater weight to larger errors, thereby penalizing significant deviations more than MAE. A lower RMSE value indicates higher prediction accuracy. The RMSE is given by “Eq.3.36”.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y_i} - y_i\right)^2} \tag{3.36}$$

Where ($\hat{y}i - yi$) represents the difference between the predicted and actual target values.

Mean Squared Error (MSE) is employed as the loss function for model training. The objective function used to optimise the model parameters is formulated in “Eq. 3.37”.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y_i} - y_i\right)^2 \tag{3.37}$$

Finally, the absolute difference in performance (Accuracy, loss) between a model's performance on a training dataset and its performance on test or validation data that hasn't been seen yet is known as the generalisation gap [95].

$$\text{Generalization Gap} = |\text{Validation Accuracy} - \text{Training Accuracy}| \quad (3.38)$$

These evaluation metrics collectively provide a comprehensive assessment of the model's classification and predictive performance in our case study 1.

3.6. Experiment Configuration

The running of our experiments implemented using Python high level language on Intel(R) Core (TM) i7-5500U CPU, 2.40 GHz, 64-bit operating system, x64-based processor on TITAN XP graphics cards donated by NVIDIA GPU RAM 0.0 / 40.0 GB, System RAM 2.4 / 83.5 GB, Disk 32.5 / 235.7 GB with 100 compute units approximately usage rate 11 per hour one day.

NVIDIA is a leading technology company specialising in graphics processing units (GPUs), artificial intelligence (AI), and high-performance computing. NVIDIA GPU develops the TITAN XP graphics cards as available hardware resources. It is a specific graphics card (GPU) used for running the experiments, and TITAN XP is A high-performance GPU (Graphics Processing Unit) commonly used for deep learning, AI, and computational workloads [96].

Many libraries are used to run deep learning algorithms, conduct studies, and solve problems. Today, existing libraries are constantly being updated, and new ones are being made available, but no single library is completely dominant. Users work with different libraries based on their experiences. Some of the libraries, scikit-surprise, NLTK, TensorFlow, and Keras packages used to conduct our deep learning studies are as follows:

TensorFlow: is Google develop an end-to-end open-source machine learning platform. It provides a comprehensive ecosystem of flexible tools, libraries, and community-driven resources to support various machine learning (ML) and deep learning (DL) applications. TensorFlow enables users to efficiently build and deploy machine learning models, from research prototypes to production-scale applications. One of its key advantages is its multi-level abstraction, allowing developers to select the most appropriate level of control depending on the complexity of the problem. For instance, high-level APIs such as Keras simplify the process of building and training neural networks. At the same time, low-level

operations offer more flexibility for researchers and advanced users who need finer control over model architecture and optimisation. Moreover, TensorFlow supports deployment across various platforms, including cloud services, edge devices, and mobile applications. It also integrates seamlessly with hardware accelerators like GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) to enhance computational efficiency. The framework is widely used in healthcare, finance, autonomous systems, and natural language processing (NLP) industries. Due to its open-source nature and strong community support, TensorFlow continues to evolve, incorporating state-of-the-art machine learning techniques and providing extensive documentation, tutorials, and pre-trained models to facilitate adoption [97].

Keras: Keras is a high-level deep learning API that provides an intuitive and user-friendly interface for building, training, and deploying neural networks. Unlike lower-level frameworks, Keras prioritises ease of use, enabling researchers and practitioners to develop complex deep learning models with minimal code. It is specifically designed for humans, not machines, making it an accessible tool for beginners and experienced machine learning engineers. One of Keras's key advantages is its consistent and modular API design, which allows users to prototype models quickly and experiment with different architectures. It supports rapid debugging by offering precise and actionable feedback to help identify errors efficiently. Additionally, Keras is highly extensible, allowing for easy addition of custom layers, loss functions, and metrics, enhancing its adaptability for cutting-edge research. Keras seamlessly integrates with TensorFlow, which serves as its backend, enabling efficient execution on both CPUs and GPUs. The framework is widely used in academic research, industry applications, and educational contexts due to its simplicity and flexibility. Moreover, Keras provides a vast ecosystem of pre-trained models and reusable code, making it well-suited for transductive learning and practical deployment in real-world scenarios [98].

NLTK: The Natural Language Toolkit (NLTK) is a comprehensive open-source library for natural language processing (NLP) in Python, widely used in corpus linguistics and text analysis. NLTK provides a broad range of tools and methods to clean and preprocess text data before analysis by other NLP models. This includes removing common words (like "the, " "is, " "in") that do not add significant meaning to the analysis, counting how often each word appears in a corpus, identifying pairs or groups of words that frequently occur together, tokenisation, stemming, lemmatisation, part-of-speech (POS) tagging, named

entity recognition (NER), and syntactic parsing, making it a fundamental library for automated text analysis and linguistic research. It includes access to predefined corpora and lexical resources, such as the WordNet database, enabling researchers to analyse linguistic structures and relationships effectively. One of NLTK's key applications is automated sentiment analysis, which facilitates classifying text data into categories such as positive, negative, or neutral sentiments. By integrating machine learning models and sentiment lexicons, NLTK enables sentiment classification across various domains, including social media analysis, customer feedback evaluation, and market trend prediction. The library also supports text preprocessing techniques such as stopword removal and word frequency analysis, which are essential for enhancing the performance of NLP models. Due to its modular structure and user-friendly interface, NLTK is widely adopted in academic research and industry applications, serving as an essential tool for text mining, natural language understanding (NLU), and computational linguistics. Its role in corpus research and automated sentiment analysis makes it a valuable resource for researchers and developers working on machine learning-based text classification and NLP-driven applications [99].

Scikit-Surprise: The Surprise library is a Python scikit designed explicitly for building recommendation systems. It provides a variety of algorithms and tools for rating prediction and collaborative filtering, making it a versatile library for anyone working on recommender systems. It supports multiple recommendation algorithms, including memory-based and model-based collaborative filtering techniques, and allows for easy integration of different similarity measures. One of the key features of the Surprise library is its straightforward installation process, which can be carried out using a standard pip command. This ease of installation makes it accessible for developers looking to prototype and test various recommendation system algorithms quickly. Designed to mirror the scikit-learn API closely, Surprise simplifies the transition from users familiar with scikit-learn to those working with recommendation systems. It provides a collection of estimators (or prediction algorithms) that can be utilised for rating prediction. These algorithms include popular ones like SVD (Singular Value Decomposition), KNN-based collaborative filtering, and baseline algorithms. Each algorithm is tailored for rating prediction in collaborative filtering-based recommendation systems. Moreover, Surprise supports various similarity indices to compare user preferences or item features, which is crucial for collaborative filtering approaches. This makes it easy for users to experiment with different similarity measures to find the most suitable one for their problem domain [100].

4. EXPERIMENTAL RESULTS

4.1. Case Study 1 (BHRQUT): Results and Experimental Validation

The effectiveness of the proposed model in Case Study 1 was evaluated using four dataset categories (C1, C2, C3, and C4), each consisting of two paths, with two cases per path. Consequently, each category contained four cases: case (1_1), case (1_2), case (2_1), and case (2_2), all aimed at classifying reviews into two labels: helpful and unhelpful. The TCF algorithm was employed for optimization to enhance data quality, and dataset balancing was achieved through the Synonym-Based Unhelpful Review Augmentation (SbURA) algorithm. The sampling structure of the datasets following these preprocessing steps is illustrated in Figure 4.1.

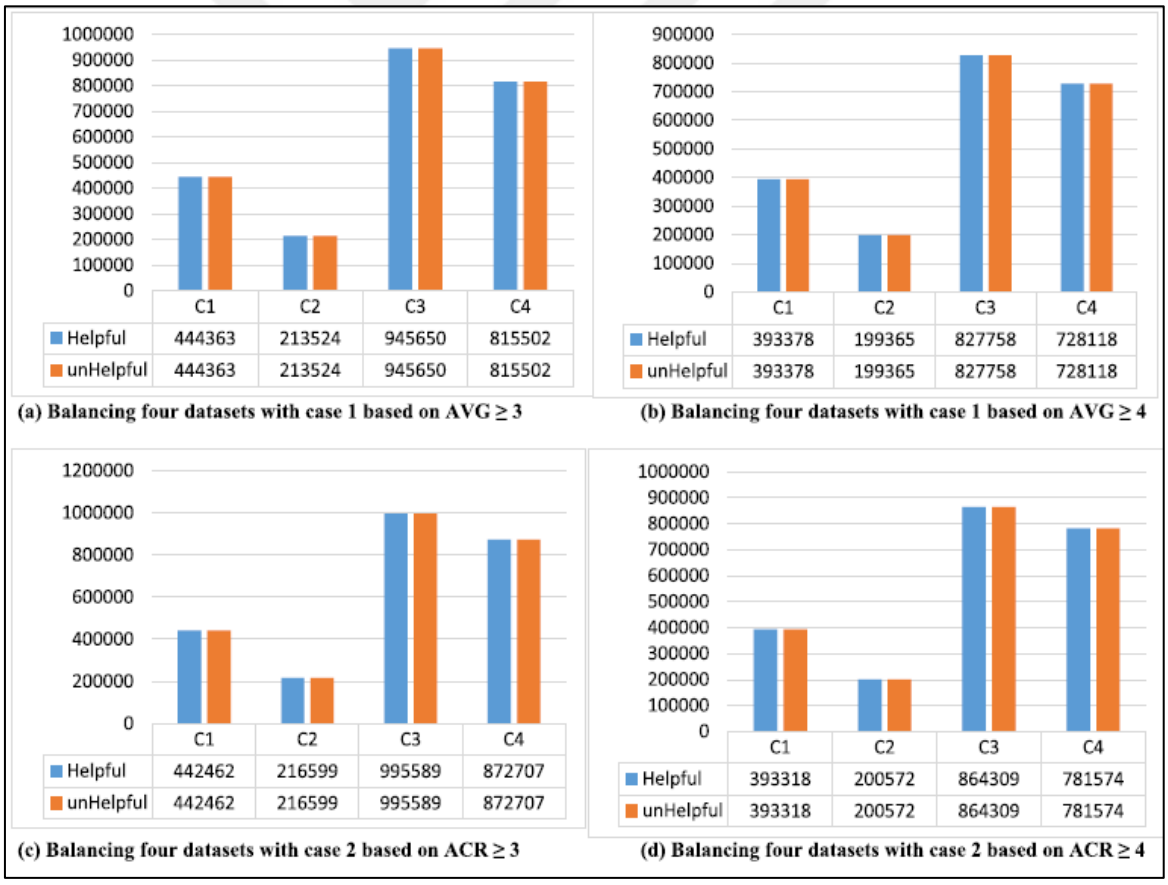


Figure 4.1. Balancing data for the four Amazon-5 score categories of datasets in path1 and path2

Subsequently, the data underwent a structured text preprocessing pipeline before being trained using a deep balanced CNN-BiLSTM model. Additionally, as presented in Table 4.1,

the evaluation metrics were determined based on a loss function (or cost function), which quantifies the deviation between predicted and actual values while considering the probability and uncertainty of predictions. This analysis provided a comprehensive assessment of the model's performance. The validation and training loss values were recorded, and experimental results indicate that case (1_1) exhibited the lowest error value across all dataset categories.

Table 4.1. Validation and training loss results of our model for each category (C1, C2, C3, and C4) in each case name

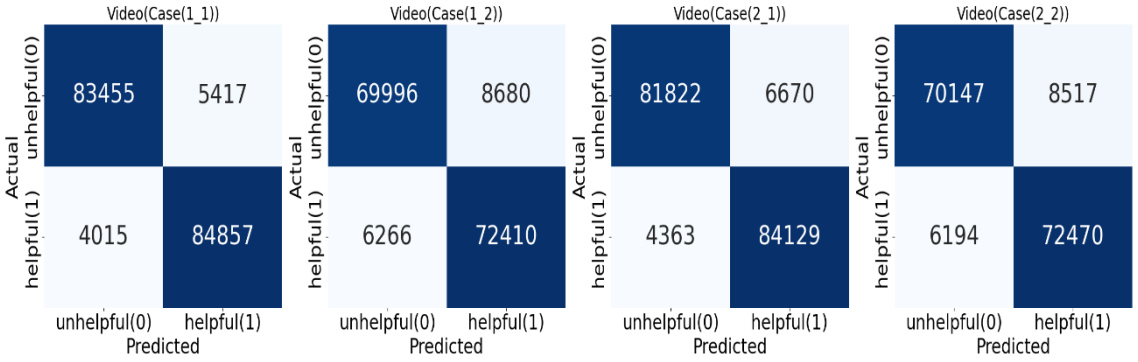
Case Name	Category of the Dataset	Loss Value	
		Val	Train
Case (1_1)	C1	0.04	0.03
	C2	0.03	0.02
	C3	0.05	0.03
	C4	0.05	0.04
Case (1_2)	C1	0.07	0.05
	C2	0.05	0.03
	C3	0.08	0.06
	C4	0.07	0.05
Case (2_1)	C1	0.05	0.03
	C2	0.03	0.02
	C3	0.05	0.04
	C4	0.05	0.04
Case (2_2)	C1	0.07	0.04
	C2	0.05	0.03
	C3	0.09	0.06
	C4	0.07	0.06

Accuracy is the most widely utilised evaluation metric in performance assessment. To evaluate the predictive capability of the proposed recommendation framework, we analysed the training accuracy, as presented in Table 4.2. Various graphical measures can be employed to illustrate a model's predictive performance. The proposed model achieved a high accuracy of 97% with a low loss value of 0.03 when tested on the C2 dataset in case (1_1). Conversely, the lowest accuracy of 88% with a higher loss value of 0.09 was observed in case (2_2) when tested on the C3 dataset. Further analysis revealed that model performance deteriorated in case (1_2) when the average rating (avg) was greater than or equal to 4, and in case (2_2) when the rating exceeded 4. Therefore, the optimal helpfulness ratio should be maintained at avg $\geq$ 3 to enhance the recommendation performance.

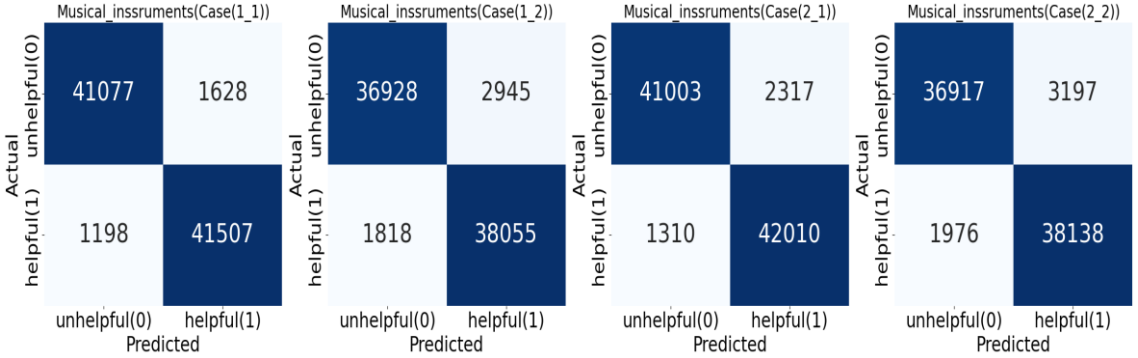
Table 4.2. Results of our model for each category (C1, C2, C3, and C4) in terms of Validation and Training (Precision, Recall, F1-score, and Accuracy) for each case name

Category of Amazon Score-5	Case Name	Precision		Recall		F1-Score		Accuracy	
		Val	Train	Val	Train.	Val	Train.	Val	Train
Cl	case (1-1)	94%	96%	95.4%	98%	95%	97%	95%	97%
	case (1-2)	89.4%	93%	92%	95%	90.6%	94%	90.4%	94%
	case (2-1)	92.6%	94%	95%	98%	94%	96%	94%	96%
	case (2-2)	89.8%	93%	92.2%	96%	91%	95%	90.8%	94%
Average								92.55%	95.25%
C2	case (1-1)	96.2%	97%	96.8%	99%	97%	98%	97%	98%
	case (1-2)	92.8%	96%	95.4%	98%	94%	97%	94%	97%
	case (2-1)	94.8%	97%	97%	99%	96%	96%	96%	98%
	case (2-2)	92.4%	95%	95%	98%	94%	97%	94%.	97%
Average								95.25%	97.5%
C3	case (1-1)	93.4%	95%	94.4%	96%	94%.	96%	94%	96%
	case (1-2)	86.6%	90%	91.6%	95%	89%	93%	88.8%	92%
	case (2-1)	92%	95%	93.8%	96%	93%	95%	93%	95%
	case (2_2)	85.6%	89%	91%	96%	88%	92%	88%	92%
Average								90.95%	93.75%
C4	case (1-1)	92.2%	93%	94.2%	97%	93%	95%	93%	95%
	case (1-2)	89.2%	91%	92.6%	95%	91%	93%	90.8%	93%
	case (2-1)	91.8%	94%	94.2%	96%	93%	95%	93%	95%
	case (2-2)	88%	91%	93%	95%	90.8%	93%	90%	93%
Average								91.7%	94%

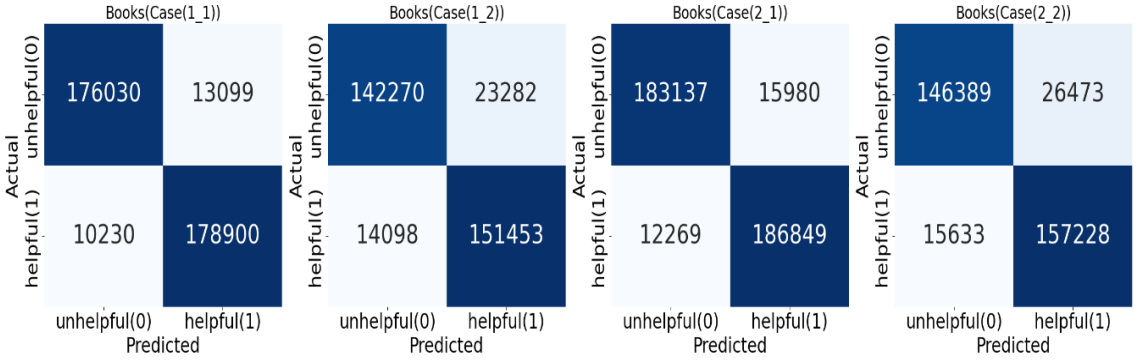
A confusion matrix is a widely adopted tool for evaluating the performance of classification models. In the context of our proposed model, the confusion matrix displays the predicted classes along the horizontal axis and the actual classes along the vertical axis, enabling a clear distinction between correct and incorrect classifications. The diagonal elements of the matrix represent the number of true positives (TP) and true negatives (TN), offering a reliable indicator of classification accuracy. This structure is instrumental in deriving the components of the classification report across different training data categories. The results derived from the confusion matrices indicated an increase in the number of accurate positive predictions across all cases and classes, thereby contributing to the overall enhancement of model accuracy and quality. Figure 4.2. presents the confusion matrices generated during the testing phase of the proposed model. Specifically, subfigures (a), (b), (c), and (d) correspond to the four cases applied to the respective datasets C1, C2, C3, and C4.



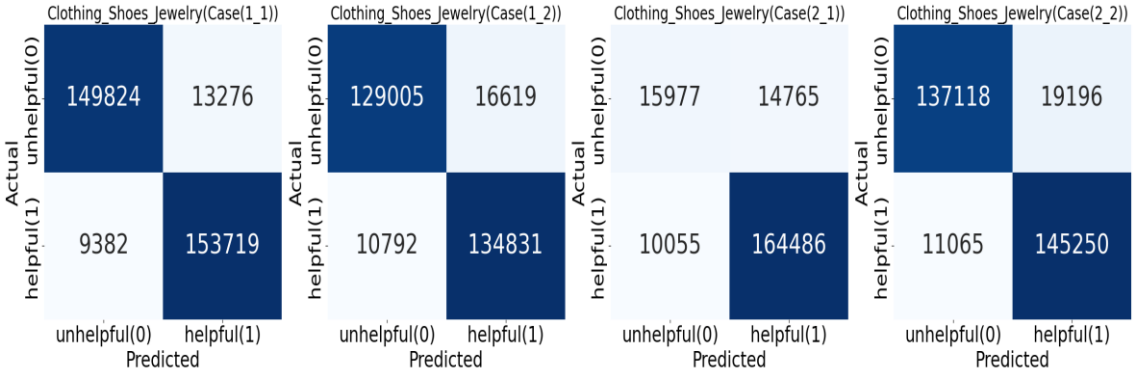
(a) Confusion matrices of the model in case (1_1), case (1_2), case (2_1), and case (2_2) for C1



(b) Confusion matrices of the model for case (1_1), case (1_2), case (2_1), and case (2_2) for C2



(c) Confusion matrices of the model for case (1_1), case (1_2), case (2_1), and case (2_2) for C3



(d) Confusion matrices of the model for case (1_1), case (1_2), case (2_1), and case (2_2) for C4

Figure 4.2. Confusion matrices (a-d) for the model of case study 1 in path1 and path2, featuring two cases for the C1, C2, C3, and C4 datasets

Table 4.3. presents a comparative analysis of various existing methods alongside the proposed approach, providing a clearer understanding of these studies through evaluation metrics such as RMSE, MSE, and MAE to measure the impact of different user opinions. Several notable models have been introduced in the literature, each incorporating distinct methodologies to enhance personalised recommendation services. The RHRN model [9] introduces a three-phase architecture: a review semantics extractor, a user profile constructor, and a user/item recommendation generator. This framework employs a hybrid CNN–BiLSTM model to classify the helpfulness of reviews based on vote value, thus assisting users in making informed purchasing decisions. The AFRAM model [63] focuses on capturing latent features from customer reviews and ratings. It incorporates a dual-path architecture consisting of CNN and LSTM networks to develop an aspect-based fashion recommendation system enhanced with an attention mechanism. The Machine-Review Generation Model [78] explores context-aware attention mechanisms applied at two levels: human-written reviews and machine-generated reviews. This model utilizes DNN and RNN frameworks to generate and explain helpful reviews. The DUPIA model [101] predicts user ratings by analyzing user reviews and considering diverse user preferences. It introduces a CNN-based approach with attention mechanisms to distinguish between user preferences and item attractiveness. Lastly, the approach in [102]employed Matrix Factorization (MF) to evaluate predictive ratings and helpful recommendations, enhancing this process by extracting item feature matrices from customer reviews using Latent Dirichlet Allocation (LDA) analysis.

Table 4.3. Performance comparison of helpfulness review analysis models based on RMSE, MSE, and MAE

Models	RMSE				MSE				MAE			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
RHRM [9]			0.592								0.468	
AFRAM [63]								1.098				0.801
Machine-generated reviews[78]	0.881											
DUPIA [101]		0.833										
Deep Feature Extractor [102]					1.581	1.475	1.221	1.211				
BHRQUT. On case (1_1) with avg>=3	0.204	0.161	0.218	0.223	0.041	0.026	0.047	0.050	0.071	0.038	0.089	0.091

Table 4.3. (continued) Performance comparison of helpfulness review analysis models based on RMSE, MSE, and MAE

Models	RMSE				MSE				MAE			
	C1	C2	C3	C4	C1	C2	C3	C4	C1	C2	C3	C4
BHRQUT. On case (1_2) with avg>=4	0.265	0.219	0.290	0.264	0.070	0.048	0.084	0.069	0.133	0.073	0.154	0.139
BHRQUT. On case (2_1) with avg>=3	0.217	0.184	0.233	0.225	0.047	0.034	0.054	0.051	0.076	0.047	0.102	0.101
BHRQUT. On case (2_2) with avg>=4	0.263	0.226	0.301	0.268	0.069	0.051	0.089	0.072	0.126	0.083	0.179	0.145

To assess the impact of sequence length on model performance, we selected the maximum text sentence length from the dataset and evaluated its effect across different cases and four datasets, as presented in Table 4.4. Our findings indicate that a maximum sequence length of 21 yields the best performance in terms of accuracy and loss function, demonstrating superior results compared to longer sequences.

Table 4.4. Effect of the max sequence length of the input sentences on the performance of the (BHRQUT) model

Dataset Name	Case Name	Max-Length	Accuracy	Loss
Cl	case (1_1)	30	95%	0.04
	case (2_1)	30	94%	0.03
	case (1_2)	30	91%	0.07
	case (2_2)	38	91%	0.07
C2	case (1_1)	21	97%	0.03
	case (2_1)	23	96%	0.03
	case (1_2)	24	94%	0.05
	case (2_2)	25	94%	0.05
C3	case (1_1)	24	94%	0.05
	case (2_1)	23	93%	0.05
	case (1_2)	22	89%	0.08
	case (2_2)	23	88%	0.09
C4	case (1_1)	23	93%	0.05
	case (2_1)	23	93%	0.05
	case (1_2)	44	91%	0.07
	case (2_2)	44	90%	0.07

In a separate study [84], a deep learning approach utilizing a CNN and an attention-based BiGRU model was proposed to enhance sentiment analysis of reviews. This method employs

deep learning algorithms to extract sentiment, distinguishing between positive and negative reviews effectively based on rating value. We examine the impact of the maximum review text length by establishing an optimal sequence length and comparing the classification performance of the RHRM [9] and SLCABG [84]models within the book category(C3). The evaluation was conducted using Precision, Recall, F1-Score, and Accuracy metrics. The results indicate that the optimal maximum sequence length was 24 for case (1_1) of our model, while the SLCABG model, with a maximum text length of 648, outperforms the RHRM model, which uses a text length of 2817. These findings are further detailed in Table 4.5.

Table 4.5. Comparison of the Impact of Maximum Input Sentence Length on the BHRQUT Model and Other Models within the Book Category (C3)

Models	measured score for helpfulness	Max•length	Precision	Recall	F1 *_score*	Accuracy
RHRM [9]	vote	2817	85.54%	88.73%	87.10%	86.14%
SLCABG [84]	rating	648	92.9%	93.8%	93.3%	93.4%
Our BHRQUT. case (1_1)	Average rating	24	93.4%	94.4%	94%	94%

4.2. Case Study 2 (PRE-DHRRF: Results and Experimental Validation

The effectiveness of the proposed model was evaluated using two categories of Amazon review datasets (DS1 and DS2) and one category of the Yelp Academic Review dataset. The evaluation involved training the datasets on two pre-trained models: Pretrained Model I (Balanced CNN-BiLSTM architecture) and Pretrained Model II (Roberta-base). For each dataset category, the models were designed to perform binary classification, categorizing reviews as either helpful (1) or unhelpful (0). We prepared the data using the steps of data preprocessing. To investigate the effect of our model on each maximum sequence review length, we used both small and large max-length datasets. This case study introduces a model that analyses customers' helpful reviews of products to improve decision-making accuracy in online shopping and avoid the problem of overfitting with balancing datasets when the system encounters noisy and irregular variations in the training dataset. The separation of the samples of the three datasets before and after balancing using the (SbURA) approach [80]is depicted in Figure 4.3.

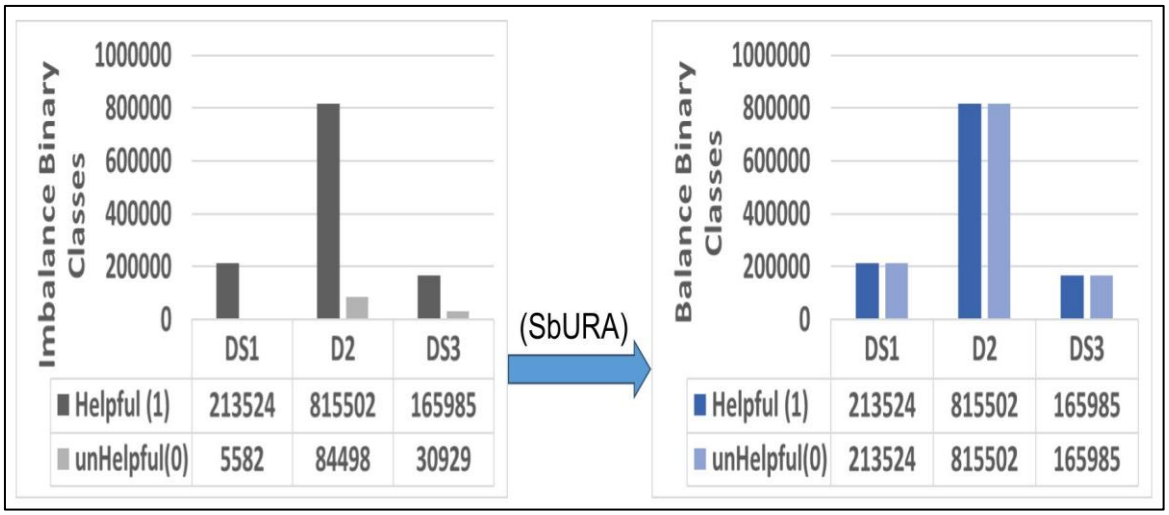


Figure 4.3. Balancing three datasets (DS1, DS2, and DS3) on the SbURA algorithm

To better classify helpful or unhelpful reviews, we preferred only invaluable reviews (average rating >= 3) and unhelpful reviews (average rating < 3) as the training data set. The average rating variable is computed from the average of the summation of actual and predicted ratings, calculated using four tendency algorithm conditions [17].

Our model achieved the best metrics in classification performance. Our results are comparable to those of other studies, Differentiated Recommendation Framework [23], the authors in this model recommend product list are generated by combining the feature extraction knowledge graph generated through data augmentation based on deep learning and knowledge transfer using the FM (Factorization Machine) model. This method provides recommendations based on ratings for active users and solves cold start problems. In the CF(NRCF) + CF(DCCF) model [103], representative and influential neighbours are selected based on users' valuable comments and consistent ratings. NRCF (Neighbour Constrained Collaborative Filtering) and DCCF (Dual Candidate Collaborative Filtering) algorithms support this selection process. Deep learning-based textual review [26], the authors explored the performance of deep learning networks and transformer models in review-based recommender systems, aiming to enhance recommendation accuracy by effectively utilising textual reviews. The Sentiment Analysis System [104] is structured into three distinct stages. In the initial stage, a web scraping module collects user-generated comments from various websites and social media platforms. The second stage involves applying sentiment analysis techniques to classify the collected comments into positive and negative categories. In the

final stage, collaborative filtering methods generate personalized recommendations based on the analysed sentiments.

In our case study 2, we used eight different statistical metrics to evaluate the classification performance of the proposed model: accuracy, precision, recall, F1 score, misclassification rate (MAE, RMSE), confusion matrix, and generalisation gap. Each experiment was performed for 100 iterations with 5-fold cross-validation over 20 epochs to obtain consistent and reliable results. This approach successfully addressed the overfitting problem by improving both model accuracy and the reliability of the exploratory analyses. In the training process, 80% of the samples in each dataset were randomly selected for training, while the remaining 20% were reserved for testing.

A confusion matrix is a detailed table designed to assess the effectiveness of supervised learning methods, particularly in machine learning and classification tasks. It provides an in-depth analysis of the classifier's performance by breaking down predictions into various categories, offering valuable insights into its accuracy and reliability Table 4.6. The evaluation metrics include the total number of samples (N-Actual), True Negatives (TN), False Positives (FP), True Positives (TP), and False Negatives (FN). The results demonstrate that Model II (Roberta-base) consistently outperforms Model I (BalancedCNN-BiLSTM) across all datasets, indicating superior classification performance. The results suggest that Model II demonstrates superior performance across all datasets by achieving higher TN and TP rates while maintaining lower FP and FN rates. This comparative analysis highlights the reliability and robustness of the Roberta-base model for classification tasks, outperforming the BalancedCNN-BiLSTM model across multiple datasets. These findings underscore the potential of pre-trained transformer-based models in delivering more accurate and efficient results in text classification tasks.

Table 4.6. Confusion matrix of our framework for each pre-trained model with three datasets

Pre-trained Model Name	Dataset	N-Actual	TN	FP	TP	FN
Model. I (BalancedCNN-BiLSTM)	DS1	85409	41171	1534	41455	1249
	DS2	326201	149004	14096	154676	8425
	DS3	66394	31359	1838	31370	1827
Model.II Roberta-base	DS1	85409	41793	912	42192	513
	DS2	326201	154197	8903	155755	7346
	DS3	66394	31456	1741	31991	1206

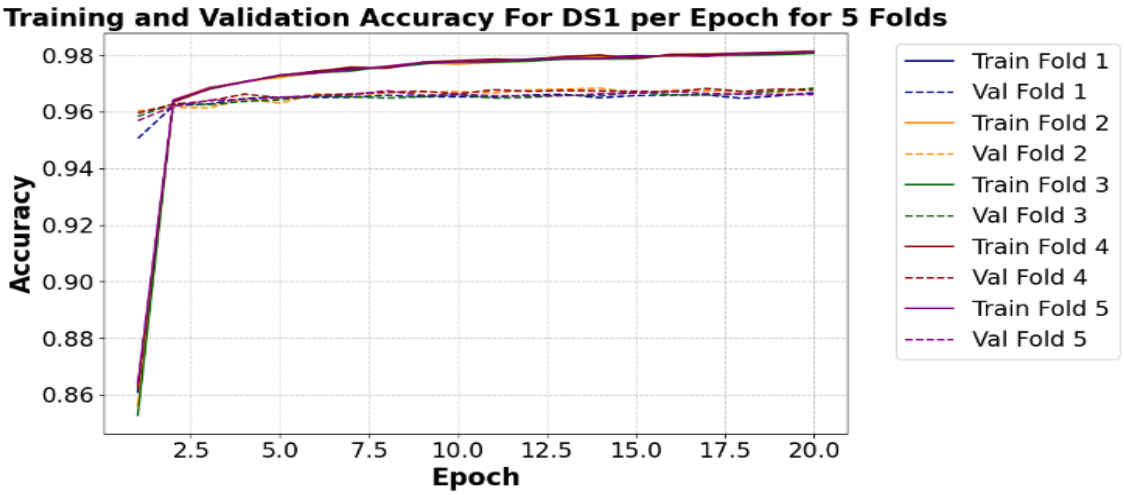
In evaluating the performance of deep learning models, particularly in the context of review classification and recommendation systems, it is essential to assess accuracy and examine how well the model generalises to unseen data. The generalisation gap, defined as the difference between training and validation accuracy, is an important gauge for the model's robustness and susceptibility to overfitting. A small generalisation gap indicates that the model performs consistently on the training and validation datasets, reflecting its ability to capture relevant patterns without memorising the data. Conversely, a significant gap may suggest overfitting, where the model learns noise or dataset-specific characteristics that do not transfer to new inputs. Therefore, incorporating the generalisation gap into model evaluation provides deeper insight into model behaviour, reliability, and real-world applicability, especially vital in recommendation systems where diverse user inputs and dynamic data patterns are prevalent. Table 4.7. presents a comparative evaluation of the generalisation gap, accuracy, and loss for two pre-trained deep learning models, Model I (Balanced CNN-BiLSTM) and Model II (RoBERTa-base), across three different datasets: DS1, DS2, and DS3. The generalisation gap is computed as the difference between validation and training accuracy, indicating the model’s ability to generalise from the training data to unseen validation data.

Table 4.7. Evaluation of the generalization gap for the proposed model across three datasets

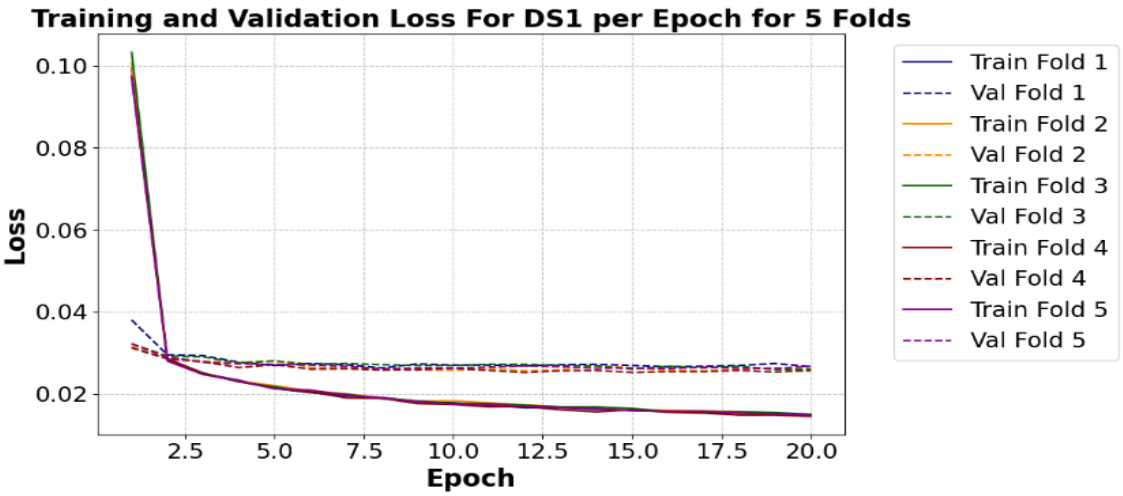
Pre-trained Model Name	Dataset	Accuracy		Generalization Gap	Loss	
		Val	Train		Val	Train
Model.I (BalancedCNN-BiLSTM)	DS1	0.97	0.98	0.01	0.027	0.022
	DS2	0.93	0.95	0.02	0.050	0.046
	DS3	0.945	0.988	0.043	0.060	0.023
Model.II (Roberta-base)	DS1	0.983	0.986	0.003	0.065	0.062
	DS2	0.95	0.954	0.004	0.114	0.110
	DS3	0.956	0.962	0.006	0.116	0.110

The training, validation accuracy, and loss subgraphs are shown in Figure 4.4. (a-f), Figure 4.5. (g-l), and (See Table 4.7.) were plotted to assess our model performance using two pre-trained models (Model I and Model II). The accuracy of DS1's training and validation using pre-trained Model I is shown in Figure 4.4 (a), which shows that the model could effectively converge as it rose continuously and plateaued around epoch 20. The training and validation loss for DS1 using pre-trained Model I is displayed in Figure 4.4. (b), The model's ability to withstand overfitting on this dataset is demonstrated by the small difference between training

and validation losses. The training and validation accuracy for DS2 utilising pre-trained Model I is shown in Figure 4.4. (c), where comparable patterns were noted, with limited divergence and high accuracy. The training and validation loss for DS2 using pre-trained Model I, which likewise converged with minimum variance, is shown in Figure 4.4. (d). Using pre-trained Model I, Figure 4.4. (e) and (f) demonstrate the minimal difference between training and validation accuracy and loss, which suggests that the model finds a balance between learning from the training data and generalising to new data. The durability of the concept is demonstrated by regularity across folds when applied to DS3.

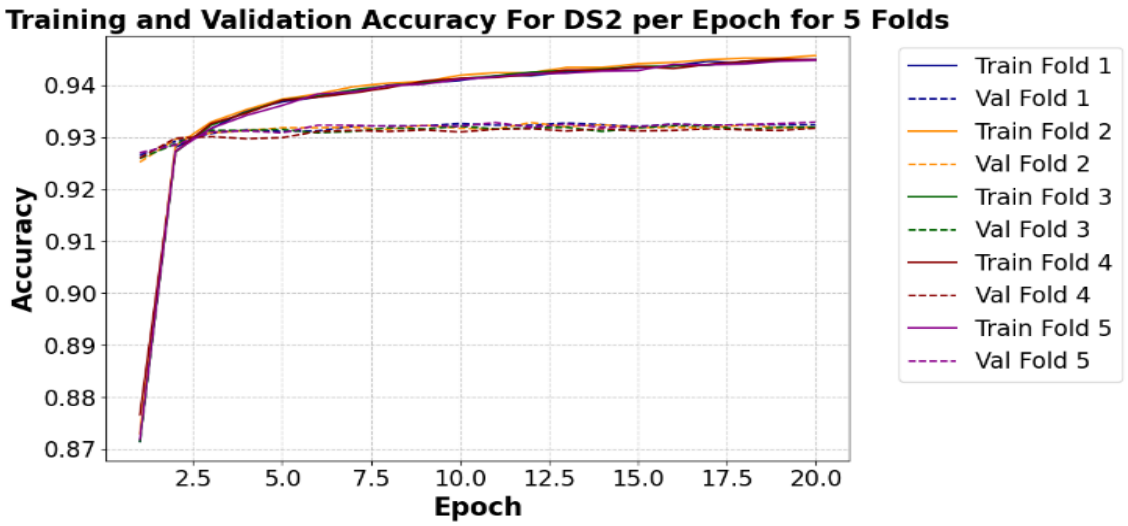


a. Accuracy of trained DS1 using pre-trained Model I

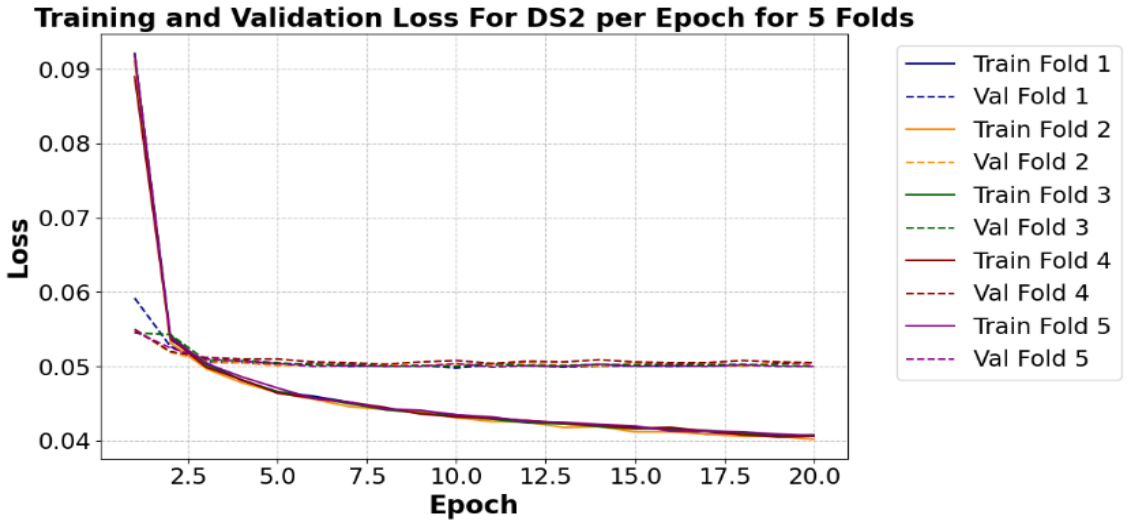


b. Loss of trained DS1 using pre-trained Model I

Figure 4.4. (a-f) Accuracy and loss during training and validation in five-fold cross-validation. Each row represents our model, tested on pre-trained Model I (Balanced CNN-BiLSTM) with the corresponding dataset (DS1, DS2, DS3) respectively in each left-column subplot (train accuracy vs. validation accuracy) and each right column (train loss vs. validation loss)

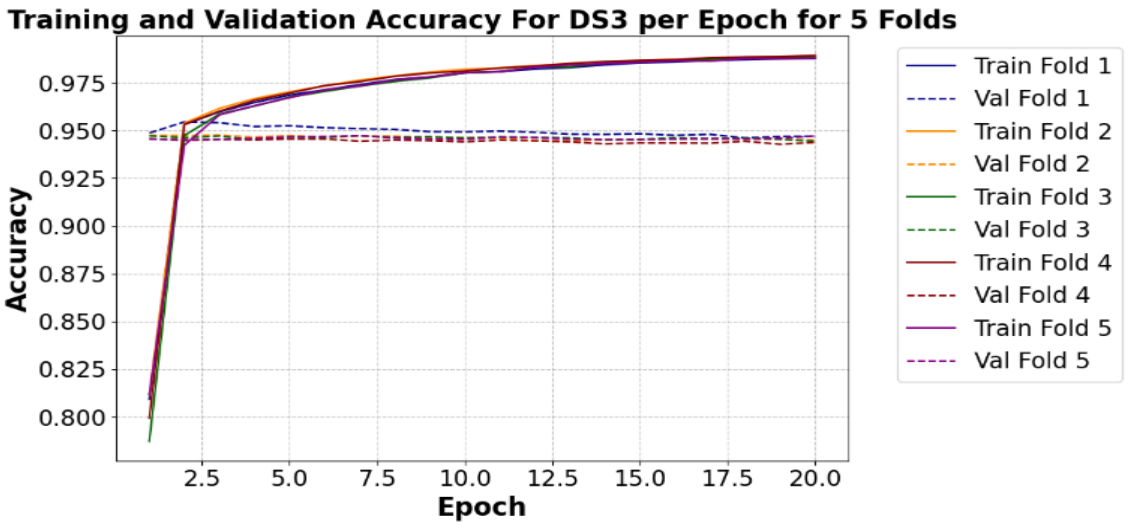


c. Accuracy of trained DS2 using pre-trained Model I

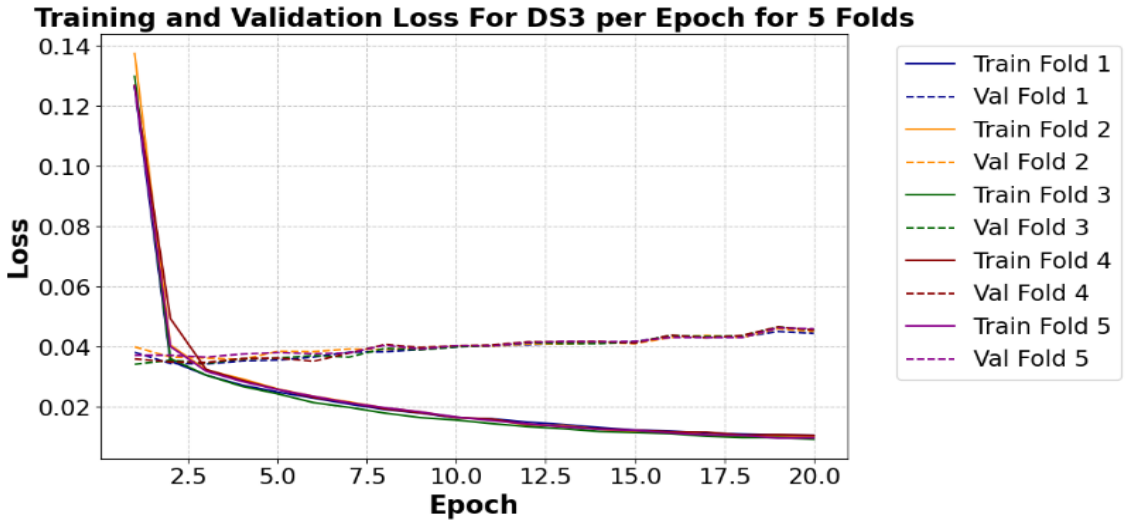


d. Loss of trained DS2 using pre-trained Model I

Figure 4.4. (continued) (a-f) Accuracy and loss during training and validation in five-fold cross-validation. Each row represents our model, tested on pre-trained Model I (Balanced CNN-BiLSTM) with the corresponding dataset (DS1, DS2, DS3) respectively in each left-column subplot (train accuracy vs. validation accuracy) and each right column (train loss vs. validation loss)



e. Accuracy of trained DS3 using pre-trained Model I

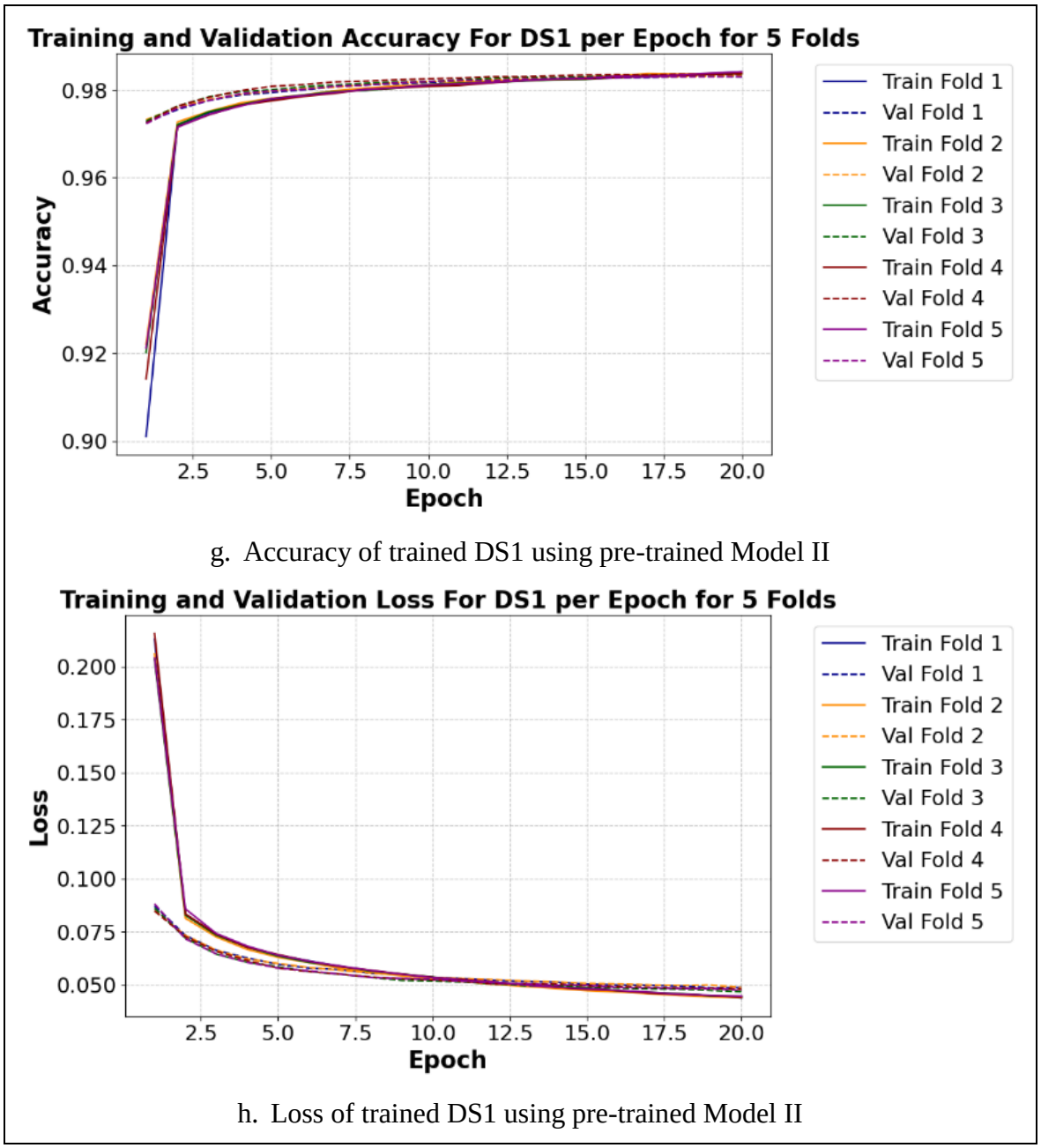


f. Loss of trained DS3 using pre-trained Model I

Figure 4.4. (continued) (a-f) Accuracy and loss during training and validation in five-fold cross-validation. Each row represents our model, tested on pre-trained Model I (Balanced CNN-BiLSTM) with the corresponding dataset (DS1, DS2, DS3) respectively in each left-column subplot (train accuracy vs. validation accuracy) and each right column (train loss vs. validation loss)

The training and validation accuracy for DS1 utilising pre-trained Model II is shown in Figure 4.5. (g), where high accuracy with minimal (0.003) variance is achieved. DS1's training and validation loss utilising pre-trained Model II is displayed in Figure 4.5. (h), where convergence took place without any indications of overfitting. Figure 4.5. (i) demonstrates the training and validation accuracy for DS2 using pre-trained Model II, reflecting consistent and stable performance across folds. Confirming the robustness of the model, Figure 4.5. (j) shows the training and validation loss for DS2 utilising pre-trained

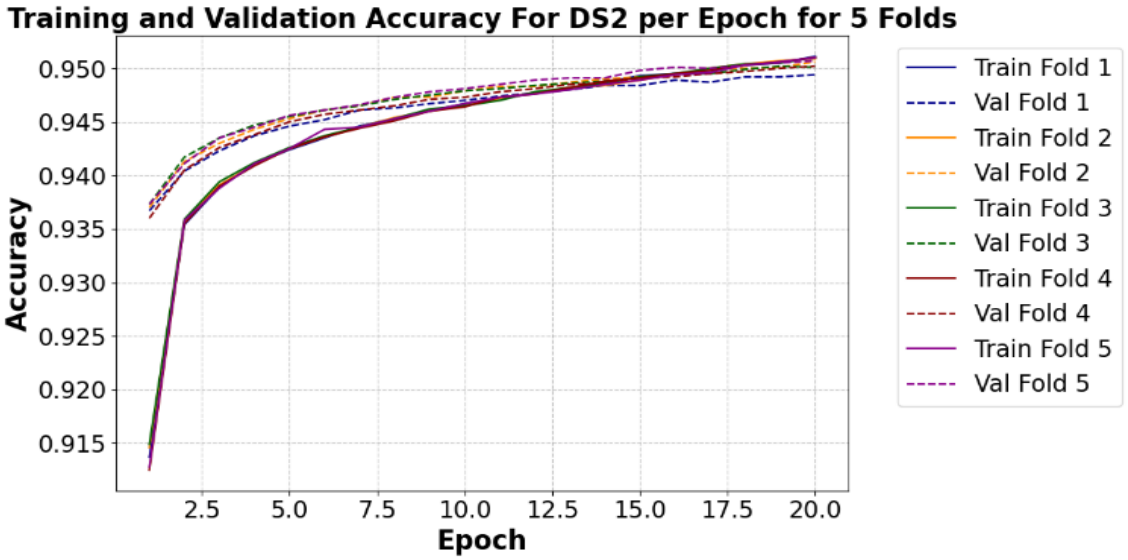
Model II. Accuracy with minimal variation is maintained when training and validating DS3 using pre-trained Model II, as shown in Figure 4.5. (k). Figure 4.5. (l) highlights the training and validation loss for DS3 using pre-trained Model II, which showed similar convergence patterns to the DS2 datasets.



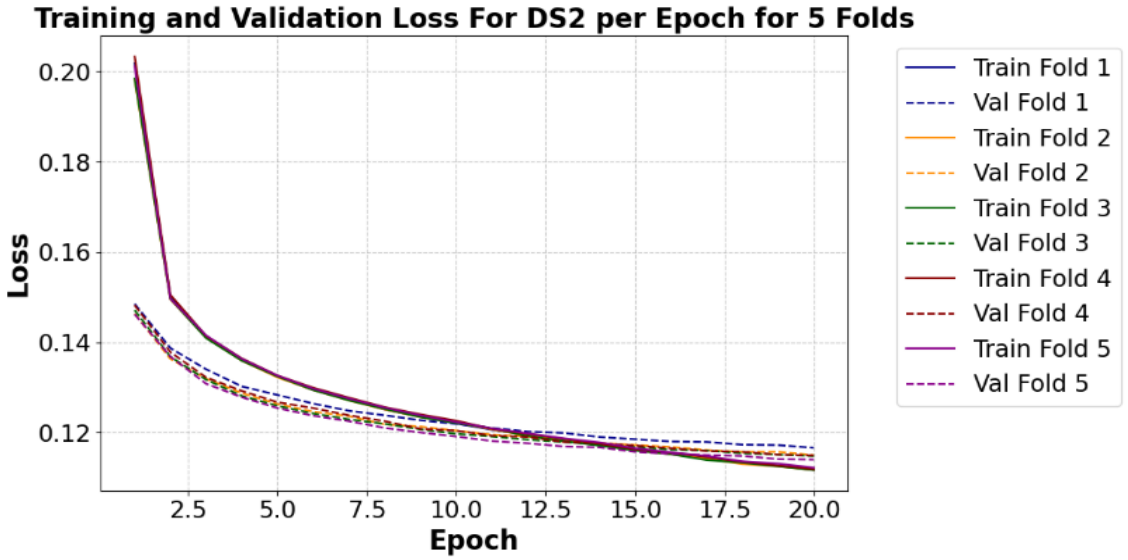
g. Accuracy of trained DS1 using pre-trained Model II

h. Loss of trained DS1 using pre-trained Model II

Figure 4.5. (g-1) Accuracy and loss during training and validation in five-fold cross-validation. Each row represents our model, tested on pre-trained Model II (Roberta-base) with the corresponding dataset (DS1, DS2, DS3) respectively in each left-column subplot (train accuracy vs. validation accuracy) and each right column (train loss vs. validation loss)

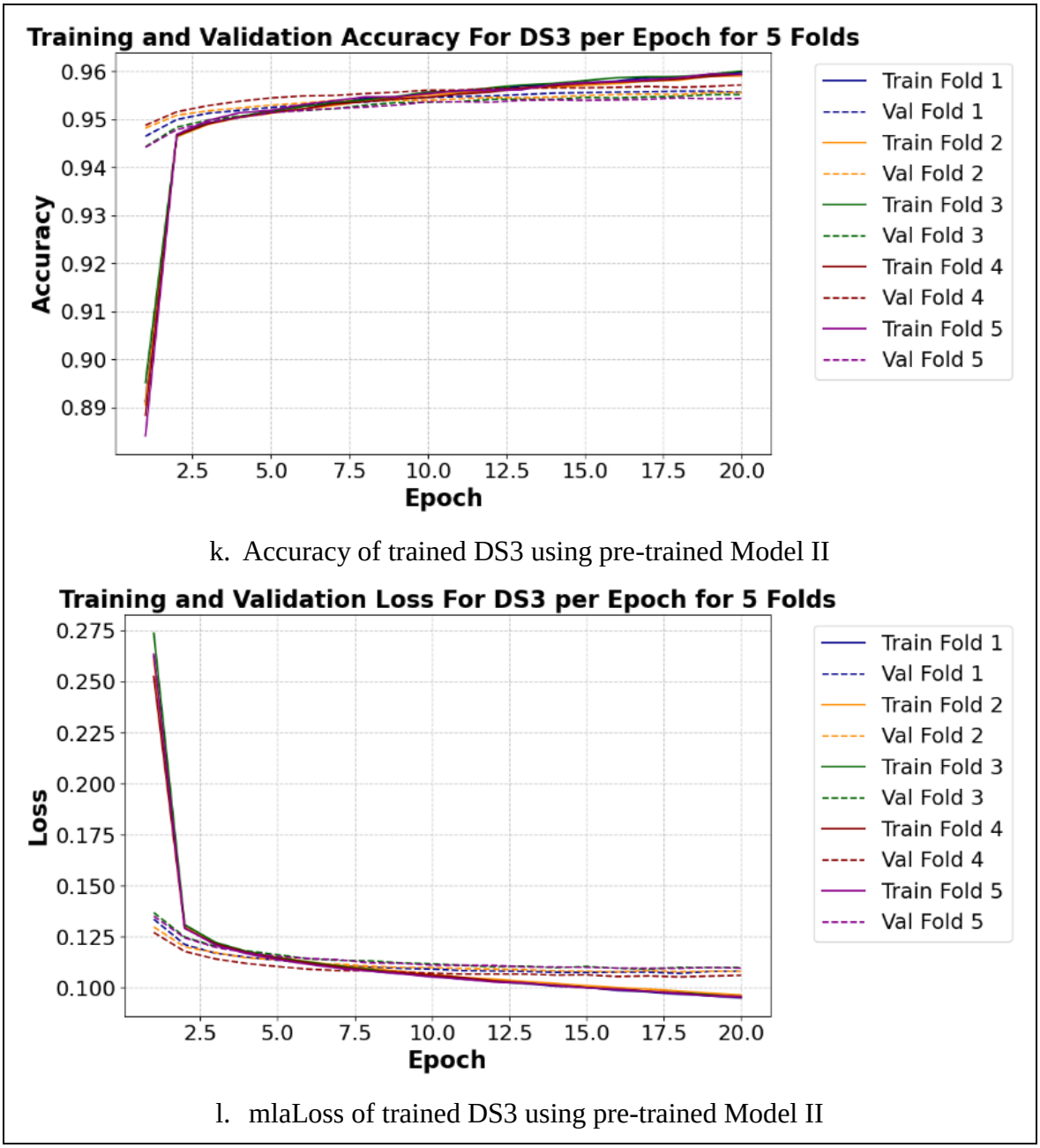


i. Accuracy of trained DS2 using pre-trained Model II



j. Loss of trained DS2 using pre-trained Model II

Figure 4.5. (continued) g-1) Accuracy and loss during training and validation in five-fold cross-validation. Each row represents our model, tested on pre-trained Model II (Roberta-base) with the corresponding dataset (DS1, DS2, DS3) respectively in each left-column subplot (train accuracy vs. validation accuracy) and each right column (train loss vs. validation loss)



k. Accuracy of trained DS3 using pre-trained Model II

l. mlaLoss of trained DS3 using pre-trained Model II

Figure 4.5. (continued) (g-1) Accuracy and loss during training and validation in five-fold cross-validation. Each row represents our model, tested on pre-trained Model II (Roberta-base) with the corresponding dataset (DS1, DS2, DS3) respectively in each left-column subplot (train accuracy vs. validation accuracy) and each right column (train loss vs. validation loss)

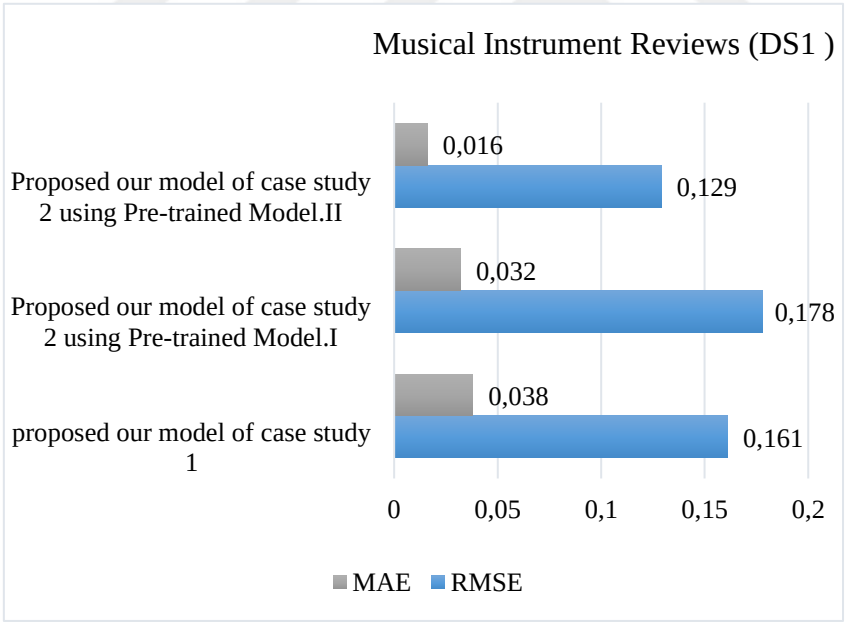
The performance results are shown in Table 4.8. For all three datasets, deep learning models with review analysis perform better regarding RMSE and MAE than deep learning models with the Proposed model using the Pre-trained Model. I proposed a model using a Pretrained Model. II. The percentage improvements are presented in Bold style and delineated in Fig. 4.6. (a-c). For instance, in Table 4.8. and Figure 4.6. (a-c), respectively, the Mean Absolute

Error (MAE) and Root Mean Square Error (RMSE) for three models are evaluated on the Musical Instrument Reviews dataset (DS1). Among the models, the proposed model utilises a Pre-trained Model.II demonstrates the best performance, achieving the lowest error rates with an MAE of 0.016 and an RMSE of 0.129. In comparison, the proposed model employs a Pre-trained Model. I exhibit slightly higher error values, recording an MAE of 0.032 and an RMSE of 0.178. Our proposed BHRQUT model in case study 1, while showing moderate performance, reports an MAE of 0.038 and an RMSE of 0.161. These results highlight the superior effectiveness of the proposed model compared to the pre-trained model.II in minimising predictive errors, making it the most reliable approach among the three. On analysing MAE and RMSE for three models evaluated on the Clothing, Shoes, and Jewellery Reviews dataset (DS2). Among the models, the proposed model utilises a Pre-trained Model.II demonstrates the best performance, achieving the lowest error rates with an MAE of 0.049 and an RMSE of 0.223. The proposed model employs a Pre-trained Model. It shows slightly higher error values, recording an MAE of 0.069 and an RMSE of 0.262. The BHRQUT model, while performing moderately well, reports an MAE of 0.091 and an RMSE of 0.225. These results emphasise the superior accuracy of the proposed model using the Pre-trained Model II, which consistently outperforms the other approaches in minimising predictive errors.

The MAE and RMSE for four models tested on the Yelp dataset (DS3), comparing the performance of two proposed models utilising pre-trained architectures against two baseline methods. Among the models, the proposed model utilises a Pre-trained Model.II performs best, achieving the lowest error rates with an MAE of 0.044 and an RMSE of 0.210. The proposed model employs a Pre-trained Model. I follow with slightly higher error rates, reporting an MAE of 0.052 and an RMSE of 0.228. In contrast, the baseline models exhibit significantly higher error rates. The deep learning-based textual review model [26]records an MAE of 0.755 and an RMSE of 1.025, while the CF(NRCF) + CF(DCCF) model [103]performs the worst, with an MAE of 0.909 and an RMSE of 1.220. These findings highlight the effectiveness of the proposed models, particularly the Pre-trained Model.II, achieving superior accuracy and minimising predictive errors compared to traditional and baseline methods for the Yelp dataset.

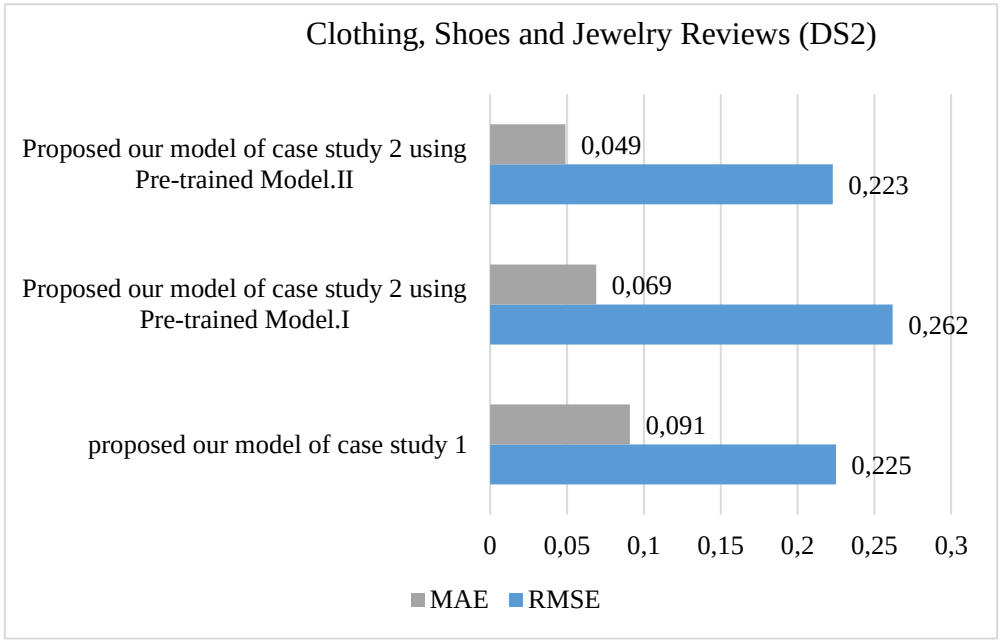
Table 4.8. Comprehensive analysis of model performance

Study	Dataset	RMSE	MAE	Accuracy	Precision	Recall	F1score
Differentiated recommendation framework [23]	DS2			0.8125			
CF(NRCF)+ CF(DCCF) [103]	DS3	1.22	0.909				
Deep learning-based textual review [26]	DS3	1.025	0.755				
Sentiment analysis system [104]	DS3			0.73	0.88	0.73	
Proposed BHRQUT in case study 1	DS1	0.161	0.038	0.97	0.962	0.968	0.97
	DS2	0.225	0.091	0.93	0.92	0.94	0.93
Proposed model in case study 2 using a Pre-trained Model. I	DS1	0.178	**0.032**	0.97	0.962	**0.97**	0.97
	DS2	0.262	**0.069**	0.93	0.92	**0.95**	0.93
	DS3	0.228	**0.052**	0.945	0.946	**0.944**	0.95
Proposed model in case study 2 using a Pre-trained Model.II	DS1	**0.129**	**0.016**	**0.983**	**0.978**	**0.988**	**0.983**
	DS2	**0.223**	**0.049**	**0.95**	**0.945**	**0.955**	**0.95**
	DS3	**0.210**	**0.044**	**0.956**	**0.948**	**0.963**	**0.956**

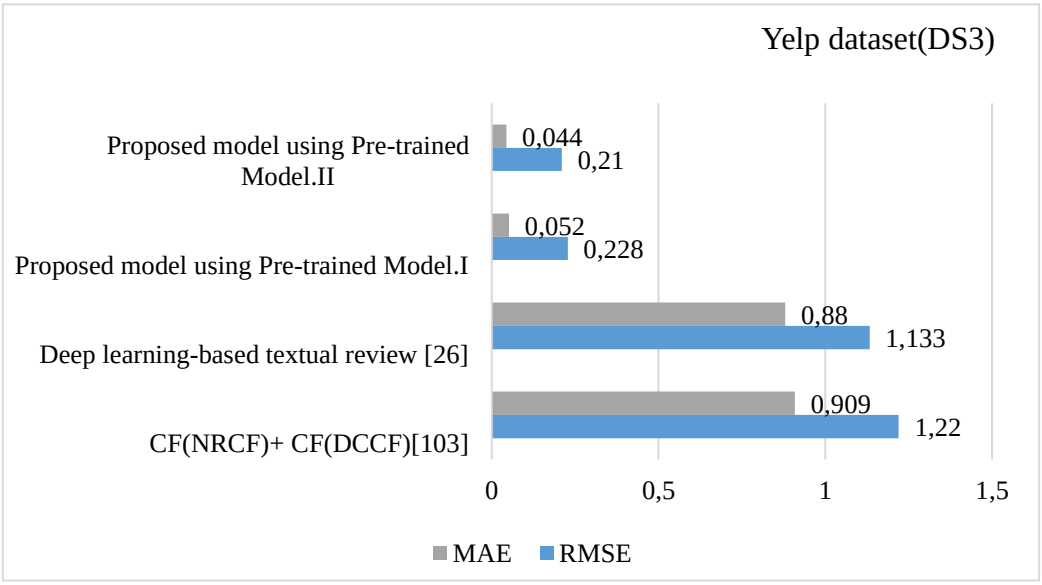


a. Comparison of error metrics for Musical Instrument Reviews (DS1)

Figure 4.6. (a-c) Comparative analysis of error metrics RMSE and MAE across three datasets DS1, DS2, and DS3



b. Comparison of error metrics for Clothing, Shoes, and Jewelry Reviews dataset(DS2)

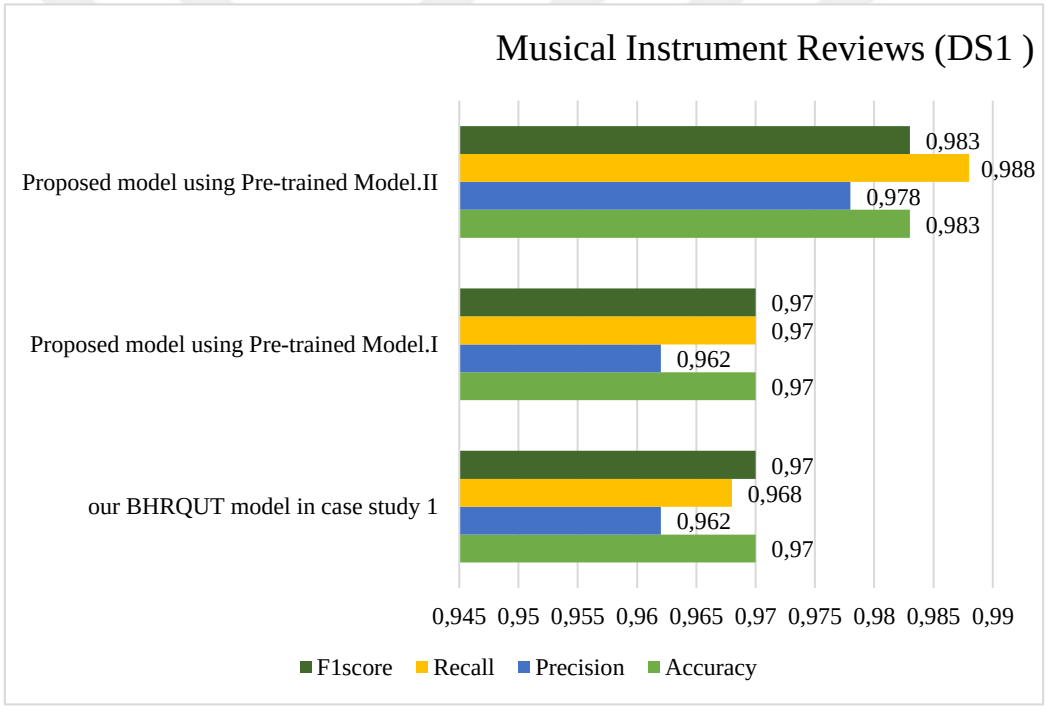


c. Comparison of error metrics for Yelp dataset (DS3)

Figure 4.6. (continued) (a-c) Comparative analysis of error metrics RMSE and MAE across three datasets DS1, DS2, and DS3

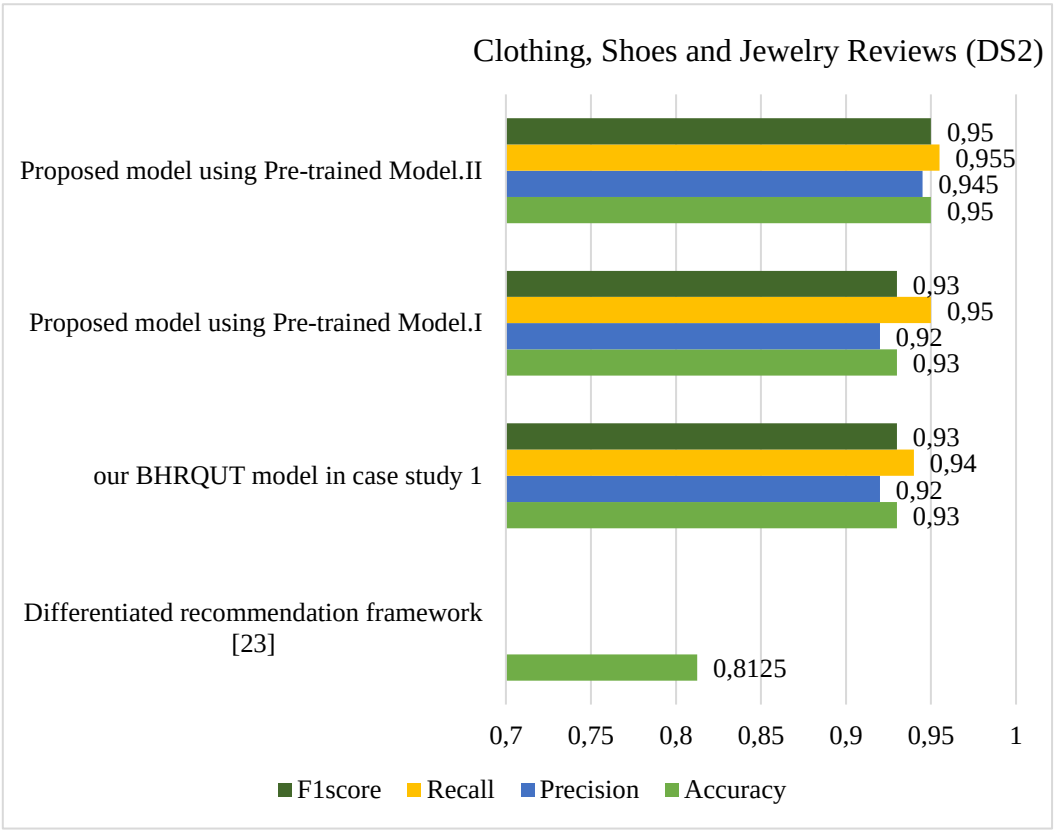
As shown in (See Table 4.8) and Fig. 4.7 (a-c), in three subfigures, the proposed model uses a Pre-trained Model.II outperformed other models (Accuracy, Precision, Recall, and F1-score equal to 98.3%, 97.8%, 98.8%, and 98.3% respectively on Musical Instrument Reviews (DS1); the Proposed model using a Pre-trained Model. I outperformed our BHRQUT with Recall (97%, 95%), whereas our BHRQUT model performed somewhat with Accuracy, Precision, and F1-score equal to the Pre-trained Model. I metrics values on

Musical Instrument Reviews (DS1) and Clothing, Shoes, and Jewelry Reviews dataset (DS2). The accuracy, Precision, and Recall of the Sentiment analysis system [104] were 0.73, 0.88, and 0.73, smaller compared to 0.956, 0.948, and 0.963 for the Proposed model using the Pre-trained Model.II in the case of the Yelp dataset (DS3). The accuracy of the Differentiated recommendation framework [23] is 0.8125 smaller than 0.93 for the Proposed model using the Pre-trained Model.I and for the Proposed model using a Pre-trained Model.II to 0.95 in the case of the Clothing, Shoes, and Jewelry Reviews dataset (DS2). dataset. When comparing the performances of the five models on accuracy, precision, recall, and F1-score, the proposed model uses a pre-trained model.II > Proposed model using Pre-trained Model.I > BHRQUT > Differentiated recommendation framework > Sentiment analysis system.

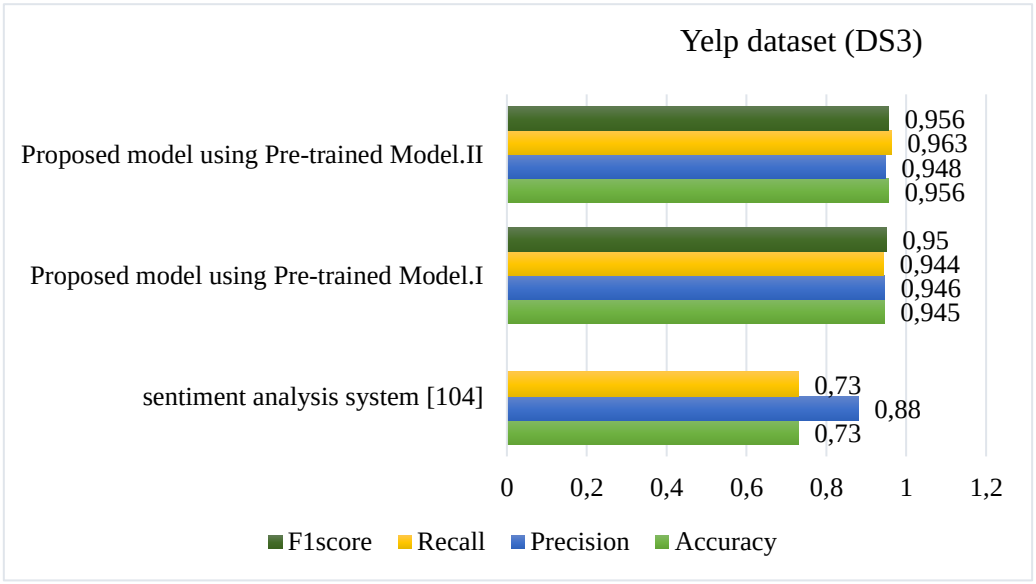


(a) Evaluation of model performance on DS1 dataset using Accuracy, Precision, Recall, and F1-Score

Figure 4.7. (a-c) Performance evaluation of the model on three datasets DS1, DS2, and DS3 using Accuracy, Precision, Recall, and F1-Score metrics



(b) Evaluation of model performance on DS2 dataset using Accuracy, Precision, Recall, and F1-Score



(c) Evaluation of model performance on DS3 dataset using Accuracy, Precision, Recall, and F1-Score metrics

Figure 4.7. (continued) (a-c) Performance evaluation of the model on three datasets DS1, DS2, and DS3 using Accuracy, Precision, Recall, and F1-Score metrics.

5. DISCUSSION AND CONCLUSION

In this thesis, we propose a new method for classifying user reviews. This method is important in e-commerce recommendation systems, where it is used to classify user opinions based on their usefulness or unhelpfulness when making decisions about purchasing goods online**.** The method strategy is based on a model from deep learning. In this research, two novel models, BHRQUT in case study 1 and PRE-DHRRF in case study 2, were proposed to enhance the analysis of user-generated reviews and support purchasing decisions within e-commerce platforms. Each model adopts a distinct deep learning strategy to classify the helpfulness of product reviews and tackle the challenges associated with review sparsity, imbalance, and contextual understanding. This section offers a comparative discussion of the key findings, performance insights, and practical implications derived from the implementation and evaluation of both models.

5.1. Analysis of the BHRQUT Model

The BHRQUT model incorporates a balanced deep learning approach to mitigate issues related to skewed review distributions. The model emphasizes the integration of user preferences and rating variations to more effectively classify the helpfulness of reviews. Utilizing a hybrid CNN–BiLSTM architecture, the system learns semantic patterns from textual reviews while simultaneously considering user bias in rating behavior. This dual-path mechanism enables BHRQUT to generate more accurate and personalized predictions for review helpfulness, supporting users in making informed purchasing decisions.

Before feeding the review text sequences into the deep learning model, the dataset undergoes a quantification process using a tendency-based classification method. This approach predicts user ratings by accounting for diverse opinion patterns, thereby enhancing the representational quality of the "helpful" feature and improving model accuracy. To address class imbalance between helpful and unhelpful reviews, a random oversampling technique, specifically the (SbURA) algorithm, is applied as a form of data augmentation. In parallel, natural language processing (NLP) tools are employed to tokenize and analyze the review summaries. For consistency in input representation, the maximum sentence length is fixed across all samples in the dataset, ensuring that each review summary is truncated or padded to a predefined word limit. To ensure robustness and prevent overfitting to specific subsets,

a 5-fold cross-validation strategy is adopted. In each fold, 80% of the data is used for training, while the remaining 20% is reserved for testing and validation. This approach allows for a comprehensive evaluation of the model's generalisation capabilities. The model is trained in over 20 epochs in each fold, offering a balanced and efficient framework for learning and performance assessment across the datasets.

This setup ensured that the model had sufficient training iterations to learn meaningful patterns while minimizing the risk of overfitting and providing a reliable evaluation of performance across different data partitions. The tokenized words are then converted into dense vector representations of fixed dimensions through the embedding layer. In this layer, semantically similar words are mapped to vectors that are close in the embedding space, facilitating the model's ability to capture nuanced relationships between terms. This process reduces the dimensionality of the input data and enhances computational efficiency, allowing the model to focus on the most salient features for improved learning and generalisation.

Subsequently, a Convolutional Neural Network (CNN) is employed to assign varying weights to different features within the user profile, thereby enabling the model to emphasise more informative attributes. This is followed by a Bidirectional Long Short-Term Memory (BiLSTM) attention mechanism, which integrates forward and backwards LSTM networks to capture contextual information from past and future word sequences. Using bidirectional LSTMs enhances the model's ability to access a more comprehensive context, improving its capacity to store and interpret data effectively. This approach significantly contributes to advancing the performance of valuable review analysis models within the domain of user review evaluation for product recommendation.

The proposed BHRQUT model was applied to four Amazon review category datasets, and its performance was evaluated based on key metrics, including RMSE, MSE, MAE, Accuracy, Recall, Precision, and F1-score. The results demonstrate that our model outperforms the baseline models across all these metrics (See Table 4.3.) and (See Table 4.5.) In four distinct cases, the BHRQUT model demonstrated superior performance, with RMSE improvements of up to 76.84% compared to the RHRM [9], machine-generated reviews [78], and DUPIA [101] models. This includes significant performance gains in the Video Games (C1), Musical Instruments (C2), and Books-5 (C3) categories. Additionally,

the BHRQUT model exhibited the best performance in terms of MSE and MAE, achieving improvements of 98.35% and 95.26%, respectively, compared to the RHRM [9], AFRAM [63], and Deep Feature Extractor [102] models. These improvements were observed across several categories, including Video Games (C1), Musical Instruments (C2), Books-5 (C3), and Clothing, Shoes, and Jewelry (C4) reviews.

We selected the maximum input sentence length for the summary review text feature in the datasets by fixing the number of words in all sentences. The proposed strategy proved to be more effective for most product categories (See Tables 4.3) and (See Table 4.5), which illustrate the impact of maximum input text length on model performance. Our findings indicate that the model performs optimally when the input text length is set to 21 tokens. Upon comparing the effects of varying sentence lengths across all model configurations, we observed that reducing the sentence length consistently led to improved accuracy.

Moreover, our proposed BHRQUT model demonstrated greater efficiency with a maximum text length of 24 tokens for the book category, outperforming the RHRM [9] model (with a max text length of 2817 tokens) in terms of Precision, Recall, F1-Score, and Accuracy by 9.19%, 6.39%, 7.92%, and 9.12%, respectively. Additionally, the BHRQUT model also outperformed the SLCABG [84] model (with a max text length of 648 tokens) in the same metrics, with improvements of 0.54%, 0.64%, 0.75%, and 0.64%, respectively. These results suggest that shorter sequence lengths contribute to enhanced model performance by reducing noise, improving accuracy, preventing overfitting, and increasing computational efficiency. Furthermore, this approach ensures that the model concentrates on the most relevant data, ultimately leading to better generalization and improved performance in real-world applications.

In this study, we employed both small and large datasets and evaluated the performance of our model against other helpfulness analysis models for each dataset size. The results demonstrate that our model outperforms others in predicting helpful reviews across all four datasets, particularly in case (1_1). The experimental findings reveal that the classification performance of the deep learning model (Balanced CNN-BiLSTM) in case (1_1) is significantly superior to all other configurations of our proposed model. However, the performance of case (1_2) and case (2_2) across all datasets was lower than that of case (1_1) and case (2_1).

Additionally, we observed that the Musical Instruments (C2) dataset exhibited the best performance in both cases (1_1) and case (2_1) for all performance metrics. This indicates that the ratio based on (avg ≥ 3) or (ACR ≥ 3) outperforms the ratio based on (avg ≥ 4) or (ACR ≥ 4). The experimental results further suggest that the proposed model is more effective in predicting reviews that assist in making purchasing decisions when compared to other competing methodologies. Specifically, the helpfulness ratio, dependent on rating (star), shows that the SLCABG model performs better than the RHRM model, which predicts helpfulness based on vote values. This conclusion holds across key evaluation metrics, including Accuracy, Precision, Recall, and F1-score (See Table 4.5). Moreover, the classification performance of the BHRQUT model, enhanced by our comprehensive TCF, DA, and Balanced CNN-BiLSTM approach, achieved an impressive 97% performance, outperforming commonly used deep learning models.

Furthermore, we investigated the impact of the TCF and DA approaches on the model's performance under the condition that (avg ≥ 3), as discussed in Section 5.1. The experimental results from the confusion matrix analysis showed that the proposed model performed better when classifying helpful reviews. Additionally, it was observed that users' tendencies are influenced more by subjective factors, such as the quantity of helpfulness ratings in reviews, rather than by the reviews themselves. This finding highlights the effectiveness of the balanced CNN-BiLSTM model in categorizing review helpfulness data, demonstrating its capacity to account for these nuanced user preferences. Next, we conducted experiments to evaluate the performance of our proposed model, which demonstrated significant improvements in classification. The model achieved impressive results, with 97% accuracy, 0.161 RMSE, 0.026 MSE, and 0.038 MAE across all cases of our proposed approach. These results surpassed those of the models in [9] and [84], which yielded 94% accuracy. Additionally, our model outperformed the models in [9], [63], [78], [101], and [102] in terms of RMSE, MSE, and MAE, with values of 0.161, 0.026, and 0.038, respectively. The experimental results highlight the significant role of user preference assessments in influencing purchasing behavior, particularly when integrated with informative review data. This finding reinforces the practical relevance of the proposed recommendation framework for real-world applications, including platforms such as Amazon, Alibaba, and Netflix. Additionally, the study provides insights into both the theoretical underpinnings and practical implications of the proposed models.

5.2. Analysis of the PRE-DHRRF Model

Building upon the insights from BHRQUT, the second model, PRE-DHRRF, extends the analytical capabilities by integrating pre-trained deep learning models and auxiliary feature processing. In this model, review text sequences are input alongside a quantified helpful feature, generated using the Tendency-based Collaborative Filtering (TCF) algorithm. TCF predicts user ratings based on varying opinion patterns to improve the representation of review helpfulness and alleviate information scarcity.

Binary classification is achieved by calculating the average rating from actual and predicted values, where reviews with an average rating ≥ 3 or above are labelled as helpful, and those below as unhelpful. The SbURA data augmentation technique addresses class imbalance, and NLP preprocessing tools are used to tokenise and format the input sequences for training. Two pre-trained models, Balanced CNN–BiLSTM (Model I) and Roberta-base (Model II), were used to evaluate classification performance across three datasets (DS1, DS2, and DS3).

(See Table 4.6) presents the key components of the confusion matrix True Negatives (TN), False Positives (FP), True Positives (TP), and False Negatives (FN) for two pretrained models: BalancedCNN-BiLSTM (Model I) and Roberta-base (Model II), across three datasets (DS1, DS2, DS3). These components directly impact crucial evaluation metrics such as Accuracy and Recall. Specifically, higher True Positives (TP) values increase accuracy, while lower False Negatives (FN) values enhance recall.

As illustrated in (See Table 4.7), (See Figure 4.4 (a-f), and (See Figure 4.5 (g-i)) the performance of Model I (Balanced CNN-BiLSTM) across the three datasets demonstrates strong generalization capability and effective handling of overfitting. For the DS1 dataset, the model achieved a validation accuracy of 97% and a training accuracy of 98%, resulting in a generalization gap of 1% (calculated as $|0.97 - 0.98| = 0.01$). Similarly, for the DS2 dataset, the validation accuracy was 93% compared to a training accuracy of 95%, producing a generalization gap of 2%. The largest gap was observed in the DS3 dataset, where the model achieved 94.5% on validation and 98.8% on training, resulting in a 4.3% generalization gap. This increase is expected given the more complex nature of longer sequences in the DS3 dataset. Nevertheless, the loss values across all datasets remain low (ranging from 0.027 to 0.060), reinforcing the model's overall robustness. The narrow

generalization gaps suggest that the Balanced CNN-BiLSTM model effectively learns from training data without significant overfitting, supporting its suitability for reliable review classification tasks. Moreover, Model II showed strong performance on datasets with longer and more complex review texts, such as DS3, demonstrating its robustness. These findings highlight that both models, especially Model II, maintain a balanced trade-off between learning and generalization, ensuring reliable prediction outcomes across different datasets. Model II (Roberta-base) demonstrates excellent performance across all three datasets with very minimal generalization gaps, indicating strong generalization and minimal overfitting. For the DS1 dataset, the model achieved a validation accuracy of 98.3% and a training accuracy of 98.6%, resulting in an exceptionally low generalization gap of 0.3% (calculated as $|0.986 - 0.983| = 0.003$). Similarly, for the DS2 dataset, the validation and training accuracies were 95.0% and 95.4%, respectively, with a gap of only 0.4%. For DS3, the model maintained its robustness with accuracies of 95.6% on validation and 96.2% on training, yielding a generalization gap of 0.6%. Although the loss values are slightly higher compared to Model I (ranging from 0.065 to 0.116), the consistently small generalization gaps reflect the model's ability to learn complex patterns without overfitting. This performance suggests that Roberta-base is highly effective, especially on datasets with longer or more complex text sequences.

The results presented in Table 4.8, Figure 4.6 (a-c), and Figure 4.7 (a-c) comprehensively compare the proposed approach with various existing methods, demonstrating significant performance improvements. Notably, the pre-trained Balanced CNN-BiLSTM model consistently outperforms the Roberta-base model across three datasets (DS1, DS2, and DS3). The proposed model exhibited remarkable enhancements in several key metrics: Accuracy reached 98.3% for DS1, 95% for DS2, and 95.6% for DS3. Precision values were 0.978 (DS1), 0.945 (DS2), and 0.948 (DS3), while Recall scores achieved 0.988 (DS1), 0.955 (DS2), and 0.963 (DS3). The F1-score for the proposed model was 0.983 for DS1, 0.950 for DS2, and 0.956 for DS3, highlighting its superior performance in these domains. These results demonstrate that the proposed model outperforms existing systems such as our BHRQUT model, the Differentiated Recommendation Framework [23], deep learning-based textual review [26], CF(NRCF)+ CF(DCCF) [103], and the Sentiment Analysis System [104] in terms of these critical evaluation metrics. Moreover, the proposed model exhibited lower error rates relative to other methods, with RMSE values of 0.129 (DS1), 0.223 (DS2), and 0.210 (DS3), as well as MAE values of 0.016 (DS1), 0.049 (DS2), and 0.044 (DS3).

Furthermore, the PRE-DHRRF model showed superior efficiency compared to our previous work, particularly for the DS1 and DS2 datasets. This is evident in its improved Recall (97%, 95%) and MAE (0.032, 0.069) metrics compared to the Balanced CNN-BiLSTM model from our prior BHRQUT. Similar performance gains were observed across other metrics, including accuracy, precision, and F1 score. These findings prove that deep learning-based approaches align closely with the statistical metrics required for effective review text classification, particularly when paired with well-prepared datasets.

We conducted the performance of the PRE-DHRRF model, which exploits differences in users' propensity to evaluate target products and employs data augmentation techniques to balance imbalanced data. Furthermore, the input data for the two pre-trained models, trained over 20 epochs with 5-fold cross-validation, are obtained from active text sequences preprocessed using natural language processing approaches. The proposed framework achieves impressive results such as 98.3% average accuracy, 97.8% precision, 98.8% recall, 98.8% recall (duplicated), 98.3% F1 score, 1.29% RMSE, and 0.16% MAE when analysing data from three different review sources.

The findings highlight the substantial potential of the proposed framework to enhance review and rating recommendation systems significantly, positioning it as an invaluable asset for e-commerce platforms. Furthermore, the relatively small generalisation gap and the model's consistently high performance demonstrate the framework's robustness and applicability in real-world contexts. These results indicate that the proposed approach achieves high accuracy and shows exceptional reliability, marking it as a promising solution for many practical applications. A critical implication of these findings lies in the potential for seamless integration into e-commerce platforms. Businesses can prioritise valuable feedback by automating the identification of helpful reviews, facilitating more informed purchasing decisions, and improving user satisfaction. Moreover, this framework could be a useful tool for content moderators, streamlining the review filtering process, reducing manual effort, and increasing operational efficiency. This would optimise resource allocation and enhance the digital platform user experience.

5.3. Comparative Overview of BHRQUT and PRE-DHRRF Models

This section provides a side-by-side comparison of the two proposed deep learning models, BHRQUT and PRE-DHRRF, as Table 5.1., to highlight their architectural differences, input data configurations, and relative advantages in review classification tasks. Both models were developed to analyze user-generated content for improving product recommendation systems, but they adopt distinct strategies in terms of feature integration, model composition, and fine-tuning.

Table 5.1. Comparative Overview of BHRQUT vs. PRE-DHRRF

Aspect	BHRQUT	PRE-DHRRF
Model Architecture	Custom Balanced CNN-BiLSTM	Pre-trained Balanced CNN-BiLSTM + RoBERTa backbone
Data Used	Amazon Reviews data (C1, C2, C3, and C4) with textual reviews (summary), user-id, item-id, and overall ratings features	Amazon Reviews (DS1, DS2) with textual reviews (summary), user-id, item-id, and overall ratings features, and Yelp Dataset (DS3) with user-id, item-id, full review text, and star ratings features
Inputs	Embedding tokenized sequences + helpful label (0/1)	Embedding tokenized sequences + helpful label (0/1) in pre-trained Balanced CNN-BiLSTM, and Positional encoding tokens+helpful label (0/1) with using RoBERTa model
Network Structure	Custom CNN-BiLSTM + Tuning Layer	All layers of the pre-trained custom CNN-BiLSTM model and the pre-trained RoBERTa-base model, including the feed-forward network (FFN) and final classification layer, were fine-tuned to optimize performance for the target classification task
Max Input Lengths	Tested on 21, 22, 23, 24, 25, 30, 38, 44 token sequence lengths	Tested on 21, 24, 690 token sequence lengths
Performance	Competitive accuracy with lower computational cost	Higher accuracy and better generalization, especially for small and long reviews
Overfitting Handling	Balanced training data + dropout (0.5)	Balanced training data, + pre-trained knowledge + lower dropout rate (0.1)
Advantages	-Lightweight - Faster training - Customised to task - Success the model with a small review length	-Superior generalization - Stronger semantic understanding - Scalable -Success the model with a small and long review length
Innovations	Combining the TCF algorithm, data augmentation, and NLP techniques with CNN for feature locality and BiLSTM for sequential modeling	Combining the TCF algorithm, data augmentation, and leveraging transfer learning from a large NLP corpus

5.4. Implications and Practical Significance

The empirical results underscore the importance of deep learning-based review analysis frameworks in real-world applications. Both BHRQUT and PRE-DHRRF significantly contribute to personalised recommendation systems by emphasising informative reviews and accurately representing user preferences. The small generalisation gap and high classification performance indicate that the proposed models are not only accurate but also robust and generalizable across datasets. From a practical standpoint, these models can be vital in e-commerce environments. By automatically identifying beneficial reviews, platforms can enhance user trust, lessen cognitive load during decision-making, and boost overall user satisfaction. Furthermore, these models can aid content moderation efforts by filtering and prioritizing valuable feedback, thereby streamlining manual review processes.

6. LIMITATION AND FUTURE WORK

While the proposed models BHRQUT and PRE-DHRRF demonstrate strong performance in analyzing review helpfulness, several technical and practical limitations must be acknowledged.

Technical limitations

- Overfitting on training data

Despite implementing regularization techniques such as dropout and cross-validation, signs of overfitting were observed, particularly in the Balanced CNN-BiLSTM (BHRQUT) model. This was evident from discrepancies between training and validation performance in certain folds, suggesting that the model may have memorized training patterns without generalizing well to unseen data.

- Input length constraints

Both models were constrained by fixed input lengths. Long reviews were truncated, potentially discarding critical contextual information.

- Data imbalance and bias

Oversampling of minority classes may result in over-representation of specific patterns, reducing the model's ability to generalize across diverse review structures. The datasets exhibited skewed distributions in helpful vs. unhelpful labels. Although oversampling techniques were applied, residual bias may persist, affecting the robustness of predictions across underrepresented categories this was especially pronounced in BHRQUT.

- Computational complexity

The PRE-DHRRF model, leveraging RoBERTa, required substantial computational resources due to its large parameter count and self-attention mechanisms. Fine-tuning all layers further increased training time and resource demands.

- Interpretability

Deep learning models, especially transformer-based architectures such as RoBERTa, are often described as "black boxes" because their internal decision-making processes are not easily understandable to humans. These models learn complex patterns from data using millions of parameters, but they do not provide transparent reasoning for their outputs. In real-world applications, particularly in domains like e-commerce or decision support systems, stakeholders (e.g., consumers, business managers, or regulators) may require explanations for why a review was classified as helpful or unhelpful. The inability to trace the model's decision back to interpretable features reduces user trust and makes it difficult to validate or audit the model's behavior.

Practical limitations

- Real-world deployment challenges

Deploying these models in real-time recommendation systems poses challenges such as latency, especially with the PRE-DHRRF model due to its high inference time. Lightweight or distilled models may be needed for practical applications.

- Domain adaptation

While the models performed well on Amazon and Yelp datasets, their effectiveness in other domains or languages has not been validated. Domain-specific retraining or adaptation may be necessary for broader applicability.

- Language drift and model degradation

The language used in customer reviews is dynamic and evolves over time. Users frequently adopt new slang, abbreviations, emojis, or product-specific terminology that may not have been present in the data used during the initial training phase. As a result, pre-trained models may struggle to accurately interpret or classify newer reviews containing such evolving language patterns. This phenomenon, often referred to as language drift, can lead to a decline in model performance over time. To maintain accuracy and relevance, it is essential to

implement continual learning strategies or conduct periodic retraining using updated datasets that reflect current linguistic trends.

The classification approach primarily relies on rating-based indicators to distinguish between helpful and unhelpful reviews. This reliance may be inadequate in domains requiring more sophisticated recommendation criteria or contextual understanding.

The model's effectiveness can be constrained by datasets with lengthy review texts. The large maximum sequence length may hinder the model’s ability to capture finer details, and the associated preprocessing steps demand substantial memory resources, potentially exceeding available hardware capacities. Moreover, the feature analysis is narrowly focused on customer preferences, neglecting other influential aspects such as product imagery, availability, or manufacturing dates, which could further enrich the understanding of review helpfulness.

Additionally, the tendency-based collaborative filtering technique, while effective, may encounter scalability issues in large datasets due to the computational overhead required to model user-item interactions. Deep learning methods, in general, also require high-performance hardware, such as GPUs, and large datasets to achieve optimal results. Unlike traditional machine learning algorithms, deep learning approaches solve problems holistically rather than modularly, making them more resource-intensive.

Lastly, the BHRQUT model, which combines convolutional layers with bidirectional LSTMs, exhibited reduced performance when processing reviews of excessive length. Such lengthy inputs were subject to truncation due to fixed input length constraints, leading to the loss of potentially valuable contextual information. Moreover, the PRE-DHRRF model, based on RoBERTa, performs well with longer sequences, particularly in certain categories within the Yelp dataset. Tokenization-based truncation still affected reviews exceeding the model’s maximum token limit (typically 512 tokens), excluding potentially relevant information. Although NLTK provides a robust foundation for natural language processing tasks, its performance may be suboptimal for large-scale deployments. It lacks native support for high-performance deep learning applications, requiring integration with frameworks like TensorFlow or PyTorch to handle complex modeling tasks effectively.

To address these limitations, several improvements can be proposed. For BHRQUT, incorporating attention mechanisms or hierarchical LSTM architectures may improve performance on varied-length reviews. Additionally, dynamically adjusting the input size or segmenting long reviews could preserve more contextual data

Future research should address these limitations by exploring hybrid classification approaches that incorporate ratings and voting signals under unified conditions to improve helpfulness detection accuracy. Integrating additional contextual features such as visual product attributes, product availability, and manufacturing data may also enhance the model's capability to assess the full scope of review usefulness.

Furthermore, future models should consider optimizing tendency-based collaborative filtering techniques or replacing them with more computationally efficient alternatives to overcome scalability challenges. Enhancing the memory efficiency of preprocessing and model training stages would also make these methods more accessible on standard hardware.

Investigating the temporal dynamics of review helpfulness is another promising direction. Reviews written earlier tend to receive higher engagement, which may introduce a sequential bias. Future studies should explicitly model review timestamps and evaluate their impact on helpfulness perception. Moreover, exploring the combination of NLTK with scalable deep learning libraries could bridge the gap between linguistic analysis and high-performance computation, facilitating more practical deployment in production environments.

Ultimately, future work should conduct comparative assessments of various deep learning algorithms to determine their suitability across different e-commerce contexts and refine the proposed models for broader applicability.

REFERENCES

1. Papazoglou, M. P., Tsalgatidou, A. (2000). Business to business electronic commerce issues and solutions. *Decision Support Systems*, *29*(4), 301-304.

2. Sheth, J. (2020). Impact of Covid-19 on consumer behavior: Will the old habits return or die?. *Journal of business research*, *117*, 280-283.

3. Hussien, F. T. A., Rahma, A. M. S. and Wahab, H. B. A. (2021, May). Recommendation systems for e-commerce systems an overview. In *Journal of Physics: Conference Series, IOP Publishing*, Vol. 1897, No. 1, p. 012024.

4. Ricci, F., Rokach, L. and Shapira, B. (2010). Introduction to recommender systems handbook. in *Recommender systems handbook*, Springer, pp. 1–35.

5. Martin, L. and Pu, P. (2014, June). Prediction of helpful reviews using emotions extraction. In *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, no. 1.

6. Margaris, D., Vassilakis, C., Spiliotopoulos, D. (2020). What makes a review a reliable rating in recommender systems?. *Information Processing & Management*, 57(6), 102304.

7. Xue, K., Wang, J. (2018, March). Collaborative filtering recommendation algorithm for user interest and relationship based on score matrix. In *International Conference on Mathematics, Modelling, Simulation and Algorithms (MMSA 2018)* (pp. 217-221). Atlantis Press.

8. Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, *25*, 99-154.

9. Li, Q., Li, X., Lee, B., & Kim, J. (2021). A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation Service. *Applied Sciences*, *11*(18), 8613.

10. Ahmed, Z., & Wang, J. (2023). A fine-grained deep learning model using embedded-CNN with BiLSTM for exploiting product sentiments. *Alexandria Engineering Journal*, *65*, 731-747.

11. Li, X., Li, Q., & Kim, J. (2023). A review helpfulness modeling mechanism for online e-commerce: Multi-channel CNN end-to-end approach. *Applied Artificial Intelligence*, *37*(1), 2166226.

12. Olatunji, I. E., Li, X., & Lam, W. (2019, November). Context-aware helpfulness prediction for online product reviews. In *Asia information retrieval symposium* (pp. 56-65). Cham: Springer International Publishing.

13. Ge, S., Qi, T., Wu, C., Wu, F., Xie, X., & Huang, Y. (2019). Helpfulness-aware review based neural recommendation. *CCF transactions on pervasive computing and interaction*, *1*, 285-295.

14. Liu, Z., Yuan, B., & Ma, Y. (2022). A multi-task dual attention deep recommendation model using ratings and review helpfulness. *Applied Intelligence*, *52*(5), 5595-5607.

15. Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2022). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(5), 4425-4445.

16. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).

17. Sreepada, R. S., Patra, B. K., Chakrabarty, A., & Chandak, S. (2018, January). Revisiting tendency based collaborative filtering for personalized recommendations. In *Proceedings of the ACM India joint international conference on data science and management of data* (pp. 230-239).

18. Zhang, L. (2023). Enhancing recommender systems via data augmentation.

19. Abulaish, M., & Sah, A. K. (2019, January). A text data augmentation approach for improving the performance of cnn. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)* (pp. 625-630). IEEE.

20. Roy, K., Goel, A., & Goyal, P. (2022, April). Effectiveness of data augmentation to identify relevant reviews for product question answering. In *Companion Proceedings of the Web Conference 2022* (298-301).

21. Abonizio, H. Q., Paraiso, E. C., & Barbon, S. (2021). Toward text data augmentation for sentiment analysis. *IEEE Transactions on Artificial Intelligence*, *3*(5), 657-668.

22. Chen, D., & Yin, C. (2022). Data augmentation for intent classification. *arXiv preprint arXiv:2206.05790*.

23. Yan, C., Chen, Y., & Zhou, L. (2019). Differentiated fashion recommendation using knowledge graph and data augmentation. *Ieee Access*, *7*, 102239-102248.

24. Li, C. R., Sanjay, A., Yang, S. W., & Lin, S. D. (2016, November). Transfer learning for sequential recommendation model. In *2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI)* (pp. 154-161). IEEE.

25. Jafri, S. I. H., Ghazali, R., Javid, I., Mahmood, Z., & Hassan, A. A. A. (2022). Deep transfer learning with multimodal embedding to tackle cold-start and sparsity issues in recommendation system. *Plos one*, 17(8), e0273486.

26. Gheewala, S., Xu, S., Yeom, S., & Maqsood, S. (2024). Exploiting deep transformer models in textual review based recommender systems. *Expert Systems with Applications*, *235*, 121120.

27. Kim, P. (2017). Matlab deep learning. *With machine learning, neural networks and artificial intelligence*, *130*(21), 151.

28. Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017). Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)* (pp. 364-371). IEEE.

29. Mu, R. (2018). A survey of recommender systems based on deep learning. *Ieee Access*, *6*, 69009-69022.

30. Itmazi, J., & Gea, M. (2006, December). The recommendation systems: Types, domains and the ability usage in learning management system. In *Proceedings of the International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan*.

31. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

32. Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, *33*(12), 6999-7019.

33. Savaş, S. (2019). *Karotis arter intima media kalınlığının derin öğrenme ile sınıflandırılması* (Doctoral dissertation, Doktora Tezi, Fen Bilimleri Enstitüsü, Gazi University: Ankara).

34. Zucchet, N., & Orvieto, A. (2024). Recurrent neural networks: vanishing and exploding gradients are not the end of the story. *Advances in Neural Information Processing Systems*, *37*, 139402-139443.

35. Semberecki, P., & Maciejewski, H. (2017, September). Deep learning methods for subject text classification of articles. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 357-360). IEEE.

36. Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the association for computational linguistics*, *4*, 357-370.

37. Rice, L., Wong, E., & Kolter, Z. (2020, November). Overfitting in adversarially robust deep learning. In *International conference on machine learning* (pp. 8093-8104). PMLR.

38. Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, *52*, 1-37.

39. Xu, N., & Hu, C. (2023). Enhancing E-Commerce Recommendation using Pre-Trained Language Model and Fine-Tuning. *arXiv preprint arXiv:2302.04443*.

40. Ni, J., Li, J., & McAuley, J. (2019, November). Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)* (pp. 188-197).

41. Yang, S., Yu, X., & Zhou, Y. (2020, June). Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example. In *2020 International workshop on electronic communication and artificial intelligence (IWECAI)* (pp. 98-101). IEEE.

42. Prasada Rao, P. V. R. D., Varakumari, S., Vineetha, B., & Satish, V. (2018). Application of data mining to e-commerce recommendation systems. *Int J Eng Technol*, *7*(2.32).

43. Smith, B., & Linden, G. (2017). Two decades of recommender systems at Amazon.com. *Ieee internet computing*, *21*(3), 12-18.

44. Bai, T., Wen, J. R., Zhang, J., & Zhao, W. X. (2017, November). A neural collaborative filtering model with interaction-based neighborhood. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1979-1982).

45. He, X., Du, X., Wang, X., Tian, F., Tang, J., & Chua, T. S. (2018). Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912*.

46. Bobadilla, J., Gutiérrez, A., Alonso, S., & González-Prieto, Á. (2024). Neural collaborative filtering classification model to obtain prediction reliabilities. *arXiv preprint arXiv:2410.16838*.

47. Nozari, R. B., Divsalar, M., Abkenar, S. A., Amiri, M. F., & Divsalar, A. (2024). A Novel Behavior-Based Recommendation System for E-commerce. *arXiv preprint arXiv:2403.18536*.

48. Gao, S., Liu, K., & Fu, Y. (2023). Unified Matrix Factorization with Dynamic Multi-view Clustering. *arXiv preprint arXiv:2308.04661*.

49. Bibas, K., Sar Shalom, O., & Jannach, D. (2023, April). Semi-supervised adversarial learning for complementary item recommendation. In *Proceedings of the ACM Web Conference 2023* (pp. 1804-1812).

50. Chen, C., Chang, K. C. C., Li, Q., & Zheng, X. (2018). Semi-supervised learning meets factorization: Learning to recommend with chain graph model. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *12*(6), 1-24.

51. Song, K., Sun, Q., Xu, C., Zheng, K., & Yang, Y. (2023). Self-supervised multi-modal sequential recommendation. *arXiv preprint arXiv:2304.13277*.

52. Xia, L., Huang, C., Huang, C., Lin, K., Yu, T., & Kao, B. (2023, April). Automated self-supervised learning for recommendation. In *Proceedings of the ACM web conference 2023* (pp. 992-1002).

53. TWang, T., Xia, L., & Huang, C. (2023). Denoised self-augmented learning for social recommendation. *arXiv preprint arXiv:2305.12685*.

54. Chen, W., Cai, F., Chen, H., & Rijke, M. D. (2019). Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems (TOIS)*, *37*(4), 1-30.

55. Xiao, Y., & Zhong, R. (2019). A hybrid recommendation algorithm based on weighted stochastic block model. *arXiv preprint arXiv:1905.03192*.

56. Ibrahim, M., Bajwa, I. S., Sarwar, N., Hajjej, F., & Sakr, H. A. (2023). An intelligent hybrid neural collaborative filtering approach for true recommendations. *IEEE Access*, *11*, 64831-64849.

57. Jia, M., Liu, F. A., Li, X., & Zhuang, X. (2023). Hybrid graph neural network recommendation based on Multi-Behavior interaction and time sequence awareness. *Electronics*, *12*(5), 1223.

58. Fang, X. (2021). Making recommendations using transfer learning. *Neural Computing and Applications*, *33*(15), 9663-9676.

59. Lin, Z., Liu, D., Pan, W., & Ming, Z. (2021, September). Transfer learning in collaborative recommendation for bias reduction. In *Proceedings of the 15th ACM Conference on Recommender Systems* (pp. 736-740).

60. Fareed, A., Hassan, S., Belhaouari, S. B., & Halim, Z. (2024). Elevating recommender systems: Cutting-edge transfer learning and embedding solutions. *Applied Soft Computing*, *166*, 112140.

61. Zhao, L., Pan, S. J., & Yang, Q. (2017). A unified framework of active transfer learning for cross-system recommendation. *Artificial Intelligence*, *245*, 38-55.

62. Tan, K. L., Lee, C. P., & Lim, K. M. (2023). A survey of sentiment analysis: Approaches, datasets, and future research. *Applied Sciences*, *13*(7), 4550.

63. Li, W., & Xu, B. (2020). Aspect-based fashion recommendation with attention mechanism. *IEEE Access*, *8*, 141814-141823.

64. Shafqat, W., & Byun, Y. C. (2019). A recommendation mechanism for under-emphasized tourist spots using topic modeling and sentiment analysis. *Sustainability*, *12*(1), 320.

65. Qu, X., Li, X., Farkas, C., & Rose, J. (2021). Review helpfulness evaluation and recommendation based on an attention model of customer expectation. *Information Retrieval Journal*, *24*, 55-83..

66. Hasan, E., Rahman, M., Ding, C., Huang, J. X., & Raza, S. (2024). Based recommender systems: a survey of approaches, challenges and future perspectives. *arXiv preprint arXiv:2405.05562*.

67. AL-Sharuee, M. T., Liu, F., & Pratama, M. (2021). Sentiment analysis: dynamic and temporal clustering of product reviews. *Applied Intelligence*, *51*, 51-70.

68. Xu, H., Liu, H., Jiao, P., & Wang, W. (2021, July). Transformer reasoning network for personalized review summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1452-1461).

69. Xu, H., Liu, H., Lv, Z., Yang, Q., & Wang, W. (2023, July). Pre-trained personalized review summarization with effective salience estimation. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 10743-10754).

70. González, F., Guzmán, G., Torres-Ruiz, M., Sidorov, G., & Mata-Rivera, F. (2021). Analysis of User Generated Content Based on a Recommender System and Augmented Reality. In *Telematics and Computing: 10th International Congress, WITCOM 2021, Virtual Event, November 8–12, 2021, Proceedings 10* (pp. 207-228). Springer International Publishing.

71. Mauro, N., Ardissono, L., & Petrone, G. (2021). User and item-aware estimation of review helpfulness. *Information Processing & Management*, *58*(1), 102434.

72. He, M., Zhang, S., & Meng, Q. (2019). Learning to style-aware Bayesian personalized ranking for visual recommendation. *IEEE Access*, *7*, 14198-14205.

73. He, R., & McAuley, J. (2016, February). VBPR: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).

74. Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

75. Lv, J., Song, B., Guo, J., Du, X., & Guizani, M. (2019). Interest-related item similarity model based on multimodal data for top-N recommendation. *IEEE access*, *7*, 12809-12821.

76. Liu, T., Yin, X., & Ni, W. (2020). Next basket recommendation model based on attribute-aware multi-level attention. *IEEE Access*, *8*, 153872-153880..

77. Vijayakumar, S. Interdisciplinary Areas of Research in Applied Linguistics.

78. Ouyang, S., & Lawlor, A. (2021). Improving explainable recommendations by deep review-based explanations. *IEEE Access*, *9*, 67444-67455.

79. Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., & He, X. (2021, October). UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management* (pp. 1253-1262).

80. JWei, J., & Zou, K. (2019). Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

81. Ćavar, I., Kavran, Z., Jolić, N., Anđelović, N., Cvitić, I., & Gović, M. (2013). Content-based Recommender System for Textual Documents Written in Croatian. In *The Second International Conference on Data Analytics*..

82. Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *ArXiv Prepr. ArXiv190711692*, 2019.

83. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

84. Yang, L., Li, Y., Wang, J., & Sherratt, R. S. (2020). Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning. *IEEE access*, *8*, 23522-23530..

85. Kim, J., Park, J., Shin, M., Lee, J., & Moon, N. (2021). The method for generating recommended candidates through prediction of multi-criteria ratings using CNN-BiLSTM. *Journal of Information Processing Systems*, *17*(4), 707-720..

86. A. S. Alhanaf, M. Farsadi, and H. H. Balik, "Fault Detection and Classification in Ring Power System with DG Penetration Using Hybrid CNN-LSTM," *IEEE Access*, 2024.

87. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

88. Yu, T., & Zhu, H. (2020). Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.

89. Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

90. Jadon, A., Patil, A., & Jadon, S. (2024, January). A comprehensive survey of regression-based loss functions for time series forecasting. In *International Conference on Data Management, Analytics & Innovation* (pp. 117-147). Singapore: Springer Nature Singapore.

91. Buja, A., Stuetzle, W., & Shen, Y. (2005). Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November*, *3*, 13.

92. Liu, P., Zhang, L., & Gulla, J. A. (2023). Pre-train, prompt, and recommendation: A comprehensive survey of language modeling paradigm adaptations in recommender systems. *Transactions of the Association for Computational Linguistics*, *11*, 1553-1571.

93. Barrios, W., & Jin, S. (2024). Multi-layer Learnable Attention Mask for Multimodal Tasks. *arXiv preprint arXiv:2406.02761*.

94. Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83-85.

95. Jiang, Y., Krishnan, D., Mobahi, H., & Bengio, S. (2018). Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*.

96. Niero, B. NVIDIA's AI Revolution: A Comprehensive Strategic Analysis.

97. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265-283)..

98. Vasilev, I. (2023). *PYTHON DEEP LEARNING-: Exploring Deep Learning Techniques and Neural Network... Architectures with Pytorch*. PACKT PUBLISHING LIMITED..

99. JYao, J. (2019, April). Automated sentiment analysis of text data with NLTK. In *Journal of physics: conference series* (Vol. 1187, No. 5, p. 052020). IOP Publishing..

100. Hug, N. (2020). Surprise: A Python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174.

101. Zhang, X., Liu, H., Chen, X., Zhong, J., & Wang, D. (2020). A novel hybrid deep recommendation system to differentiate user's preference and item's attractiveness. *Information Sciences*, *519*, 306-316.

102. Shoja, B. M., & Tabrizi, N. (2019). Customer reviews analysis with deep neural networks for e-commerce recommender systems. *IEEE access*, *7*, 119121-119130..

103. Lee, S., Shin, H., Choi, I., & Kim, J. (2022). Analyzing the impact of components of yelp. com on recommender system performance: case of Austin. *IEEE Access*, *10*, 128066-128076..

104. Hassan, A. K. A., & Abdulwahhab, A. B. A. (2017). Reviews Sentiment analysis for collaborative recommender system. *Kurdistan journal of applied research*, 87-91.



Gazili olmak ayrıcalıktır