T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

RS-DSINET: TAVSİYE SİSTEMLERİ DERECELENDİRME PROBLEMİ İÇİN GELİŞTİRİLMİŞ YENİ BİR DERİN ÖĞRENME MODELİ

Veysel TÜRK

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ŞANLIURFA
2022

İÇİNDEKİLER

Sayfa No

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
ŞEKİLLER DİZİNİ iv
ÇİZELGELER DİZİNİ v
SİMGELER ve KISALTMALAR DİZİNİ vii
1. GİRİŞ 1
2. ÖNCEKİ ÇALIŞMALAR 6
3. MATERYAL ve YÖNTEM 13
3.1. Tavsiye Sistemi 14
3.1.1. Tavsiye sistemi uygulamaları 17
3.1.2. Tavsiye sistemi benzerlik ölçütleri 20
3.1.3. Tavsiye sistemi problemleri 21
3.2. Makine Öğrenmesi 23
3.2.1. Denetimli öğrenme 24
3.2.1.1 K-en yakın komşuluk (K-nearest neighbors) 26
3.2.1.2 Destek vektör makinesi (support vector machine) 27
3.2.1.3 Rastgele orman (random forest) 28
3.2.1.4 Karar ağaçları (decision tree) 29
3.2.1.5 Olasılıksal dereceli azalma (stochastic gradient descent) 30
3.2.2. Denetimsiz öğrenme 30
3.2.2.1 Temel bileşen analizi (principal component analysis) 30
3.2.3. Pekiştirmeli öğrenme 31
3.3. Derin Öğrenme 31
3.3.1. Derin öğrenme yöntemleri 40
3.3.1.1 Basit tekrarlayan sinir ağları (simple recurrent neural network) 41
3.3.1.2 Uzun-kısa süreli bellek (long-short term memory) 42
3.3.1.3 Çift yönlü uzun-kısa süreli bellek (bidirectional long-short term memory) 43
3.3.1.4 Kapılı tekrarlayan hücre (gated recurrent unit) 44
3.3.1.5 Evrişimli sinir ağı (convolutional neural network) 45
3.3.2. Derin öğrenme algoritmalarında yaşanan problemler 47
3.3.3. Önerilen RS-DSINet derin öğrenme algoritması 49
3.3.4. Önerilen derin öğrenme algoritmalarında kullanılan diğer teknikler 61
3.3.5. Derin öğrenme algoritmalarının eğitim ve test süreci 62
3.3.6. Derin öğrenme algoritmalarının performans denetimi 64
3.3.7. Derin öğrenme algoritmalarının eğitimi sürecinde kullanılan hiper-parametreler 64
3.4. Çapraz Doğrulama (cross-validation) 66
3.5. Kullanılan Veri Setleri 67
3.6. Veri Seti Ön İşleme 69
3.7. Değerlendirme Ölçütleri 70
4. ARAŞTIRMA BULGULARI ve TARTIŞMA 72
4.1. MovieLens Veri Seti 72
4.1.1. Derin öğrenme 72
4.1.1.1 5 katlı çapraz doğrulama 79
4.1.1.2 Derin öğrenme parametre seçimi 82
4.1.2. Makine öğrenmesi 84
4.1.3. Geleneksel tavsiye sistemi yöntemleri 85
4.2. Goodbooks-10K Veri Seti 86
4.2.1. Derin öğrenme 86
4.2.1.1 5 katlı çapraz doğrulama 91
4.2.2. Makine öğrenmesi 92
4.3. FilmTV Movies Veri Seti 94
4.3.1. Derin öğrenme 94

4.3.2. Makine öğrenmesi 95
4.4. IMDb Movies Extensive Veri Seti 95
4.4.1. Derin öğrenme 95
4.4.2. Makine öğrenmesi 105
4.5. Literatür Karşılaştırılması....105
5. SONUÇLAR ve ÖNERİLER....108
5.1. Sonuçlar....108
5.2. Öneriler....109
KAYNAKLAR 111

ÖZET

Yüksek Lisans Tezi

RS-DSINET: TAVSİYE SİSTEMLERİ DERECELENDİRME PROBLEMİ İÇİN GELİŞTİRİLMİŞ YENİ BİR DERİN ÖĞRENME MODELİ

Veysel TÜRK

Harran Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. İbrahim Berkan AYDİLEK
YIL: 2022, Sayfa: 118

Web teknolojilerinde yaşanan gelişmeler, çevrimiçi ortamda işlenmiş/işlenmemiş büyük miktarda veri birikmesine neden olmuştur. Biriken bu veri yığınları arasında kullanıcıların, ihtiyaç duyduğu bilgiye erişiminde problemler yaşanabilmektedir. Çevrimiçi ortamda, bilgiye erişimde yaşanan sorunların çözümünde, kullanıcıların gerçek zamanlı sorgulamasına ihtiyaç duymadan kullanıcının gözlemlenen geçmiş davranışlarını dikkate alarak kişileştirilmiş öğeler önermeyi amaçlayan bir filtreleme tekniği olan tavsiye sistemleri uygulanabilir. Tavsiye sistemi uygulamalarında, geleneksel yöntemlerden içerik bazlı ve işbirlikçi filtreleme yöntemleri sıklıkla kullanılmaktadır. Bununla birlikte doğrusal/doğrusal olmayan problemlerin çözümünde başarılı sonuçlar üreten ve veri kümelerinde gizli kalan bilginin keşfedilmesinde önemli bir yaklaşım olan yapay zeka teknolojilerinden makine öğrenme teknolojisi potansiyel bir çözüm tekniği olarak dikkat çekmektedir. Bu çalışmada, denetimli makine öğrenmesi tekniklerinden derin sinir ağları, geleneksel makine öğrenme algoritmaları, geleneksel tavsiye sistemi uygulamalarından içerik, işbirlikçi filtreleme teknikleri ile film/kitap derecelendirme uygulaması geliştirilmiştir. Çalışma kapsamında derin sinir ağları ile önerilen uygulamada tekrarlayan sinir ağları (recurrent neural networks - RNN) ve türevleri, ek özellik vektörlerinin yoğun kullanımı, artık ağlarla geliştirilen RS-DSINet modeli ve RNN mimarileriyle geliştirilen hibrit modeller kullanılmıştır. Geleneksel makine öğrenme algoritmalarından karar ağaçları (decision tree - DT), rastgele orman (random forest - RF), olasılıksal dereceli azalma (stochastic gradient descent - SGD), destek vektör makinesi (support vector machine - SVM), K-en yakın komşuluk (K-nearest neighbors - KNN) kullanılmıştır. Deneysel süreçte önerilen yöntemlerin analizinde MovieLens 10M, MovieLens 20M, FilmTV movies, IMDb movies extensive ve Goodbooks-10K veri kümeleri kullanılmıştır. Ampirik sonuçların değerlendirmesinde ortalama mutlak hata (mean absolute error - MAE), ortalama hata karesi (mean squared error - MSE), ortalama karekök sapması (root mean square error - RMSE) ve ortanca mutlak hata (median absolute error - MdAE) hata ölçüm metriklerinden yararlanılmıştır. MovieLens 10M veri setiyle gerçekleştirilen deneysel analizde RS-DSINet + SimpleRNN modeli 0.3663 MAE hata metriği ile başarılı sonuç üretmiştir. Tez kapsamında, deneysel sonuçlardan elde edilen bulgulardan yararlanılarak önerilen RS-DSINet derin öğrenme modeli film/kitap derecelendirme uygulamalarında tatmin edici sonuçlar ürettiği gözlemlenmiştir. RS-DSINet modeli derin sinir ağlarında yaşanan güncel sorunlardan aşırı öğrenme, eksik öğrenme, gradyan kaybolması probleminin çözümünde yeni bir bakış açısı getirebilir, mimarinin geliştirilebilir pratikte ve ekonomik yapısı dolayısıyla veri bilimcilerine fayda sağlayabilir.

ANAHTAR KELİMELER: Derecelendirme problemi, Geleneksel makine öğrenme algoritmaları, RS-DSINet derin öğrenme modeli, Tavsiye sistemi, Tekrarlayan sinir ağları

ABSTRACT

MSc Thesis

RS-DSINET: A NEW DEEP LEARNING MODEL DEVELOPED FOR THE RATING PROBLEM OF RECOMMENDATION SYSTEMS

Veysel TÜRK

Harran University
Graduate School of Natural and Applied Sciences
Department Computer Engineering

Supervisor: Assoc. Dr. İbrahim Berkan AYDİLEK
Year: 2022, Page: 118

Developments in web technologies have led to the accumulation of large amounts of processed/unprocessed data online. Among these accumulated data piles, users may experience difficulty accessing the information they need. Recommendation systems, a filtering technique that aims to suggest personalized items by taking into account the observed past behaviors of the user, without the need for a real-time query of the users, can be applied in solving the problems experienced in accessing information in online environment. Content-based and collaborative filtering methods, which are traditional, are frequently used in recommendation system applications. However, machine learning technology, one of the artificial intelligence technologies, which produces successful results in solving linear/non-linear problems and is an essential approach to discovering hidden information in datasets, draws attention as a potential solution technique. This study deep neural networks from supervised machine learning techniques, traditional machine learning algorithms, content from traditional recommendation system applications, collaborative filtering techniques, and movie/book rating applications. Within the scope of the study, recurrent neural networks (RNN) and their derivatives, intensive use of additional feature vectors, RS-DSINet architecture developed with residual networks, and hybrid models developed with RNN architectures were used in the proposed application with deep neural networks. In addition, traditional machine learning algorithms were used, such as the decision tree (DT), random forest (RF), stochastic gradient descent (SGD), support vector machine (SVM), and K-nearest neighbors (KNN). In the analysis of the proposed methods in the experimental process, MovieLens 10M, MovieLens 20M, FilmTV movies, IMDb movies extensive, and Goodbooks-10K datasets were used. Mean absolute error (MAE), mean squared error (MSE), root mean square error (RMSE), and median absolute error measurement (MdAE) metrics were used to evaluate the empirical results. In the experimental analysis performed with the MovieLens 10M dataset, the RS-DSINet + SimpleRNN architecture produced a successful result with an MAE error metric of 0.3663. Within the scope of the thesis, it has been observed that the RS-DSINet deep learning model proposed using the findings obtained from the experimental results produces satisfactory results in film/book rating applications. The RS-DSINet architecture can bring a new perspective to solving the current problems of deep neural networks, such as over-learning, under-learning, and gradient loss. Furthermore, it can benefit data scientists due to the architecture's developmental practical, and economic structure. Recommendation systems, which is a filtering technique that aims to suggest personalized items by taking into account the observed past behaviors of the user, without the need for the real-time query of the users, can be applied in solving the problems experienced in accessing information in the online environment.

KEY WORDS: Rating problem, Traditional machine learning algorithms, RS-DSINet deep learning model, Recommender system, Recurrent neural networks

TEŞEKKÜR

Yüksek lisans ders döneminde, tez konusunun seçiminde, tez hazırlama sürecinde, akademik çalışmalarımda bilgi ve tecrübesini paylaşarak yol gösteren tez danışmanım Doç. Dr. İbrahim Berkan AYDİLEK hocama destek ve rehberliğinden dolayı teşekkürlerimi sunuyorum.

Yüksek lisans döneminde maddi ve manevi desteklerini hiç esirgemeyen aileme, değerli dostlarım Arş. Gör. Cengiz KINA, Halil ARSLAN, Sedat AKBAŞ, Ömer AĞIRMAN, Ekrem AKSU, Ferit AKGÖK ve Cihan BOZKURT'a teşekkür ederim.

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 3.1. Tez akış şeması 13
Şekil 3.2. Filtreleme teknikleri 15
Şekil 3.3. İçerik tabanlı filtreleme 18
Şekil 3.4. İşbirlikçi filtreleme 19
Şekil 3.5. Makine öğrenme modelleri 24
Şekil 3.6. KNN algoritması 27
Şekil 3.7. SVM algoritması 28
Şekil 3.8. İleri beslemeli yapay sinir ağı 32
Şekil 3.9. Bir nöronun yapısı 33
Şekil 3.10. Sigmoid aktivasyon fonksiyonu grafiği 34
Şekil 3.11. Tanh aktivasyon fonksiyonu grafiği 35
Şekil 3.12. ReLU aktivasyon fonksiyonu grafiği 36
Şekil 3.13. Leaky ReLU aktivasyon fonksiyonu grafiği 37
Şekil 3.14. Swish ReLU aktivasyon fonksiyonu grafiği 38
Şekil 3.15. Softmax aktivasyon fonksiyonu grafiği 39
Şekil 3.16. SimpleRNN mimarisi 42
Şekil 3.17. LSTM mimarisi 43
Şekil 3.18. BiLSTM mimarisi 44
Şekil 3.19. GRU mimarisi 45
Şekil 3.20. CNN mimarisi 46
Şekil 3.21. FCCMLP mimarisi örneği 47
Şekil 3.22. Overfitting 48
Şekil 3.23. Underfitting 48
Şekil 3.24. M-1 modelinin ağ yapısı 51
Şekil 3.25. M-2 modelinin ağ yapısı 53
Şekil 3.26. M-3 modelinin ağ yapısı 55
Şekil 3.27. M-4 modelinin ağ yapısı 57
Şekil 3.28. M-5 modelinin ağ yapısı 59
Şekil 3.29. M-6 modelinin ağ yapısı 61
Şekil 3.30. CV akış şeması 67
Şekil 3.31. Ön işlem adımları 70
Şekil 4.1. MovieLens 10M eğitim veri setinde (a) M-3, (b) M-6, (c) M-Hibrit-1, (d) M-Hibrit-2 derin sinir ağlarının training loss ve validation loss grafikleri 74
Şekil 4.2. MovieLens 10M eğitim veri setinde (a) M-3, (b) M-6, (c) M-Hibrit-1, (d) M-Hibrit-2 derin sinir ağlarının training MAE ve validation MAE grafikleri 76
Şekil 4.3. Derin sinir ağ mimarilerinin test sonuçları (MAE ve RMSE hata metrikleriyle) 77
Şekil 4.4. Derin sinir ağ mimarilerinin test sonuçları (MAE, RMSE ve MdAE hata metriklerinin ortalama değerleriyle) 80
Şekil 4.5. Derin sinir ağ mimarilerinin test sonuçları (MAE ve RMSE hata metrikleriyle) 82
Şekil 4.6. Geleneksel makine öğrenme algoritmalarının test sonuçları (MAE ve RMSE hata metrikleriyle) 85
Şekil 4.7. Goodbooks-10K eğitim veri setinde (a) M-5, (b) M-6, (c) M-Hibrit-2, (d) M-Hibrit-3 derin sinir ağ mimarilerinin training loss ve validation loss grafikleri 87
Şekil 4.8. Goodbooks-10K eğitim veri setinde (a) M-5, (b) M-6, (c) M-Hibrit-2, (d) M-Hibrit-3 derin sinir ağ mimarilerinin training MAE ve validation MAE grafikleri 89
Şekil 4.9. Derin sinir ağlarının test sonuçları (MAE ve RMSE hata metrikleriyle) 90
Şekil 4.10. Derin sinir ağ mimarilerinin test sonuçları (MAE, RMSE ve MdAE hata metriklerinin ortalama değerleriyle) 92
Şekil 4.11. Makine öğrenme algoritmalarının test sonuçları (MAE ve RMSE hata metrikleri) 93
Şekil 4.12. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle) 97
Şekil 4.13. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle) 99
Şekil 4.14. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle) 101
Şekil 4.15. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle) 103

ÇİZELGELER DİZİNİ

Sayfa No

Çizelge 3.1. Denetimli öğrenme algoritmalarına ait hiper-parametreler 25
Çizelge 3.2. İleri beslemeli derin sinir ağının maliyet fonksiyonun hesaplanması 46
Çizelge 3.3. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-1 modelinin mimari detayları 50
Çizelge 3.4. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-2 modelinin mimari detayları 52
Çizelge 3.5. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-3 modelinin mimari detayları 54
Çizelge 3.6. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-4 modelinin mimari detayları 56
Çizelge 3.7. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-5 modelinin mimari detayları 58
Çizelge 3.8. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-6 modelinin mimari detayları 60
Çizelge 3.9. RS-DSINet modelinin eğitim ve test süreci sözde kodu 63
Çizelge 3.10. Hiper-parametre 65
Çizelge 3.11. Veri seti detayları 67
Çizelge 4.1. MovieLens 10M veri kümesiyle ve derin sinir ağları ile geliştirilen derecelendirme uygulamasının ampirik sonuçları (eğitim-doğrulama-test yaklaşımıyla) 73
Çizelge 4.2. MovieLens 10M veri kümesinde farklı sayıdaki eğitim/test oranları ile M-6 modeli kullanılarak geliştirilen uygulamanın deneysel sonuçları 78
Çizelge 4.3. 5-fold cross validation yöntemiyle MovieLens 10M veri kümesi kullanılarak önerilen M (1-6), M-Hibrit (1-3) mimarilerinin performansı 80
Çizelge 4.4. MovieLens 25M veri kümesi ile eğitilen derin sinir ağ mimarilerinin test süreci sonucundaki performansı (eğitim-doğrulama-test yaklaşımıyla) 81
Çizelge 4.5. M-6 modelinin optimal optimizasyon algoritması, loss ve metrik parametrelerinin belirlenmesi 83
Çizelge 4.6. M-6 modelinin epoch, batch size parametrelerinin belirlenmesi 83
Çizelge 4.7. MovieLens 10M veri kümesiyle geliştirilen derecelendirme uygulamasında geleneksel makine öğrenme algoritmalarının performansı 84
Çizelge 4.8. MovieLens 10M veri kümesiyle geliştirilen derecelendirme uygulamasında CBF ve CF tekniklerinin performansı 86
Çizelge 4.9. Goodbooks-10K veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama ve test veri kümeleri ile gerçekleştirilen çalışmanın ampirik sonuçları (eğitim-doğrulama-test yaklaşımıyla) 87
Çizelge 4.10. 5-fold cross validation yöntemiyle Goodbooks-10K veri kümesi kullanılarak önerilen M (1-6), M-Hibrit (1-3) modellerin performansı 91
Çizelge 4.11. Goodbooks-10K veri kümesiyle geliştirilen derecelendirme uygulamasında makine öğrenme algoritmalarının performansı 93
Çizelge 4.12. FilmTV Movies veri kümesiyle gerçekleştirilen derecelendirme uygulamasında derin öğrenme modellerinin performansı (eğitim-doğrulama-test yaklaşımıyla) .. 94
Çizelge 4.13. FilmTV Movies veri kümesiyle geliştirilen derecelendirme uygulamasında makine öğrenme algoritmalarının performansı 95
Çizelge 4.14. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile) 96
Çizelge 4.15. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama (eğitim sırasındaki test verileri) veri kümeleri ile yapılan test çalışmasının deneysel sonuçları ve çalışma zamanı (tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile) 98
Çizelge 4.16. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (18 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile) 99

Çizelge 4.17. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama sonuçları (eğitim sırasındaki test verileri) ve çalışma zamanı (18 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)..... 100
Çizelge 4.18. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (30 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile) 101
Çizelge 4.19. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama sonuçları (eğitim sırasındaki test verileri) ve çalışma zamanı (30 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)..... 102
Çizelge 4.20. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (45 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile) 103
Çizelge 4.21. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama sonuçları (eğitim sırasındaki test verileri) ve çalışma zamanı (45 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)..... 104
Çizelge 4.22. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında geleneksel makine öğrenmesi algoritmalarının ampirik sonuçları (tüm kullanıcıların ağırlık ortalaması ile)..... 105
Çizelge 4.23. Önerilen derin öğrenme modelinin literatürdeki yöntemlerle karşılaştırılması. 106

SİMGELER ve KISALTMALAR DİZİNİ

BiLSTM	Bidirectional Long Short-Term Memory (Çift Yönlü Uzun-Kısa Süreli Bellek)
CBF	Content Based Filtering (İçerik Tabanlı Filtreleme)
CF	Collaborative Filtering (İşbirlikçi Filtreleme Teknikleri)
CNN	Convolutional Neural Network (Evrişimli Sinir Ağı)
CS	Cosine Similarity (Kosinüs Benzerliği)
CV	Cross-Validation (Çapraz Doğrulama)
DNN	Deep Neural Network (Derin Sinir Ağları)
DT	Decision Tree (Karar Ağaçları)
FCCMLP	Fully Connected Multi-Layer Perceptron (Tam Bağlantılı Çok Katmanlı Algılayıcı)
GRU	Gated Recurrent Unit (Kapılı Tekrarlayan Hücre)
HF	Hybrid Filtering (Hibrit filtreleme)
KNN	K-Nearest Neighbors (K-En Yakın Komşuluk)
LSTM	Long-Short Term Memory (Uzun- Kısa Süreli Bellek)
MAE	Mean Absolute Error (Ortalama Mutlak Hata)
MdAE	Median Absolute Error (Ortanca Mutlak Hata)
MF	Matrix Factorization (Matris Çarpanlara Ayırma Yöntemi)
MSE	Mean Squared Error (Ortalama Hata Karesi)
PCA	Principal Component Analysis (Temel Bileşen Analizi)
RF	Random Forest (Rastgele Orman)
RMSE	Root Mean Square Error (Ortalama Karekök Sapması)
RNN	Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
SGD	Stochastic Gradient Descent (Olasılıksal Dereceli Azalma)
SimpleRNN	Simple Recurrent Neural Networks (Basit Tekrarlayan Sinir Ağları)
SVM	Support Vector Machine (Destek Vektör Makinesi)

1. GİRİŞ

Bilgi teknolojilerindeki gelişmeler, web teknolojilerinin yaygınlaşması, web servislerinin yaygın kullanımı ve akıllı cihazların kullanımın artmasından dolayı çevrimiçi ortamda yapılandırılmış/yapılandırılmamış büyük verinin birikmesine neden olmuştur. Burada, çevrimiçi ortamda veride yaşanan üstel büyümeyi gözlemleyebilmek için 2021 yılında Amerikalı şirket DOMO'nun Veri Asla Uyumaz (Data Never Sleeps) 9.0 kapsamında yaptığı araştırma incelenmiştir. Yapılan araştırmaya göre sadece bir dakika Instagram uygulamasında 65.000 fotoğraf paylaşıldığı, Twitter uygulamasında 575.000 tweet atıldığı, TikTok uygulamasında 167 milyon video izlendiği, Google da 5.7 milyon arama yapıldığı, Discord uygulamasında 668.000 mesaj paylaşıldığı, Microsoft Teams uygulamasında 100.000 bağlantı kurulduğu, Zoom uygulamasında 856 toplantı yapıldığı ve Amazonda 283.000 dolar harcama yapıldığı kaydedilmiştir (DOMO, 2021). Veride yaşanan üstel büyüme, kullanıcılara daha fazla seçenek sunsa da aynı zamanda kullanıcıların, aşırı bilgi yüklemesi problemi ile karşı karşıya kalmasına neden olmuştur (Sejwal ve Abulaish, 2020; Duong ve ark., 2020). Büyük verinin birikmesi sonucu yaşanan aşırı bilgi yüklemesi sorununun çözümünde, araştırmacılar, kullanıcılara yalnızca ihtiyaç veya ilgi duyduğu öğeleri sunabilmek ve alakasız öğelerin filtrelenmesi için çeşitli filtreleme tekniklerini önermişlerdir (Sejwal ve Abulaish, 2021). Burada önerilen filtreleme tekniklerini, kişileştirilmiş ve kişileştirilmemiş filtreleme teknikleri olarak iki başlık altında toplayabilmek mümkündür. Kişileştirilmiş filtreleme teknikleri arasında; kişileştirilmiş arama motorları, akıllı kişisel asistanlar, kullanıcıların gözlemlenen geçmiş davranışlarına, ilgi alanlarına, tercihlerine göre veya etkileşim geçmişlerine göre kişisel öğeler önermeyi amaçlayan bir bilgi filtreleme tekniği olan tavsiye sistemi gibi teknolojiler yer almaktadır (Roy ve Ding, 2021; Musto ve ark., 2021; Airen ve Agrawal, 2022; Wang ve ark., 2021a; Rajendran ve Sundarraj, 2021) Kişileştirilmemiş filtreleme teknikleri arasında; e-ticaret siteleri, haber siteleri, kütüphane otomasyon sistemleri, kitap satış siteleri gibi pek çok web site ve uygulamalarında sıklıkla kullanılan ve bilgiye erişimde önemli filtreleme tekniklerinden arama motorları, e-ticaret sitelerinde ürünün özelliklerine (örneğin;

ürün markası, ürün fiyat aralığı, ürünün rengi, ürün değerlendirme puan vb.) göre uygulanan filtreleme teknikleri, kategoriler gibi teknolojiler yer almaktadır. Kişileştirilmiş filtreleme tekniklerinden tavsiye sistemlerinin, çevrimiçi ortamda kullanımı hızla artmaktadır. Son yıllarda tavsiye sistemleri, e-ticaret hizmetleri, turizm endüstrisi, müzik/film tavsiyesi, sosyal ağ uygulamaları, elektronik kitap tavsiyesi gibi birçok web uygulamalarında sıklıkla kullanılmaktadır (Tahmasebi ve ark., 2021; Guo ve ark., 2020). Gerçek dünyada Amazon'da önerilen ürünler, Netflix'teki film-belgesel-oyun önerilerinde, Spotify'da önerilen müzik parçaları, YouTube'da önerilen videolar, Facebook'ta arkadaş önerme, sosyal medyadaki haber akışının kişileştirilmesi gibi web site ve uygulamalarında ticari şirketler kullanıcıların ilgi veya ihtiyaç duyduğu öğelere erişebilmesi için tavsiye sistemlerinden yararlanmaktadır (Afoudi ve ark., 2021; Jesse ve Jannach, 2021; Anwaar ve ark., 2018).

Modern toplumlarda filmler önemli eğlence kaynağıdır. Son yıllarda geniş ekranların tanıtılması, IMAX teknolojisinin gelişmesiyle popülerliği artmıştır (Ning ve ark., 2020). Günümüzde her yıl dünya genelinde 4000'den fazla film üretiliyor (Ning ve ark., 2020). Film sayılarında, film derecelendirme bilgisinde, filmlere yapılan yorumların artması büyük verinin birikmesine yol açmaktadır. Burada, büyük verinin analizinde tavsiye sistemleri yöntemlerinden film tavsiye sistemi popüler bir çözüm tekniğidir (Thakker ve ark., 2021). Büyük film katalogları arasında kullanıcıların manuel analiz süreciyle ilgi duyduğu filmlere ulaşabilmesi zor olabilir. Burada, binlerce film arasında kullanıcıların tercihlerini, geçmiş deneyimlerinden veya demografik bilgilerinden yararlanarak film tavsiye sistemleri ilgili öğeleri öneri başlığı altında otomatik olarak sunmaktadır (Sujithra Alias Kanmani ve ark., 2021). Geniş çapta tartışılan ve tavsiye sistemlerinde önemli çalışmalardan kitap tavsiye sistemi, büyük kitap katalogları arasında kullanıcılara ilgi duyduğu uygun kitapları tavsiye etmeyi amaçlayan bir filtreleme tekniğidir (Wang ve ark., 2021b).

Tavsiye sistemleri uygulamalarında içerik bazlı filtreleme, işbirlikçi filtreleme teknikleri yaygın olarak kullanılan yöntemlerdir (Yassine ve ark., 2021). İçerik bazlı filtreleme tekniğinde öğeler ile ilgili özellikler analiz edilerek kullanıcılara benzer ürünler tavsiye edilir (Saraswat ve ark., 2020; Choudhury ve ark., 2021). İşbirlikçi

filtreleme tekniğinde kullanıcılara uygun ürün önermek için kullanıcı (kullanıcı-kullanıcı benzerliği) veya ürün benzerliği (ürün-ürün benzerliği) yöntemleri kullanılır (Yassine ve ark., 2021). Büyük veri kümelerinden ilgili kullanıcının ilgi veya ihtiyaç duyduğu doğru bilgilere ulaşmasını sağlayabilmek için sıklıkla geleneksel yöntemler kullanılmaktadır. Bununla birlikte genellikle büyük veri kümelerinin kullanıldığı çalışmalarda yüksek performans gösteren makine öğrenme alt başlıklarında derin sinir ağları karmaşık veya karmaşık olmayan örüntüleri işleyebilme ve ham bilgilerden bilgi keşfi sürecinde başarılı sonuçlar üretebilmesi sayesinde tavsiye sistemi alanında potansiyel bir çözüm tekniğidir (Tahmasebi ve ark., 2021). Derin öğrenme tabanlı tavsiye sistemleri, derecelendirme bilgilerini kullanılmasının yanı sıra içerik tabanlı bilgilerden ürün incelemeleri ve öğe açıklamaları gibi bilgilerini kullanabilmesi sayesinde tavsiye sisteminde başarılı sonuçlar üretebilir (Fang, 2021). Derin sinir ağlarında küçük boyuttaki veri kümelerinin kullanılması, etiketli veri eksikliği, model tasarım süreçlerindeki hatalar nedeniyle gradyan kaybolması, aşırı öğrenme (overfitting), eksik öğrenme (underfitting) gibi güncel problemler yaşanmaktadır (de Dios Rojas Olvera ve ark., 2022; Zhu ve ark., 2021a). Sinir ağlarında yaşanan sorunların çözümünde, model tasarım sürecinde ek özellik vektörleri kullanılabilir. Veri ekleme, veri artırımı yöntemleri uygulanabilir.

Bu çalışmada, film ve kitap derecelendirme problemlerinin çözümde yapay zeka teknolojileri ve geleneksel tavsiye sistemleri yöntemlerden içerik tabanlı filtreleme (content based filtering - CBF), işbirlikçi filtreleme teknikleri (collaborative filtering - CF) önerilmiştir. Önerilen çalışmanın birinci aşamasında; GoogLeNet mimarisi inception modülü, ResNet residual ağ yapısı, DenseNet mimarisi dense blok mimari yaklaşımlarından ilham alınarak uzun-kısa süreli bellek (long-short term memory - LSTM), çift yönlü uzun-kısa süreli bellek (bidirectional long-short term memory - BiLSTM), evrişimli sinir ağı (convolutional neural network - CNN), kapılı tekrarlayan hücre (gated recurrent unit - GRU), tam bağlantılı çok katmanlı algılayıcı (fully connected multi-layer perceptron - FCCMLP), basit tekrarlayan sinir ağları (simple recurrent neural networks - SimpleRNN) katmanları ile geliştirilen altı farklı RS-DSINet derin öğrenme modeli ve tekrarlayan sinir ağları (recurrent neural networks - RNN) ve türevleri, CNN, FCCMLP mimarileri ile geliştirilen hibrit yöntemler

önerilmiştir. Derin sinir ağları ile geliştirilen uygulamada, mimarilerin eğitim ve test sürecinde eğitim-doğrulama-test yaklaşımı ve 5 katlı çapraz doğrulama tekniklerinden yararlanılmıştır. İkinci aşamada; geleneksel makine öğrenme algoritmalarından karar ağaçları (decision tree - DT), rastgele orman (random forest - RF), olasılıksal dereceli azalma (stochastic gradient descent - SGD), destek vektör makinesi (support vector machine - SVM), K-en yakın komşuluk (K-nearest neighbors - KNN) yöntemleri uygulanmıştır. Üçüncü ve son aşamada; geleneksel tavsiye sistemi yöntemlerden içerin tabanlı filtreleme ve işbirlikçi filtreleme teknikleri kullanılmıştır. Film derecelendirme uygulamalarında MovieLens 10M, MovieLens 20M, FilmTV movies ve IMDb movies extensive veri kümeleri kullanılmıştır. Kitap derecelendirme uygulamasında Goodbooks-10K veri kümesi kullanılmıştır. Tez kapsamında, derecelendirme problemi ön işlem adımlarında veri kümelerinde yer alan yazar, tür, dil gibi nominal değerlerin sayısal dönüşümünde one hot encoding, süre, yayın yılı, filmtv puan öz niteliklerinin standart bir aralık dönüşümünü sağlamak için min-max normalizasyonu uygulanmıştır. Son olarak, ön işlem adımlarının ilk iki işleminden elde edilen büyük boyutlu vektörün işlem hacmini azaltmak için temel bileşen analizi (principal component analysis - PCA) yöntemi uygulanmıştır. Deneysel analizlerde, MovieLens 10M, MovieLens 20M, FilmTV movies veri setleri ve Goodbooks-10K veri setlerinde ağırlıklı puan ortalaması kullanılırken IMDb veri setinde ise tüm kullanıcıların ağırlıklı oy ortalamaları, tüm erkek/kadın kullanıcıların ağırlıklı oy ortalamaları, 18, 30, 45 yaş grubundaki tüm (erkek/kadın) kullanıcıların filmlere verdiği oyların ortalaması ve son olarak her yaş grubundaki kadınların/erkeklerin filmlere verdiği oyların ortalaması kullanılmıştır. Deneysel sonuçların yorumlanmasında, ortalama mutlak hata (mean absolute error - MAE), ortalama hata karesi (mean squared error - MSE), ortalama karekök sapması (root mean square error - RMSE) ve ortanca mutlak hata (median absolute error - MdAE) hata metrikleri kullanılmıştır. Ampirik sonuçlardan elde edilen bilgilere göre yapay zeka teknolojileri ile önerilen derecelendirme uygulamalarının daha başarılı sonuçlar ürettiği gözlemlenmiştir. Yapay zeka teknolojileri arasında derin sinir ağlarıyla önerilen derecelendirme uygulamalarının genellikle daha başarılı sonuçlar üretmiştir.

Bu tez çalışmasında; Bölüm 2'de, yapay zeka teknolojileri ve geleneksel yöntemler ile önerilen derecelendirme uygulamalarına ilişkin literatürde yapılan çalışmalar sunulmuştur. Bölüm 3'de, tavsiye sistemi, tavsiye sistemi uygulamaları, tavsiye sistemleri benzerlik ölçütleri, geleneksel tavsiye sistemi uygulamalarında yaşanan yaygın problemler, makine öğrenme algoritmalarının teorik çerçevesi, tez kapsamında önerilen RS-DSINet derin öğrenme modeli ayrıntılı analizi, derin sinir ağlarının teorik çerçevesi, derin sinir ağlarının eğitimi ve test süreci, tez kapsamında kullanılan veri setleri, ön işlem adımları, performans metrikleriyle materyal ve yöntemler verilmiştir. Bölüm 4'te, yapay zeka teknolojileri ve geleneksel yöntemler ile geliştirilen derecelendirme uygulamalarının deneysel sonuçları ve yorumlaması verilmiştir. Son olarak Bölüm 5'te, sonuç ve öneriler sunulmuştur.

2. ÖNCEKİ ÇALIŞMALAR

Büyük miktardaki heterojen verinin çevrimiçi ortamda kontrolsüz şekilde birikmesi aşırı bilgi yüklemesi problemine neden olmaktadır. Yaşanan problem, kullanıcıların ihtiyaç duyduğu veriye erişiminde sınırlamalar getirebilir. Karşılaşılan problemlerin çözümünde, kişileştirilmiş filtreleme tekniklerinden tavsiye sistemleri popüler bir çözüm tekniği olabilir. Tavsiye sistemleri, etkileşimli web siteleri (örneğin sosyal ağlar, forum, blog vs.), e ticaret ürün tavsiyelerinde, makale önerilerinde, kitap tavsiyesi gibi birçok alanda yaygın olarak kullanılmaktadır (Afoudi ve ark., 2021; Jesse ve Jannach, 2021; Anwaar ve ark., 2018). Tavsiye sistemleri problemlerinde sıklıkla geleneksel tavsiye sistemi yöntemlerden CF, CBF, hibrit filtreleme (hybrid filtering - HF) ve matris çarpanlara ayırma yöntemi (matrix factorization - MF) teknikleri uygulanmaktadır. CF tekniğinde yaşanan soğuk başlatma, veri seyrekliği, ölçeklenebilirlik sorunu ve CBF tekniğinde yetersiz veri sorunu nedeniyle tavsiye sistemlerinde uygulamalarında yaşanan performans kaybı sorunlarının çözümlerinde HF ve MF yöntemleri uygulanmıştır. Burada HF yönteminde artan hesaplama karmaşıklığı ve MF teknolojisinin doğrusal olmayan problemlerin çözümünde etkili olamaması tavsiye sistemleri uygulamalarında araştırmacıları yeni ve etkili yöntem arayışına girmiştir. Son yıllarda, tavsiye sistemleri uygulamalarında derin sinir ağları popüler bir çözüm tekniği olarak dikkat çekmektedir. Bu bölümde, geleneksel yöntemler ve yapay zeka teknolojileri ile önerilen tavsiye sistemleri konu kapsamında literatür çalışmaları sunulmuştur.

Zhu ve ark. (2021b), kişileştirilmiş tavsiye sistemleri için işbirlikçi otomatik kodlayıcı (“collaborative autoencoder for personalized recommendation” - CAPR) önermişlerdir. Önerilen yöntem kullanıcı ve öğe tabanlı verilerin farklı öznitelikleriyle iki farklı otomatik kodlayıcı modelinden yararlanılarak verilerdeki farklı özelliklerin keşfinde uygulanmıştır. CAPR modeli ile önerilen film derecelendirme uygulamasında MovieLens 100K, MovieLens 1M, MovieTweetings-10K veri kümeleri kullanılmıştır. MovieLens 100K veri kümesi ile geliştirilen uygulamada CAPR 0.685 MAE hata

değeri üretirken, MovieLens 1M veri kümesinde 0.656 MAE ve son olarak MovieTweetings-10K veri kümesinde 0.611 MAE hata değeri bulunmuştur.

Chen ve ark. (2021b), kullanıcı öğe matrisinden yararlanarak komşuluk tabanlı modeller (“neighborhood based models” - NBMs) makine öğrenme yöntemi ile tavsiye sistemi derecelendirme probleminin çözümünde çekirdek tabanlı matris tamamlama (“kernel based matrix completion” - KMC) modeli önerilmişlerdir. Önerilen modelin derecelendirme problemindeki performansını iyileştirmek için kosinüs benzerliği (cosine similarity - CS), pearson korelasyon katsayısı (pearson correlation coefficient - PCC), ortalama kare farkı (mean square difference - MSD) benzerlik ölçütleri ve gauss çekirdeği uygulanmıştır. Ayrıca yazarlar KMC modelinin performansını iyileştirmek için modele çok çekirdekli bir çerçeve ekleyerek KMC için çok çekirdekli bir çerçeve (multi-kernel framework for KMC - M-KMC) modelini geliştirmişlerdir. Önerilen yöntemler Filmtrust, Epinions, Amusic, Jester, Ciao, BookCrossing, MovieLens 1M olmak üzere yedi farklı veri kümeleri ile kapsamlı analiz edilmiştir. BookCrossing veri kümesi önerilen uygulamada KMC, M-KMC sırasıyla 1.3734 MAE ve 1.3618 MAE hata değerlerini üretmiştir. Ciao veri kümesi önerilen uygulamada KMC, M-KMC sırasıyla 0.7336 MAE ve 0.7288 MAE hata değerlerini üretmiştir. Amusic veri kümesi önerilen uygulamada KMC, M-KMC sırasıyla 0.6477 MAE ve 0.6351 MAE hata değerlerini üretmiştir.

Shambour (2021), çok kriterli öneri sistemleri uygulamalarında derin sinir ağ tabanlı, otomatik kodlayıcı tabanlı çoklu kriter (“autoencoder based multi criteria” - AEMC) mimarisini önermiştir. Önerilen mimari kullanıcı-öğe matrisinden yararlanarak, doğrusal olmayan karmaşık yapılı gizli ilişkilerin keşfiyle ve önemsiz verilerin tespit edilmesiyle tavsiye sistemleri uygulamalarının doğruluğunun iyileştirilmesi hedeflemiştir. Çalışma kapsamında önerilen mimari TripAdvisor ve Yahoo! Movies veri kümeleri ile deneysel olarak analiz edilmiştir. TripAdvisor veri seti ile geliştirilen uygulamada AEMC mimarisi 0.8712 MAE hata değeri üretirken Yahoo! Movies veri kümesi ile geliştirilen uygulamada AEMC mimarisi 0.6435 MAE hata değeri üretmiştir.

Anwar ve ark. (2022), tavsiye sistemlerinde yaşanan soğuk başlatma ve seyreklik sorunlarını önleyebilmek için model ve bellek tabanlı işbirlikçi filtreleme tekniklerini uygulamışlardır. Model tabanlı işbirlikçi filtreleme tekniğinde ortak kümeleme (co-clustering - CC) yöntemi uygulanmıştır. Bellek tabanlı işbirlikçi filtreleme tekniğinde standart K-en yakın komşu (KNNBaseline), temel K-en yakın komşu (KNNBasic), tekil değer ayrışımı (singular value decomposition-SVD), SVD++ ve ortalama K-en yakın komşu (KNNWithMeans) makine öğrenme algoritmalarından yararlanılmıştır. Önerilen yöntemler FilmTrust veri seti ile test edilmiştir. Deneysel süreçte uygulanan modellerin detaylı analizi için çapraz doğrulama (cross validation - CV) yöntemini uygulamışlardır. Çalışma kapsamında 5, 10, 15 - katlı CV (5, 10, 15 - fold CV) model doğrulama tekniği uygulanmıştır. FilmTrust veri kümesi ile önerilen 15-fold CV uygulamasında 0.6245 MAE hata değeri ile KNNBaseline en başarılı model olmuştur. CC yöntemi 5-fold CV uygulamasında 0.6818 MAE hata değeri ile çalışma kapsamında başarısız bir performans göstermiştir. KNNBaseline algoritması literatür kapsamındaki sosyal güven ile tavsiye sistemi ve autoencoder kullanarak derin öğrenme (“recommender system with social trust and deep learning using autoEncoder” - AutoTrustRec), işbirlikçi kullanıcı ağı yerleştirme (“collaborative user network embedding”- CUNE), örtük güven tavsiye yaklaşımı (“implicit trust recommendation approach” - ITRA) yöntemleri ile karşılaştırılmıştır. Ampirik sonuçlara göre AutoTrustRec 0.7010 MAE hata değeri, CUNE 0.7980 MAE hata değeri, ITRA 0.9050 MAE hata değeri üretirken, çalışma kapsamında önerilen KNNBaseline algoritması 0.6245 MAE hata değeri üreterek en başarılı model olmuştur.

Fkih (2021), tavsiye sistemi uygulamalarında yaygın olarak kullanılan PCC, dengelenmis kosinüs vektörü (adjusted cosine vector - ACV), öklit benzerliği (euclidean similarity - ES), vektör benzerliği (vector similarity - VS), Manhattan benzerliği (Manhattan similarity - MS), jaccard indeksi (jaccard index - JI), dengelenmiş sıra indeksi (adjusted rand index - ARI), Spearman sıra-düzen korelasyon katsayısı (Spearman rank-order correlation coefficient - SROCC), dengelenmiş karşılıklı bilgi (adjusted mutual information - AMI), Kendall’ın tau’su (Kendall’s tau - KsT), geliştirilmiş pearson korelasyon katsayısı ağırlığı ile kullanıcı derecelendirme

tercihi davranışı (“improved pearson correlation coefficient weighted with user rating preference behavior” - IPWR), Chebyshev benzerliği (Chebyshev similarity - ChS), kullanıcı derecelendirme tercihleriyle tamamlanan geliştirilmiş üçgen benzerliği (“improved triangle similarity complemented with user rating preferences” - ITR) 13 benzerlik ölçütleri ile kullanıcı tabanlı ve öğe tabanlı CF tekniği ile tavsiye sistemi uygulaması önermiştir. Önerilen yaklaşımlar MovieLens 100K, MovieLens 1M ve Jester veri kümeleri ile test edilmiştir. Ampirik sonuçlara göre öğe tabanlı tavsiye sisteminde AMI yöntemi başarılı sonuç verirken, kullanıcı tabanlı tavsiye sisteminde ITR ve IPWR yöntemleri başarılı olmuştur. Öğe tabanlı yaklaşımda AMI yöntemi Jester veri setinde 3.2814 MAE, MovieLens 1M veri setinde 0.7244 MAE ve MovieLens 100K veri setinde 0.7458 MAE hata değeri ile başarılı sonuçlar üretmiştir. Kullanıcı tabanlı tavsiye sisteminde ITR yöntemi MovieLens 100K veri setinde 0.7863 MAE, MovieLens1M veri setinde 0.7315 en düşük MAE hata değeri üreterek başarılı olurken, Jester veri setinde IPWR yöntemi 3.2561 MAE hata değeri ile en başarılı model olmuştur.

Anwar ve Uma (2021), SVD++ algoritması ve işbirlikçi filtreleme tekniği ile film tavsiye sistemi önermiştir. SVD++ önerilen yöntem KNN, SVD ve CC makine öğrenme yaklaşımları ile karşılaştırılmıştır. Önerilen yaklaşımlar MovieLens 100K veri kümesi ile test edilmiştir. Önerilen yöntemlerin performans analizi MAE, RMSE hata metrikleri ile gerçekleştirilmiştir. Çalışma kapsamında en düşük hata oranına sahip SVD++ algoritması, film derecelendirme probleminde 0.9201 RMSE ve 0.7219 MAE hata metriğini üretmiştir.

Sun ve ark. (2022), tavsiye sistemi problemlerin çözümünde metrik çarpanlara ayırma yöntemi ve derin öğrenme algoritmaları ile geliştirilen sinirsel metrik çarpanlara ayırma çerçevesi (“neural metric factorization framework” - NMetricF) yöntemini önermiştir. Çalışma kapsamında NMetricF yönteminde kullanıcı-öğeler arasındaki mesafenin ölçümü için öklit uzaklığı yönteminin etkisi araştırılmıştır. Deneysel süreç Movielens-Latest, Movielens 100K, Movielens 1M, FilmTrust, EachMovie, Goodbooks-10K olmak üzere altı veri seti ile gerçekleştirilmiştir. Movielens-1M veri setiyle gerçekleştirilen deneysel sonuçlara göre NMetricF yöntemi

0.731 RMSE hata değeri üretirken, Goodbooks veri setiyle gerçekleştirilen deneysel sonuçlara göre NMetricF 0.747 RMSE hata değeri üretmiştir.

Sinha ve Dhanalakshmi (2022), çok kriterli tavsiye sistemlerinin başarısını artırmak için çoklu kriterlerden elde edilen puan bilgilerden yararlanarak yeni ve etkili bir yöntem araştırmıştır. Yazarlar çok kriterli uygulamalarda kullanıcılar arasındaki doğrusal olmayan gizli ilişkilerin keşfi için MF, derin sinir ağları (deep neural network - DNN) ve sosyal örümcek algoritması (social spider optimization - SSO) ile hibrit bir yöntem önermiştir. Önerilen yöntem Yahoo! ve TripAdvisor veri setleri test edilmiştir. Yahoo! veri setiyle gerçekleştirilen deney sonuçlarına DNN-MF + SSO hibrit yöntemi 0.6147 MAE hata değeri ile çalışma kapsamındaki en başarılı sonucu üretmiştir. TripAdvisor veri setiyle gerçekleştirilen deneysel sonuçlara göre DNN-MF + SSO hibrit yöntemi 0.7361 MAE hata değeri ile en başarılı sonucu üretmiştir.

Sharma ve ark. (2022), kümeleme ve işbirlikçi filtreleme tekniğinden yararlanılarak tavsiye sistemi uygulaması geliştirilmiştir. Yazarlar kümeleme yöntemiyle geliştirilen uygulamanın optimizasyonunu sağlamak için sürü zekası tabanlı kümeleme yöntemini önermişlerdir. Burada süre zekası optimizasyon yöntemlerinden guguk kuşu kümeleme algoritması (cuckoo clustered algorithm - CCA), ateşböceği kümeleme algoritması (firefly clustered algorithm - FCA), serçe arama algoritması (sparrow clustered algorithm - SCA) ve balina kümeleme algoritması (whale clustered algorithm - WCA) kullanılmıştır. Önerilen yaklaşımlar MovieLens 100K, MovieLens 1M, Epinions, Jester veri kümeleri ile test edilmiştir. MovieLens 100K veri setiyle gerçekleştirilen deneysel sonuçlara göre FCA 0.695 MAE hata değeri ile en başarılı sonucu üretmiştir. MovieLens 1M veri setiyle gerçekleştirilen deneysel sonuçlara göre SCA 0.654 MAE hata değeri ile en başarılı sonucu üretmiştir. Epinions veri setiyle gerçekleştirilen deneysel sonuçlara göre SCA 0.78 MAE hata değeri ile en başarılı sonucu üretmiştir. Jester veri setiyle gerçekleştirilen deneysel sonuçlara göre SCA 3.09 MAE hata değeri ile en başarılı sonuçtur.

Harshvardhan ve ark. (2021), tavsiye sistemi derecelendirme problemlerin çözümünde denetimsiz boltzmann makine tabanlı zamana duyarlı tavsiye sistemi ("unsupervised boltzmann machine based time aware recommender system" - UBMTR) yöntemi önermişlerdir. UBMTR yönteminde girdi verileri film derecelendirmeleri ve zamansal bilgilerinden oluşmaktadır. Önerilen yaklaşım MovieLens 100K veri kümesiyle test edilmiştir. MovieLens 100K ile yapılan deneysel analizde önerilen UBMTR yöntemi 0.76 MSE hata değeri ile başarılı bir performans göstermiştir.

Behera ve Nain (2022), tavsiye sistemleri uygulamalarında doğrusal olmayan problemlerin çözümü sürecinde bir MF yöntemi olan bir öğenin meta verileri derin işbirlikçi öneri sistemine dönüştürme (metadata of an item into a deep collaborative recommender system - ME-DCRS) metodunu önermiştir. ME-DCRS iki aşamadan oluşmaktadır: Birinci aşamada kullanıcılar ve öğelere ait veriler belirlenir, ikinci aşamada elde edilen tüm metadata verileri birleştirilir son olarak toplanan veriler DNN'de girdi verileri olarak kullanılmaktadır. Önerilen yaklaşım MovieLens 100K, MovieLens 1M, MovieLens latest-small veri kümeleri ile test edilmiştir. MovieLens 1M veri setiyle gerçekleştirilen deneysel sonuçlara göre önerilen yöntem 0.573 MAE hata değeri üretmiştir. ME-DCRS yönteminin detaylı analizi için 5-fold CV yöntemi uygulanmıştır. Deneysel sonuca göre 0.5681 MAE hata değeri ile başarılı bir sonuç üretmiştir.

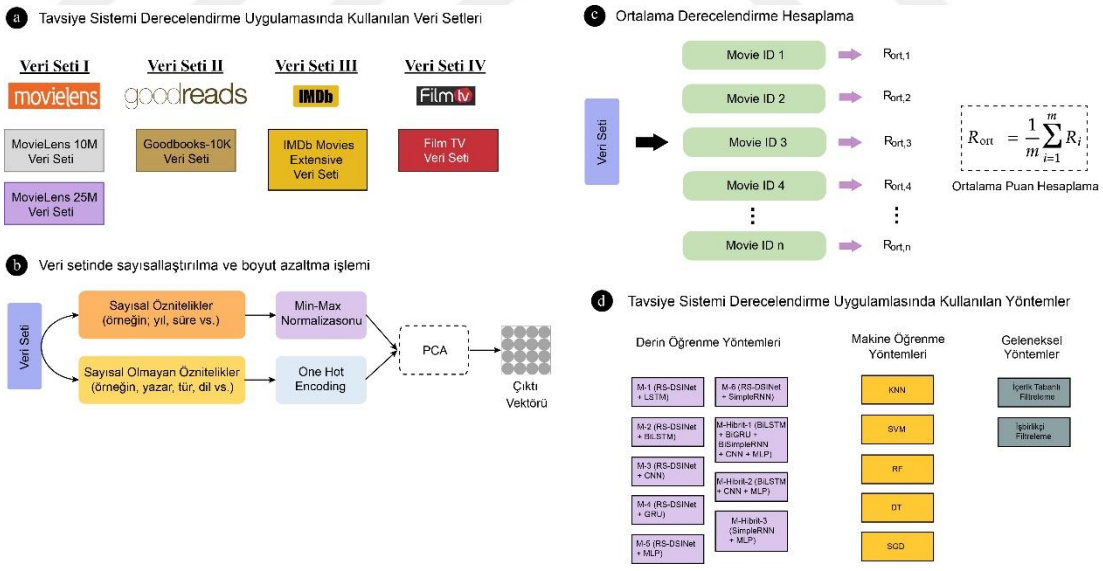
Aljunid ve Dh (2020), işbirlikçi filtreleme yönteminde yaşanan seyreklik sorunun çözümünde derin işbirlikçi tabanlı, işbirlikçi tavsiye sistemlerinde derin öğrenme yöntemi ("deep learning method of collaborative recommender systems" - DCLSR) yöntemini önermişlerdir. Önerilen yaklaşım Movielens-100K ve Movielens-1M veri kümeleriyle test edilmiştir. MovieLens 100K veri setiyle gerçekleştirilen deneysel sonuçlara göre önerilen yöntem 0.917 RMSE hata değeri üretmiştir. MovieLens 1M veri setiyle gerçekleştirilen deneysel sonuçlara göre önerilen yöntem 0.903 RMSE hata değeri vermiştir.

Zhang ve ark. (2022), tavsiye sistemleri derecelendirme tahminlerinde kullanıcı incelemeleri ve derecelendirmeler kullanılmıştır. Burada yazarlar kullanıcı ve öğeler arasındaki incelemeler ve derecelendirmeleri modellemek için grafik evrişimsel ağ (graph convolutional network - GCN) önermişlerdir. Grafiğe dayalı kullanıcı-öğe özelliklerinin öğrenilmesi ve derecelendirme, incelemeleri birleştirmek için atfedilen grafik evrişimsel ağ metodu (attributed GCN method - AGCN) önerilmiştir. Önerilen yöntem 6 Amazon veri seti (digital music, automotive, Kindle store, toys and games, instant video, movies and TV) ve Epinions veri kümeleriyle test edilmiştir. Digital music, automotive, Kindle store, toys and games, instant video, movies and TV veri setleriyle gerçekleştirilen deneysel sonuçlara göre önerilen yöntem sırasıyla 0.6432, 0.5858, 0.5354, 0.6158, 0.6790, 0.7191 MAE hata değerlerini üretmiştir. Epinions veri setiyle gerçekleştirilen deneysel sonuçlara göre önerilen yöntem 0.9098 MAE hata değeri üretmiştir.

Nápoles ve ark. (2020), tavsiye sistemi derecelendirme probleminin çözümünde model tabanlı ve bellek tabanlı işbirlikçi filtreleme teknikleri ile kullanıcı öğe matrislerinden yararlanarak RNN mimarilerinden uzun vadeli/dönemli bilişsel ağlar (“long-term cognitive networks” - LTCNs) sinir ağını önermişlerdir. Önerilen LTCNs mimarisi sigmoid nöronlar (sigmoid neurons - SNs), gauss çekirdek nöronu (gaussian kernel neuron - GKN) ve PCC yöntemleri ile geliştirilmiştir. Model tabanlı işbirlikçi filtreleme yönteminde otokodlayıcı (autoencoders - AE), varyasyonel otomatik kodlayıcı (variational autoencoders - VAE), MF ve SVD yaklaşımları uygulanırken, bellek tabanlı işbirlikçi filtreleme yönteminde KNN makine öğrenme yöntemi uygulanmıştır. Önerilen yaklaşımlar MovieLens10M, Anime Recommendations ve Netflix Prize veri kümeleriyle test edilmiştir. Anime Recommendations veri kümesinden rastgele seçilen 500 veriyle gerçekleştirilen deneysel sonuçlara LTCN mimarisi 0.0132 MSE hata metrik değeri ile en başarılı model olmuştur. MovieLens10M veri kümesinden rastgele seçilen 500 veriyle gerçekleştirilen deneysel sonuçlara AE mimarisi 0.0059 MSE hata metrik değeri ile en başarılı model olmuştur. Netflix Prize veri kümesinden rastgele seçilen 500 veriyle gerçekleştirilen deneysel sonuçlara LTCN mimarisi 0.0046 MSE hata metrik değeri ile en başarılı model olmuştur.

3. MATERYAL ve YÖNTEM

Bu bölümde, tavsiye sistemi, tavsiye sistemi uygulama alanları ve film-kitap tavsiye sistemi hakkındaki bilgiler, tavsiye sistemi uygulamaları hakkındaki teknik ve teorik bilgiler ve tavsiye sistemi uygulamalarında yaşanan problemler Bölüm 3.1'de, geleneksel makine öğrenme algoritmalarının teorik çerçevesi Bölüm 3.2'de, derin öğrenme algoritmalarının teorik çerçevesi, tez kapsamında önerilen RS-DSINet derin öğrenme modellerinin ayrıntılı analizi, mimarilerin bileşenleri, önerilen derin öğrenme modelinin eğitim ve test süreci, derin öğrenme modellerinde kullanılan hiper-parametrelerin teorik bilgisi ve derin öğrenme algoritmalarında yaşanan problemler, Bölüm 3.3'de, CV yöntemi Bölüm 3.4'de, tez kapsamında kullanılan veri setleri Bölüm 3.5'de, veri setine uygulanan ön işlemlerin teorik ve teknik bilgisi Bölüm 3.6'da, RS-DSINet derin öğrenme modellerinin deneysel analizinde kullanılan farklı performans ölçütlerinin teorik ve teknik bilgisi Bölüm 3.7'de sunulmuştur. Tez akış şema özeti Şekil 3.1.'de gösterilmiştir.



Şekil 3.1. Tez akış şeması

3.1. Tavsiye Sistemi

Çevrimiçi ortamda biriken büyük miktardaki heterojen veri, aşırı bilgi yüklemesi sorununa yol açmaktadır (Sejwal ve Abulaish, 2020). Bu durum bilgiye erişim sürecinde kısıtlamalara neden olabilir. Burada yaşanan sorunun çözümünde sıklıkla, kişileştirilmemiş ve kişileştirilmiş başlıkları altında yer alan bilgi filtreleme teknolojilerinden yararlanılmaktadır.

Kullanıcıların gerçek zamanlı sorgulamalarına (bilgi girişlerine) ihtiyaç duyan kişileştirilmemiş filtreleme teknikleri, aşırı bilgi yüklemesi sorunu karşısındaki kullanıcılara başarılı çözümler sağlamaktır (Fan ve ark., 2008). Kişileştirilmemiş filtreleme teknikleri, günümüzde çeşitli web site ve uygulamalarında sıklıkla kullanılmaktadır. Bu teknolojiler, bilgi filtreleme sürecinde başarılı çözümler sunması dolayısıyla önemli çözüm teknikleri olsa da buna karşın kullanıcıların ilgi alanlarına göre ihtiyaçları doğrultusundaki içeriklerin üretiminde yetersiz kalmakta önerilen ürünler tüm kullanıcılar için aynıdır (Azeroual ve Koltay, 2022). Kişileştirilmemiş filtreleme teknikleri arasında:

- Kitap satış siteleri, e-ticaret siteleri, haber siteleri, akademik siteler gibi web site ve uygulamalarında, kütüphane otomasyon sistemleri, stok takip otomasyonu, kurum ve kuruluş otomasyon sistemleri gibi birçok alanda sıklıkla kullanılan arama motorları,
- E-ticaret site ve uygulamalarında ilgili ürünlerin özelliklerine göre (örneğin; renk, marka, ürün değerlendirme puanları vb.) uygulanan filtreleme teknikleri,
- Haber siteleri, e-ticaret sitelerindeki kategoriler,
- Sosyal ağlar, akademik sitelerde, haber sitlerinde yer alan etiket teknolojileri gibi filtreleme teknikleri yer almaktadır.

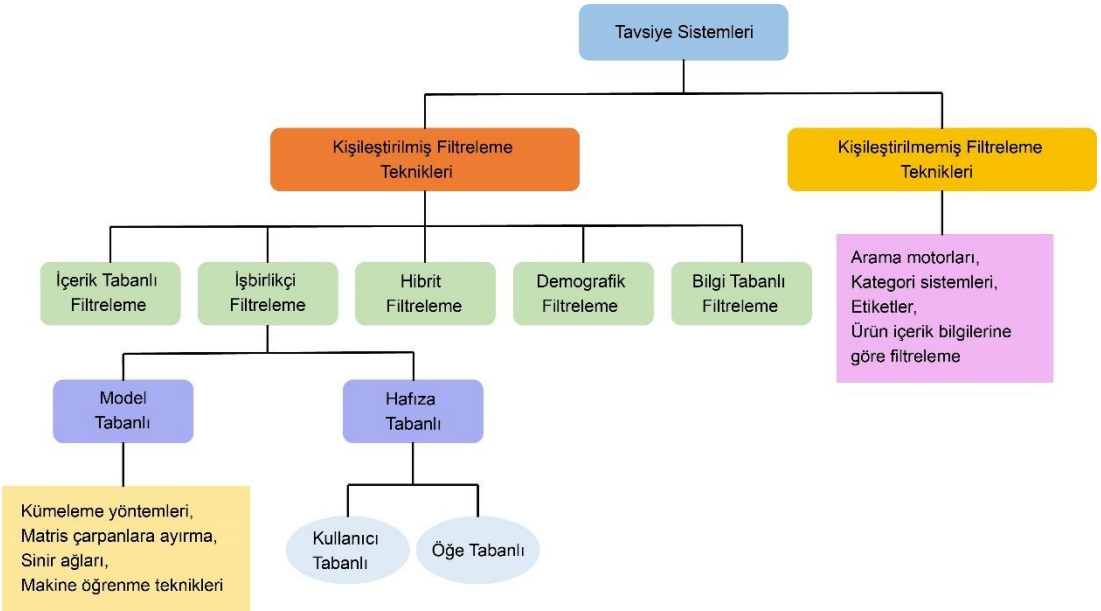
Kişisel veri üretiminde yetersiz kalan kişileştirilmemiş filtreleme tekniklerine çözüm olarak araştırmacı/geliştirici/uzmanlar tarafından önerilen kişileştirilmiş filtreleme teknikleri sıklıkla kullanılmaktadır. Kişileştirilmiş filtreleme teknikleri

genellikle kullanıcıların gerçek zamanlı sorgulamalarına ihtiyaç duymadan öneriler sunmaktadır. Bu teknikler arasında: kişileştirilmiş arama motorları, tavsiye sistemleri, akıllı kişisel asistanlar gibi teknolojiler yer almaktadır (Musto ve ark., 2021).

Kişileştirilmiş filtreleme teknikleri arasında sıklıkla başvurulan yöntem tavsiye sistemleridir. Veri madenciliği teknolojilerinden tavsiye sistemi, büyük heterojen veri kümeleri karşısındaki kullanıcılara gerçek zamanlı sorgulamalara (bilgi girişlerine) ihtiyaç duymadan, kullanıcının gözlemlenen geçmiş davranışları incelenerek ilgili kullanıcının ilgi alanlarına göre ihtiyaçları doğrultusundaki içeriklerin üretilmesinde yardımcı olmayı amaçlayan bir karar destek sistemidir (Yassine ve ark., 2021). Tavsiye sistemlerinin başlıca görevleri:

- İlgili kullanıcıların deneyimleri, tercihleri hakkında bilgi toplamak,
- Toplanan verilerden ilgili kullanıcının tercih edeceği ürünleri seçmesine, bulmasına yardımcı olabilecek model oluşturma yer alır (Yassine ve ark., 2021).

Şekil 3.2.'de kişileştirilmiş ve kişileştirilmemiş filtreleme tekniklerinde yer alan teknolojiler verilmiştir.



Şekil 3.2. Filtreleme teknikleri (Thakker ve ark. (2021) çalışmasından uyarlandı)

Tavsiye sistemleri eBay, Amazon, Flipkart gibi e-ticaret sitelerinde, Twitter, Facebook gibi sosyal medya web site ve uygulamalarında, film platformları, dijital kütüphaneler, çevrimiçi video ve müzik akış platformlarında (örneğin, Spotify, Youtube vs.), e-öğrenme portalları, e-turizm gibi birçok alanda kişisel içerik üretme kapsamında yenilikçi çözüm önerileri sunmaktadır (Thakker ve ark., 2021).

Filmler modern dünyada önemli bir eğlence kaynağıdır (Ning ve ark., 2020). Son yıllarda IMAX teknolojisinin ve geniş ekran teknolojilerinin gelişmesi ile popülerliği artmaktadır (Ning ve ark., 2020). Her yıl dünya genelinde ortalama 4000'den fazla film üretilmektedir (Ning ve ark., 2020). Film sayılarında, web sitelerinde ilgili filmlerin derecelendirme bilgisinde ve ilgili filmlere yönelik yorumlarında yaşanan artış sonucunda büyük verinin çevrimiçi ortamda depolanmasını sağlamıştır (Roy ve Ding, 2021). Veride yaşanan üstel büyüme veriye erişim sürecinde ve kullanıcıların ilgi alanlarına göre içeriklere ulaşmasında problemlere neden olabilir. Burada veriye erişim sürecinde yaşanan problemlerin çözümünde tavsiye sistemi teknolojilerinden film tavsiye sistemi uygulanabilir (Roy ve Ding, 2021). Film tavsiye sistemleri film puan tahminlerinde ve film öneri listeleri oluşturmada sıklıkla başvurulan bir yöntemdir. Film derecelendirme, filmlerin kalite değerlendirmesinde ve ilgili filmlerin izlenip izlenmemesini belirleyebilen temel bir kriterdir (Ning ve ark., 2020).

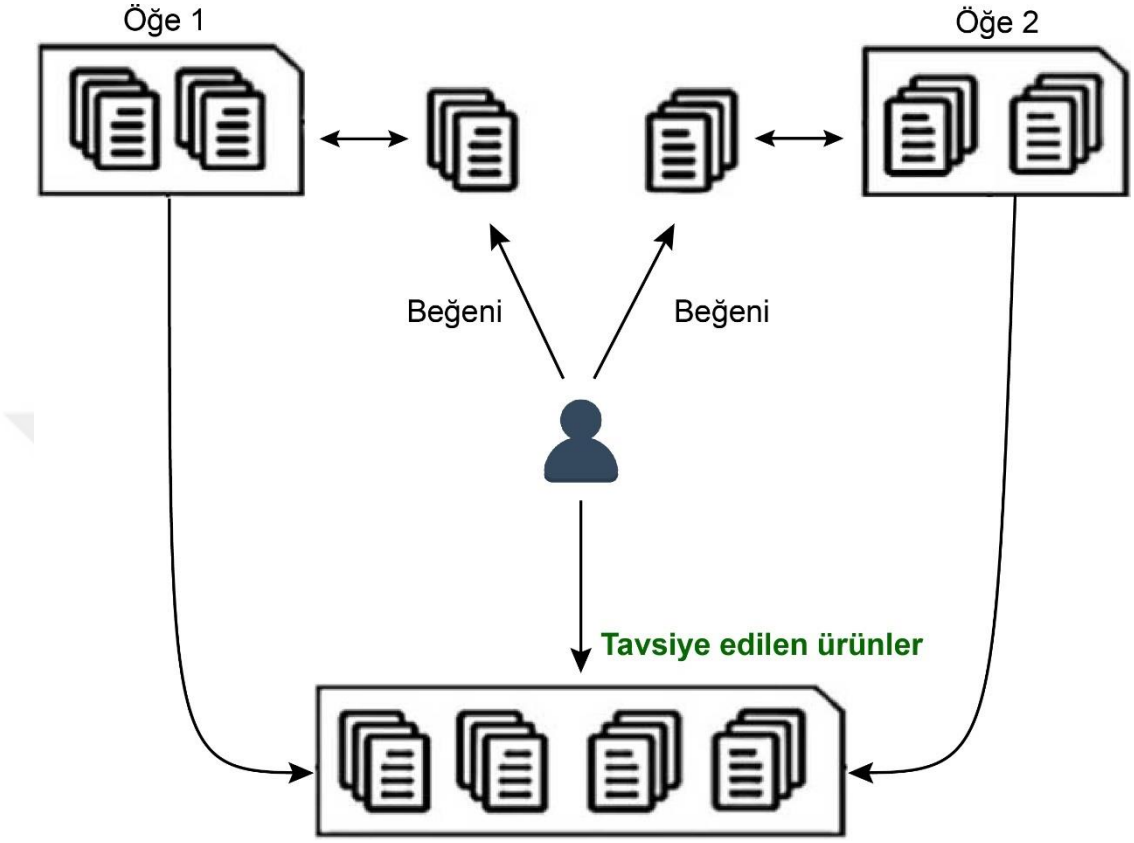
Son dönemlerde, dijital kütüphane kaynaklarında yaşanan artış bilgi edinme sürecinde önemli olsa da geniş koleksiyonlar arasında kullanıcıların ilgi veya ihtiyaçları doğrultusundaki kaynaklara ulaşımda sorunlar yaşamasına neden olabilir. Burada, yaşanan problemlerin çözümünde tavsiye sistemi teknolojilerinden kitap tavsiye sistemi kullanıcıların ilgi alanlarına göre ihtiyaçları doğrultusundaki içeriklere ulaşmasında popüler bir teknolojidir (Ullah ve ark., 2020). Kitap tavsiye sistemi, kitap öneri listeleri ve kitap puan tahminlerinde güncel bir uygulama şeklidir.

3.1.1. Tavsiye sistemi uygulamaları

Tavsiye sistemlerinde uygulamalarında, geleneksel tavsiye sistemi yöntemlerden CBF, CF, HF ile birlikte MF, demografik filtreleme (DF), bilgi tabanlı filtreleme (BTF) gibi farklı yöntemler kullanılmaktadır (Thakker ve ark., 2021). Tavsiye sistemleri uygulamalarında CBF, CF, HF yaygın olarak kullanılırken BTF ve DF yöntemleri genellikle içerik ve işbirlikçi filtreleme teknikleriyle birlikte kullanılmaktadır (Peška ve ark., 2019). Son yıllarda, doğrusal problemlerin çözümünde başarılı sonuçlar üretebilen MF yöntemi tavsiye sistemleri problemlerine sıklıkla uygulanmaktadır (Behera ve Nain, 2022).

İçerik tabanlı filtreleme: Kullanıcıların geçmiş davranışlarının ve tercihlerinin dikkate alındığı bu yöntemde, öğelerin içerik bilgilerinden yararlanılarak ilgili kullanıcıların geçmişte beğendiği/tercih ettiği öğelere benzer öğeler önerilmektedir (Guo ve ark., 2020). CBF yaklaşımın temeli, kullanıcı tarafından tercih edilen öğeler ile önerilecek öğeler arasında, öğelerin içerik bilgilerine göre yapılan benzerlik hesaplamaları sonucunda en uygun öğelerin önerilmesine dayanır. CBF yöntemin birçok avantajı bulunmaktadır. Diğer kullanıcıların profil bilgilerinin dikkate alınmadığı bu yöntemde, öğelere ait yeterli içerik bilgisi olması durumunda tavsiye sistemlerinde yaşanan soğuk başlatma sorunu önlenebilir (Guo ve ark., 2020; Zhu ve ark., 2021b). Bu yöntemde sadece ilgili kullanıcıların tercihleri dikkate alındığından, genellikle işbirlikçi filtreleme tekniklerinde yaşanan ve kişisel veri paylaşımından doğabilecek veri gizliliği ve güvenlik sorunları yaşanmaz (Roy ve Dutta, 2022). CBF yöntemi zamanla değişen kullanıcı tercihlerine göre önerileri listelerini, dinamik ve hızlı bir şekilde yenileyebilecek bir potansiyele sahiptir (Roy ve Dutta, 2022). CBF yönteminin dezavantajı arasında, geribildirimin zor olduğu alanlarda veri toplanmasında yaşanan güçlükler nedeniyle öğelerdeki eksik bilgi probleminin çözülememesi CBF yöntemi ile geliştirilen tavsiye sistemleri uygulamalarında performans düşüklüğüne neden olabilir (Roy ve Dutta, 2022). Şekil 3.3.'de CBF yönteminin çalışma mantığı verilmiştir. Şekil 3.3.'e göre CBF, kullanıcının geçmiş davranışlarından ve öğelere ait içerik bilgisinden yararlanarak Film A'yı izleyen ilgili

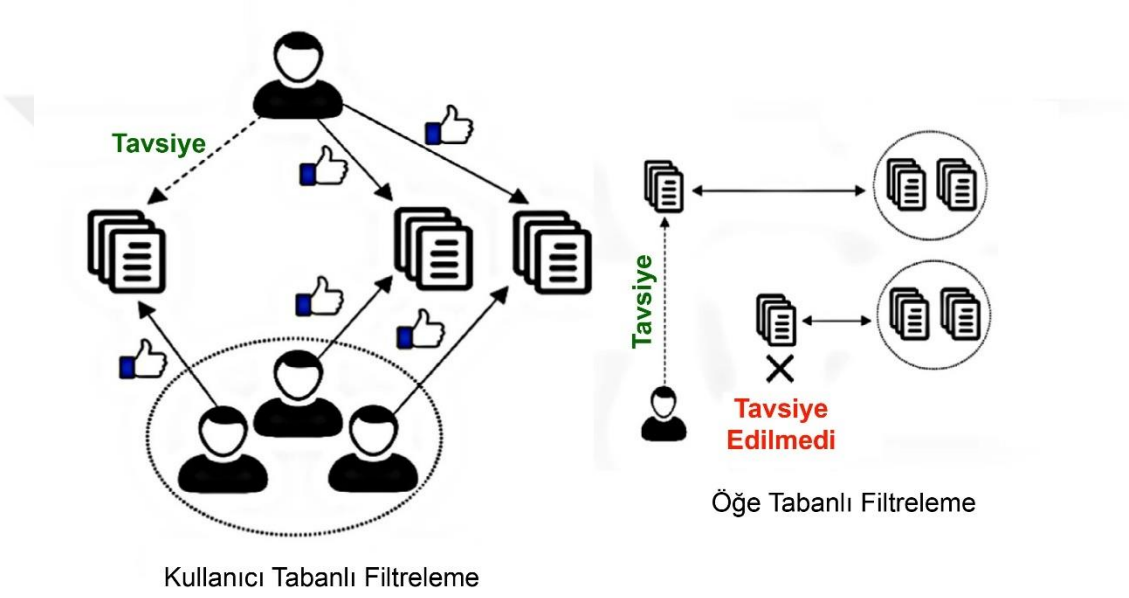
kullanıcıya Film A'ya ait özniteliklere benzer içeriğe sahip Film B'yi ilgili kullanıcıya önermektedir.



Şekil 3.3. İçerik tabanlı filtreleme (Roy ve Dutta (2022) çalışmasından uyarlandı)

İşbirlikçi filtreleme: Bu yöntemde, kullanıcılar (kullanıcı benzerliği) ve öğeler (öğe benzerliği) arasındaki benzerlik ölçüsü kullanılır (Roy ve Dutta, 2022). Diğer kullanıcıların görüşlerinin, tercihlerinin dikkate alındığı CF yaklaşımın temeli kullanıcı veya öğelere en yakın kullanıcılar veya öğeler tespit edilmesi, ardından ilgili kullanıcıya en uygun öğelerin önerilmesine dayanmaktadır. CF yöntemin avantajları arasında; öğelere ait yeterli içerik bilgisi bulunmadığı durumlarda da başarılı sonuçlar üretebilmesi, diğer kullanıcıların tercihleri dikkate alındığından ürünlerin gerçek kalite değerlendirilmesinde önemli bir yaklaşım olması, yeni öğeleri keşfederek ilgili kullanıcıların ilgi alanlarının genişlenmesine yardımcı olması yer almaktadır (Zhu ve ark., 2021b; Roy ve Dutta, 2022; Papadakis ve ark., 2022; Ke ve ark., 2021). CF yönteminde yaşanan başlıca sorunlar; diğer kullanıcıların tercihleri dikkate alındığından kişisel veri paylaşımından dolayı ortaya çıkabilecek veri gizliliği ve

güvenliği sorunlarını, veri seyrekliği, ölçeklenebilirlik, soğuk başlatma, kararsız kullanıcılar (gray sheep) problemleri yer almaktadır (Roy ve Dutta, 2022). Şekil 3.4.'de CF yönteminin çalışma mantığı verilmiştir. İçerik tabanlı filtreleme tekniği ile öğe tabanlı işbirlikçi filtreleme tekniği arasındaki fark; içerik tabanı filtreleme tekniği öğelerin meta verilerinden yararlanarak öğeler önerirken, öğe tabanlı işbirlikçi filtreleme tekniğinde ilgili kullanıcının satın aldığı ürünler, incelediği içerikler ile birlikte diğer kullanıcıların tercihleri dikkate alarak öneriler sunulur. Öğe tabanlı işbirlikçi filtreleme tekniğinde öğelerin metaverileri dikkate alınmaz.



Şekil 3.4. İşbirlikçi filtreleme (Roy ve Dutta (2022) çalışmasından uyarlandı)

Hibrit filtreleme: HF yönteminde amaç, kişisel tavsiye sistemleri uygulamalarının performansını geliştirmek için birden fazla tekniğinin bir arada kullanılmasıdır (Nápoles ve ark., 2020). Yöntemin avantajları arasında; genellikle birçok tavsiye sistemi uygulamalarında doğruluk ve performans artışını sağlayabilmesi, tavsiye sistemi uygulamalarındaki problemlerden veri seyrekliği, ölçeklenebilirlik, soğuk başlatma gibi sorunların çözümünde avantaj sağlaması yer almaktadır (Zhu ve ark., 2021b). Bu yöntemde, birden fazla modelin birleştirilmesi sonucunda model yapıları nedeniyle artan hesaplama karmaşıklığı hibrit modellerin dezavantajını oluşturabilir (Zhu ve ark., 2021b).

Demografik filtreleme: Bu yöntemde, kullanıcı kişisel özelliklerine (örneğin cinsiyet, yaş, ülke, eğitim vs.) göre sahip kullanıcıların ortak görüşleri, tercihlerine göre önerilerde bulunma amaçlanmıştır (Chicaiza ve Valdiviezo-Diaz, 2021). Yöntemin avantajı arasında karmaşık yapıdaki bilgilere ihtiyaç duymaması yer alırken, dezavantajları arasında, kişisel kullanıcı verilerin toplanması sırasında eksik bilgilerin girilmesi, kullanıcı kişisel verilerin toplanması sonucunda yaşanabilecek güvenlik ve gizlilik sorunları, filtrelenme işleminde kullanılan genel veriler dolayısıyla öneri listelerin genel içeriklere sahip olması yer alabilir (Chicaiza ve Valdiviezo-Diaz, 2021).

Bilgi tabanlı filtreleme: Bu yöntemde, kullanıcı tercihleri ve tercih edilen öğeler arasındaki ilişkiler belirlenir ardından elde edilen bilgiler ile alan bilgilerine dayanarak kullanıcılara öğeler tavsiye edilir (Chicaiza ve Valdiviezo-Diaz, 2021). KBF uygulamasında kullanıcı hakkında, öğe hakkında ve kullanıcı-öğe arasındaki ilişkilerden yararlanmaktadır (Chicaiza ve Valdiviezo-Diaz, 2021).

Matris çarpanlara ayırma yöntemi: MF yöntemi tavsiye sistemi kullanıcı derecelendirme matrisi tahmin uygulamasında önerilmiştir. Burada, derecelendirme matrisi kullanıcı matrisi ve öğe matrislerinin iç çarpımından oluşmaktadır (Zhang ve ark., 2021).

3.1.2. Tavsiye sistemi benzerlik ölçütleri

Tavsiye sistemleri uygulamalarında kullanılan benzerlik ölçütleri arasında; kosinüs benzerliği, pearson korelasyonu, öklid benzerliği, Manhattan benzerliği, jaccard indeksi, kendall’s tau korelasyonu, vektör benzerliği, dengelenmiş sıra indeksi, Spearman sıra-düzen korelasyon katsayısı, dengelenmiş karşılıklı bilgi, Chebyshev benzerliği yer almaktadır. Tez kapsamında CBF ve CF filtreleme teknikleri ile geliştirilen tavsiye sistemi uygulamasında kosinüs benzerliği yönteminden yararlanılmıştır.

Kosinüs benzerliği, çok boyutlu uzayda iki vektör arasındaki açının kosinüsüdür. Burada her bir vektör kullanıcıların önceki değerlendirmelerinden oluşan öznitelik vektörlerinden oluşmaktadır (Ray ve ark., 2021). Vektörler arasındaki açı azaldıkça benzerlik oranı artmaktadır. A ve B kullanıcıları arasındaki kosinüs benzerliği aşağıdaki gibi hesaplanır (Denklem 3.1).

$$\text{sim}\,(\mathrm{A},\mathrm{B}) = \cos\,(\Theta) = \frac{\mathrm{A}\,.\,\mathrm{B}}{\|\mathrm{A}\|\,.\,\|\mathrm{B}\|} = \frac{\sum_{i=1}^{n} \mathrm{A_i B_i}}{\sqrt{\sum_{i=1}^{n} \mathrm{A_i^2}}\,\sqrt{\sum_{i=1}^{n} \mathrm{B_i^2}}} \qquad (3.1)$$

Denklem (3.1)'de A ve B farklı iki kullanıcıyı temsil eden çok boyutlu vektör, "." vektörel çarpımı, $\|A\|$ ve $\|B\|$ sırasıyla A ve B vektörlerinin uzunluğunu ifade eder.

3.1.3. Tavsiye sistemi problemleri

Tavsiye sistemi öneri listeleri oluşturma veya tahmin sonuçları, veri toplama ve model oluşturma sürecinden oluşmaktadır. Burada veri toplama aşamasında; geri bildirimin zor olduğu alanlarda veri kümelerinde kullanıcı ve öğeler ile ilgili eksik bilgilerin yer alması, tavsiye sistemlerinde derecelendirme ve değerlendirmelerinde yaşanabilecek manipülasyonlar tavsiye sistemi uygulamalarının doğruluğunu etkilemektedir. Tavsiye sistemi uygulamalarında model oluşturma sürecinde genellikle seyreklik (sparsity), ölçeklenebilirlik (scalability), soğuk başlatma (cold start), kararsız kullanıcılar (gray sheep), benzerlik (synonymy), yanlış yönlendirme (shilling attacks), gecikme (latency), popülerlik yanlılığı (popularity bias) gibi problemler ile karşılaşılmaktadır (Roy ve Dutta, 2022; Nguyen ve ark., 2020; Joy ve Pillai, 2021).

Seyreklik (Roy ve Dutta, 2022; Nguyen ve ark., 2020; Joy ve Pillai, 2021) – Seyreklik sorunu, veri kümelerindeki beklenen değerlerin eksik olması nedeniyle oraya çıkmaktadır. Genellikle büyük boyutlu veri kümeleriyle gerçekleştirilen veri analizlerinde karşılaşılan bir sorundur. Tavsiye sistemlerinde, kullanıcı derecelendirmelerin yetersiz olduğu durumlarda ortaya çıkmaktadır. Yaşanan sorun

tavsiye sistemlerinin öneri doğruluğunu olumsuz etkilemektedir. Seyreklik sorunun çözümünde genellikle CF, DF, HF, SVD teknikleri uygulanmaktadır.

Ölçeklenebilirlik (Roy ve Dutta, 2022; Nguyen ve ark., 2020) - Bu sorun, tavsiye sistemi uygulamalarında girdi veri miktarının hızla artması sonucunda ortaya çıkmaktadır. Son yıllarda, veride yaşanan üstel büyüme sorunun tavsiye sistemlerinde yaygınlaşmasına neden olmaktadır. Ölçeklenebilirlik probleminin çözümünde genellikle boyut azaltma ve kümeleme yöntemleri uygulanmaktadır.

Soğuk başlatma (Roy ve Dutta, 2022; Joy ve Pillai, 2021) - Bu sorun, tavsiye sistemi uygulamalarında yeni kullanıcı/öğe eklenmesi sonucunda ortaya çıkmaktadır. Sorunun çözümü için yeni kullanıcılardan öğe tercihleri ve öğe derecelendirmeleri ile ilgili geri bildirimler alınabilir, ilgili kullanıcılardan demografik bilgileri toplanabilir elde edilen bilgiler yardımıyla öneriler sunulabilir, popüler öğeler önerilebilir, CBF, HF yöntemleri uygulanabilir.

Kararsız kullanıcılar (Roy ve Dutta, 2022) - CF tekniğinde yaşanan gray sheep sorunu, ilgili kullanıcıya benzer hiçbir kullanıcının bulunmaması/eşlenmemesi sonucunda ortaya çıkmaktadır. Sorunun çözümünde genellikle, kullanıcı profili ve öğe özelliklerinin kullanımı ile CBF tekniği uygulanmaktadır.

Benzerlik (Roy ve Dutta, 2022) - Benzer öğelerin sistem üzerinde farklı tanımlanmaları sonucunda ortaya çıkan benzerlik problemi, tavsiye sistemi uygulamalarının karar verme sürecinde doğruluğunu etkileyebilir. Sorunun çözümünde DF, SVD teknikleri uygulanabilir.

Yanlış yönlendirme (Roy ve Dutta, 2022) - Bu sorun, sahte profil hesaplarına sahip kişilerin tavsiye sistemi uygulamalarına yaptığı saldırıların sonucunda ortaya çıkmaktadır. Burada, tavsiye sistemlerinde yapılan saldırılarda bazı ürünlerin popülaritesi artabilir veya azalabilir. Bu durum sonucunda, kişisel ürün önermede sistemlerin güvenirliliği azalmaktadır. Yaşanan sorunun çözümünde sahte kullanıcı

profillerin ve saldırganların tespit edilip sistemden derecelendirme bilgileri ile birlikte kaldırılması gerekmektedir.

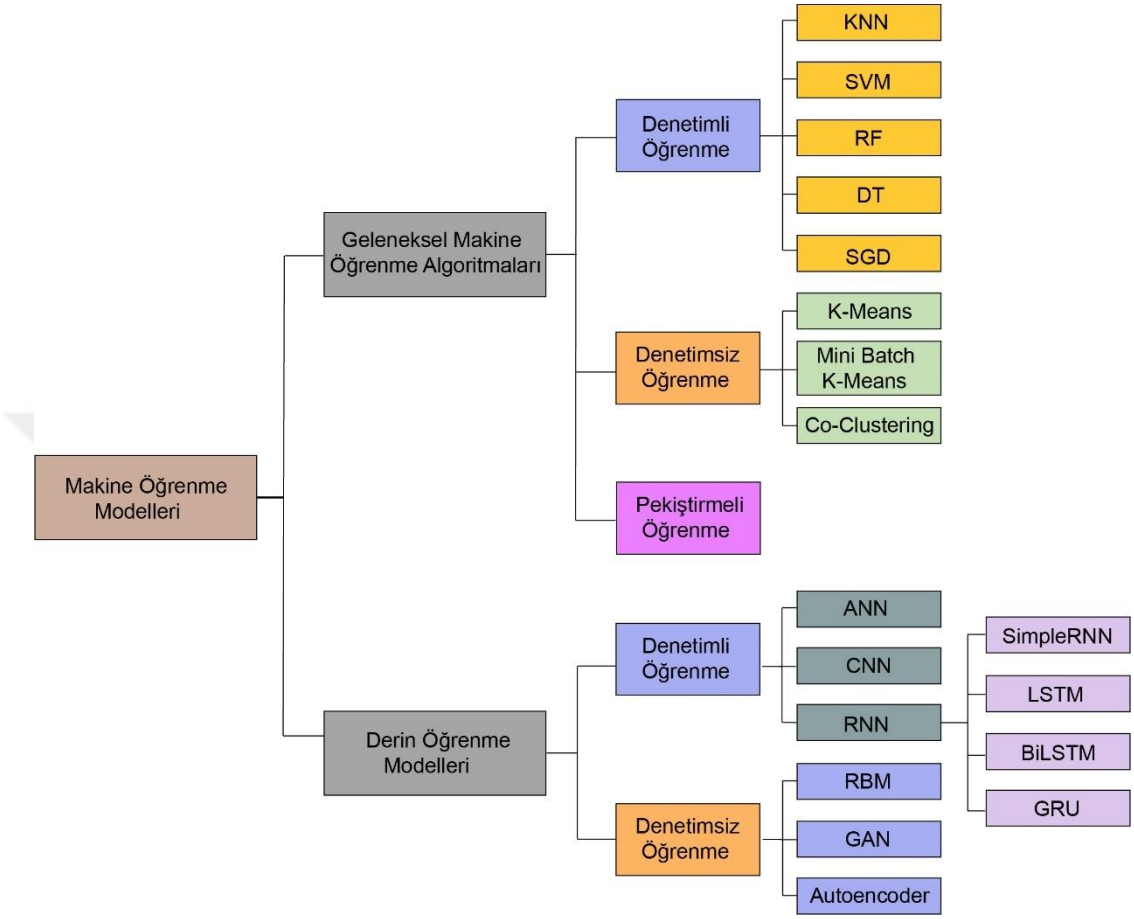
Gecikme (Roy ve Dutta, 2022) - CF tekniği ile geliştirilen tavsiye sistemleri uygulamalarında, veri tabanına yeni ürünlerin sık sık eklenmesi sonucunda gecikme sorunu ortaya çıkmaktadır. Burada eklenen yeni öğelerin derecelendirme bilgisine sahip olmaması dolayısıyla ilgili ürünler öneri listelerinde yer alamamaktadır. Yaşanan sorunu çözmek ve uygulamaların performansını artırmak için kümeleme teknikleri, CBF tekniği uygulanabilir.

Popüler yanlılığı (Yalçın ve Bilge, 2022; Kowald ve Lacic, 2022) - CF tekniğinde yaşanan problemlerden biridir: popüler öğeler popüler olmayan öğelere göre daha çok öne çıkarılır ve ilgili popüler öğeler aşırı tavsiye edilir. Burada ilgili kullanıcıların ilgilenebilecekleri ancak daha az popüler olması nedeniyle daha az tavsiye edilen öğeler doğru öneri listelerin oluşturulmasında dezavantaj oluşturabilir. Tavsiye uygulamalarında önerilen öğelerin görünürlüğünü artırmak amacıyla sahte hesaplar tarafından, belirli öğelerin derecelendirmeleri artırabilir, sahte incelemeler eklenebilir. Bu durum tavsiye sistemi uygulamalarının doğruluğunu olumsuz etkileyecektir. Yaşanan sorunu çözmek için kümeleme tekniği, makine öğrenme algoritmaları uygulanabilir.

3.2. Makine Öğrenmesi

Yapay zeka teknolojilerinden makine öğrenmesi, yapılandırılmış, yapılandırılmamış homojen, heterojen veri kümelerinden öğrenen, gizli örüntüleri/bilgileri keşfinde, sınıflandırma, regresyon, kümeleme, boyut azaltma gibi birçok uygulamada sıklıkla kullanılan tahmine dayalı matematiksel bir modelleme tekniğidir (Khanal ve ark., 2020). Makine öğrenme algoritmaları bilgi edinme veya problem çözme sürecinde doğrudan veri kümelerinden yararlanmaktadır. Makine öğrenmesi genel olarak denetimli, denetimsiz, pekiştirmeli öğrenme olmak üzere üç başlık altında toplanmaktadır. Şekil 3.5.'de makine öğrenme modellerinin alt başlıkları verilmiştir. Şekil 3.5.'de artificial neural network (yapay sinir ağları - ANN) ve

restricted boltzmann machines (kısıtlı boltzmann makineleri - RBM) temsil etmektedir.



Şekil 3.5. Makine öğrenme modelleri (Liu ve ark. (2019) çalışmasından uyarlandı)

3.2.1. Denetimli öğrenme

Denetimli öğrenme, etiketlenmiş veri kümelerinden özniteliklerinin/örüntülerin öğrenmesinde önerilen bir makine öğrenme teknolojisidir (Portugal ve ark., 2018). Bu yöntemde, girdi vektörleri ve her bir girdi vektörüne ait çıktı vektörü ile öğrenme gerçekleşmektedir (Portugal ve ark., 2018). Girdi-çıktı çiftlerine göre eğitilen makine öğrenme algoritmaları test aşamasında test için ayrılan veri kümeleri ile sınıflandırma, regresyon, tahmin süreçlerinde kullanılmaktadır (Portugal ve ark., 2018). Çalışma kapsamında, geliştirilen tavsiye sistemi derecelendirme uygulamasında denetimli makine öğrenme algoritmalarından KNN, SVM, RF, DT, SGD yöntemleri

kullanılmıştır. Algoritmalarda kullanılan hiper-parametreler Çizelge 3.1.'de gösterilmiştir.

Çizelge 3.1. Denetimli öğrenme algoritmalarına ait hiper-parametreler

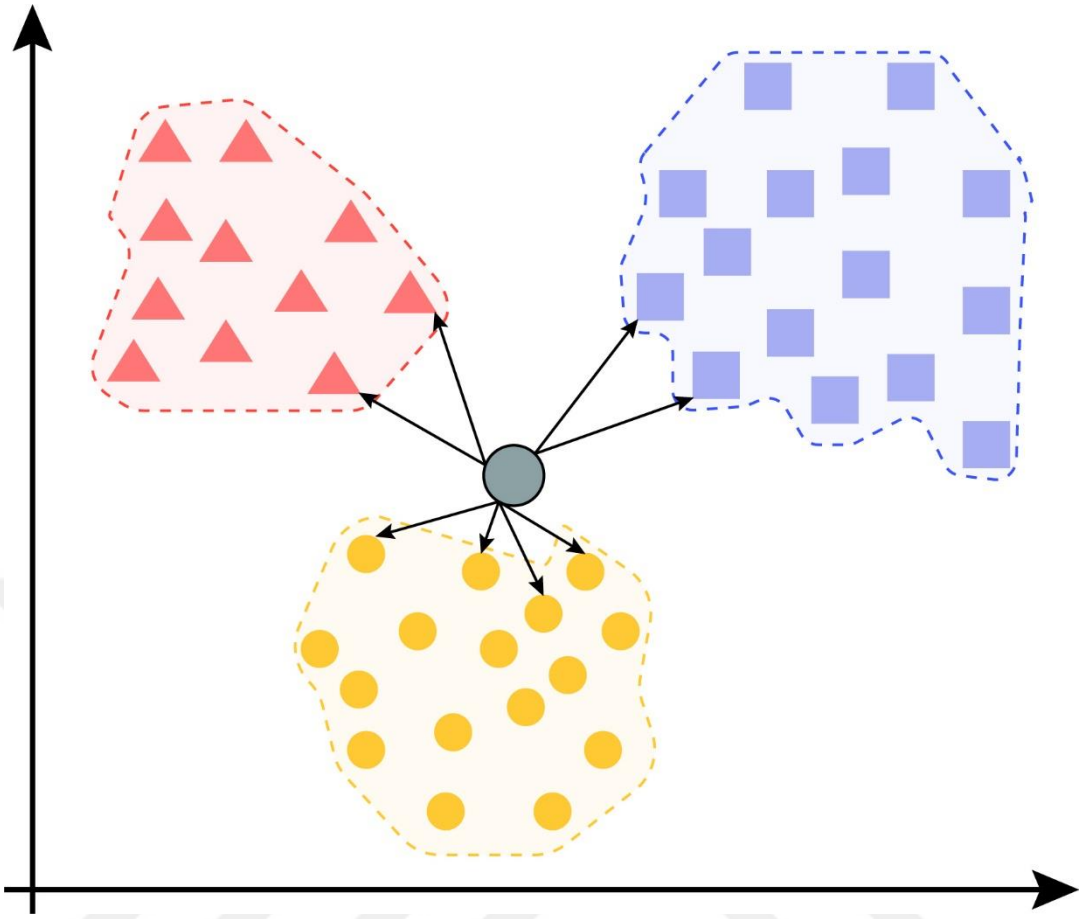
Denetimli makine öğrenme algoritması	**Hiper-parametreler**	**Açıklama**	**Parametre**
KNN	Neighbors	Komşu sayısı	11
	Weights	Ağırlık fonksiyonu	Distance
	Algorithm	Komşu seçiminde kullanılan hesaplama için kullanılan algoritma	Auto
	Leaf size	Yaprak boyutu	50
	P	Mesafe ölçüm metriğin belirlenmesi	2
	Metric	Mesafe metriği	Euclidean
SVR	Gamma	Çekirdek katsayısı	0.1
	C	Düzenlileştirme parametresi	10
	Kernel	SVR çekirdek türü	RBF
	Epsilon	Tolerans marjı	0.2
RF	Estimators	Ağaç sayısı	140
	Bootstrap	Örneklem oluşturma	True
	Criterion	Bölünmede kullanılan ölçüt	Absolute Error
	Max features	En iyi bölünme için kullanılması gereken maximum özellik sayısı	log 2
DT	Criterion	Bölünmede kullanılan ölçüt	Absolute Error
	Splitter	Her düğümdeki bölme stratejisi	Best
	Max features	Maximum özellik sayısı	log 2
	Random state	Örneklemin belirlenmesi	0
SGD	Penalty	Düzenlileştirme algoritması	L2
	Loss	Kayıp	squared_error
	Alpha	Düzenlileştirme katsayısı	0.01
	Early stopping	Eğitim sürecini erken sonlandırma	True

3.2.1.1 K-en yakın komşuluk (K-nearest neighbors)

KNN algoritması, sınıflandırma ve regresyon problemlerinde en çok kullanılan makine öğrenme algoritmalarından biridir. KNN yönteminde, nesnelerin sahip olduğu özelliklere göre sınıflara ayrılır (Aha ve ark., 1991). Burada, her bir nesne ile k komşu değeri kadar kendisine en yakın özellikler/komşular arasında uzaklıklar hesaplanır ardından sınıflara ayrılır (Aha ve ark., 1991). Uzaklık hesaplama işleminde genellikle öklid fonksiyonu kullanılır. Ayrıca Minkowski, Manhattan, Jacckard indeksi, Hamming, Mahalanobis gibi uzaklık matrisleri de kullanılabilir. KNN algoritmasının temel çalışma mantığı Şekil 3.6.'da verilmiştir. Tez kapsamında, KNN algoritması ile önerilen tavsiye sistemi derecelendirme uygulamasında, Scikit-learn makine öğrenme kütüphanesi Neighbors modülündeki KNeighborsRegressor fonksiyonu kullanılmıştır. KNN algoritmasında kullanılan hiper-parametreler Çizelge 3.1.'de verilmiştir. Çizelge 3.1.'e göre deneysel sonuçlar sonucunda KNN algoritmasında uzaklık hesaplama işleminde öklid uzaklığı, KNN algoritmasında k komşu sayısı 11 olarak belirlenmiştir. Öklid uzaklığı aşağıdaki gibi hesaplanır (Denklem 3.2).

$$d\left(x_i, x_j\right) = \sqrt{\sum_{k=1}^{m}\left(x_{ik} - x_{jk}\right)^2} \tag{3.2}$$

Denklem (3.2)'de $d\left(x_i, x_j\right)$ uzaklığı, x_i $(x_{i1}, x_{i2}, \ldots, x_{im})$ ve x_j $(x_{j1}, x_{j2}, \ldots, x_{jm})$ veri örneklerini temsil etmektedir.

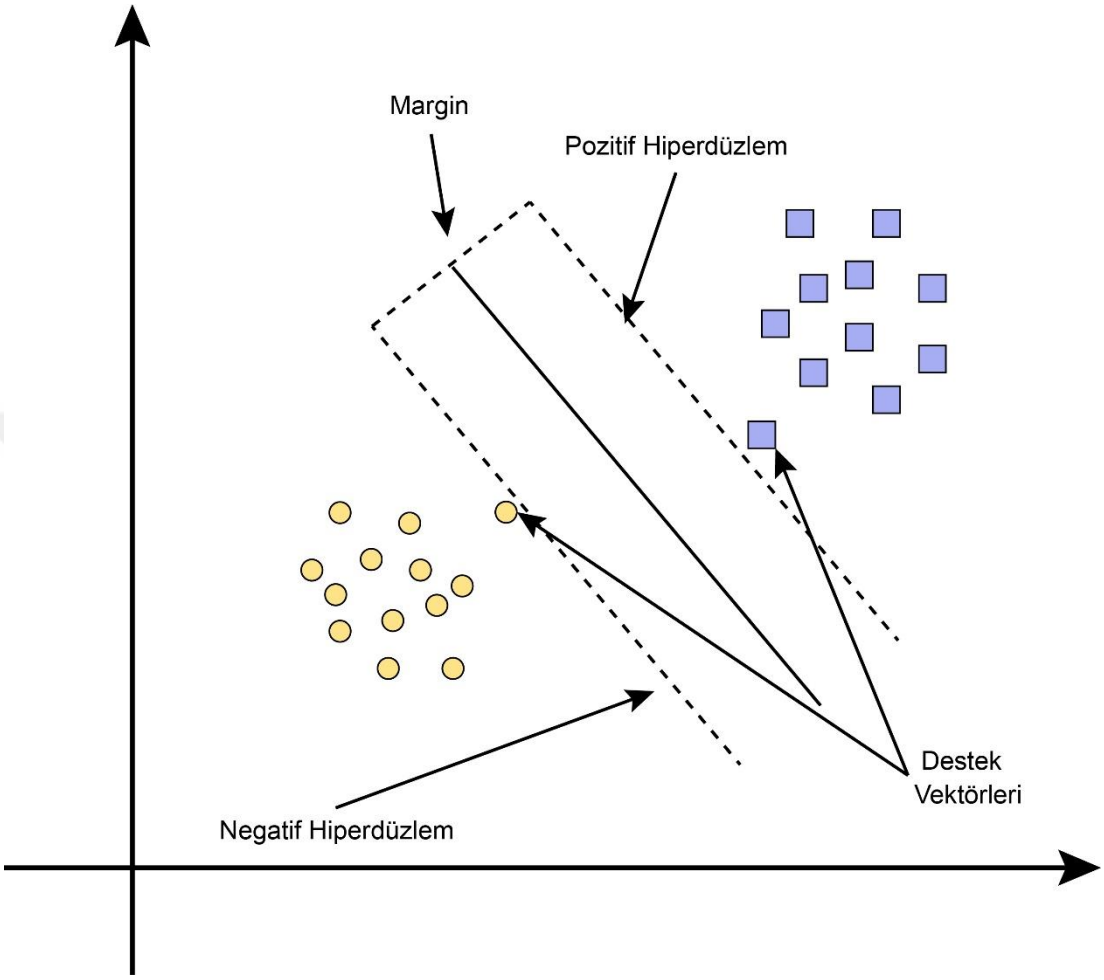


Şekil 3.6. KNN algoritması (Asharf ve ark. (2020) çalışmasından uyarlandı)

3.2.1.2 Destek vektör makinesi (support vector machine)

Vapnik tarafından önerilen SVM, popüler denetimli makine öğrenme tekniklerinden biridir (Vapnik, 1999). SVM algoritması ile amaçlanan; veri setindeki sınıflar arasını en geniş aralık ile en uygun şekilde ayırmaktır (Cervantes ve ark., 2020). Burada SVM, veri seti özelliklerini/noktaları bir doğru parçası/hiperdüzlem ile ayırır ve karar sınırları belirler (Cervantes ve ark., 2020). SVM doğrusal, doğrusal olmayan birçok probleme uygulanabilir. SVM algoritmasının temel çalışma mantığı Şekil 3.7.'de verilmiştir. Tez kapsamında SVM algoritması ile önerilen tavsiye sistemi derecelendirme uygulamasında, Scikit-learn makine öğrenme kütüphanesi SVM modülündeki SVR fonksiyonu kullanılmıştır. SVR algoritmasında kullanılan hiper-parametreler Çizelge 3.1.'de verilmiştir. Çizelge 3.1.'e göre deneysel sonuçlar

sonucunda SVM algoritmasındaki C, gamma, kernel, epsilon değeri sırasıyla 10, 0.1, rbf, 0.2 olarak belirlenmiştir.



Şekil 3.7. SVM algoritması (Toledo-Pérez ve ark. (2019) çalışmasından uyarlandı)

3.2.1.3 Rastgele orman (random forest)

Breiman (2001) önerdiği denetimli makine öğrenme algoritmalarından RF, mimarinin temeli oluşturan çok sayıdaki karar ağaçlarından oluşmaktadır. Bir karar ağacı kök düğümü ve düğümlerden oluşmaktadır. RF algoritmasında ağaçlardaki kök düğümünün bulunması ve düğümlerdeki bölme işlemleri rastgele gerçekleşmektedir (Chen ve ark., 2021a). RF algoritması ile geliştirilen uygulama sonucunda rastgele bir orman oluşmaktadır. Burada algoritmadaki ağaç sayısı algoritmanın performansı ile doğru orantılıdır (Chen ve ark., 2021a). Ağaç sayısı artırıldıkça

sınıflandırma/regresyon performansı artabilir (Chen ve ark., 2021a). RF algoritması sınıflandırma ve regresyon problemlerinde sıklıkla kullanılmaktadır (Chen ve ark., 2021a). Veri kümelerindeki sınıfı oluşturan değişken, kategorik ise sınıflandırma işlemi uygulanır, sınıf değişkeni sürekli ise regresyon ağacı yöntemi uygulanır. Tez kapsamında RF algoritması ile önerilen tavsiye sistemi derecelendirme uygulamasında, Scikit-learn makine öğrenme kütüphanesi ensemble modülündeki RandomForestRegressor fonksiyonu kullanılmıştır. RandomForestRegressor algoritmasında kullanılan hiper-parametreler Çizelge 3.1.'de gösterilmiştir. Çizelge 3.1.'e göre deneysel sonuçlar sonucunda RF algoritmasındaki estimators, bootstrap, criterion, max features parametreleri sırasıyla 140, True, absolute_error, log2 olarak belirlenmiştir.

3.2.1.4 Karar ağaçları (decision tree)

Denetimli makine öğrenme algoritmalarından DT Quinlan (1986) tarafından önerilmiştir. Sınıflandırma ve regresyon problemlerine sıklıkla uygulanmaktadır. Karar ağacının temelini kök, dal, düğüm ve yaprak oluşturmaktadır. Karar ağaçlarının tasarlanma sürecindeki en önemli, adım ağaçtaki olası dallanmalar için gerekli kriterlerin belirlenme aşamasıdır (Sarker ve ark., 2019). Burada, entropi ve bilgi kazancı sıklıkla kullanılmaktadır (Sarker ve ark., 2019). Entropi ve bilgi kazancı yöntemleri ile geliştirilen algoritmalar arasında ID3, C4.5, C5.0 algoritmaları yer almaktadır (Sarker ve ark., 2019). Tez kapsamında, DT algoritması ile önerilen tavsiye sistemi derecelendirme uygulamasında, Scikit-learn makine öğrenme kütüphanesi tree modülündeki DecisionTreeRegressor fonksiyonu kullanılmıştır. DecisionTreeRegressor algoritmasında kullanılan hiper-parametreler Çizelge 3.1.'de sunulmuştur. Çizelge 3.1.'e göre deneysel sonuçlar sonucunda DT algoritmasındaki criterion, splitter, max features, random state parametreleri sırasıyla absolute error, best, log 2, 0 olarak belirlenmiştir.

3.2.1.5 Olasılıksal dereceli azalma (stochastic gradient descent)

Büyük veri kümelerinde başarılı sonuçlar üretebilen SGD algoritması, makine öğrenme uygulamalarında popüler bir yaklaşımdır (Neu, 2021). Ayrıca geri yayılım sürecinde gradyan hesaplamalarındaki verimliliği dolayısıyla derin sinir ağının eğitiminde önemli bir yöntemdir (Neu, 2021). Çalışmada, SGD algoritması ile önerilen tavsiye sistemi derecelendirme uygulamasında, Scikit-learn makine öğrenme kütüphanesi linear_model modülündeki SGDRegressor fonksiyonu kullanılmıştır. SGDRegressor algoritmasında kullanılan hiper-parametreler Çizelge 3.1.'de verilmiştir. Çizelge 3.1.'e göre deneysel sonuçlar sonucunda SGD algoritmasındaki penalty, alpha, early_stopping parametreleri sırasıyla l2, 0.01, True olarak belirlenmiştir.

3.2.2. Denetimsiz öğrenme

Denetimsiz öğrenme, etiketlenmemiş veri kümelerinden özniteliklerinin/örüntülerin öğrenmesinde önerilen bir makine öğrenme teknolojisidir (Portugal ve ark., 2018). Denetimsiz öğrenme yönteminde denetleyici bir mekanizma yoktur. Bu yöntemdeki amaç girdi veri kümelerinde yer alan ve önceden bilinmeyen gizli örüntüleri/benzerlik ilişkileri/düzenleri bulmak ve bulunan ilişkilerden yararlanarak, etiketlenmemiş girdi veri kümelerini etiket gruplarına dahil etmektir (Portugal ve ark., 2018). Çalışmada, geliştirilen tavsiye sistemi derecelendirme probleminde denetimsiz makine öğrenme algoritmalarından PCA yöntemi kullanılmıştır.

3.2.2.1 Temel bileşen analizi (principal component analysis)

PCA, birden fazla birbiriyle ilişkili özellikten oluşan veri kümelerinin boyutunu küçültmeyi sağlayan bir veri dönüştürme tekniğidir (Omuya ve ark., 2021). PCA uygulama sonucunda veriler daha az sayıdaki değişkenlerle ifade edilebilir (Omuya ve ark., 2021). Tez kapsamında PCA algoritması ile önerilen tavsiye sistemi derecelendirme uygulamasında, Scikit-learn makine öğrenme kütüphanesi

decomposition modülündeki PCA fonksiyonu kullanılmıştır. PCA algoritmasında n_components, svd_solver hiper-parametreleri kullanılmıştır. Parametre değerleri sırasıyla 18, auto olarak belirlenmiştir.

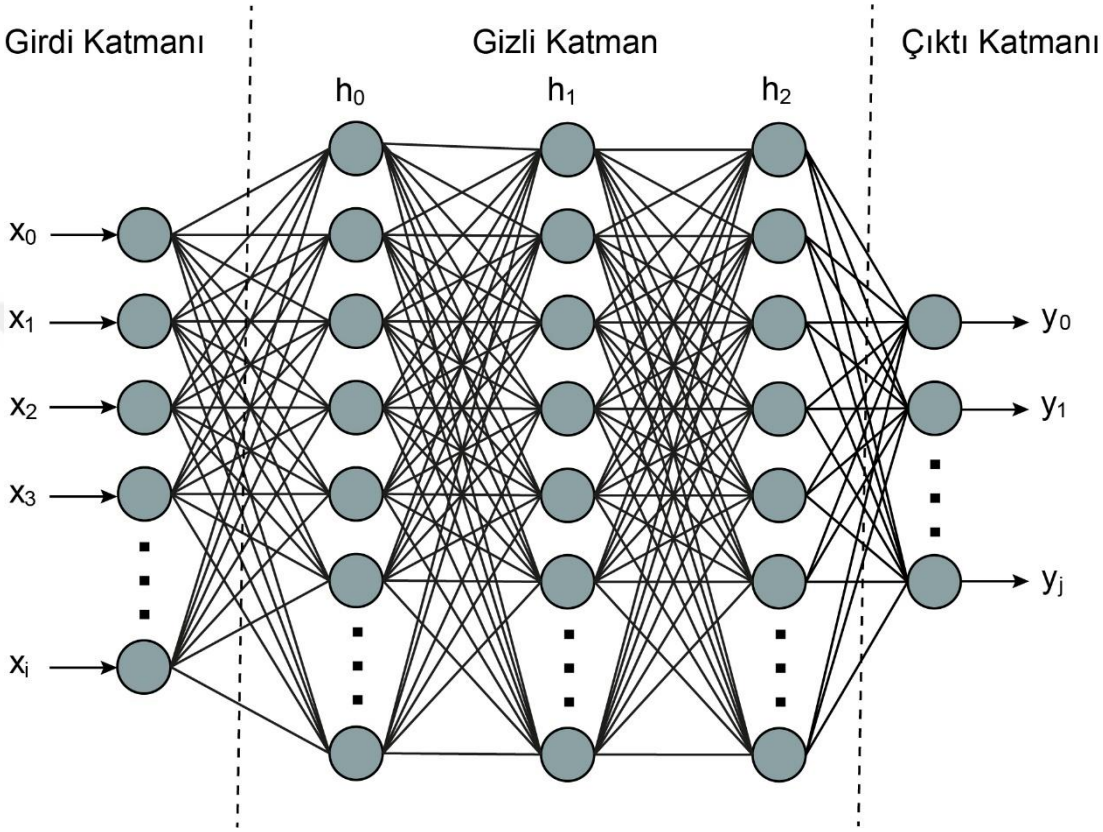
3.2.3. Pekiştirmeli öğrenme

Makine öğrenme yöntemlerinden pekiştirmeli öğrenme, denetimli ve denetimsiz öğrenmeden farklı olarak eğitim veri kümesi olmadan herhangi bir ortama yerleştirilen ajanın (mobil robotun) akıllı bir davranış oluşturmak için çevre ile etkileşimden yararlanarak deneme yanılma yaklaşımıyla eğitilmesi ve eğitim sonucunda en uygun karara ulaşılması sürecidir (Gronauer ve Diepold, 2022). Bu öğrenmede yönteminde ajanlar bulunduğu ortamda zaman içinde kendi başlarına öğrenme ve adapte olma yeteneğine sahiptirler (Gronauer ve Diepold, 2022).

3.3. Derin Öğrenme

Yapay zeka ve makine öğrenme alanlarının güncel uygulamalarından derin öğrenme, genel olarak yapay sinir ağlarına dayanmaktadır (Batmaz ve ark., 2019). Yapay sinir ağlarından farklı olarak derin öğrenme algoritmaları gizli katman ve nöron sayıları daha fazladır. Sinir ağı ve derin öğrenme teknolojisi yaklaşık olarak 50 yıldır olmasına rağmen derin sinir ağ modellerinin eğitimi için gereken büyük veri ve yüksek hesaplama gücüne sahip derin öğrenme teknolojisinin eğitimi için ihtiyaç duyulan donanım eksikliği nedeniyle derin sinir ağların etkinliği son on yılda ortaya çıkmaya başlamıştır (Batmaz ve ark., 2019). Burada, derin sinir ağlarının son 10 yılda aktif olarak kullanılmasının önünü açan gelişmeler arasında büyük veri kümelerinin toplanması ve GPU, CPU gibi yüksek işlem gereksinimlerini karşılayabilecek donanımların geliştirilmesi yer almaktadır (Batmaz ve ark., 2019). Son yıllarda derin öğrenme algoritmaları, büyük veri kümelerindeki karmaşık örüntülerin/yapıların keşfinde başarılı olması sayesinde birçok bilim ve işletme alanlarında uygulanmaktadır (LeCun ve ark., 2015). Derin sinir ağları görüntü işleme, ses tanıma, parçacık hızlandırıcı verileri analiz etmede, duygu analizi, çeviri, tavsiye sistemleri gibi birçok uygulamada sıklıkla kullanılmaktadır (LeCun ve ark., 2015).

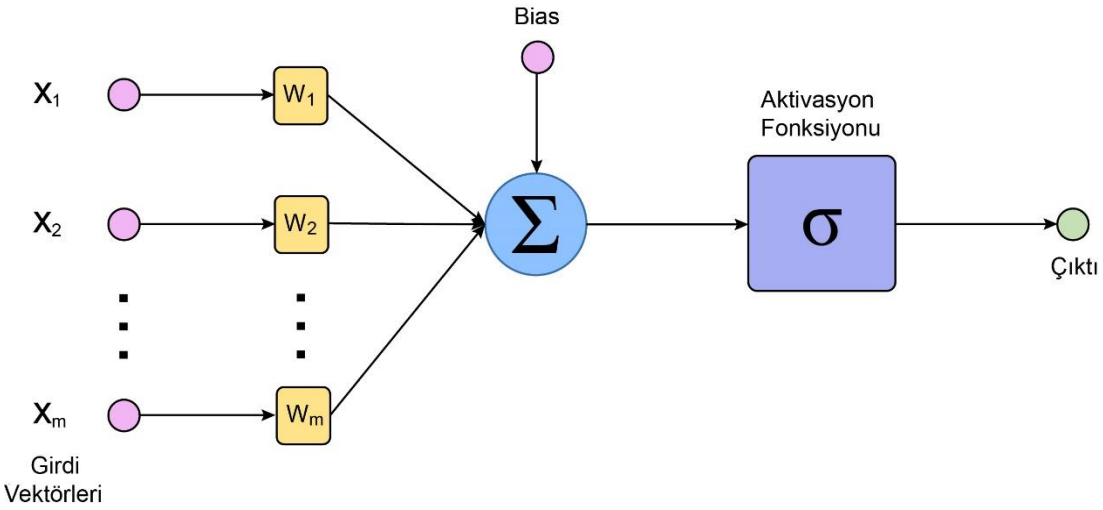
Yapay sinir ağları girdi, gizli ve çıktı katmanlarındaki nöronlardan oluşmaktadır (Şekil 3.8.). Burada, gizli katmandaki her bir nöron girdi bilgilerini önceki nöronun çıktı vektöründen oluşmaktadır. Genel olarak her bir nörondaki matematiksel işlem Denklem 3.3'de gösterilmiştir. Şekil 3.9.'da bir nöronun yapısı verilmiştir.



Şekil 3.8. İleri beslemeli yapay sinir ağı (Bre ve ark. (2018) çalışmasından uyarlandı)

$$y_k = \varphi\left(b + \sum_{i=0}^{m} w_i x_i\right) \tag{3.3}$$

Denklem (3.3)'de b bias, w ağırlık, x girdi vektörü, y çıktı vektörünü temsil etmektedir.



Şekil 3.9. Bir nöronun yapısı (Groensfelder ve ark. (2019) çalışmasından uyarlandı)

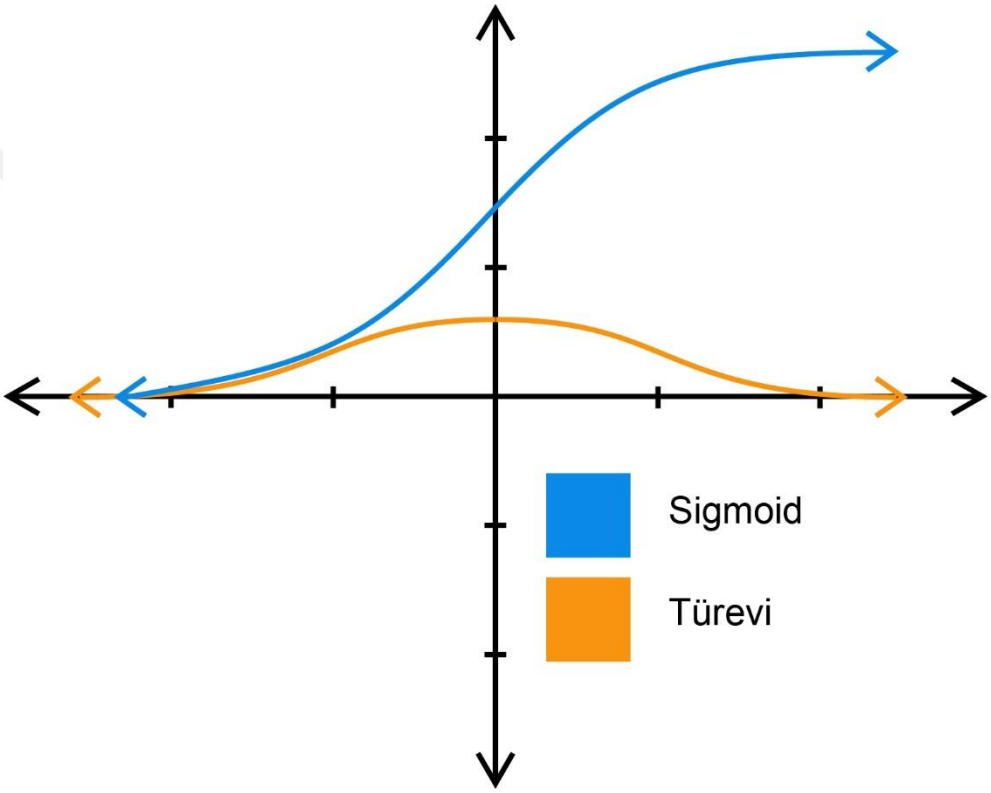
Şekil 3.9.'da bir nöronda gerçekleşen matematiksel işlem sürecinde; girdi vektörleri her bir girdi vektörüne ait ağırlık vektörü ile çarpılır ve bias değeri toplanır. Son olarak, aktivasyon fonksiyonu kullanılarak çıktı üretilir. Burada, yapay sinir ağlarındaki öğrenme işlemi ağırlık ve bias değerlerinin güncellenmesi ile gerçekleşmektedir. Ağırlık ve bias değişkenlerinin güncellenmesindeki amaç; yapay sinir ağındaki hatanın minimize edilmesidir. Yapay sinir ağlarında nöronlarda gerçekleşen matematiksel işlem ardında aktivasyon fonksiyonlarının kullanılmasının nedeni doğrusal olmayan karmaşık yapılı görüntü, sinyal, duygu analizi gibi karmaşık yapılı veri kümeleri ile öğrenme işleminin gerçekleşebilmesini sağlamaktadır. Yapay sinir ağlarında ağırlık vektörleri ile ilgili hata değerlerinin hesaplanması işlemi geri yayılım sürecinde gerçekleşmektedir. Burada geri yayılımda ağırlık güncellenmesinden sorumlu gradyanların, kaybolması problemini önlemek için aktivasyon fonksiyonları kullanılabilir. Derin sinir ağlarında sıklıkla kullanılan aktivasyon fonksiyonları arasında Sigmoid, hiperbolik tanjant (tanh), doğrultulmuş lineer birim (rectified linear unit - ReLU), sızıntılı doğrultulmuş lineer birim (leaky ReLU), kendinden geçitli (Swish), Softmax fonksiyonu yer almaktadır. Tez kapsamında önerilen derin sinir ağ mimarilerinde, ReLU aktivasyon fonksiyonu kullanılmıştır.

Sigmoid aktivasyon fonksiyonu: Sıklıkla kullanılan aktivasyon fonksiyonlarından sigmoid, verileri (0,1) aralığına dönüştürmektedir (Kandel ve

Castelli 2020). Sigmoid fonksiyonun matematiksel ifadesi Denklem 3.4'de, sigmoid fonksiyonun türev sonucundaki matematiksel ifadesi Denklem 3.5'de, fonksiyonun grafiği Şekil 3.10.'da verilmiştir.

$$\sigma\,(x) = \frac{1}{1 + e^{-x}} \tag{3.4}$$

$$\sigma'\,(x) = \sigma\,(x)\left(1 - \sigma\,(x)\right) = \frac{e^{-x}}{(1 + e^{-x})^2} \tag{3.5}$$



Şekil 3.10. Sigmoid aktivasyon fonksiyonu grafiği (Wang ve ark. (2022) çalışmasından uyarlandı)

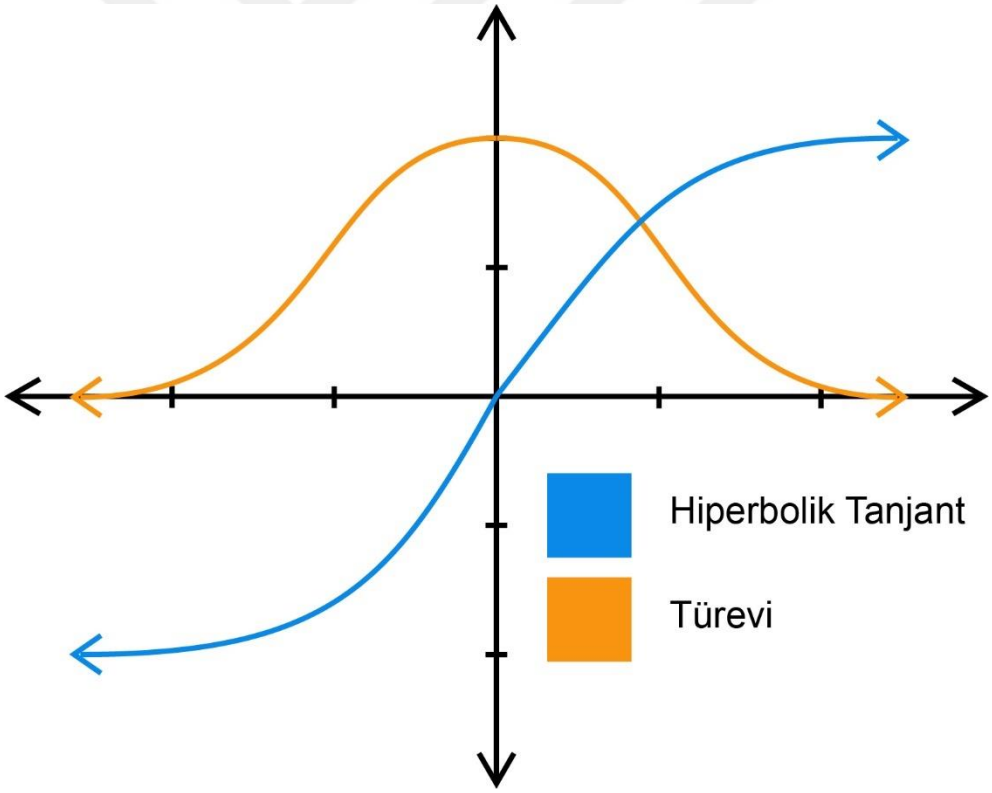
Şekil 3.10.'da sigmoid fonksiyonu grafiğine göre sigmoid fonksiyonun türevi uç noktalarda sıfıra yakınsamış bu durum geri yayılım sürecinde ağırlık vektörlerinin güncellenmesini sağlayan gradyanların kaybolmasına neden olabilir. Sinir ağlarında yaşanan gradyan kaybolması probleminin sonucunda ağının öğrenmesi durabilir.

Tanh aktivasyon fonksiyonu: Tanh fonksiyonu, sigmoid aktivasyon fonksiyonuna benzer bir yapıya sahiptir (Sewak ve ark., 2020). Sigmoid fonksiyonundan farklı olarak fonksiyonun değer aralığı (-1, +1) olarak tanımlanmış ve

orijin etrafında simetriktir (Kandel ve Castelli 2020; Sewak ve ark., 2020; Rahman ve ark., 2021). Ayrıca türev işlemi sonucunda sigmoid fonksiyonuna göre daha geniş değer aralığına sahip olması dolayısıyla öğrenme daha hızlı gerçekleşmektedir (Şekil 3.10. ve Şekil 3.11.). Türev sonucunda tanh fonksiyonu, uç noktalarda sıfıra yakınsar bu durum gradyan kaybolması problemine neden olmaktadır (Şekil 3.11.). Tanh fonksiyonun matematiksel ifadesi Denklem 3.6'da, tanh fonksiyonun türev sonucundaki matematiksel ifadesi Denklem 3.7'de, fonksiyonun grafiği Şekil 3.11.'de gösterilmiştir.

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.6}$$

$$f'(x) = 1 - f(x)^2 = \frac{4}{(e^{x} + e^{-x})^2} \tag{3.7}$$

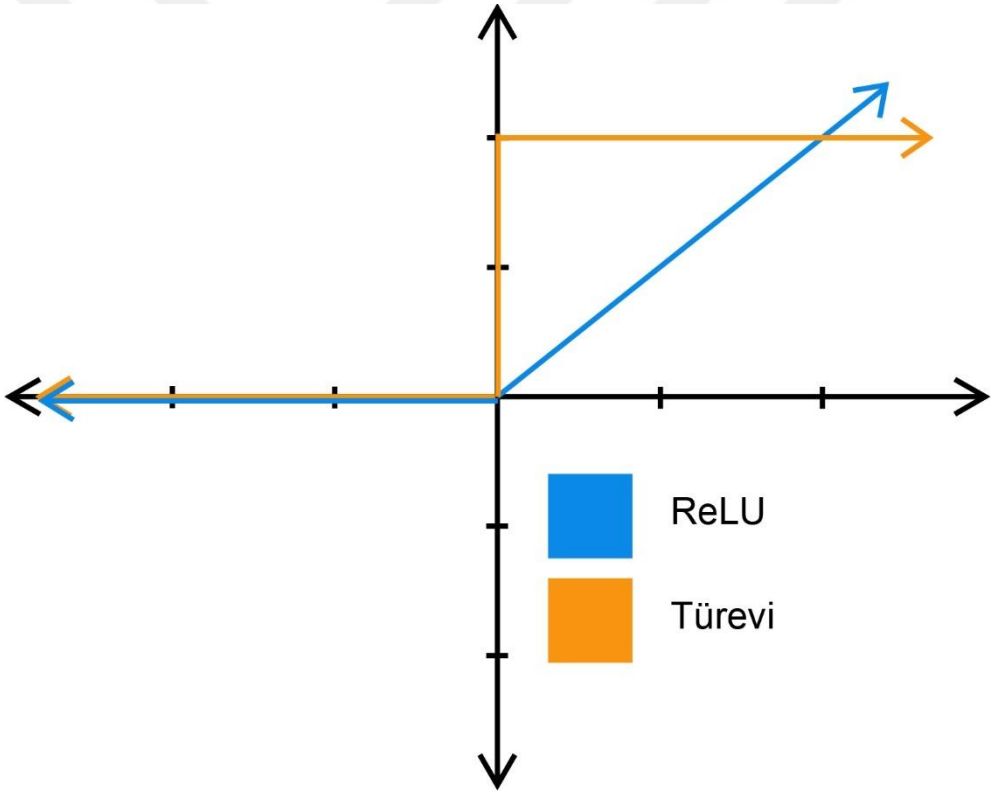


Şekil 3.11. Tanh aktivasyon fonksiyonu grafiği (Wang ve ark. (2022) çalışmasından uyarlandı)

ReLU aktivasyon fonksiyonu: Derin sinir ağlarında yaygın olarak kullanılan ReLU, doğrusal olmayan bir aktivasyon fonksiyonudur (Kandel ve Castelli 2020; Jahan ve ark., 2022). Fonksiyonun değer aralığı [0, +∞) olarak tanımlanmıştır. ReLU

aktivasyon fonksiyonu negatif eksende (x-ekseninde) sıfır değeri almaktadır (Sewak ve ark., 2020; Jahan ve ark., 2022) Bu durum sonucunda sinir ağlarındaki tüm nöronlar aynı anda etkin değildir (Jahan ve ark., 2022). Dolayısıyla bu durum ağın daha hızlı çalışmasına olanak tanıyacaktır (Rahman ve ark., 2021). Ancak ReLU aktivasyon fonksiyonun sıfır olduğu alanların türevi de sıfır olduğundan ağda gradyan kaybolması problemi yaşanmasına neden olmaktadır dolayısıyla burada öğrenme işlemi gerçekleşemeyecektir (Jahan ve ark., 2022). ReLU fonksiyonun matematiksel ifadesi Denklem 3.8'de, fonksiyonun grafiği Şekil 3.12.'de verilmiştir.

$$f(x) = \max(0, x) = \begin{cases} 0, & x < 0 \text{ ise} \\ x, & x \geq 0 \text{ ise} \end{cases} \quad (3.8)$$



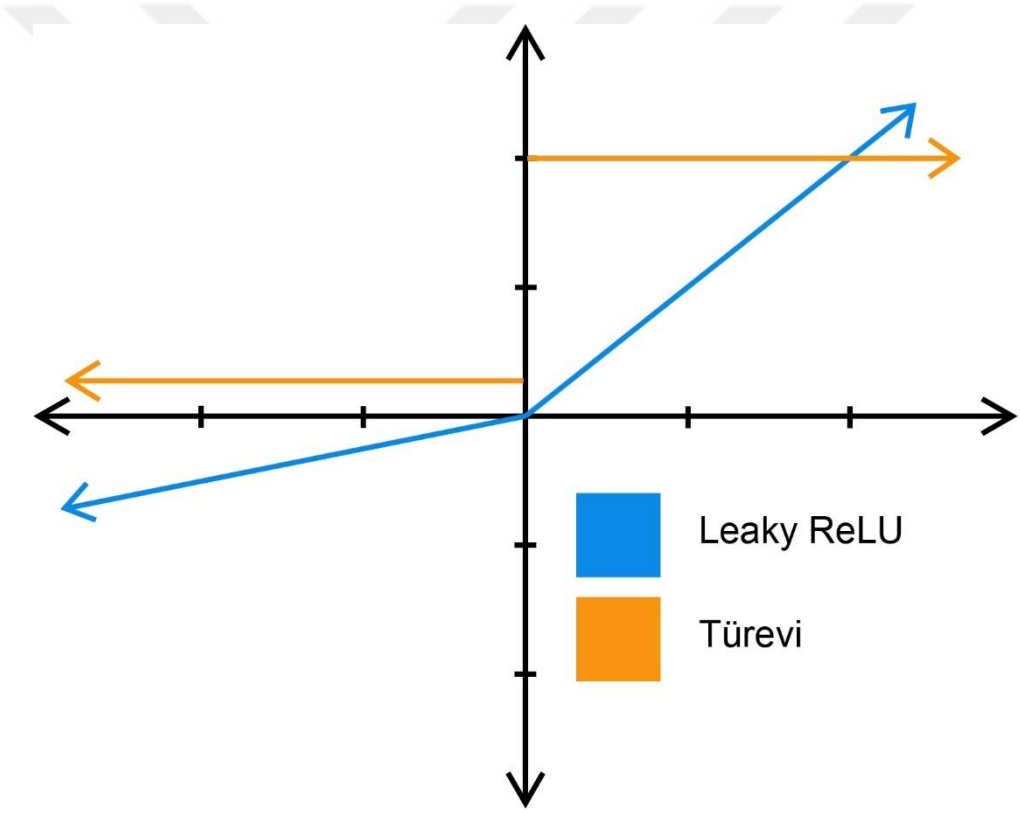
Şekil 3.12. ReLU aktivasyon fonksiyonu grafiği (Wang ve ark. (2022) çalışmasından uyarlandı)

Leaky ReLU aktivasyon fonksiyonu: ReLU aktivasyon fonksiyonunda negatif eksende yaşanan öğrenme probleminin çözümünde Leaky ReLU aktivasyon fonksiyonu önerilmiştir (Sewak ve ark., 2021; Jahan ve ark., 2022). Fonksiyonun değer aralığı (-∞, +∞) olarak tanımlanmıştır (Sewak ve ark., 2021). Leaky ReLU

aktivasyon fonksiyonu negatif alanda a sızıntı parametre değeri ile türevin sıfır olması engellenmiştir (Sewak ve ark., 2021). Bu durum sonucunda geri yayılım sürecinde negatif alanda da öğrenme işlemi gerçekleşmektedir. Leaky ReLU fonksiyonun matematiksel ifadesi Denklem 3.9'da, fonksiyonun grafiği Şekil 3.13.'de verilmiştir.

$$f(x) = \max(0, x) = \begin{cases} ax, & x < 0 \text{ ise} \\ x, & x \geq 0 \text{ ise} \end{cases} \quad (3.9)$$

Denklem (3.9)'da a sabit bir değerdir ve genellikle 0.01 değeri kullanılır (Sewak ve ark., 2021).



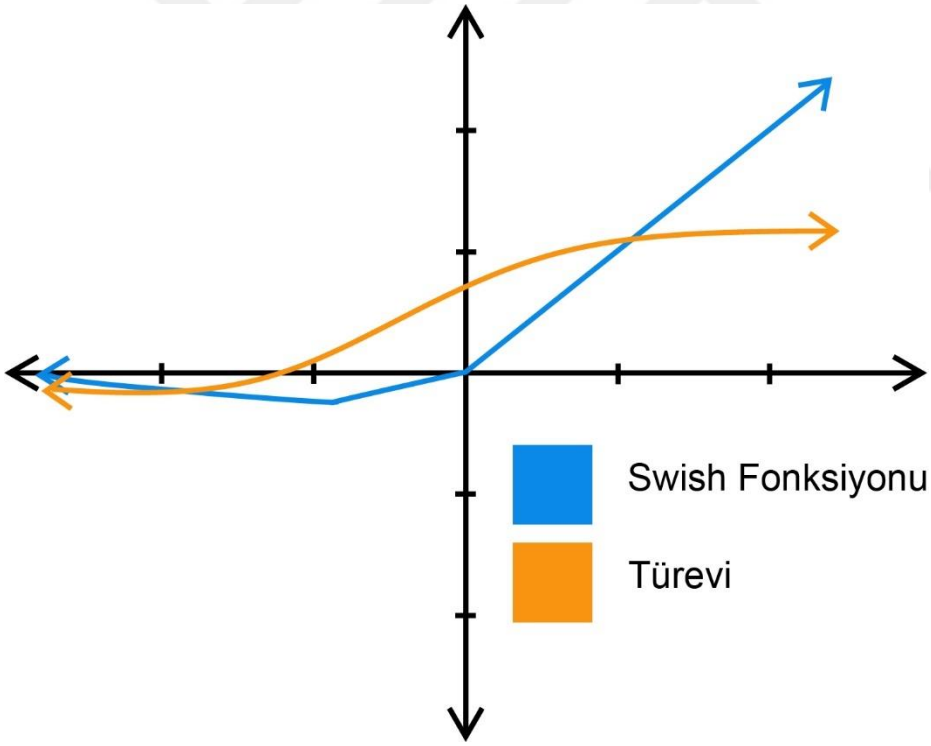
Şekil 3.13. Leaky ReLU aktivasyon fonksiyonu grafiği (Mastromichalakis (2020) çalışmasından uyarlandı)

Swish aktivasyon fonksiyonu: Google araştırmacıları tarafından önerilen Swish aktivasyon fonksiyonun değer aralığı (-∞, +∞) olarak tanımlanmıştır (Ramachandran ve ark., 2021). ReLU aktivasyon fonksiyonundan farklı olarak negatif alanda da değer almaktadır (Ramachandran ve ark., 2021). Swish fonksiyonun

matematiksel ifadesi Denklem 3.10'da, fonksiyonun grafiği Şekil 3.14.'de verilmiştir. Şekil 3.14. Zhu ve ark. (2021) "Logish: A new nonlinear nonmonotonic activation function for convolutional neural network" başlıklı makalesinden uyarlanmıştır.

$$f(x) = x \text{ sigmoid } (\beta x) = \frac{x}{1 + e^{-\beta x}} \tag{3.10}$$

Denklem (3.10)'da β değeri sabit bir değer veya öğrenilmiş bir parametre olabilir (Ramachandran ve ark., 2021). β=0 olursa f (x) doğrusal bir fonksiyon olur, $\beta \rightarrow \infty$ olursa sigmoid bileşeni 0-1 değer aralığına yaklaşır bu nedenle Swish fonksiyonu ReLU aktivasyon değerine benzer, β= 1 olursa düzgün enterpolasyon sağlanmış olur (Ramachandran ve ark., 2021). Enterpolasyonun derecesi, β öğrenilmiş bir parametre olarak ayarlanırsa, model tarafından kontrol edilebilir (Ramachandran ve ark., 2021).

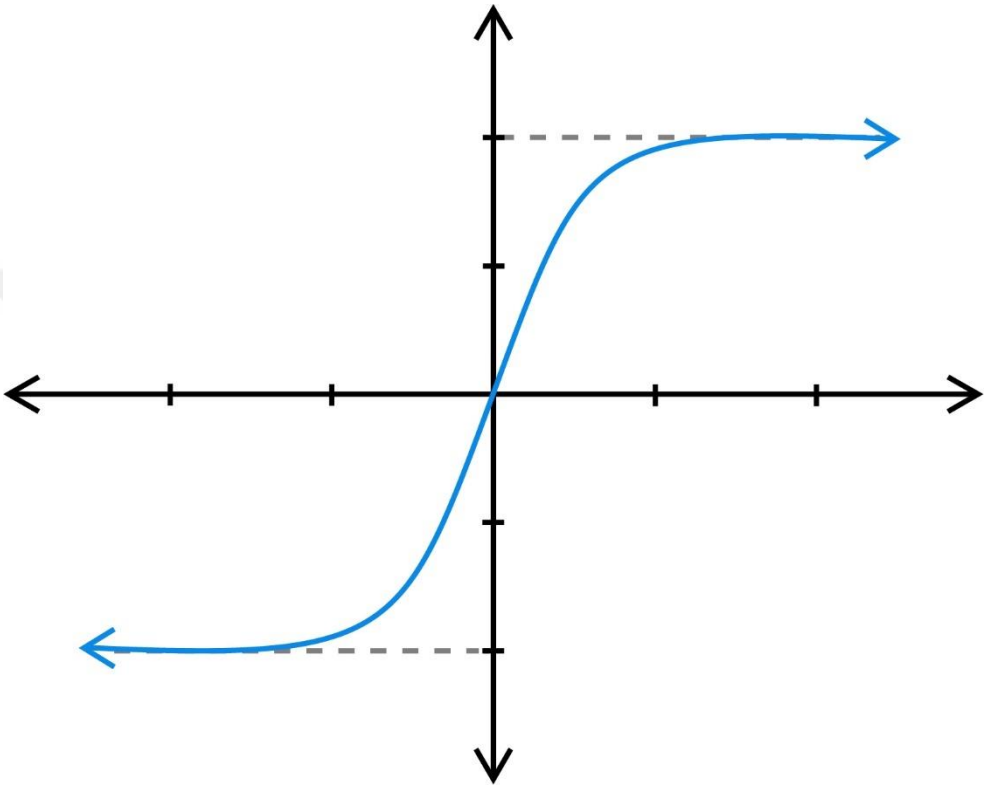


Şekil 3.14. Swish ReLU aktivasyon fonksiyonu grafiği (Fatima ve ark. (2022) çalışmasından uyarlandı)

Softmax aktivasyon fonksiyonu: Softmax aktivasyon fonksiyonu genellikle çoklu sınıflandırma probleminde uygulanmaktadır. Softmax fonksiyonun çıktısı her bir sınıfa ait (0-1) değerleri arasında olasılıksal sonuçlardan oluşmaktadır (Kandel ve

Castelli, 2020). Softmax fonksiyonun matematiksel ifadesi Denklem 3.11'de, fonksiyonun grafiği Şekil 3.15.'de gösterilmiştir.

$$\sigma\,(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.11}$$



Şekil 3.15. Softmax aktivasyon fonksiyonu grafiği (Es-Sabery ve ark. (2021) çalışmasından uyarlandı)

İleri Yayılım ve Geri Yayılım (Forward propagation ve Backpropagation): İleri beslemeli sinir ağlarında giriş sinyalleri tek yönlü sinyal akışıyla girdi katmanından çıkış katmanına doğru iletilmektedir (Que ve ark., 2022). İleri beslemeli sinir ağlarında bir katmanın çıktısı (ilgili katmandaki nöronların çıktısı) sonraki katmandaki nöronlarda girdi vektörü olarak kullanılmaktadır. İleri beslemeli sinir ağlarında girdi katmanında işlenen girdi vektörleri ara katmanlarda işlenerek elde edilen özellik vektörleri çıkış katmanda girdi vektörü olarak kullanılır burada özellik matrisleri işlenerek sinir ağının çıktısı/tahminleri oluşturmaktadır (Que ve ark., 2022). Çıktı katmanında tahmin edilen değerler beklenen değerlerden farklı ise hata değerinin minimum veya sıfır olabilmesi için geri yayılım yöntemi uygulanmaktadır (Que ve

ark., 2022). Burada geri yayılım sürecinde hata değerini azaltmak için rastgele seçilen ağırlık vektörleri ve bias değerleri güncellenir (Qiu ve ark., 2021).

Sinir ağları ile önerilen uygulamalarda modellerin tahmin sürecinde hata değerini minimize etmek için hata kontrolü yapılır. Burada, kayıp (loss) fonksiyonlarıyla (örneğin Cross-Entropy Kayıp Fonksiyonu, MSE, MAE, Binary-Crossentropy vs.) hata değerleri hesaplanır, optimizasyon algoritmalarıyla (örneğin dereceli azalma (gradient descent), adam, RMS-Prop vs.) beklenen sonuca yakınsamaya dolayısıyla hata değeri minimize edilmeye çalışılır (Que ve ark., 2022; Qiu ve ark., 2021). Hata değerinin minimize edilme sürecinde, loss fonksiyonları hata değerlerini hesaplarken, optimizasyon ile en düşük hata değerini üretecek en uygun değeri/değerleri bulmaya çalışır.

3.3.1. Derin öğrenme yöntemleri

Derin öğrenme algoritmaları kullanım alanlarına göre (örneğin görüntü işleme, duygu analizi, ses tanıma vs.) farklı teknolojilere sahiptir. Bu teknolojiler arasında derin boltzmann makinesi (deep boltzmann machine - DBM), yapay sinir ağları (artificial neural network - ANN), derin sinir ağları (deep neural networks - DNN), yinelenen sinir ağları (recurrent neural networks - RNN), evrişimsel sinir ağları (convolutional neural networks - CNN) yer almaktadır. Tez kapsamında önerilen derin sinir ağlarında ANN, RNN ve türevleri, CNN algoritmaları kullanılmıştır.

Bu bölümde; film ve kitap derecelendirme uygulaması için geliştirilen RS-DSINet derin öğrenme modelinin omurgasını oluşturan algoritmalardan SimpleRNN (Lipton ve ark., 2015), BiLSTM (Graves ve Schmidhuber, 2005), LSTM (Hochreiter ve Schmidhuber, 1997), GRU (Cho ve ark., 2014), CNN (Le Cun ve ark., 1989) ve FCCMLP (Murtagh, 1991) derin sinir ağlarının teorik çerçevesi verilmiştir.

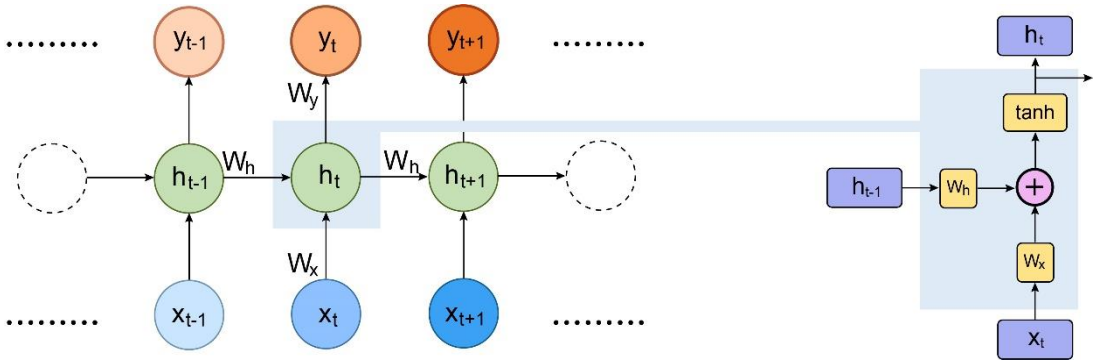
3.3.1.1 Basit tekrarlayan sinir ağları (simple recurrent neural network)

İleri beslemeli sinir ağında girdi sinyalleri tek yönlü olarak gizli katmanından çıktı katmanına doğru iletilmektedir. Burada iletim katmanlarda yer alan nöronlar tarafından gerçekleşmektedir. Gizli katmandaki her bir nöron önceki katmandan gelen veri ile eğitilir, ardından elde edilen özellik vektörleri sonraki nöronun girdi verisi olarak kullanılmaktadır. Sinir ağlarında yer alan nöronlarda döngü yoktur. Dolayısıyla mevcut nöron sadece önceki nöronun çıktı verisi ile eğitilmektedir. Burada ileri beslemeli sinir ağların eğitim sürecinde önceki bilgiler dikkate alınmaz. Bu durum, sinir ağların karar verme/tahmin sürecini olumsuz etkileyebilir. Sinir ağlarının eğitiminde yaşanan problemlerin çözümünde RNN ve türevleri potansiyel bir çözüm tekniğidir. RNN mimarileri sıralı verileri işlemede önemli bir teknolojidir. RNN mimarilerinin temelini yinelemeli ağlar oluşturmaktadır. Burada bilgiler bir döngü boyunca hareket eder. RNN mimarilerinin omurgasını kısa süreli bellek teknolojisi oluşturmaktadır. Bellekler önceki ve mevcut girdileri hafızada tutarak karar verme/tahmin sürecinde sinir ağların verimliğini artırabilir. RNN teknolojisinin dezavantajları arasında; uzun vadeli bağımlılık sorunu, gradyan kaybolması ve önemli önemsiz tüm bilgiler belleklerde tutulduğundan işlem hacminde artış yaşanmasına neden olabilir. RNN teknolojisinde yaşanan sorunların çözümünde RNN türevlerinden LSTM, BiLSTM, GRU mimarileri kullanılabilir. SimpleRNN tanh katmanından oluşan en basit zaman serisi mimarisidir. SimpleRNN mimarisinin ağ yapısı Şekil 3.16.'da verilmiştir. Tez kapsamında tensorflow keras kütüphanesi layers modülündeki SimpleRNN fonksiyonu kullanılmıştır. SimpleRNN modelindeki matematiksel işlemler Denklem (3.12-3.13)'de verilmiştir.

$$\mathrm{h}_t = tanh\,(W_h x_t + U_h h_{t-1}) \tag{3.12}$$

$$\mathrm{y}_t = (W_y h_t) \tag{3.13}$$

Denklem (3.12-3.13)'de x_t giriş vektörü, y_t çıktı vektörü, W_h, W_y ağırlık matrisleri, h_{t-1} önceki çıktı, U_h kısa vadeli durumun ağırlık matrisini, h_t gizli durumun hesaplanmasıdır.



Şekil 3.16. SimpleRNN mimarisi (Khalifa ve ark. (2021) çalışmasından uyarlandı)

3.3.1.2 Uzun-kısa süreli bellek (long-short term memory)

LSTM, 1997'de Hochreiter ve Schmidhuber tarafından önerilmiştir. RNN mimarisindeki uzun vadeli bağımlılık, gradyan kaybolması problemlerin çözümünde önerilmiştir. LSTM bellek hücrelerinden oluşmaktadır. Bellek teknolojisi mimarinin karar mekanizmasıdır. Bellekler bilgi kaybolması/gradyan kaybolması probleminin çözümünde önerilmiştir. Mimarideki her bellek hücresi giriş, unutma ve çıkış kapılarından oluşmaktadır. Giriş ve unutma kapısı güncellenme işleminde, çıkış kapısı sonraki bellek hücresinin giriş vektörünü veya mimarinin tahmin vektörünü oluşturmaktadır. LSTM mimarisinin ağ yapısı Şekil 3.17.'de verilmiştir. Çalışmada, tensorflow keras kütüphanesi layers modülündeki LSTM fonksiyonu kullanılmıştır. LSTM modelindeki matematiksel işlemler Denklem (3.14-3.19)'da verilmiştir. LSTM mimarisindeki matematiksel işlemler:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (3.14)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3.15)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3.16)$$

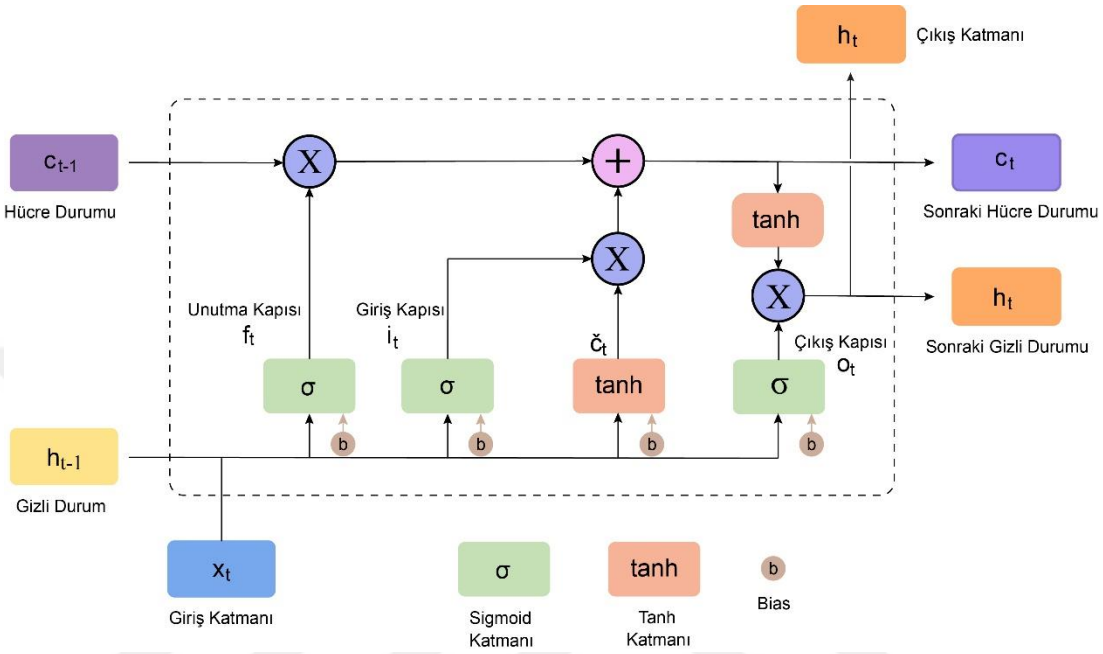
$$\check{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (3.17)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \check{c}_t \quad (3.18)$$

$$h_t = o_t \odot tanh(c_t) \quad (3.19)$$

Denklem (3.14-3.19)'da x_t giriş vektörü, h_t gizli durum, σ sigmoid fonksiyonu, $\check{c}_t$ hücre girişi aktivasyon vektörü, c_{t-1}, c_tsırasıyla *t-1* ve *t* zaman aralığında hücre

durumları, i_t, o_t, f_t sırasıyla giriş, çıkış ve unutma kapısı, W_i, W_o, W_c, W_f ağırlık matrisleri, b_i, b_o, b_c, b_f bias değeri, U_i, U_o, U_c, U_f kısa vadeli durumun ağırlık matrisi, $\odot$ vektörel çarpımı göstermektedir.



Şekil 3.17. LSTM mimarisi (Le ve ark. (2019) çalışmasından uyarlandı)

3.3.1.3 Çift yönlü uzun-kısa süreli bellek (bidirectional long-short term memory)

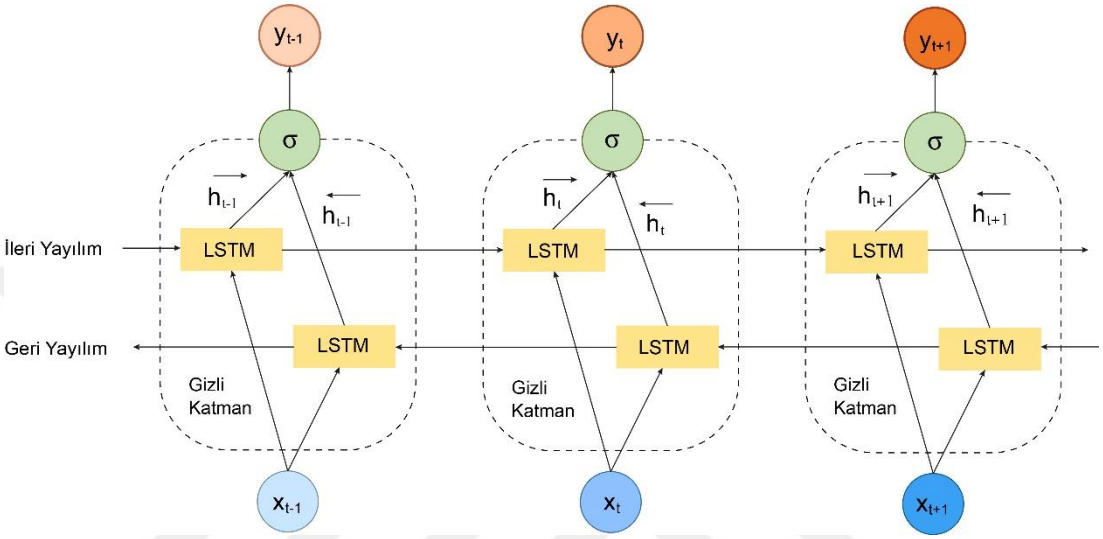
LSTM tek yönlü bir mimaridir. Dolayısıyla yalnızca bilginin geçmiş zaman aralığındaki özelliği işleyebilir. Bu durumda bilginin gelecekteki durumu ihmal edilir. Bu durumun sonucunda LSTM mimarisinde bilgi kaybı yaşanabilir. Sorunun çözümünde BiLSTM mimarisi kullanılabilir. BiLSTM mimarisi yapısındaki güçlü bellekler ile geçmiş ve gelecekte tüm bilgileri işleyebilen çift yönlü bir mimaridir. BiLSTM mimarisinin ağ yapısı Şekil 3.18.'de verilmiştir. Tez kapsamında, tensorflow keras kütüphanesi layers modülündeki LSTM ve Bidirectional fonksiyonları kullanılmıştır. BiLSTM modelindeki matematiksel işlemler Denklem (3.20-3.22)'de verilmiştir.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (3.20)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3.21)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3.22)$$

Denklem (3.20-3.22)'de x_t, y_t sırasıyla giriş ve çıkış vektörü, $\mathcal{H}$ genellikle sigmoid fonksiyonu. $\vec{h}_t, \overleftarrow{h}_t$ sırasıyla ileriye ve geriye doğru bilgileri işler.



Şekil 3.18. BiLSTM mimarisi (Cui ve ark. (2020) çalışmasından uyarlandı)

3.3.1.4 Kapılı tekrarlayan hücre (gated recurrent unit)

LSTM mimarilerinden GRU, Cho ve ark. 2014 yılında önermiştir. GRU mimarisi LSTM mimarisine göre parametre sayısı daha azdır. Dolayısıyla eğitim süreci daha kısadır. Mimari güncelleme ve sıfırlama kapılarından oluşmaktadır. Güncelleme kapısı LSTM bellek mimarisindeki giriş ve unutma kapılarına benzer burada hangi bilginin tutulacağına veya silineceğine karar verilir. Sıfırlama kapısı hangi bilginin tutulacağına karar verir. GRU mimarisinin ağ yapısı Şekil 3.19.'da gösterilmiştir. Tez kapsamında, tensorflow keras kütüphanesi layers modülündeki GRU fonksiyonu kullanılmıştır. GRU modelindeki matematiksel işlemler Denklem (3.23-3.26)'da verilmiştir.

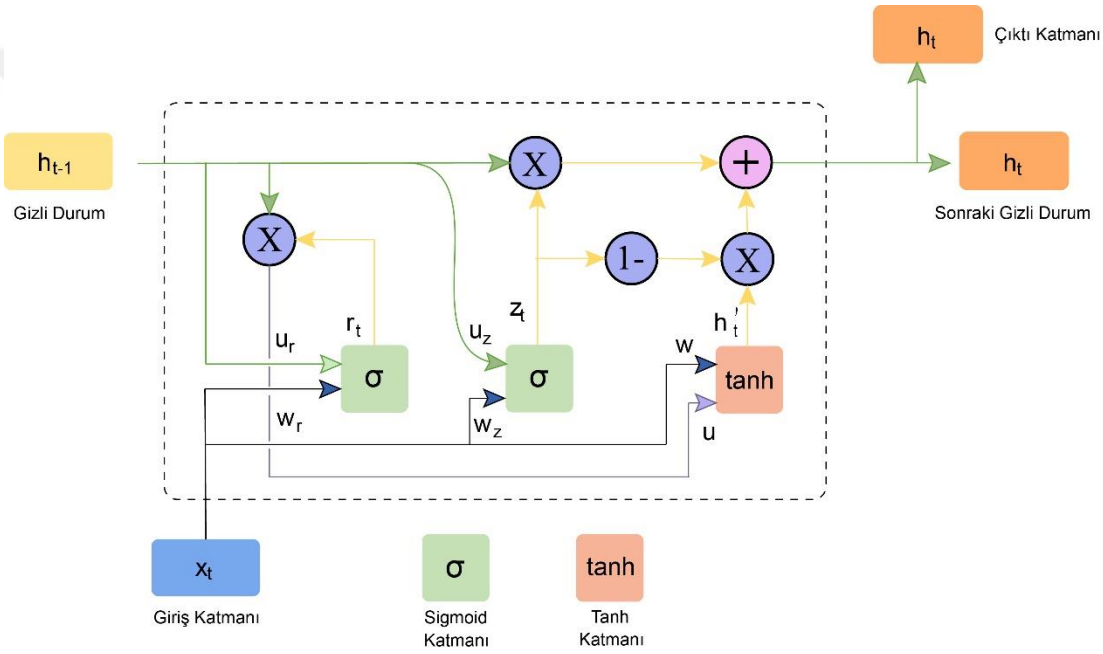
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (3.23)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (3.24)$$

$$\hat{h}_t = \phi(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (3.25)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (3.26)$$

Denklem (3.23-3.26)'da x_t giriş h_t çıkış vektörü, r_t sıfırlama z_t güncelleme kapısı, h_{t-1} önceki çıktı $\hat{h}_t$ beklenen aktivasyon vektörü, ϕ hiperbolik tanjant σ sigmoid aktivasyon fonksiyonu, W_r, W_z, W_h ağırlık matrisleri, b_r, b_z, b_h bias değerlerini, U_r, U_z, U_h kısa vadeli durumlarının ağırlık matrisleri, $\odot$ vektörel çarpımı göstermektedir.

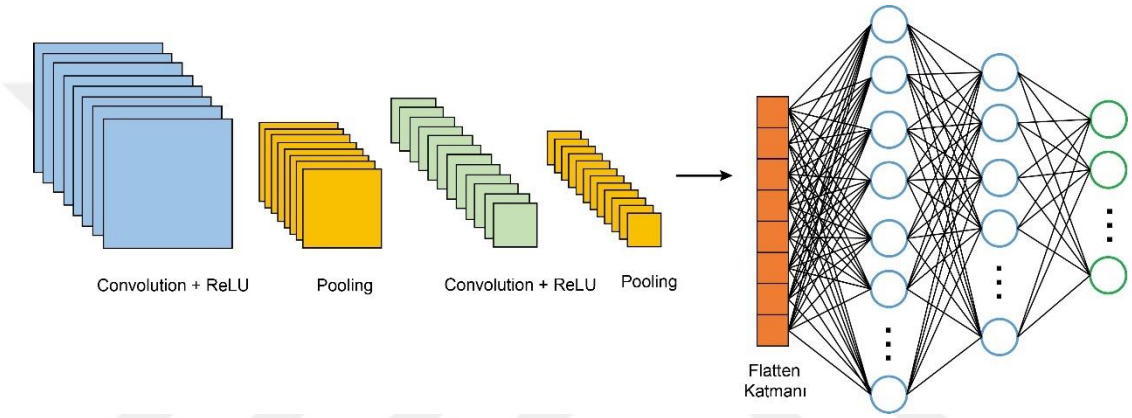


Şekil 3.19. GRU mimarisi (Zhao ve ark. (2019) çalışmasından uyarlandı)

3.3.1.5 Evrişimli sinir ağı (convolutional neural network)

CNN, Le Cun ve ark. 1989 yılında önermiştir. Genellikle görüntü sınıflandırma, görüntü segmentasyonu gibi görsel veri analizinde kullanılmaktadır. Mimari temel olarak evrişim katmanı (convolutional layer - CL), havuzlama katmanı (pooling layer - PL), tam bağlı katmanlardan (fully connected layer - FCC) oluşmaktadır. CL görüntülerden öz nitelik çıkarımı için filtrelemeler uygulanır. İşlem hacminin azaltılmasında kullanılan havuzlama katmanı görüntünün boyutunu azaltmaktadır. Evrişimli sinir ağlarının genellikle sonunda bulunur ve ağın çıktı/sınıflandırıcı

katmanını oluşturmaktadır. CNN tabanlı mimarilerde geri yayılım sürecinde FCC katmanındaki ağırlık, bias değerleri ve evrişim katmanlarındaki filtre katsayıları güncellenir. Evrişimli sinir ağlarında sıklıkla kullanılan diğer bileşenler şöyle: stride (adım sayısı), aktivasyon katmanı, padding (piksel ekleme) ve toplu normalleştirme (batch normalization). Örnek bir CNN mimarisinin ağ yapısı Şekil 3.20.'de verilmiştir. Tez kapsamında tensorflow keras kütüphanesi layers modülündeki Conv2D, GlobalAveragePooling2D, GlobalMaxPooling2D, BatchNormalization (BN) fonksiyonları kullanılmıştır.



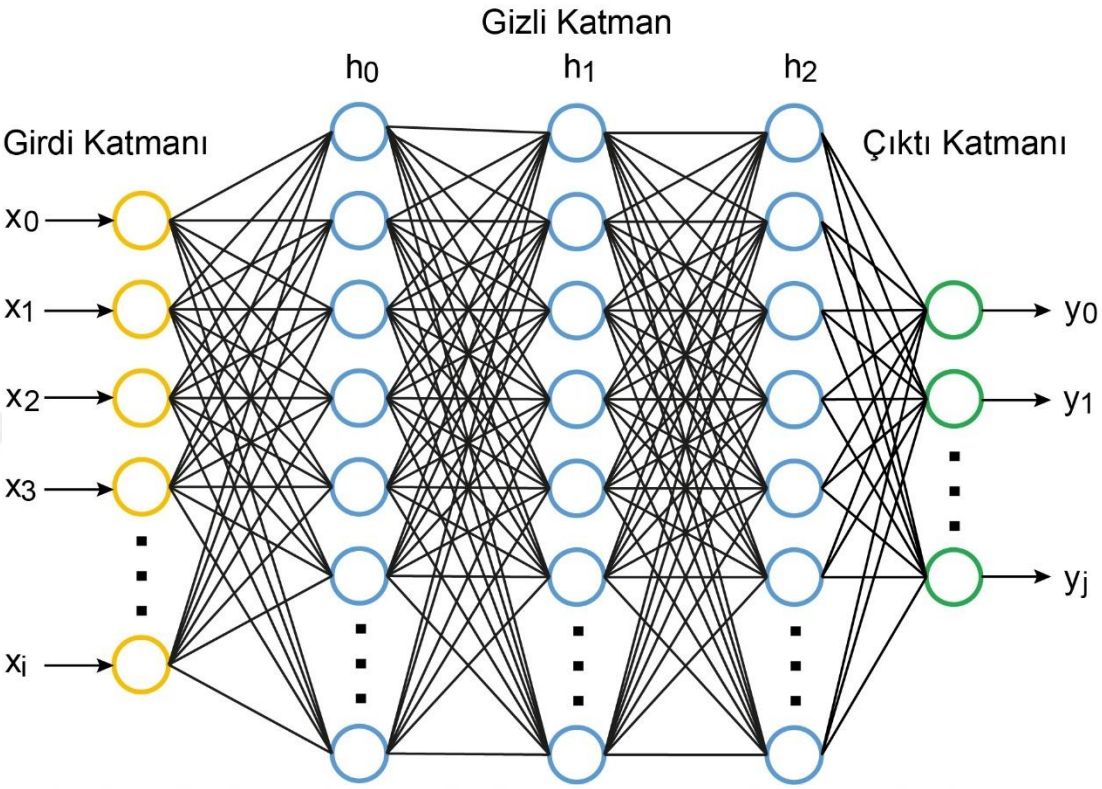
Şekil 3.20. CNN mimarisi (Phung ve Rhee (2019) çalışmasından uyarlandı)

İleri beslemeli derin sinir ağının maliyet fonksiyonun hesaplanmasını içeren sözde kodu Çizelge 3.2.'de verilmiştir.

Çizelge 3.2. İleri beslemeli derin sinir ağının maliyet fonksiyonun hesaplanması

x: girdi vektörü, y: çıktı vektörü, z: derinlik
W_i : i ∈ {1, 2, …, z} ağırlık matrisi, b_i : i ∈ {1, 2, …, z} bias
h_0 = x
for t = 1, …, z **do**
$a_t = b_t + W_t\, h_{(t-1)}$
$h_t = f\,(a_t)$
end for
$\hat{y} = h_z$
$J = L\,(y, \hat{y})$ + λΩ (Θ) (J maliyet, L kayıp, y gerçek değer, $\hat{y}$ tahmin değeri)

Şekil 3.21.'de 5 katmandan (1 girdi katmanı, 3 gizli katman, 1 çıktı katmanı) oluşan örnek bir FCCMLP mimarisi verilmiştir.



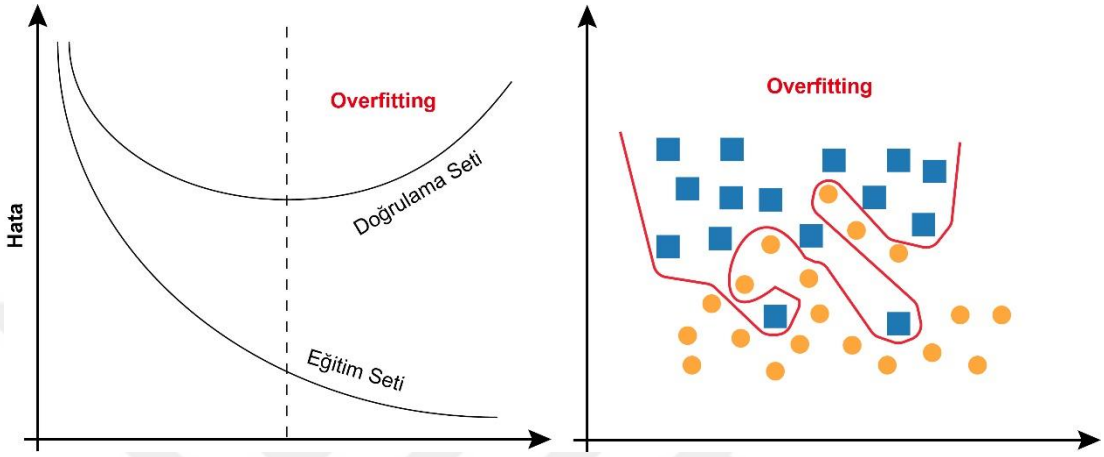
Şekil 3.21. FCCMLP mimarisi örneği (Moayedi ve Mosavi (2021) çalışmasından uyarlandı)

3.3.2. Derin öğrenme algoritmalarında yaşanan problemler

Derin sinir ağlarının başarısı genellikle girdi verileri ile doğru orantılıdır. Büyük veri kümeleri ile eğitilen sinir ağlarının hata oranı daha az dolayısıyla problemin teşhisinde performansı yüksek olabilir. Derin sinir ağlarında eğitim veri setinin küçük veri kümelerinden oluşması, etiketli veri kümelerinin eksikliği, modelin tasarım süreçleri nedeniyle aşırı öğrenme (overfitting), eksik öğrenme (underfitting) problemleri yaşanabilir.

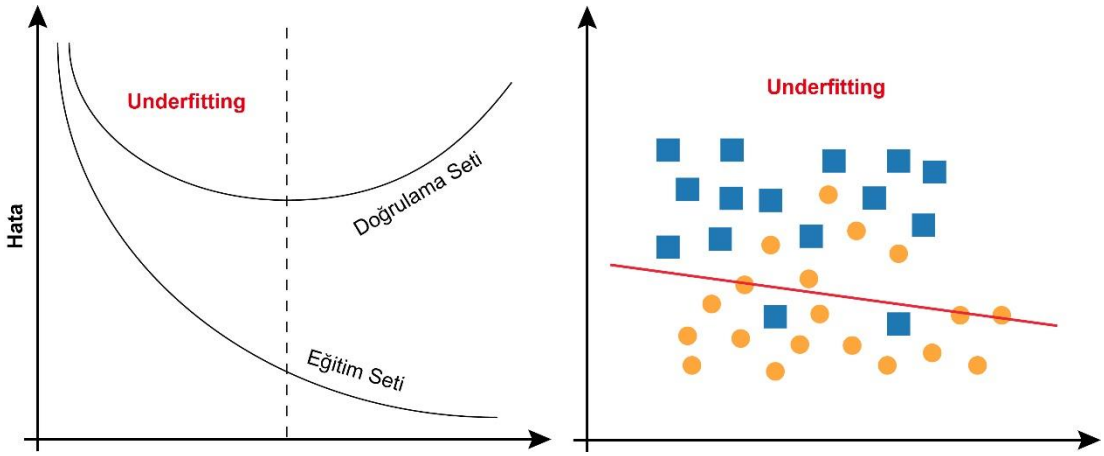
Aşırı öğrenme (overfitting): Modelin problem çözüm sürecinde veri kümesini öğrenmesi yerine ezberlenmesi sürecidir (de Dios Rojas Olvera ve ark., 2022; Zhu ve ark., 2021a). Bu durumda model eğitim veri kümesi ile başarılı sonuçlar üretse de test veri kümesinde ile gerçekleşen tahmin sürecinde başarısız olacaktır. Problemin çözüm

sürecinde veri ekleme, veri artırımı (data augmentation), modellerin karmaşıklığını azaltmak için düzenlileştirme (regularization) yöntemlerinin (lasso ve ridge) uygulanması, erken durdurma (early stopping) yöntemi uygulanabilir (Zhu ve ark., 2021a). Şekil 3.22.'de overfitting problemi özetlenmiştir.



Şekil 3.22. Overfitting (Solanes ve Radua (2022) çalışmasından uyarlandı)

Eksik öğrenme (underfitting): Modelin eğitim veri kümesini yeterince öğrenemediği durumdur. Bu durumda model eğitim ve test veri kümesinde hata oranı yüksektir (de Dios Rojas Olvera ve ark., 2022; Zhu ve ark., 2021a). Problemin çözüm sürecinde modelin karmaşıklığı artırılabilir, verilerdeki gürültüler kaldırılabilir (Zhu ve ark., 2021a). Şekil 3.23.'de underfitting problemi özetlenmiştir.



Şekil 3.23. Underfitting (Solanes ve Radua (2022) çalışmasından uyarlandı)

3.3.3. Önerilen RS-DSINet derin öğrenme algoritması

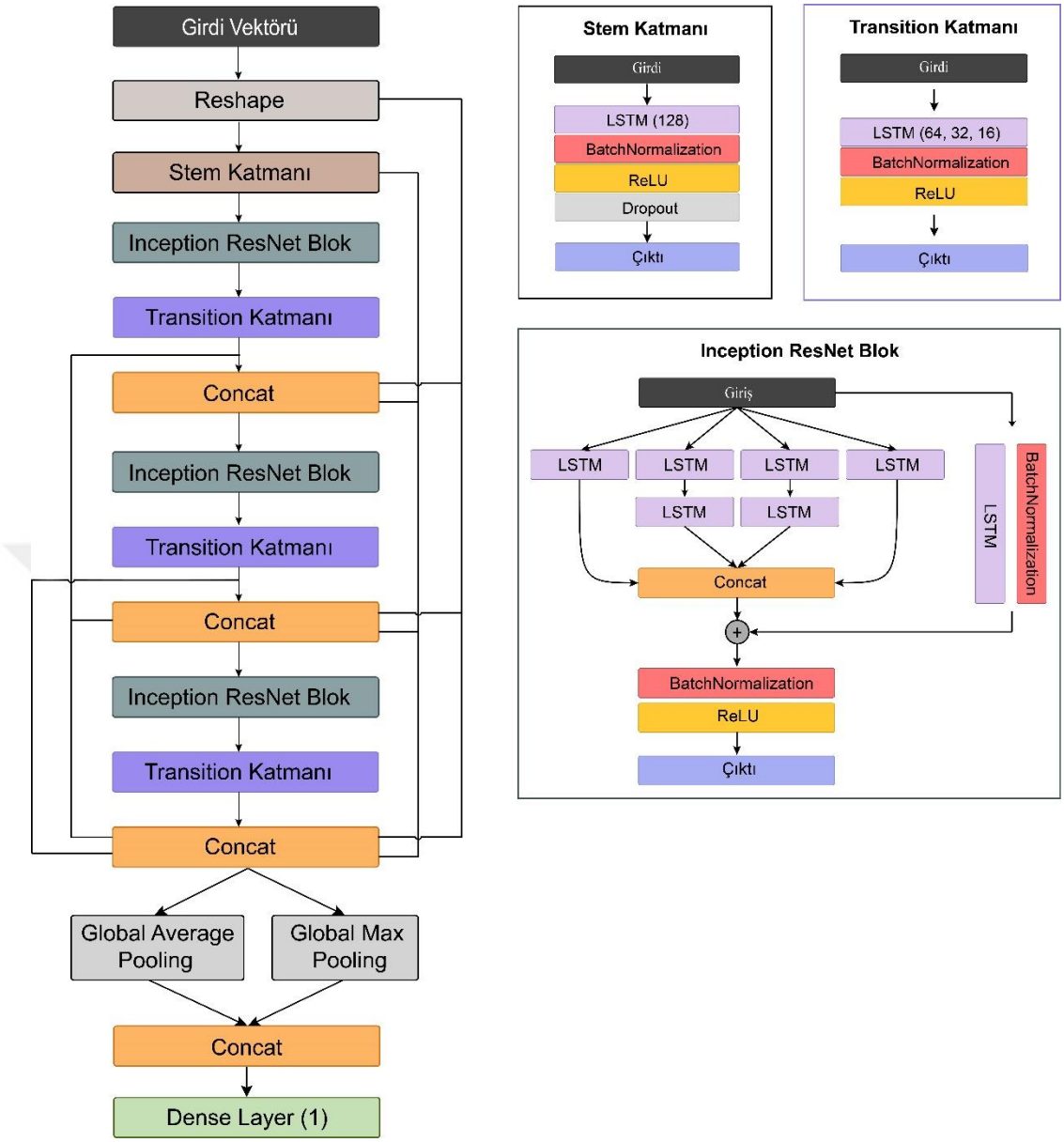
Bu bölümde RS-DSINet derin öğrenme modelinin geliştirme sürecine odaklanmıştır. RS-DSINet modeli RNN ve türevleri (SimpleRNN, LSTM, BiLSTM, GRU), CNN ve FCCMLP katmanları ile geliştirilmiştir. RS-DSINet modeli, derin öğrenme yönteminde yaşanan güncel sorunlardan gradyan kaybolması problemini önlemek için özellik vektör kullanımını artırarak geliştirilmiştir. Mimari temel olarak tensorflow keras kütüphanesi ile Inception Modulü (Szegedy ve ark., 2015) ek özellik vektörü, DenseNet (Huang ve ark., 2017) mimarisi özellik vektörü ve ResNet (He ve ark., 2016) mimarisi artık ağlardan yararlanılarak dizayn edilmiştir. Gradyan kaybolması problemi derin sinir ağlarının eğitimini olumsuz etkilemektedir. Burada sorunun çözümünde; aktivasyon fonksiyonları (ReLU, Leaky ReLU, Swish), artık ağlar (ResNet bloğu), ek özellik vektörleri (Inception ve DenseNet bloğu), Batch normalization, RNN mimarilerinden LSTM ve GRU katmanları kullanılabilir. Önerilen RS-DSINet modelinde gradyan kaybolması problemini önlemek için batch normalization, artık ağlar, ReLU aktivasyon fonksiyonu, RNN türevleri (LSTM, BiLSTM, GRU, SimpleRNN) ve özellik vektörlerinin yeniden kullanımına odaklanarak geliştirilmiştir.

Tez kapsamında önerilen M-1 (RS-DSINet + LSTM) modeli, MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen çalışmada nx18 giriş boyutlarından, yeniden boyutlandırma (reshape) ve kök (stem) katmanından, üç ResNet-Inception bloğundan, üç Geçiş Katmanı (Transition Layer-TL) bloklarından, dört birleştirme (concatenate) katmanından, global ortalama havuzlama (global average pooling-GAP), global maksimum havuzlama (global max pooling-GMP) ve son olarak çıktı katmanından oluşmaktadır. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada; stem katmanı, LSTM (çıkış vektörü 128), batch normalization, ReLU fonksiyonu ve dropout (eşik değeri 0.2) katmanından oluşmaktadır. Stem bloğunun çıktı vektörü 3x128 boyutundadır. Mimarinin ResNet-Inception bloğu çıktı katman boyutu 128, 64, 48, 32, 32, 16, 16 olan yedi LSTM katmanından, iki batch normalization, ReLU fonksiyonundan, birleştirme (concatenate) katmanından ve ekleme (add) katmanından oluşmaktadır.

Birinci ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası, ikinci ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası, üçüncü ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası elde edilmiştir. Mimarinin TL bloğu çıktı katman boyutu 64, 32, 16 olan LSTM katmanından, batch normalization ve ReLU fonksiyonundan oluşmaktadır. Birinci TL bloğunun çıktısında 3x64 öznitelik haritası, ikinci TL bloğunun çıktısında 3x32 öznitelik haritası, üçüncü TL bloğunun çıktısında 3x16 öznitelik haritası elde edilmiştir. ResNet-Inception bloğu, TL bloğu ve concatenate katmanından elde edilen özellik vektörleri GAP ve GMP fonksiyonları kullanılarak 246 boyutlu çıktı vektörü üretilmiştir. Elde edilen iki vektör concatenate fonksiyonu ile birleştirilmiştir. Son olarak elde edilen özellik vektörü Dense katmanı ile tek boyutlu bir çıktı üretilmiştir. Önerilen M-1 modelinin eğitilmiş parametre sayısı 1.035.853'tür. Mimarinin detayları Çizelge 3.3.'de, ağ yapısı Şekil 3.24.'de verilmiştir.

Çizelge 3.3. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-1 modelinin mimari detayları

Adımlar	**Katmanlar**	**Çıktı boyutu**
Giriş katmanı	Giriş vektörü	18
Kök (stem) Katmanı	Reshape	3x6
	LSTM (128) + BN + ReLU	3x128
	Dropout (rate=0.2)	3x128
ResNet-Inception bloğu TL ve Concatenate	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (64)	3x64
	Concatenate	3x198
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (32)	3x32
	Concatenate	3x230
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (16)	3x16
	Concatenate	3x246
	GlobalMaxPooling1D, GlobalAveragePooling1D	246, 246
	Concatenate	492
Çıkış katmanı	FullyConnected	1



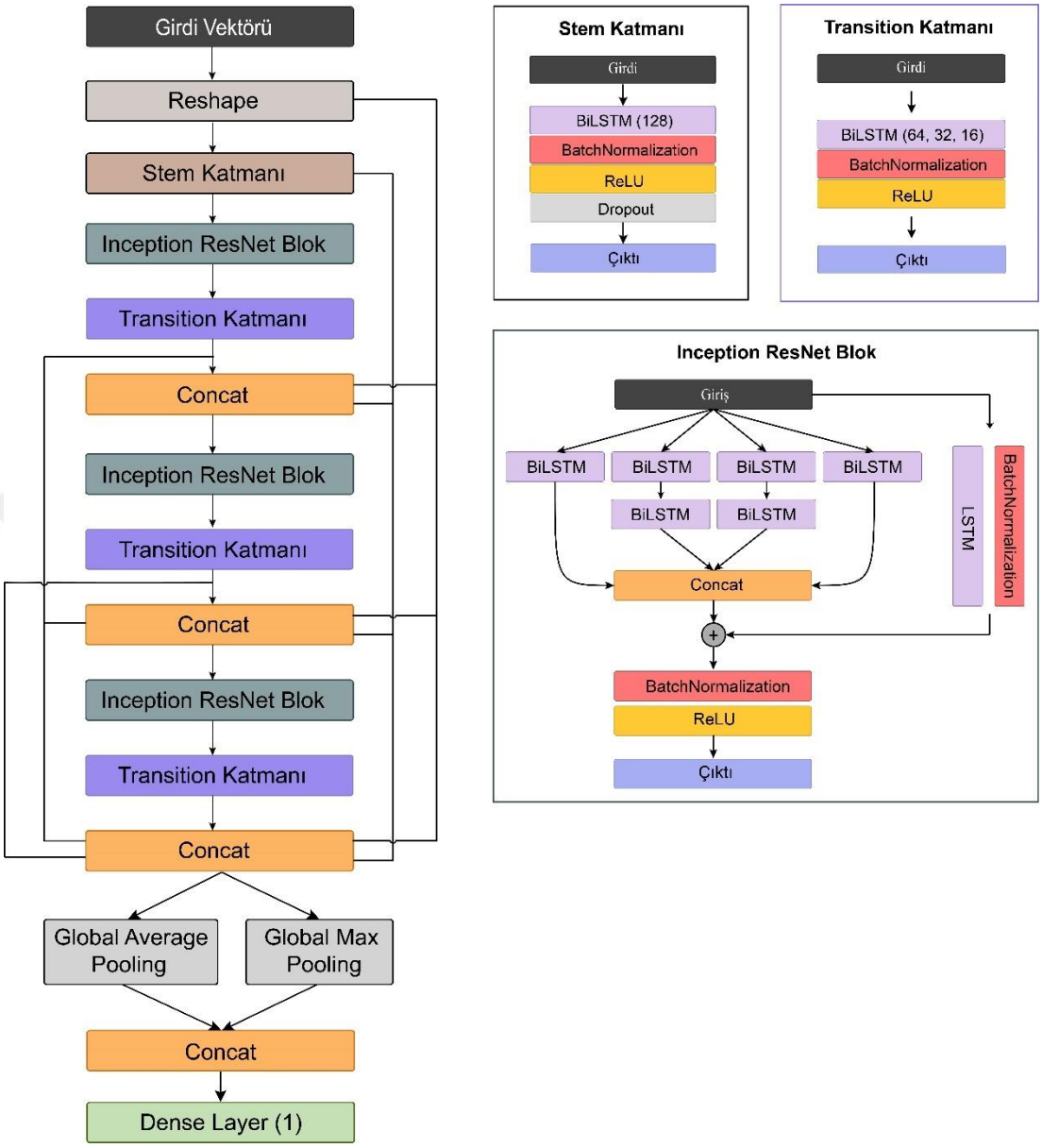
Şekil 3.24. M-1 modelinin ağ yapısı

Tez kapsamında önerilen M-2 (RS-DSINet + BiLSTM) modeli, MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada nx18 giriş boyutlarından, reshape ve stem katmanından, üç ResNet-Inception bloğundan, üç TL bloklarından, dört concatenate katmanından, GAP, GMP ve son olarak çıktı katmanından oluşmaktadır. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada; stem katmanı, BiLSTM (çıkış vektörü 256), batch normalization, ReLU fonksiyonu ve dropout (0.2) katmanından oluşmaktadır. Stem bloğunun çıktı vektörü 3x256 boyutundadır.

Mimarinin ResNet-Inception bloğu çıktı katman boyutu 256, 128, 96, 64, 64, 32, 32 olan yedi BiLSTM katmanından, iki batch normalization, ReLU fonksiyonundan, concatenate katmanından ve add katmanından oluşmaktadır. Birinci ResNet-Inception bloğunun çıktısında 3x256 öznitelik haritası, ikinci ResNet-Inception bloğunun çıktısında 3x256 öznitelik haritası, üçüncü ResNet-Inception bloğunun çıktısında 3x256 öznitelik haritası elde edilmiştir. Mimarinin TL bloğu çıktı katman boyutu 64, 32, 16 olan BiLSTM katmanından, batch normalization ve ReLU fonksiyonundan oluşmaktadır. Birinci TL bloğunun çıktısında 3x128 öznitelik haritası, ikinci TL bloğunun çıktısında 3x64 öznitelik haritası, üçüncü TL bloğunun çıktısında 3x32 öznitelik haritası elde edilmiştir. ResNet-Inception bloğu, TL bloğu ve concatenate katmanından elde edilen özellik vektörleri GAP ve GMP fonksiyonları kullanılarak 486 boyutlu çıktı vektörü üretilmiştir. Elde edilen iki vektör concatenate fonksiyonu ile birleştirilmiştir. Son olarak elde edilen özellik vektörü Dense katmanı ile tek boyutlu bir çıktı üretilmiştir. Önerilen M-2 modelinin eğitilmiş parametre sayısı 3.710.093'tür. Mimarinin detayları Çizelge 3.4.'de, ağ yapısı Şekil 3.25.'de verilmiştir.

Çizelge 3.4. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-2 modelinin mimari detayları

Adımlar	**Katmanlar**	**Çıktı boyutu**
Giriş katmanı	Giriş vektörü	18
Kök (stem) Katmanı	Reshape	3x6
	BiLSTM (128) + BN + ReLU	3x256
	Dropout(rate=0.2)	3x256
ResNet-Inception bloğu TL ve Concatenate	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (64)	3x128
	Concatenate	3x390
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (32)	3x64
	Concatenate	3x454
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (16)	3x32
	Concatenate	3x486
	GlobalMaxPooling1D, GlobalAveragePooling1D	486, 486
	Concatenate	972
Çıkış katmanı	FullyConnected	1



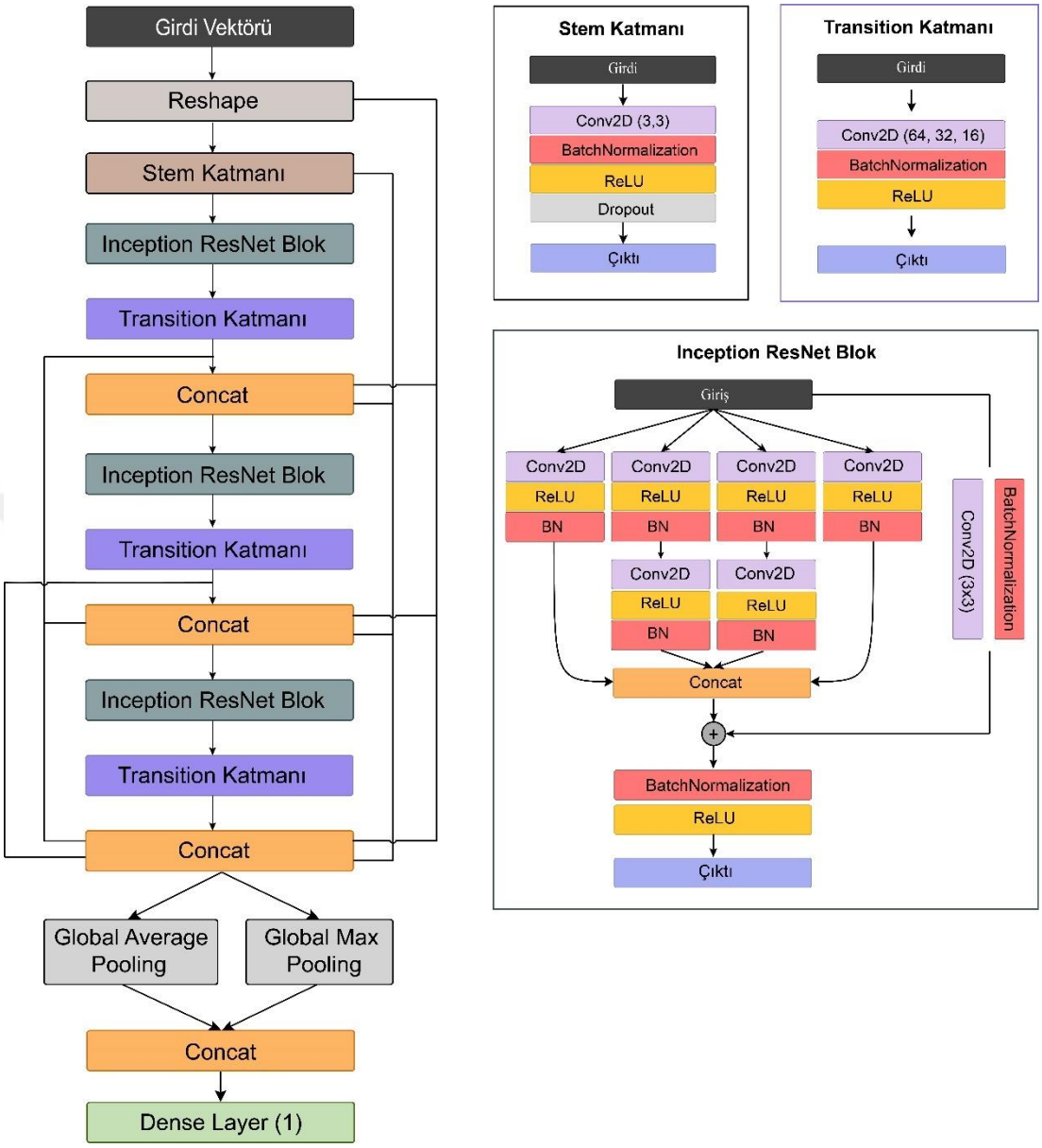
Şekil 3.25. M-2 modelinin ağ yapısı

Tez kapsamında önerilen M-3 (RS-DSINet + CNN) modeli, MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada nx18 giriş boyutlarından, reshape ve stem katmanından, üç ResNet-Inception bloğundan, üç TL bloklarından, dört concatenate katmanından, GAP, GMP ve son olarak çıktı katmanından oluşmaktadır. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada; stem katmanı, Conv2D (çıkış vektörü 128), çekirdek boyutu (kernel size) (3x3) batch normalization, ReLU fonksiyonu ve dropout

(0.2) katmanından oluşmaktadır. Stem bloğunun çıktı vektörü 3x6x128 boyutundadır. Mimarinin ResNet-Inception bloğunda yer alan Conv2D 128, 64, 48, 32, 32, 16, 16 gibi filtrelerden, iki batch normalization, ReLU fonksiyonundan, concatenate katmanından ve add katmanından oluşmaktadır. Birinci ResNet-Inception bloğunun çıktısında 3x6x128 öznitelik haritası, ikinci ResNet-Inception bloğunun çıktısında 3x6x128 öznitelik haritası, üçüncü ResNet-Inception bloğunun çıktısında 3x6x128 öznitelik haritası elde edilmiştir. Mimarinin TL bloğunda yer alan Conv2D 64, 32, 16 gibi farklı filtrelerden, batch normalization ve ReLU fonksiyonundan oluşmaktadır. Birinci TL bloğunun çıktısında 3x6x64 öznitelik haritası, ikinci TL bloğunun çıktısında 3x6x32 öznitelik haritası, üçüncü TL bloğunun çıktısında 3x6x16 öznitelik haritası elde edilmiştir. ResNet-Inception bloğu, TL bloğu ve concatenate katmanından elde edilen özellik vektörleri GAP ve GMP fonksiyonları kullanılarak 241 boyutlu çıktı vektörü üretilmiştir. Elde edilen iki vektör concatenate fonksiyonu ile birleştirilmiştir. Son olarak elde edilen özellik vektörü Dense katmanı ile tek boyutlu bir çıktı üretilmiştir. Önerilen M-3 modelinin eğitilmiş parametre sayısı 1,411,203'tür. Mimarinin detayları Çizelge 3.5.'de, ağ yapısı Şekil 3.26.'da verilmiştir.

Çizelge 3.5. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-3 modelinin mimari detayları

Adımlar	**Katmanlar**	**Çıktı boyutu**
Giriş katmanı	Giriş vektörü	18
Kök (stem) Katmanı	Reshape	3x6x1
	GRU (128) + BN + ReLU	3x6x128
	Dropout(rate=0.2)	3x6x128
ResNet-Inception bloğu TL ve Concatenate	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (64)	3x6x64
	Concatenate	3x6x193
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (32)	3x6x32
	Concatenate	3x6x225
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (16)	3x6x16
	Concatenate	3x6x241
	GlobalMaxPooling1D, GlobalAveragePooling1D	241, 241
	Concatenate	482
Çıkış katmanı	FullyConnected	1



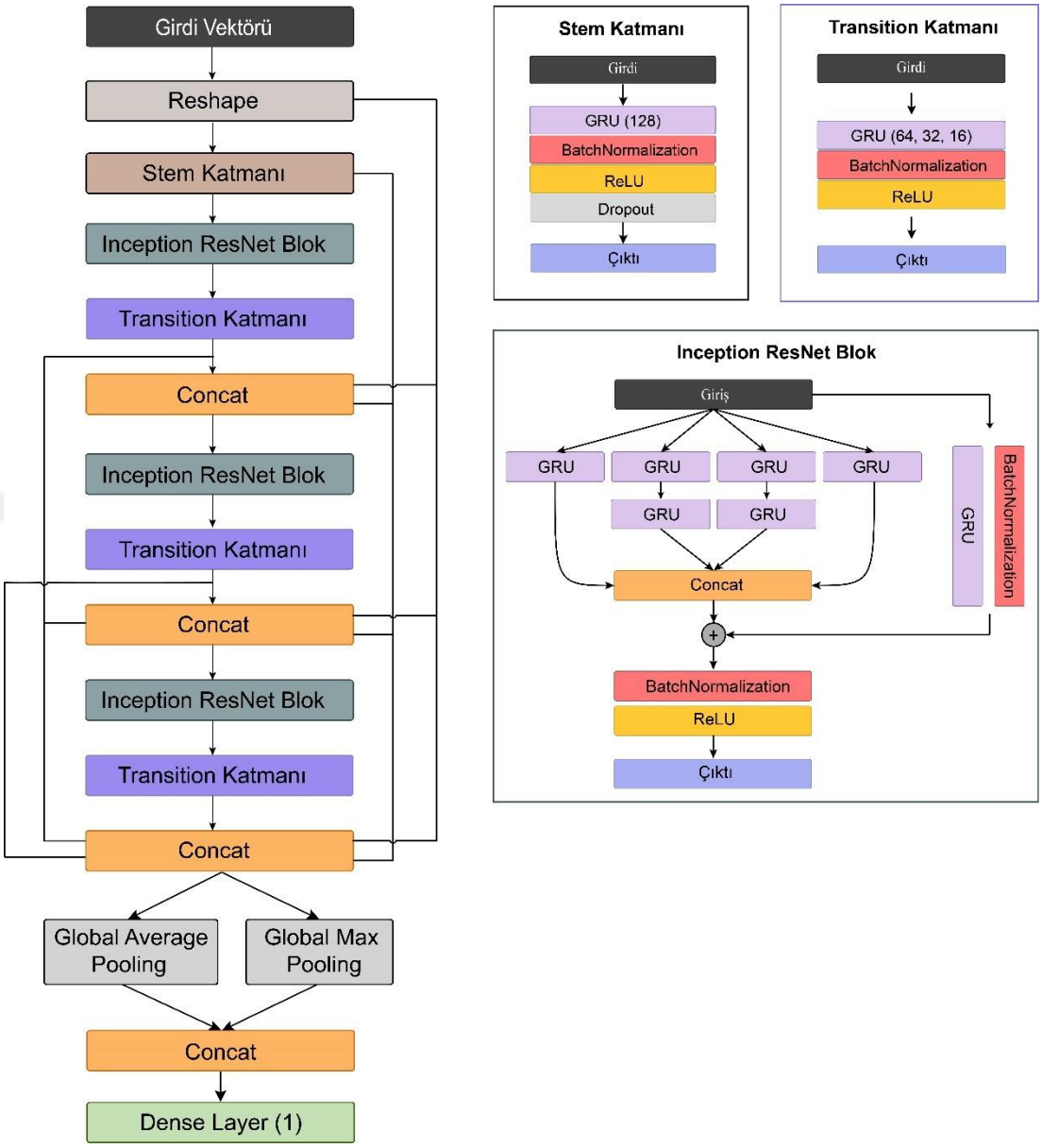
Şekil 3.26. M-3 modelinin ağ yapısı

Tez kapsamında önerilen M-4 (RS-DSINet + GRU) modeli, MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada nx18 giriş boyutlarından, reshape ve stem katmanından, üç ResNet-Inception bloğundan, üç TL bloklarından, dört concatenate katmanından, GAP, GMP ve son olarak çıktı katmanından oluşmaktadır. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada; stem katmanı, GRU (çıkış vektörü 128), batch normalization, ReLU fonksiyonu ve dropout (0.2) katmanından oluşmaktadır. Stem

bloğunun çıktı vektörü 3x128 boyutundadır. Mimarinin ResNet-Inception bloğu çıktı katman boyutu 128, 64, 48, 32, 32, 16, 16 olan yedi GRU katmanından, iki batch normalization, ReLU fonksiyonundan, concatenate katmanından ve add katmanından oluşmaktadır. Birinci ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası, ikinci ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası, üçüncü ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası elde edilmiştir. Mimarinin TL bloğu çıktı katman boyutu 64, 32, 16 olan GRU katmanından, batch normalization ve ReLU fonksiyonundan oluşmaktadır. Birinci TL bloğunun çıktısında 3x64 öznitelik haritası, ikinci TL bloğunun çıktısında 3x32 öznitelik haritası, üçüncü TL bloğunun çıktısında 3x16 öznitelik haritası elde edilmiştir. ResNet-Inception bloğu, TL bloğu ve concatenate katmanından elde edilen özellik vektörleri GAP ve GMP fonksiyonları kullanılarak 246 boyutlu çıktı vektörü üretilmiştir. Elde edilen iki vektör concatenate fonksiyonu ile birleştirilmiştir. Son olarak elde edilen özellik vektörü Dense katmanı ile tek boyutlu bir çıktı üretilmiştir. Önerilen M-4 modelinin eğitilmiş parametre sayısı 781.261'tür. Mimarinin detayları Çizelge 3.6.'da, ağ yapısı Şekil 3.27.'de verilmiştir.

Çizelge 3.6. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-4 modelinin mimari detayları

Adımlar	**Katmanlar**	**Çıktı boyutu**
Giriş katmanı	Giriş vektörü	18
Kök (stem) Katmanı	Reshape	3x6
	GRU (128) + BN + ReLU	3x128
	Dropout(rate=0.2)	3x128
ResNet-Inception bloğu TL ve Concatenate	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (64)	3x64
	Concatenate	3x198
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (32)	3x32
	Concatenate	3x230
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (16)	3x16
	Concatenate	3x246
	GlobalMaxPooling1D, GlobalAveragePooling1D	246, 246
	Concatenate	492
Çıkış katmanı	FullyConnected	1



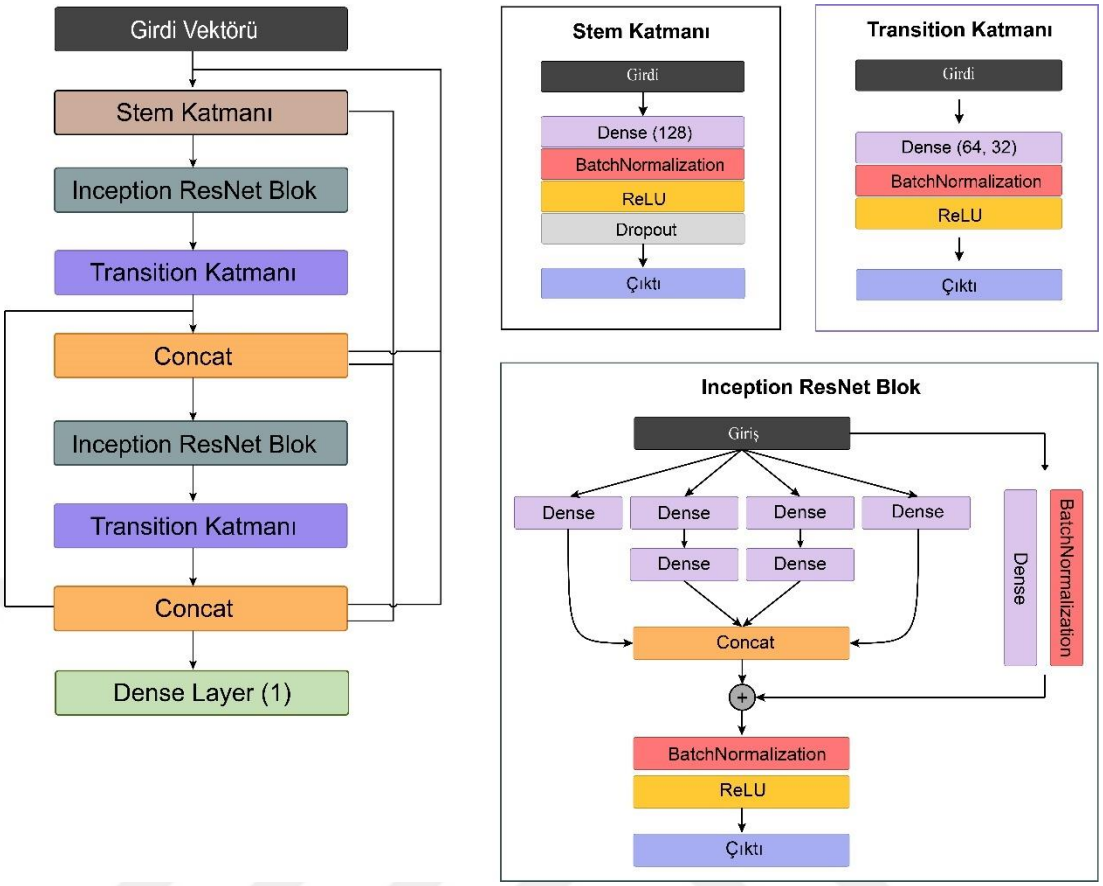
Şekil 3.27. M-4 modelinin ağ yapısı

Tezde kapsamında önerilen M-5 (RS-DSINet + FCCMLP) modeli çalışmada, tensorflow keras kütüphanesi layers modülündeki Dense fonksiyonu kullanılarak geliştirilmiştir. Mimari MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada nx18 giriş boyutlarından ve stem katmanından, iki ResNet-Inception bloğundan, iki TL bloklarından, üç concatenate katmanından, GAP, GMP ve son olarak çıktı katmanından oluşmaktadır. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada; stem

katmanı, Dense (çıkış vektörü 128), batch normalization, ReLU fonksiyonu ve dropout (0.2) katmanından oluşmaktadır. Stem bloğunun çıktı vektörü 128 boyutundadır. Mimarinin ResNet-Inception bloğu çıktı katman boyutu 128, 64, 48, 32, 32, 16, 16 olan yedi Dense katmanından, iki batch normalization, ReLU fonksiyonundan, concatenate katmanından ve add katmanından oluşmaktadır. Birinci ResNet-Inception bloğunun çıktısında 128 öznitelik haritası, ikinci ResNet-Inception bloğunun çıktısında 128 öznitelik haritası elde edilmiştir. Mimarinin TL bloğu çıktı katman boyutu 64, 32 olan Dense katmanından, batch normalization ve ReLU fonksiyonundan oluşmaktadır. Birinci TL bloğunun çıktısında 64 öznitelik haritası, ikinci TL bloğunun çıktısında 32 öznitelik haritası elde edilmiştir. ResNet-Inception bloğu, TL bloğu ve concatenate katmanından elde edilen özellik vektörleri kullanılarak Dense katmanı ile tek boyutlu bir çıktı üretilmiştir. Önerilen M-5 modelinin eğitilmiş parametre sayısı 105,491'tür. Mimarinin detayları Çizelge 3.7.'de, ağ yapısı Şekil 3.28.'de verilmiştir.

Çizelge 3.7. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-5 modelinin mimari detayları

Adımlar	**Katmanlar**	**Çıktı boyutu**
Giriş katmanı	Giriş vektörü	18
	Dense (128) + BN + ReLU	128
	Dropout(rate=0.2)	128
ResNet-Inception bloğu TL ve Concatenate	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (64)	64
	Concatenate	210
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (32)	32
	Concatenate	242
Çıkış katmanı	FullyConnected	1



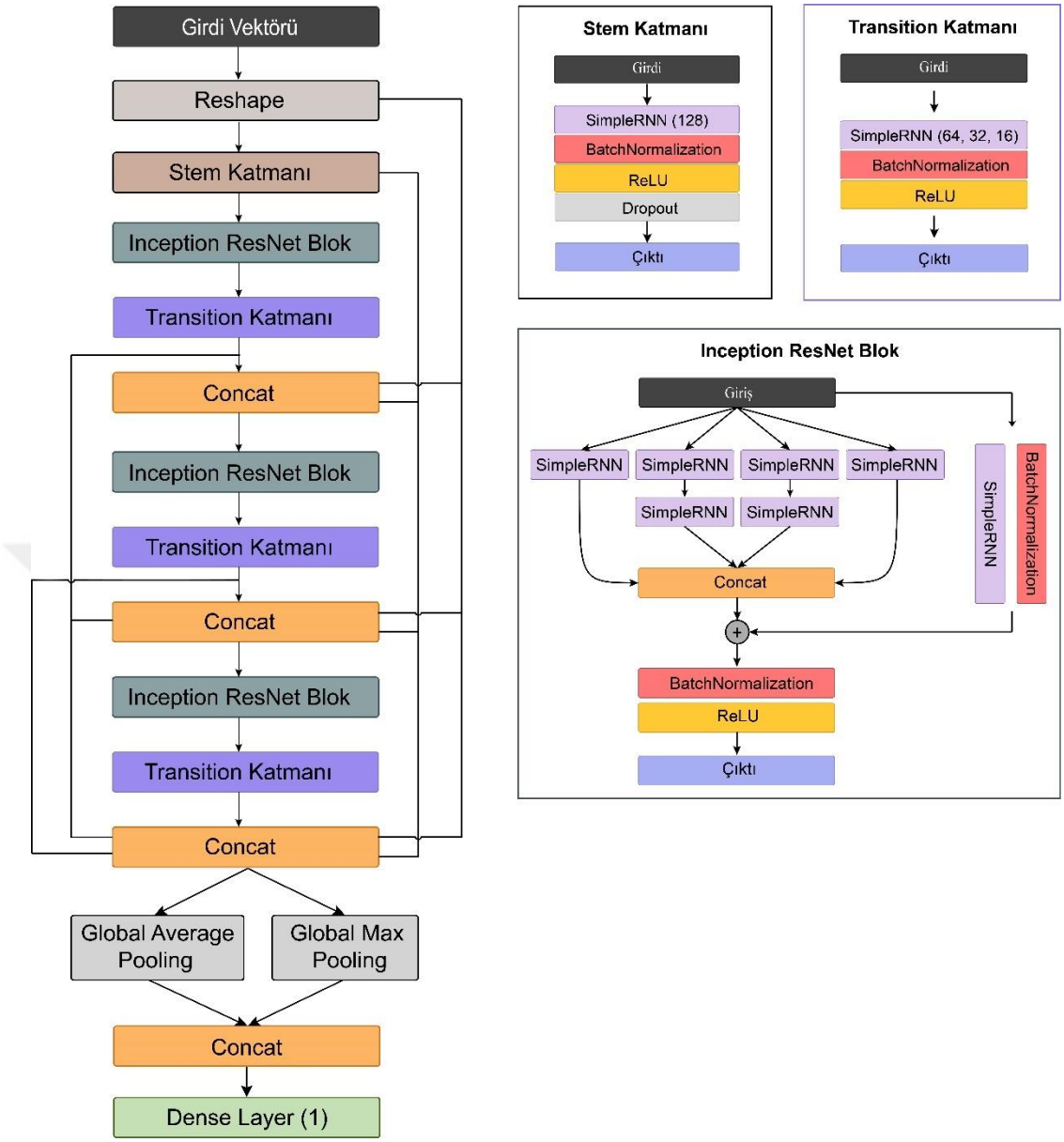
Şekil 3.28. M-5 modelinin ağ yapısı

Tez kapsamında önerilen M-6 (RS-DSINet + SimpleRNN) modeli, MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada nx18 giriş boyutlarından, reshape ve stem katmanından, üç ResNet-Inception bloğundan, üç TL bloklarından, dört concatenate katmanından, GAP, GMP ve son olarak çıktı katmanından oluşmaktadır. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada; stem katmanı, SimpleRNN (çıkış vektörü 128), batch normalization, ReLU fonksiyonu ve dropout (0.2) katmanından oluşmaktadır. Stem bloğunun çıktı vektörü 3x128 boyutundadır. Mimarinin ResNet-Inception bloğu çıktı katman boyutu 128, 64, 48, 32, 32, 16, 16 olan yedi SimpleRNN katmanından, iki batch normalization, ReLU fonksiyonundan, concatenate katmanından ve add katmanından oluşmaktadır. Birinci ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası, ikinci ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası, üçüncü ResNet-Inception bloğunun çıktısında 3x128 öznitelik haritası elde edilmiştir. Mimarinin TL bloğu çıktı katman boyutu 64,

32, 16 olan SimpleRNN katmanından, batch normalization ve ReLU fonksiyonundan oluşmaktadır. Birinci TL bloğunun çıktısında 3x64 öznitelik haritası, ikinci TL bloğunun çıktısında 3x32 öznitelik haritası, üçüncü TL bloğunun çıktısında 3x16 öznitelik haritası elde edilmiştir. ResNet-Inception bloğu, TL bloğu ve concatenate katmanından elde edilen özellik vektörleri GAP ve GMP fonksiyonları kullanılarak 246 boyutlu çıktı vektörü üretilmiştir. Elde edilen iki vektör concatenate fonksiyonu ile birleştirilmiştir. Son olarak elde edilen özellik vektörü Dense katmanı ile tek boyutlu bir çıktı üretilmiştir. Önerilen M-6 modelinin eğitilmiş parametre sayısı 260.845'tür. Mimarinin detayları Çizelge 3.8.'de, ağ yapısı Şekil 3.29.'da verilmiştir.

Çizelge 3.8. MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada, M-6 modelinin mimari detayları

Adımlar	**Katmanlar**	**Çıktı boyutu**
Giriş katmanı	Giriş vektörü	18
Kök (stem) Katmanı	Reshape	3x6
	SimpleRNN (128) + BN + ReLU	3x128
	Dropout(rate=0.2)	3x128
ResNet-Inception bloğu TL ve Concatenate	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (64)	3x64
	Concatenate	3x198
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (32)	3x32
	Concatenate	3x230
	ResNet-Inception-bloğu (128, 16, 32, 64, 48, 32, 16), TL (16)	3x16
	Concatenate	3x246
	GlobalMaxPooling1D, GlobalAveragePooling1D	246, 246
	Concatenate	492
Çıkış katmanı	FullyConnected	1



Şekil 3.29. M-6 modelinin ağ yapısı

3.3.4. Önerilen derin öğrenme algoritmalarında kullanılan diğer teknikler

Bu bölümde, önerilen RS-DSINet derin öğrenme modelinde kullanılan diğer teknikerden pooling layer, batch normalization, add layer, concatenate layer ve dropout parametrelerinin teorik çerçevesi verilmiştir.

Havuzlama (pooling) katmanı: Derin sinir ağların parametre sayısını azaltmak ve hesaplama maliyetini azaltmak için uygulanan bir yöntemdir. Tez kapsamında global max pooling ve global average pooling yöntemleri kullanılmıştır.

Toplu normalleştirme (batch normalization): Derin sinir ağlarının kararlı hale gelmesi için önerilmiştir. Deri sinir ağ mimarilerinde yaşanan gradyan kaybolması problemin çözümünde etkili bir yöntemdir.

Ekleme (add) katmanı: Aynı boyutlara sahip vektörlerin toplanması işlemidir.

Birleştirme (concatenate) katmanı: Aynı boyutlara sahip vektörlerin birleştirilmesi işlemidir.

Bırakma (dropout) katmanı: Aşırı öğrenme probleminin çözümü için önerilmiştir. Rastgele seçilen nöronların eşik değerine göre sinir ağının eğitiminde yer almaması işlemidir.

3.3.5. Derin öğrenme algoritmalarının eğitim ve test süreci

Çalışmada, geleneksel filtreleme yöntemleri ve yapay zeka teknolojileri ile önerilen film/kitap tavsiye sistemi uygulamalarında kullanılan MovieLens 10M, MovieLens 25M, Goodbooks-10K, IMDb movies extensive ve FilmTV movies veri kümelerinim eğitim (training) test oranı 9:1'dir. Eğitim aşamasında doğrulama (validation) test seti için eğitim setinin %10'u kullanılmıştır. Derin sinir ağları ile önerilen çalışmada eğitim/doğrulama veri kümeleri ile eğitim gerçekleştirilmiştir. Test veri seti eğitilen derin sinir ağlarının performans analizi gerçekleştirilmiştir. Çalışma kapsamında derin öğrenme algoritmaları, MovieLens 10M, MovieLens 25M, Goodbooks-10K veri kümeleri ile geliştirilen uygulamada eğitim aşamasında nx18, IMDb movies extensive ve FilmTV movies veri kümeleri ile geliştirilen uygulamada eğitim aşamasında nx5400 boyutlu giriş vektörü ile 100 döngü (epoch) boyunca eğitilmiştir. Sinir ağlarında kullanılan diğer hiper parametreler; minimum öğreme oranı (minimum learning rate) = 1e−03, dropout = 0.2, küme boyutu (batch size) =

128, optimizasyon (optimization) = “adam”, loss = “mse”, en iyisini kaydet (save best only) = “true”, mode (mode) = “min”, bekleme (patience) = 2, faktör (factor) = 0.5, metrik (metrics) = “mae”, gözleme (monitör) = “validation mae/val_mae” (Çizelge 3.10.). Önerilen derin sinir ağlarının hata değerlerini hesaplamak için MSE hata metriği kullanılmıştır. Hata değerinin minimize edilmesinde adam optimizasyon algoritması kullanılmıştır. Derin öğrenme algoritmalarının eğitim aşamasında öğrenmenin durduğu 2 (patience değeri) döngü boyunca validation mae değerinin azalmadığı durumunda learning rate değeri factor parametresi oranında azaltılmıştır. Derin sinir ağlarından döngü boyunca ağırlık vektörleri elde edilmiş burada en düşük validation MAE hata metriğine sahip özellik vektörü test sürecinde kullanılmıştır. Önerilen derin sinir ağ mimarilerini eğitim ve test aşamalarındaki sözde kodu Çizelge 3.9.’da verilmiştir.

Çizelge 3.9. RS-DSINet modelinin eğitim ve test süreci sözde kodu

Algoritma RS-DSINet modelinin eğitim ve test süreçleri

Girdiler: Ön işlem adımları (film ve kitap derecelendirmeleri belirleme, one hot encoding uygulaması, 0-1 normalizasyonu, PCA uygulaması), veri kümesinin eğitim, doğrulama ve test kümelerine ayrılması, d döngü sayısının belirlenmesi

Çıktılar: m eğitilmiş model; film ve kitap derecelendirme tahminleri, istatiksel analizler

1: (X) ← (veri seti öz nitelikler (örneğin şehir, tür, dil vs), target of books set (örneğin dil, yazar, kitap sayısı))
2: (Y) ← (derecelendirme [MovieLens 10M, MovieLens 25M, Goodbooks-10K (1-5), IMDb movies extensive, FilmTV movies (1-10)])
3: (train_x, train_y), (test_x, test_y) ← veri seti bölme (split) işlemi ((X, Y), split size=0.1)
4: (val_x, val_y) ← validation split ((train_x, train_y), split size=0.1)
5: **for** i = 1 to t, **do**
6: m(t) ← model train (adam, mse (train_x, train_y))
7: m(e) ← model evaluate (m(t), (val_x, val_y))
8: **end for**
9: mbest ← validation mae metriğine göre en iyi modelin kaydedilmesi {(m(t), m(e)), Iteration = 1, 2, …, I}
10: tahminler ← model-tahminleri (mbest, (test_x))
11: sonuçlar (örneğin., mae, mse, rmse vs) ← istatiksel analiz (test_y, tahminler)

3.3.6. Derin öğrenme algoritmalarının performans denetimi

Tez kapsamında, MovieLens (MovieLens 10M, MovieLens 25M), IMDb movies extensive, FilmTV movies film veri kümeleri film derecelendirme uygulaması, Goodbooks-10K kitap veri kümesi ile kitap derecelendirme uygulaması geliştirilmiştir. Tez kapsamında önerilen derin öğrenme modellerinden M-1 (RS-DSINet + LSTM), M-2 (RS-DSINet + BiLSTM), M-3 (RS-DSINet + CNN), M-4 (RS-DSINet + GRU), M-5 (RS-DSINet + FCCMLP), M-6 (RS-DSINet + SimpleRNN) ve hibrit yöntemlerden M-Hibrit-1 (BiLSTM + BiGRU + BiSimpleRNN + CNN + FCCMLP), M-Hibrit-2 (BiLSTM + CNN + FCCMLP), M-Hibrit-3 (SimpleRNN + FCCMLP) modelleri ile birinci aşamada tüm veri setlerinde eğitim-doğrulama-test (training-validation-test) yaklaşımı ile derecelendirme uygulaması geliştirilmiştir. İkinci aşamada MovieLens 10M ve Goodbooks-10K veri kümeleri ile geliştirilen uygulamada önerilen algoritmaların detaylı performans analizi için 5-fold CV yöntemi uygulanmıştır. Film ve kitap derecelendirme uygulamasında önerilen sinir ağlarının eğitim ve test süreci performans analizinde hata ölçüm metriklerinden MAE, RMSE, MSE ve MdAE metrikleri, eğitim/doğrulama (training/validation) MAE grafiklerinden yararlanılmıştır.

3.3.7. Derin öğrenme algoritmalarının eğitimi sürecinde kullanılan hiper-parametreler

Bu bölümde, deneysel süreç analizleri sonucunda elde edilen bilgiler doğrultusunda M (1-6) ve tez kapsamında önerilen hibrit mimarilerinden M-Hibrit-1, M-Hibrit-2, M-Hibrit-3 mimarilerinde kullanılan optimal hiper-parametreler Çizelge 3.10.'da verilmiştir. Çizelge 3.10.'da derin öğrenme modellerinde kullanılan parametrelerin sayısı, epoch, input shape, learning rate, dropout, batch size, optimizer, loss, save best only, mode, patience, factor, metrics, monitor parametreleri ve ilgili parametrelerin optimal değerleri yer almaktadır. Ayrıca bu bölümde çalışma kapsamında derin öğrenme modellerinde kullanılan parametrelerin teorik çerçevesi verilmiştir.

Çizelge 3.10. Hiper-parametre

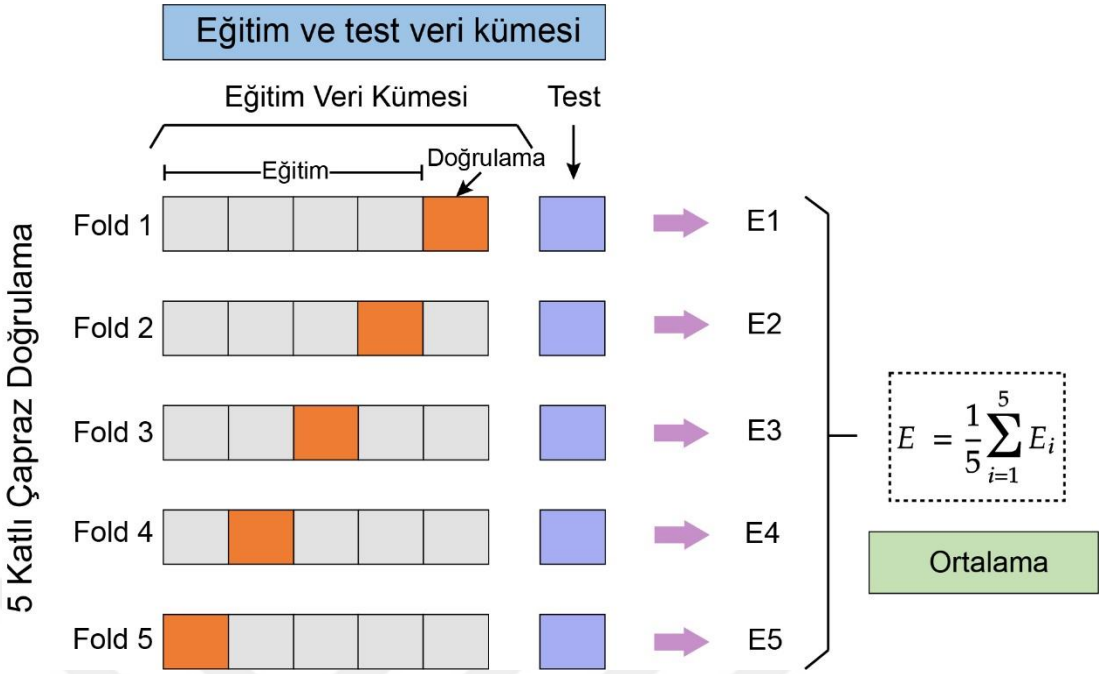
Hiper-parametre	**Değer**
M-1 eğitilmiş parametre sayısı	MovieLens 10M, MovieLens 25M, Goodbooks-10K (1.035.853), IMDb movies extensive ve FilmTV movies (14.164.849)
M-2 eğitilmiş parametre sayısı	MovieLens 10M, MovieLens 25M, Goodbooks-10K (3.710.093), IMDb movies extensive ve FilmTV movies (29.957.297)
M-3 eğitilmiş parametre sayısı	MovieLens 10M, MovieLens 25M, Goodbooks-10K (1.411.203), IMDb movies extensive ve FilmTV movies (1.460.469)
M-4 eğitilmiş parametre sayısı	MovieLens 10M, MovieLens 25M, Goodbooks-10K (781.261), IMDb movies extensive ve FilmTV movies (10.630.705)
M-5 eğitilmiş parametre sayısı	MovieLens 10M, MovieLens 25M, Goodbooks-10K (105.491), IMDb movies extensive ve FilmTV movies (2.091.449)
M-6 eğitilmiş parametre sayısı	MovieLens 10M, MovieLens 25M, Goodbooks-10K (260.845), IMDb movies extensive ve FilmTV movies (421.885)
M-Hibrit-1 eğitilmiş parametre sayısı	MovieLens 10M, Goodbooks-10K (1,033,793)
M-Hibrit-2 eğitilmiş parametre sayısı	MovieLens 10M, Goodbooks-10K (420,289)
M-Hibrit-3 eğitilmiş parametre sayısı	MovieLens 10M, Goodbooks-10K (182,657)
Epoch	100
Input Shape	18
Learning rate	1e-03
Dropout	0.2
Batch size	128
Optimizer	‘adam’
Loss	‘mse’
Save Best Only	true
Mode	‘min’
Patience	2
Factor	0.5
Metrics	'mae'
Monitor	‘val_mae’

Çizelge 3.10.’da, epoch; döngü sayısı. Deneysel süreç sonucunda elde edilen bilgilere göre optimal döngü sayısı 100 olarak belirlenmiştir. Input shape; mimari giriş katman boyutu. Learning rate; geri yayılım yakınsama oranı. Deneysel süreç kapsamında learning rate değeri 1e-03 olarak belirlenmiştir. Batch size; aynı anda kullanılan alt örnek sayısı. Deneysel süreç sonucunda elde edilen bilgilere göre optimal

batch size 128 olarak belirlenmiştir. Optimizer; Derin sinir ağlarındaki hata değerini minimize edilmesinde kullanılmaktadır. Deneysel süreç sonucunda elde edilen bilgilere göre optimal optimizasyon fonksiyonu Kingma and Ba, 2014 yılında önerdiği adam (adaptive momentum) fonksiyonu belirlenmiştir. Loss; derin sinir ağlarının tahminleri ve gerçek değer arasındaki hata değerinin hesaplanması. Çalışmada mse hata metriği kullanılmıştır. Patience; derin sinir ağlarının eğitiminin durduğu döngü sayısı. Tez kapsamında patience değeri 2 olarak belirlenmiştir. Factor; öğrenme oranı azalma miktarı. Tez kapsamında 0.5 factor değeri kullanılmıştır. Metrics; derin sinir ağlarının eğitim, test sürecindeki değerlendirme kriteri. Tez kapsamında mae metriği kullanılmıştır. Monitor-mode-save_best_only; Derin sinir ağlarının eğitiminde iterasyon sayısında ağırlık vektörleri/özellik vektörleri elde edilmektedir. Eğitim sonucunda elde edilen ağırlık vektörleri test sürecinde kullanılmaktadır. Tez kapsamında en düşük (mode = ‘min’), validation mae metriğine (monitor: val_mae) göre elde edilen ağırlık vektörü test aşamasında kullanılmıştır.

3.4. Çapraz Doğrulama (cross-validation)

Model doğrulama tekniklerinden CV yöntemi derin sinir ağlarının başarısını göstermektedir. CV yönteminde veriler k parçaya ayrılır. Derin sinir ağının eğitiminde “k-1” parça kullanılırken geriye kalan parça sinir ağının testinde kullanılmaktadır. Tez kapsamında deneysel süreçte uygulanan cv yönteminin akış şeması Şekil 3.30.’da verilmiştir.



Şekil 3.30. CV akış şeması (Guo ve ark. (2021) çalışmasından uyarlandı)

3.5. Kullanılan Veri Setleri

Bu çalışmada film derecelendirme uygulamalarında MovieLens 10M, MovieLens 20M, FilmTV movies ve IMDb movies extensive veri kümeleri kitap derecelendirme uygulamasında Goodbooks-10K veri kümesi kullanılmıştır. Çizelge 3.11.'de veri kümelerinin detayları verilmiştir.

Çizelge 3.11. Veri seti detayları (Der.: Derecelendirme)

Veri seti	Öğe	Der.	Puan aralığı	Veri seti linkleri	Erişim tarihi
MovieLens 10M	10.681	10.000.054	1-5	https://grouplens.org/datasets/movielens/10m/	30.12.2021
MovieLens 25M	62.423	25.000.095	1-5	https://grouplens.org/datasets/movielens/25m/	28.11.2021
Goodbooks-10K	10.000	5.976.479	1-5	https://www.kaggle.com/datasets/zygmunt/goodbooks-10k	19.11.2019
IMDb movies extensive	85.855	85.855	1-10	https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset	14.09.2021
FilmTV movies	36.096	37.711	1-10	https://www.kaggle.com/stefanoleone992/filmtv-movies-dataset	06.11.2021

MovieLens 10M: Veri kümesi 71567 kullanıcı, 10681 film ve 1-5 aralığındaki 10.000.054 derecelendirme bilgisinden oluşmaktadır. Veri kümesi etiket (tags), film (movies), derecelendirme (ratings) dosyalarından oluşmaktadır. Tez kapsamında film ve derecelendirme dosyaları kullanılmıştır. Film veri kümesinde film id, film adı (film çıkış tarihi) ve tür öz nitelikleri yer almaktadır. Derecelendirme veri kümesinde film id, kullanıcı id, derecelendirme ve oy kullanma tarihi olmak üzere 4 öznitelik yer almaktadır. Tez kapsamında film türleri, filmin çıkış tarihi ve derecelendirme öz nitelikleri kullanılmıştır.

MovieLens 25M: Veri kümesi 162541 kullanıcı, 62423 film ve 1-5 aralığındaki 25000095 derecelendirme bilgisinden oluşmaktadır. Veri kümesi etiket, film, derecelendirme, film linkleri (links), genom-puanlar (genome-scores) ve genom-etiket (genome-tags) dosyalarından oluşmaktadır. Tez kapsamında film ve derecelendirme dosyaları kullanılmıştır. Film veri kümesinde film id, film adı (film çıkış tarihi) ve tür öz nitelikleri yer almaktadır. Derecelendirme veri kümesinde film id, kullanıcı id, derecelendirme ve oy kullanma tarihi olmak üzere 4 öznitelik yer almaktadır. Tez kapsamında film türleri, filmin çıkış tarihi ve derecelendirme öz nitelikleri kullanılmıştır.

Goodbooks-10K: Veri kümesinde 53424 kullanıcı, 10.000 film ve 1-5 aralığındaki 5.976.479 derecelendirme bilgisinden oluşmaktadır. Veri kümesi derecelendirme, kitaplar (books), etiket dosyaları (tags, book tags), okunan kitaplar (to read) dosyalarından oluşmaktadır. Kitap veri kümesi kitap id, isbn, yazar, ortalama derecelendirme gibi 22 öznitelikten oluşmaktadır. Tez kapsamında yazar, dil, derecelendirme, kitap sayısı, kitapların yayın yılı öz nitelikleri kullanılmıştır.

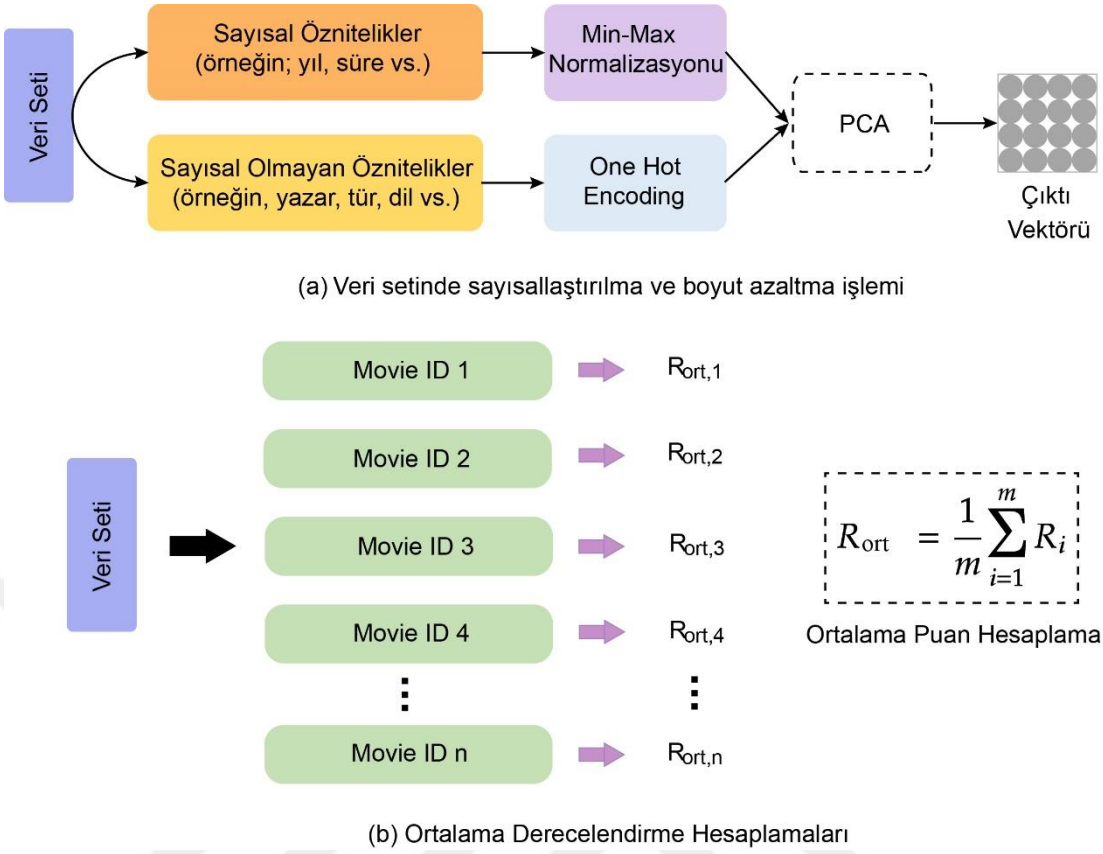
IMDb movies extensive: IMDb veri kümesi “IMDb movies”, “IMDb ratings”, “IMDb names” ve “IMDb title_principals” dosyalarından oluşmaktadır. “IMDb movies” veri kümesi 85.855 film (1894-2020 yılları arası yayınlanmış filmler) ve “imdb id”, “film adı”, “yönetmen”, “süre”, “tür”, “oyuncular”, “dil”, “ortalama oylar” gibi 22 öznitelikten oluşmaktadır. “IMDb ratings” veri kümesi 85.855 derecelendirme bilgisi ve “imdb id”, “filmlere verilen 1-10 arasındaki toplam oylar”, “ağırlıklı oy

ortalaması", "1-10 arası oy kullanan kullanıcı sayısı", "18-30-45 yaşındaki tüm kullanıcıların, yalnızca erkek ve kadın kullanıcıların oy sayısı" gibi 49 öznitelikten oluşmaktadır. "IMDb names" veri kümesi 17 özniteliğe sahip 314.482 kişisel veriden oluşmaktadır. "IMDb title_principals" veri kümesi 6 özniteliğe sahip 835.513 veriden oluşmaktadır. Veri kümesi IMDb movies-IMDb names dosyaları arasındaki bağlantıları içerir.

FilmTV movies veri seti: FilmTV veri kümesi 36,096 farklı filmlerden ve 1-10 aralığındaki 37,711 derecelendirme bilgisinden oluşmaktadır. Veri kümesi "filmtv_id", "türler", "filmin çekildiği ülkeler", "oyuncular", "yönetmenler", "ortalama oylar" gibi 19 öznitelikten oluşmaktadır. Tez kapsamında yıl, süre, tür, şehir, yönetmenler, oyuncular, filmtv tarafından verilen puan öz nitelikleri kullanılmıştır.

3.6. Veri Seti Ön İşleme

Çalışmada uygulanan ön işlem adımlarının birinci aşamasında, nominal değerlerin (yazar, tür, dil, oyuncular, yönetmenler) sayısal dönüşümü için one hot encoding yöntemi uygulanmıştır. One hot encoding çoklu kategoriye sahip değişkenlerin binary (0, 1) şeklinde gösterilmesidir. İkinci aşamada, film veri kümeleri veya kitap veri kümesinde yer alan süre, yayın yılı, filmtv puan öz niteliklerinin [0-1] değer aralığına dönüşüm sağlamak için min-max normalizasyonu uygulanmıştır. Dönüşüm sırasında veriler arasındaki oran bozulmadan işlem yürütülmektedir. Üçüncü aşamada, birinci ve ikinci aşamada elde edilen büyük boyutlu vektörlerin, işlem hacmini azaltmak için PCA yöntemi uygulanmıştır. Son aşamada, MovieLens 10M ve MovieLens 25M veri kümelerindeki filmlere ait ortalama puanlar hesaplanmıştır. Tez kapsamında uygulanan ön işlem adımlarının akış şeması Şekil 3.31.'de gösterilmiştir.



Şekil 3.31. Ön işlem adımları

3.7. Değerlendirme Ölçütleri

Tez kapsamında önerilen film ve kitap tavsiye sistemi derecelendirme çalışmalarının performans değerlendirilmesinde regresyon analizinde sıklıkla kullanılan MAE, MSE, RMSE ve MdAE hata ölçüm metrikleri kullanılmıştır.

MAE: Gerçek ve tahmin değerleri arasındaki mutlak farkın ortalamasını hesaplar. MAE hata metriğinin matematiksel ifadesi Denklem 3.27'de verilmiştir.

MSE: Gerçek ve tahmin değerleri arasındaki ortalama kare farkını hesaplar. MSE metriği hataların varyansını ölçer. MSE hata metriğinin matematiksel ifadesi Denklem 3.28'de verilmiştir.

RMSE: Hataların ortalama kare farkının kareköküdür. RMSE metriği hataların standart sapmasını ölçer. RMSE hata metriğinin matematiksel ifadesi Denklem 3.29'da verilmiştir.

MdAE: Gerçek ve tahmin değerleri arasındaki mutlak farkın medyanı ile hesaplanır. MdAE hata metriğinin matematiksel ifadesi Denklem 3.30'da verilmiştir.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |\mathrm{y}_i - \hat{\mathrm{y}}_i|}{n} \tag{3.27}$$

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (\mathrm{y}_i - \hat{\mathrm{y}}_i)^2}{n} \tag{3.28}$$

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

$$\mathrm{MdAE}\,(\mathrm{y}, \hat{\mathrm{y}}) = median\,(|\mathrm{y}_1 - \hat{\mathrm{y}}_1|, \ldots, |\mathrm{y}_n - \hat{\mathrm{y}}_n|) \tag{3.30}$$

Denklem (3.27-3.30)'da "n" veri setlerindeki toplam derecelendirme sayısı, "y" gerçek derecelendirme değerini, "ŷ" tahmin değerini, göstermektedir.

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Bu bölümde; MovieLens 10M, MovieLens 25M, Goodbooks-10K, FilmTV movies, IMDb movies extensive veri kümeleri kullanılarak önerilen M (1-6), M-Hibrit-1, M-Hibrit-2, M-Hibrit-3 derin sinir ağları, geleneksel makine öğrenme algoritmaları (KNN, SVR, RF, DT, SGD), geleneksel tavsiye sistemi uygulamaları (CBF, CF) ile geliştirilen tavsiye sistemi derecelendirme uygulamasının deneysel sonuçları verilmiştir. Derin sinir ağları geliştirilen derecelendirme uygulamasının birinci aşamasında MovieLens 10M, MovieLens 25M, Goodbooks-10K, FilmTV movies, IMDb movies extensive veri kümeleri ile eğitim-doğrulama-test yaklaşımı, ikinci aşamada MovieLens 10M, Goodbooks-10K veri kümeleri ile 5-fold CV tekniği kullanılmıştır. Bu bölümde, ampirik sonuçlar verilmiştir. Ayrıca derin öğrenme modelleri parametre seçiminin ampirik sonuçları, modellerin training ve validation loss grafikleri, training ve validation MAE grafikleri verilmiştir. Tez kapsamında deneysel sonuç değerlendirilmesinde MAE, MSE, RMSE, MdAE hata ölçüm metrikleri kullanılmıştır.

4.1. MovieLens Veri Seti

Film tavsiye uygulamalarında sıklıkla kullanılan MovieLens (MovieLens 10M, MovieLens 25M) veri kümeleri ile derin sinir ağları, makine öğrenme, CBF, CF yöntemleri kullanılarak tavsiye sistemi derecelendirme uygulamaları geliştirilmiştir. Bu bölümde ampirik sonuçlar verilmiştir.

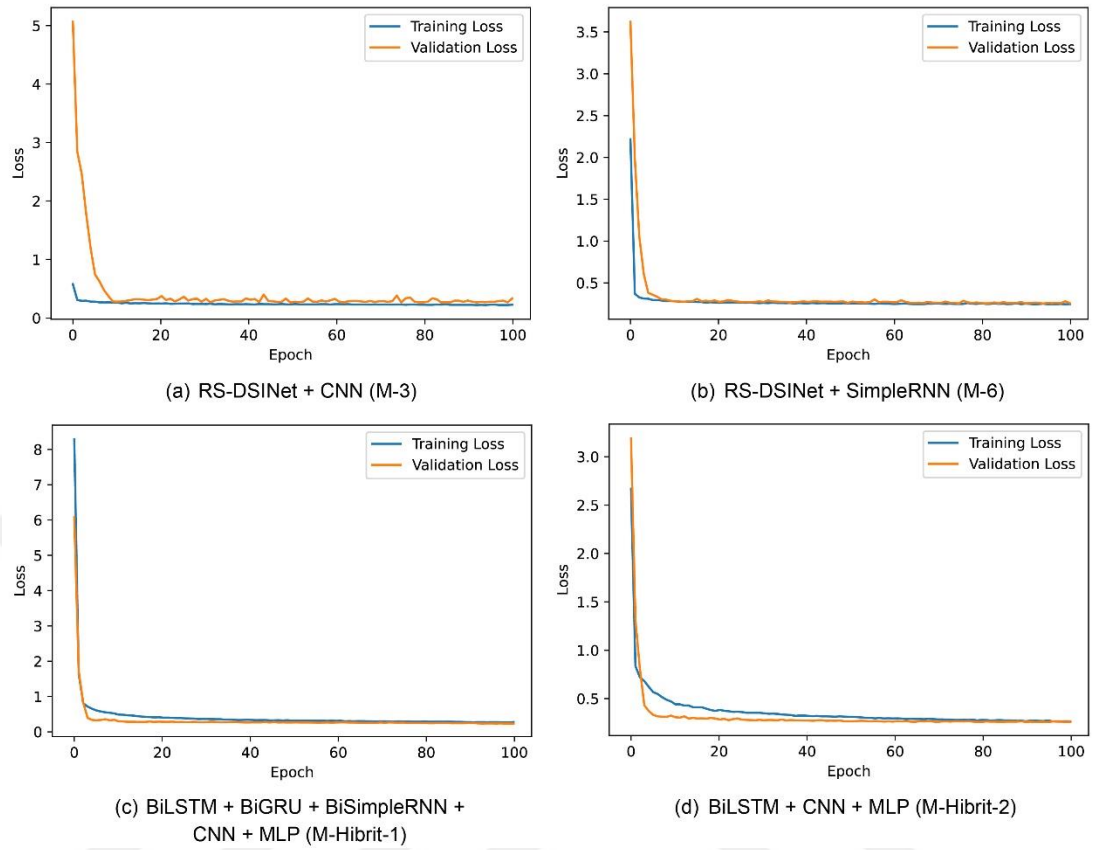
4.1.1. Derin öğrenme

MovieLens 10M veri kümesi kullanılarak M (1-6), M-Hibrit (1-3) derin sinir ağları ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve modellerin eğitim sürecindeki çalışma zamanları (ÇZ) Çizelge 4.1.'de verilmiştir. Derin sinir ağlarının eğitim ve test sürecinde, veri kümesi 9(:1):1 (eğitim/doğrulama/test) oranında ayrılmıştır. Validation veri kümesiyle

gerçekleştirilen çalışmada modellerin performans analizinde M-Hibrit-1 (BiLSTM + BiGRU + BiSimpleRNN + CNN + FCCMLP) modeli 0.3931 validation MAE hata metrik değeri üretmiştir (Çizelge 4.1.). Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, M-Hibrit-1 (BiLSTM + BiGRU + BiSimpleRNN + CNN + FCCMLP) modeli 0.3635 MAE hata metrik değeri üretmiştir. M-Hibrit-2 (BiLSTM + CNN + FCCMLP) hibrit modeli 0.3629 MAE hata metrik değerine göre en başarılı model olurken, M-2 (RS-DSINet + BiLSTM) modeli 0.3714 MAE metrik değerine göre en düşük performansa sahip model olmuştur (Çizelge 4.1.). Modellerin eğitim sürecindeki ÇZ incelendiğinde M-5 modelinin eğitimi 1 dakika (dk) 43 saniyede (sn), M-2 modelinin eğitimi 21 dk 39 sn'de tamamlanmıştır (Çizelge 4.1.). MovieLens 10M veri kümesi ile eğitilen derin sinir ağlarının eğitim sürecindeki kayıp (training loss) ve doğrulama kaybı (validation loss) grafikleri Şekil 4.1.'de, MAE ve validation MAE grafikleri Şekil 4.2.'de verilmiştir. Derin sinir ağları ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.3.'de gösterilmiştir.

Çizelge 4.1. MovieLens 10M veri kümesiyle ve derin sinir ağları ile geliştirilen derecelendirme uygulamasının ampirik sonuçları (eğitim-doğrulama-test yaklaşımıyla)

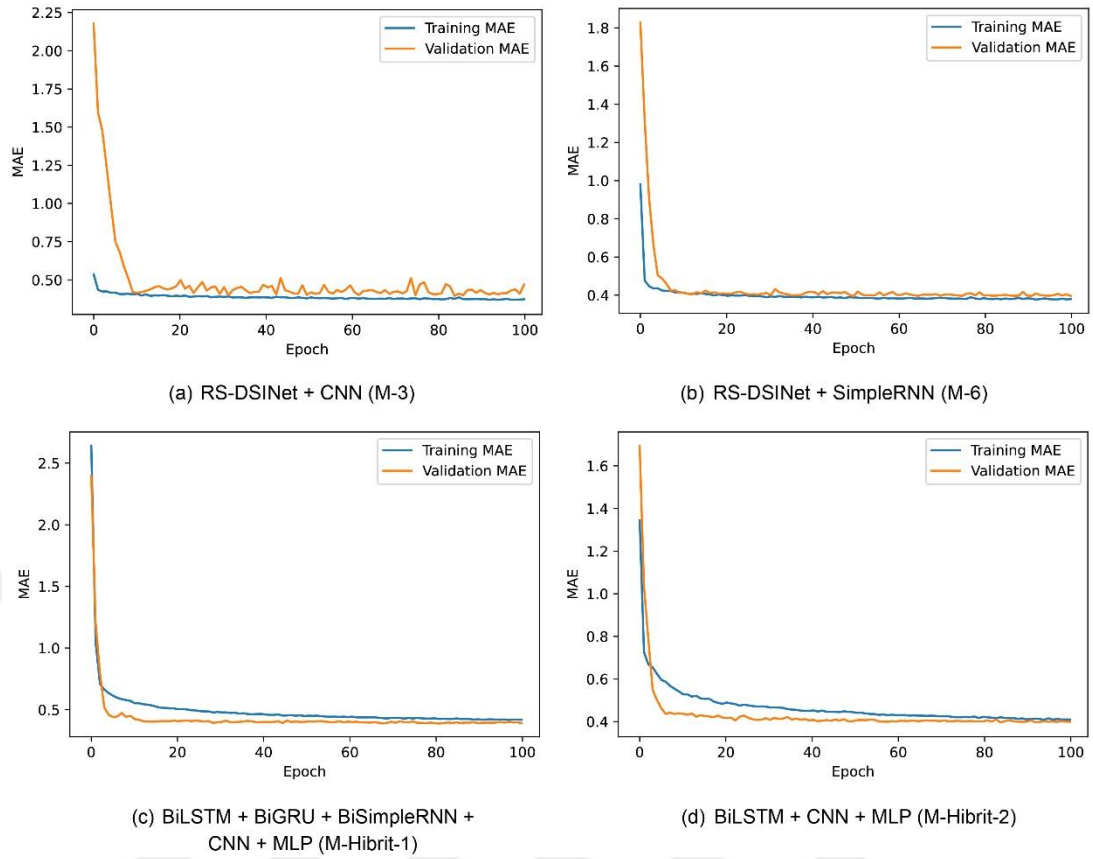
Model	**Derin sinir ağ mimarileri**	**Doğrulama MAE**	**Test MAE**	**Test MSE**	**Test RMSE**	**Test MdAE**	**ÇZ**
M-1	LSTM	0.3972	0.3707	0.2233	0.4726	0.3020	17 dk 44 sn
M-2	BiLSTM	0.3973	0.3714	0.2220	0.4711	0.3113	21 dk 39 sn
M-3	CNN	0.3982	0.3678	0.2218	0.4710	0.3008	6 dk 56 sn
M-4	GRU	0.3946	0.3700	0.2269	0.4764	**0.2979**	14 dk 17 sn
M-5	FCCMLP	0.3961	0.3684	0.2200	0.4691	0.3105	1 dk 43 sn
M-6	SimpleRNN	0.3942	0.3663	0.2185	0.4675	0.3001	8 dk 35 sn
M-Hibrit-1	BiLSTM + BiGRU + BiSimpleRNN + CNN + FCCMLP	**0.3931**	0.3635	**0.2150**	**0.4637**	0.3038	7 dk 9 sn
M-Hibrit-2	BiLSTM + CNN + FCCMLP	0.3938	**0.3629**	0.2158	0.4646	0.3017	3 dk 34 sn
M-Hibrit-3	SimpleRNN + FCCMLP	0.3958	0.3649	0.2172	0.4660	0.3045	1 dk 46 sn



Şekil 4.1. MovieLens 10M eğitim veri setinde (a) M-3, (b) M-6, (c) M-Hibrit-1, (d) M-Hibrit-2 derin sinir ağlarının training loss ve validation loss grafikleri

Şekil 4.1. (a)'ya göre M-3 modelinin en düşük loss değeri 0.2212, en yüksek loss değeri 0.5800, en düşük validation loss değeri 0.2689, en yüksek validation loss değeri 5.0644'tür. Loss değerinde 3. döngüye kadar önemli bir azalma gözlemlenmiştir. 4. döngüden sonra 19. döngüye kadar genellikle azalma görülürken 20. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 97. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 12. döngüye kadar önemli bir azalma gözlemlenirken, 13. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 32. döngüde elde etmiştir. Şekil 4.1. (b)'ye göre M-6 modelinin en düşük loss değeri 0.2353, en yüksek loss değeri 2.2163, en düşük validation loss değeri 0.2655, en yüksek validation loss değeri 3.6213'tür. Loss değerinde 9. döngüye kadar önemli bir azalma gözlemlenmiştir. 10. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 92. döngüde model en düşük loss değerini

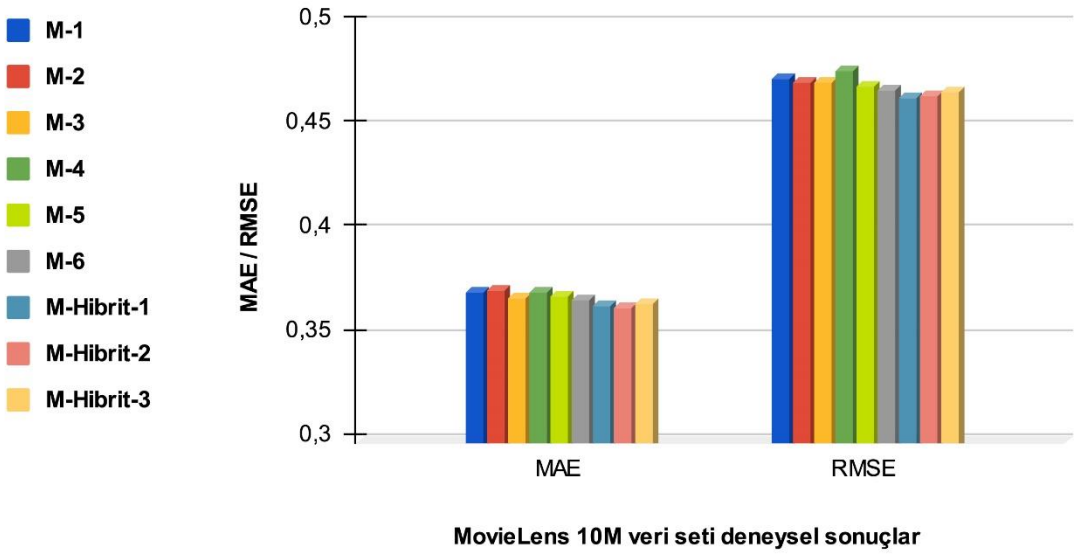
üretmiştir. Validation loss değerinde 11. döngüye kadar önemli bir azalma gözlemlenirken, 12. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 94. döngüde elde etmiştir. Şekil 4.1. (c)'ye göre M-Hibrit-1 modelinin en düşük loss değeri 0.2806, en yüksek loss değeri 8.2855, en düşük validation loss değeri 0.2630, en yüksek validation loss değeri 6.0888'dir. Loss değerinde 19. döngüye kadar önemli bir azalma gözlemlenmiştir. 20. döngüden sonra 29. döngüye kadar azalma gözlemlenmiştir. 30. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 97. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 6. döngüye kadar önemli bir azalma gözlemlenirken, 7. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 93. döngüde elde etmiştir. Şekil 4.1. (d)'ye göre M-Hibrit-2 modelinin en düşük loss değeri 0.2699, en yüksek loss değeri 2.6700, en düşük validation loss değeri 0.2642, en yüksek validation loss değeri 3.1908'dir. Loss değerinde 11. döngüye kadar önemli bir azalma gözlemlenmiştir. 12. döngüden sonra 72. döngüye kadar fonksiyon grafiğinde artış gözlemlense de model genellikle azalış eğilimindedir. 73. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 98. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 9. döngüye kadar önemli bir azalma gözlemlenirken, 10. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 95. döngüde elde etmiştir.



Şekil 4.2. MovieLens 10M eğitim veri setinde (a) M-3, (b) M-6, (c) M-Hibrit-1, (d) M-Hibrit-2 derin sinir ağlarının training MAE ve validation MAE grafikleri

Şekil 4.2. (a)'ya göre M-3 modelinin en düşük MAE hata değeri 0.3641, en yüksek MAE değeri 0.5369, en düşük validation MAE hata değeri 0.3982, en yüksek validation MAE hata değeri 2.1779'dir. MAE değerinde 2. döngüye kadar önemli bir azalma gözlemlenmiştir. 3. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 98. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 11. döngüye kadar önemli bir azalma gözlemlenirken, 11. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 84. döngüde elde etmiştir. Şekil 4.2. (b)'ye göre M-6 modelinin en düşük MAE değeri 0.3779, en yüksek MAE değeri 0.9801, en düşük validation MAE değeri 0.3942, en yüksek validation MAE değeri 1.8291'dir. MAE değerinde 4. döngüye kadar önemli bir azalma gözlemlenmiştir. 5. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 92. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 9. döngüye kadar önemli bir azalma

gözlemlenirken, 10. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 50. döngüde elde etmiştir. Şekil 4.2. (c)'ye göre M-Hibrit-1 modelinin en düşük MAE değeri 0.4153, en yüksek MAE değeri 2.6406, en düşük validation MAE değeri 0.3931, en yüksek validation MAE değeri 2.3992'dir. MAE değerinde 21. döngüye kadar önemli bir azalma gözlemlenmiştir. 22. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış ve azalışlar gözlemlense de model genellikle azalış eğilimindedir. 97. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 6. döngüye kadar önemli bir azalma gözlemlenirken, 7. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 81. döngüde elde etmiştir. Şekil 4.2. (d)'ye göre M-Hibrit-2 modelinin en düşük MAE değeri 0.4077, en yüksek MAE değeri 1.3444, en düşük validation MAE değeri 0.3938, en yüksek validation mae değeri 1.6942'dir. MAE değerinde 11. döngüye kadar önemli bir azalma gözlemlenmiştir. 12. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlense de model genellikle azalış eğilimindedir. 94. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 7. döngüye kadar önemli bir azalma gözlemlenirken, 8. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 84. döngüde elde etmiştir.



Şekil 4.3. Derin sinir ağ mimarilerinin test sonuçları (MAE ve RMSE hata metrikleriyle)

Şekil 4.3.'e göre MovieLens 10M veri seti geliştirilen uygulamada MAE metrik değerine göre M-Hibrit-2 en başarılı model olurken M-2 en başarısız model olmuştur. RMSE metrik değerine göre M-Hibrit-1 en başarılı model olurken M-4 en başarısız model olmuştur.

MovieLens 10M veri seti ile önerilen derecelendirme uygulamasında farklı sayıdaki eğitim (doğrulama)/test oranları kullanılarak yapılan deneysel çalışmanın sonuçları ve modellerin eğitim sürecindeki ÇZ Çizelge 4.2.'de verilmiştir. Deneysel süreçte M-6 derin öğrenme modeli kullanılmıştır. Çizelge 4.2.'e göre eğitim (doğrulama)/test oranın 9(:1):1 olduğu durumda M-6 modeli doğrulama veri kümesi ile gerçekleştirilen analiz çalışmasında 0.3942 MAE hata değeri üretirken, test veri kümesi gerçekleştirilen analiz çalışmasında 0.3663 MAE hata değeri üretmiştir. Eğitim (doğrulama)/test oranın 7(:1):3 olduğu durumda M-6 modeli doğrulama veri kümesi ile gerçekleştirilen analiz çalışmasında 0.3894 MAE hata değeri üretirken, test veri kümesi gerçekleştirilen analiz çalışmasında 0.3882 MAE hata değeri üretmiştir. Son olarak eğitim (doğrulama)/test oranın 5(:1):5 olduğu durumda M-6 modeli doğrulama veri kümesi ile gerçekleştirilen analiz çalışmasında 0.3836 MAE hata değeri üretirken, test veri kümesi gerçekleştirilen analiz çalışmasında 0.3888 MAE hata değeri üretmiştir. Modellerin eğitim sürecindeki ÇZ incelendiğinde eğitim (doğrulama)/test oranın 9(:1):1 olduğu durumda M-6 modelinin eğitimi 8 dk 35 sn, eğitim (doğrulama)/test oranın 5(:1):5 olduğu durumda M-6 modelinin eğitimi 5 dk 6 sn'de tamamlanmıştır (Çizelge 4.2.).

Çizelge 4.2. MovieLens 10M veri kümesinde farklı sayıdaki eğitim/test oranları ile M-6 modeli kullanılarak geliştirilen uygulamanın deneysel sonuçları

(%) Eğitim (%) doğrulama) / (%) test oranı	Doğrulama MAE	Test MAE	Test MSE	Test RMSE	Test MdAE	ÇZ
90 (10)-10	0.3942	**0.3663**	**0.2185**	**0.4675**	**0.3001**	8 dk 35 sn
80 (10)-20	0.3921	0.3815	0.2402	0.4901	0.3163	8 dk 27 sn
70 (10)-30	0.3894	0.3882	0.2441	0.4941	0.3263	7 dk 14 sn
60 (10)-40	0.3881	0.3850	0.2422	0.4921	0.3204	5 dk 37 sn
50 (10)-50	**0.3836**	0.3888	0.2460	0.4959	0.3249	5 dk 6 sn

4.1.1.1 5 katlı çapraz doğrulama

Film tavsiye sistemi derecelendirme probleminde önerilen M (1-6) derin sinir ağlarının ve M-Hibrit-1, M-Hibrit-2, M-Hibrit-3 hibrit yöntemlerinin detaylı performans analizi için 5-fold CV yöntemi uygulanmıştır. Deneysel sonuçlar ve modellerin eğitim sürecindeki ÇZ Çizelge 4.3.'de verilmiştir.

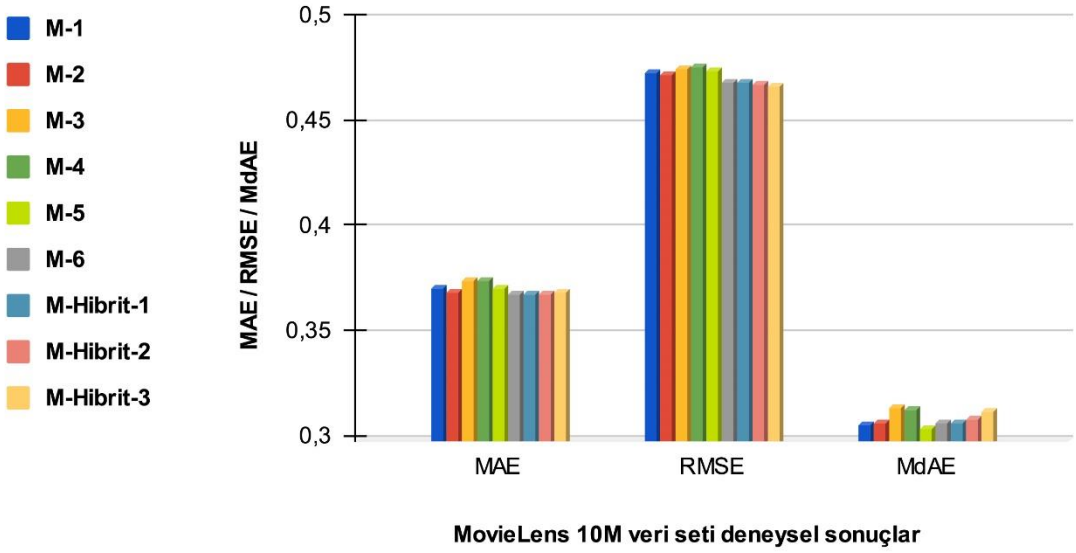
Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, MAE metriğinin ortalama (ORT) değerine göre M-6 en başarılı model olmuştur. M-6 Fold 1'de 0.3667, Fold 2'de 0.3687, Fold 3'de 0.3690, Fold 4'de 0.3660 ve Fold 5'de 0.3722 MAE hata değeri üretmiştir. M-6 modelinin ORT MAE hata değeri 0.3685, MAE metrik değerine göre standart sapması (SS) 0.0021'dir. M-3 modeli 0.3756 ORT MAE metrik değeriyle en düşük başarıya sahip model olmuştur. Fold 1'de 0.3738, Fold 2'de 0.3748, Fold 3'de 0.3726, Fold 4'de 0.3767 ve Fold 5'de 0.3801 MAE hata değeri üretmiştir. M-3 modelinin MAE metrik değerine göre standart sapması (SS) 0.0026'dir.

Uygulama kapsamında önerilen hibrit modelleri arasında M-Hibrit-1 ORT MAE metrik değerine göre en başarılı model olmuştur. M-Hibrit-1 Fold 1'de 0.3677, Fold 2'de 0.3681, Fold 3'de 0.3688, Fold 4'de 0.3683 ve Fold 5'de 0.3711 MAE hata değeri üretmiştir. M-Hibrit-1 modelinin ORT MAE hata değeri 0.3688, MAE metrik değerine göre standart sapması (SS) 0.0012'dir. Önerilen hibrit modellerden M-Hibrit-3 modeli 0.3696 ORT MAE metrik değeri ile en düşük başarıya sahip model olmuştur. Fold 1'de 0.3700, Fold 2'de 0.3692, Fold 3'de 0.3679, Fold 4'de 0.3718 ve Fold 5'de 0.3688 MAE hata değeri üretmiştir. M-Hibrit-3 modelinin MAE metrik değerine göre standart sapması (SS) 0.0013'tür.

Modellerin eğitim sürecindeki ÇZ incelendiğinde M-5 modelinin eğitimi 8 dk 40 sn, M-2 modelinin eğitimi 101 dk 15 sn'de tamamlanmıştır. Derin sinir ağları ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE, RMSE, MdAE hata metrik değerlerinin sütun grafiği Şekil 4.4.'de verilmiştir.

Çizelge 4.3. 5-fold cross validation yöntemiyle MovieLens 10M veri kümesi kullanılarak önerilen M (1-6), M-Hibrit (1-3) mimarilerinin performansı

Model	Metrik	F1	F2	F3	F4	F5	Ort	SS	ÇZ
M-1	MAE	0.3758	0.3685	0.3711	0.3707	0.3727	0.3718	0.0024	93 dk 30 sn
	RMSE	0.4724	0.4700	0.4772	0.4754	0.4768	0.4744	0.0027	
	MdAE	0.3165	0.3121	**0.2963**	0.3090	0.3013	0.3070	0.0073	
M-2	MAE	**0.3663**	0.3702	0.3734	0.3697	0.3712	0.3702	0.0023	101 dk 15 sn
	RMSE	0.4680	0.4717	0.4783	0.4694	0.4769	0.4729	0.0041	
	MdAE	0.3136	0.3075	0.3071	0.3099	**0.3010**	0.3078	0.0041	
M-3	MAE	0.3738	0.3748	0.3726	0.3767	0.3801	0.3756	0.0026	30 dk 45 sn
	RMSE	0.4722	0.4722	0.4748	0.4750	0.4847	0.4758	0.0046	
	MdAE	0.3146	0.3082	0.3102	0.3211	0.3206	0.3149	0.0053	
M-4	MAE	0.3785	0.3699	0.3706	0.3735	0.3823	0.3750	0.0048	74 dk 45 sn
	RMSE	0.4759	0.4712	0.4751	0.4785	0.4835	0.4768	0.0041	
	MdAE	0.3149	0.3091	0.3055	0.3049	0.3357	0.3140	0.0114	
M-5	MAE	0.3704	0.3672	0.3721	0.3724	0.3763	0.3717	0.0030	8 dk 40 sn
	RMSE	0.4726	0.4703	0.4804	0.4716	0.4819	0.4754	0.0048	
	MdAE	**0.2993**	**0.3007**	0.3032	0.3120	0.3067	**0.3044**	0.0046	
M-6	MAE	0.3667	0.3687	0.3690	**0.3660**	0.3722	**0.3685**	0.0021	35 dk 30 sn
	RMSE	0.4686	0.4686	0.4739	**0.4664**	0.4721	0.4699	0.0027	
	MdAE	0.3022	0.3063	0.3102	0.3039	0.3151	0.3075	0.0046	
M-Hibrit-1	MAE	0.3677	0.3681	0.3688	0.3683	0.3711	0.3688	0.0012	37 dk 10 sn
	RMSE	0.4680	0.4668	0.4684	0.4691	0.4737	0.4692	0.0024	
	MdAE	0.3116	0.3111	0.3072	**0.3019**	0.3078	0.3079	0.0035	
M-Hibrit-2	MAE	0.3726	**0.3653**	**0.3661**	0.3694	0.3716	0.3690	0.0029	17 dk 25 sn
	RMSE	**0.4677**	0.4673	0.4697	0.4673	0.4715	0.4687	0.0017	
	MdAE	0.3180	0.3022	0.3043	0.3074	0.3144	0.3093	0.0060	
M-Hibrit-3	MAE	0.3700	0.3692	0.3679	0.3718	**0.3688**	0.3696	0.0013	10 dk 35 sn
	RMSE	**0.4677**	**0.4653**	**0.4684**	0.4680	**0.4696**	**0.4678**	0.0014	
	MdAE	0.3152	0.3176	0.3092	0.3204	0.3041	0.3133	0.0059	



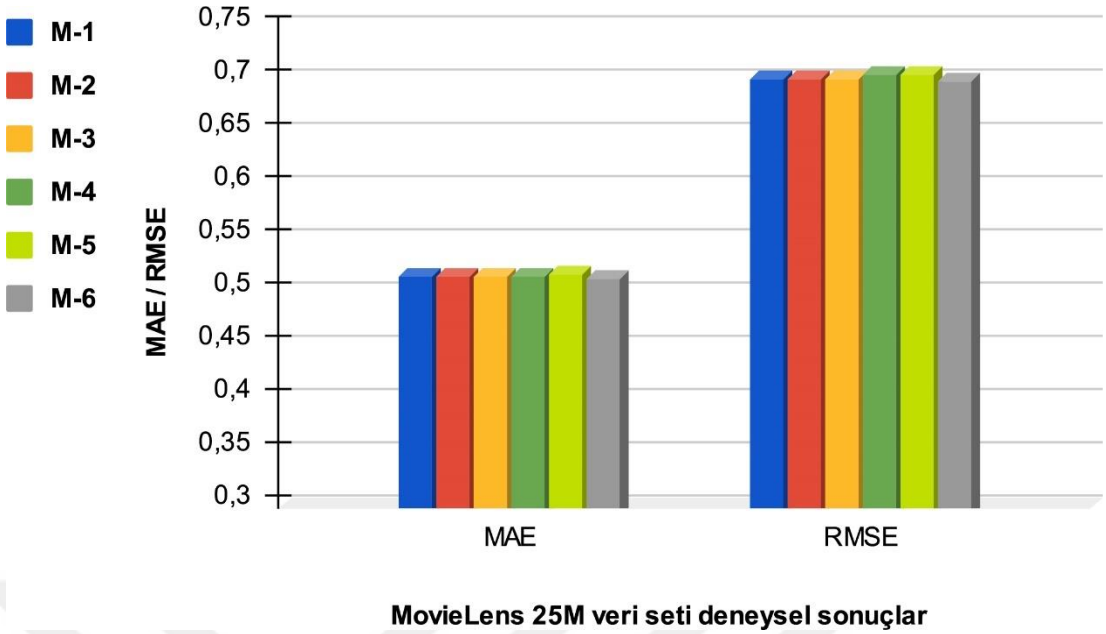
Şekil 4.4. Derin sinir ağ mimarilerinin test sonuçları (MAE, RMSE ve MdAE hata metriklerinin ortalama değerleriyle)

Şekil 4.4.'e göre MovieLens 10M veri seti geliştirilen 5-fold CV uygulamasında MAE metrik değerine göre M-6 en başarılı model olurken M-3 en başarısız model olmuştur. RMSE metrik değerine göre M-Hibrit-3 en başarılı model olurken M-4 en başarısız model olmuştur.

MovieLens 25M veri kümesiyle geliştirilen tavsiye sistemi derecelendirme uygulamasında M (1-6) derin sinir ağ mimarileri uygulanmıştır. Uygulama kapsamında veri kümesinin eğitim/doğrulama/test oranları 9(:1):1'dir. Deneysel sonuçlar ve ÇZ Çizelge 4.4.'de verilmiştir. M-5 modeli doğrulama veri kümesi ile gerçekleştirilen analiz çalışmasında 0.5227 MAE hata değeri üretirken, test veri kümesi gerçekleştirilen analiz çalışmasında 0.5146 MAE hata değeri üretmiştir. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, M-6 modeli 0.5112 MAE hata metrik değerine göre en başarılı model olurken, M-5 modeli 0.5146 MAE metrik değerine göre en düşük başarıya sahip model olmuştur (Çizelge 4.4.). Modellerin eğitim sürecindeki ÇZ incelendiğinde M-5 modelinin eğitimi 5 dk 42 sn, M-2 modelinin eğitimi 103 dk 14 sn'de tamamlanmıştır (Çizelge 4.4.). Derin öğrenme mimarileri ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.5.'de verilmiştir.

Çizelge 4.4. MovieLens 25M veri kümesi ile eğitilen derin sinir ağ mimarilerinin test süreci sonucundaki performansı (eğitim-doğrulama-test yaklaşımıyla)

Model	Doğrulama MAE	Test MAE	Test MSE	Test RMSE	Test MdAE	ÇZ
M-1	0.5249	0.5136	0.4875	0.6982	0.3831	50 dk 39 sn
M-2	0.5246	0.5127	0.4894	0.6996	0.3795	103 dk 14 sn
M-3	0.5238	0.5125	0.4900	0.7000	**0.3774**	33 dk 49 sn
M-4	0.5232	0.5136	0.4933	0.7024	0.3814	40 dk 19 sn
M-5	**0.5227**	0.5146	0.4951	0.7037	0.3871	5 dk 42 sn
M-6	0.5235	**0.5112**	**0.4862**	**0.6973**	0.3827	36 dk 10 sn



Şekil 4.5. Derin sinir ağ mimarilerinin test sonuçları (MAE ve RMSE hata metrikleriyle)

Şekil 4.5.'e göre MovieLens 25M veri seti geliştirilen uygulamada MAE metrik değerine göre M-6 en başarılı model olurken M-5 en başarısız model olmuştur. RMSE metrik değerine göre M-6 en başarılı model olurken M-5 en başarısız model olmuştur.

4.1.1.2 Derin öğrenme parametre seçimi

Derin öğrenme algoritmalarında kullanılan hiper-parametreler, deneysel çalışmalar sonucunda belirlenmiştir. Deneysel çalışma MovieLens 10M veri kümesi ve M-6 modeli ile gerçekleştirilmiştir.

M-6 modelinin eğitim sürecinin derleme (compile) aşamasında adam, adadelta, rmsprop, adagrad, sgd, nadam, adamax, optimizasyon algoritmaları kullanılmıştır. Ayrıca mse, mae loss fonksiyonların ve mse, mae metrik parametreleri kullanılmıştır. Deneysel sonuçlar Çizelge 4.5.'de verilmiştir.

M-6 modelinin eğitim sürecinde kullanılan diğer parametrelerden epoch ve batch size parametrelerin deneysel sonuçları Çizelge 4.6.'da verilmiştir.

Çizelge 4.5. M-6 modelinin optimal optimizasyon algoritması, loss ve metrik parametrelerinin belirlenmesi

Model	Kayıp	Metrik	Test MAE	Test MSE	Test RMSE	Test MdAE
Adam	**MSE**	**MAE**	**0.3663**	**0.2185**	**0.4675**	**0.3001**
Adam	MSE	MSE	0.3745	0.2233	0.4725	0.3175
Adam	MAE	MSE	0.3818	0.2295	0.4790	0.3146
Adam	MAE	MAE	0.3754	0.2312	0.4809	0.3082
RMSprop	MSE	MAE	0.3687	0.2234	0.4726	0.3024
Adadelta	MSE	MAE	0.4586	0.3219	0.5674	0.3855
Adagrad	MSE	MAE	0.4345	0.2898	0.5384	0.3727
Adamax	MSE	MAE	0.3698	0.2236	0.4729	0.2994
Nadam	MSE	MAE	0.3712	0.2217	0.4708	0.3142
SGD	MSE	MAE	0.3766	0.2272	0.4767	0.3161

Çizelge 4.5.'e göre M-6 modelinin derleme aşamasında optimizasyon, loss, metrik, iterasyon, batch size, learning rate değerlerinin sırasıyla adam, mse, mae, 100, 128, 0.001 olduğu çalışmada M-6 modeli test sürecinde 0.3663 MAE hata değeri ile en başarılı sonucu üretmiştir. M-6 modelinin derleme aşamasında optimizasyon, loss, metrik, iterasyon, batch size, learning rate değerlerinin sırasıyla adadelta, mse, mae, 100, 128, 0.001 olduğu çalışmada M-6 modeli test sürecinde 0.4586 MAE hata değeri ile en başarısız sonucu üretmiştir.

Çizelge 4.6. M-6 modelinin epoch, batch size parametrelerinin belirlenmesi

Epoch	MAE	MSE	RMSE	MdAE
20 Epoch	0.3862	0.2313	0.4810	0.3407
40 Epoch	0.3753	0.2303	0.4799	0.3091
60 Epoch	0.3705	0.2248	0.4741	0.3064
80 Epoch	0.3768	0.2363	0.4861	0.3041
100 Epoch	**0.3663**	**0.2185**	**0.4675**	**0.3001**
Batch size	**MAE**	**MSE**	**RMSE**	**MdAE**
16	0.3679	0.2235	0.4727	0.3109
32	0.3709	0.2237	0.4730	0.3055
64	0.3690	0.2192	0.4682	0.3172
128	**0.3663**	**0.2185**	**0.4675**	0.3001
256	0.3697	0.2267	0.4762	**0.2975**
512	0.3822	0.2343	0.4841	0.3095
1024	0.3744	0.2250	0.4743	0.3038

Çizelge 4.6.'ya göre M-6 modelinin derleme aşamasında optimizasyon, loss, metrik, iterasyon, batch size, learning rate değerlerinin sırasıyla adam, mse, mae, 100, 128, 0.001 olduğu çalışmada M-6 modeli test sürecinde 0.3663 MAE hata değeri ile

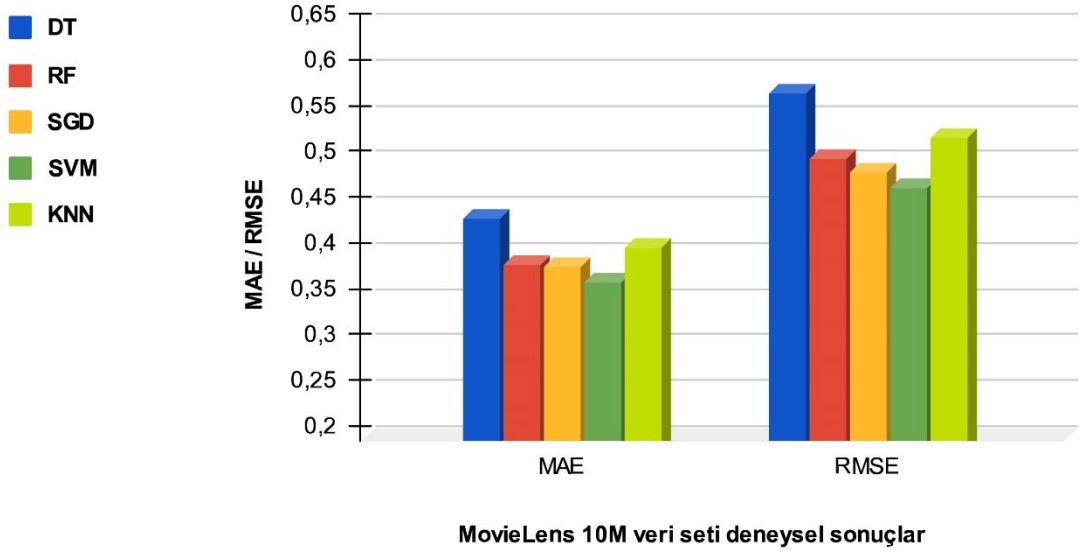
en başarılı sonucu üretmiştir. M-6 modelinin derleme aşamasında optimizasyon, loss, metrik, iterasyon, batch size, learning rate değerlerinin sırasıyla adam, mse, mae, 20, 128, 0.001 olduğu çalışmada M-6 modeli test sürecinde 0.3862 MAE hata değeriyle ve optimizasyon, loss, metrik, iterasyon, batch size, learning rate değerlerinin sırasıyla adam, mse, mae, 100, 512, 0.001 olduğu çalışmada M-6 modeli test sürecinde 0.3822 MAE hata değeri ile en başarısız sonuçları üretmiştir.

4.1.2. Makine öğrenmesi

MovieLens 10M veri kümesi kullanılarak DT, RF, SGD, SVM, KNN geleneksel makine öğrenme algoritmaları ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.7.'de verilmiştir. Makine öğrenme algoritmaları ile geliştirilen uygulamada eğitim ve test sürecinde veri kümesi 9:1 (eğitim/test) oranında ayrılmıştır. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, SVM 0.3662 MAE hata metrik değerine göre en başarılı model olurken, DT 0.4360 MAE metrik değerine göre en düşük başarıya sahip model olmuştur (Çizelge 4.7.). Geleneksel makine öğrenme algoritmalarının eğitim sürecindeki ÇZ incelendiğinde KNN 0.01 sn, RF 2 dk 9 sn'de eğitimi tamamlamıştır (Çizelge 4.7.). Makine öğrenme algoritmaları ile geliştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.6.'da verilmiştir.

Çizelge 4.7. MovieLens 10M veri kümesiyle geliştirilen derecelendirme uygulamasında geleneksel makine öğrenme algoritmalarının performansı

Model	MAE	MSE	RMSE	MdAE	ÇZ
DT	0.4360	0.3277	0.5724	0.3500	1.76 sn
RF	0.3852	0.2517	0.5017	0.3096	2 dk 9 sn
SGD	0.3869	0.2368	0.4866	0.3308	0.04 sn
SVM	**0.3662**	**0.2212**	**0.4703**	**0.3010**	4.44 sn
KNN	0.4056	0.2761	0.5255	0.3308	0.01 sn



Şekil 4.6. Geleneksel makine öğrenme algoritmalarının test sonuçları (MAE ve RMSE hata metrikleriyle)

Şekil 4.6.'a göre MovieLens 10M veri seti ve makine öğrenme algoritmaları ile geliştirilen uygulamada MAE metrik değerine göre SVM en başarılı model olurken DT en başarısız model olmuştur. RMSE metrik değerine göre SVM en başarılı model olurken DT en başarısız model olmuştur.

4.1.3. Geleneksel tavsiye sistemi yöntemleri

MovieLens 10M veri kümesi kullanılarak geleneksel tavsiye sistemi uygulamalarından CBF, CF yöntemleri ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.8.'de verilmiştir. CBF tekniği uygulamada 0.7563 MAE hata metrik değerini üretmiştir. İşbirlikçi tabanlı filtreleme tekniklerinden öğe tabanlı CF yöntemi 3.3956 MAE hata metrik değerini üretmiştir. İşbirlikçi tabanlı filtreleme tekniklerinden kullanıcı tabanlı CF yöntemi 3.3652 MAE hata metrik değerini üretmiştir (Çizelge 4.8.). Geleneksel tavsiye sistemi uygulamlarının çalışma zamanları incelendiğinde CBF yöntemi 420 dk, öğe tabanlı CF ve kullanıcı tabanlı CF 4 dk 41 sn'de eğitim süreçlerini tamamlamıştır.

Çizelge 4.8. MovieLens 10M veri kümesiyle geliştirilen derecelendirme uygulamasında CBF ve CF tekniklerinin performansı

Model	MAE	MSE	RMSE	MdAE	ÇZ
CBF	**0.7563**	**0.9170**	**0.9576**	**0.6213**	420 dk
Öğe Tabanlı CF	3.3956	12.7134	3.5656	3.5986	4 dk 41 sn
Kullanıcı Tabanlı CF	3.3652	12.5063	3.5364	3.5518	4 dk 41 sn

4.2. Goodbooks-10K Veri Seti

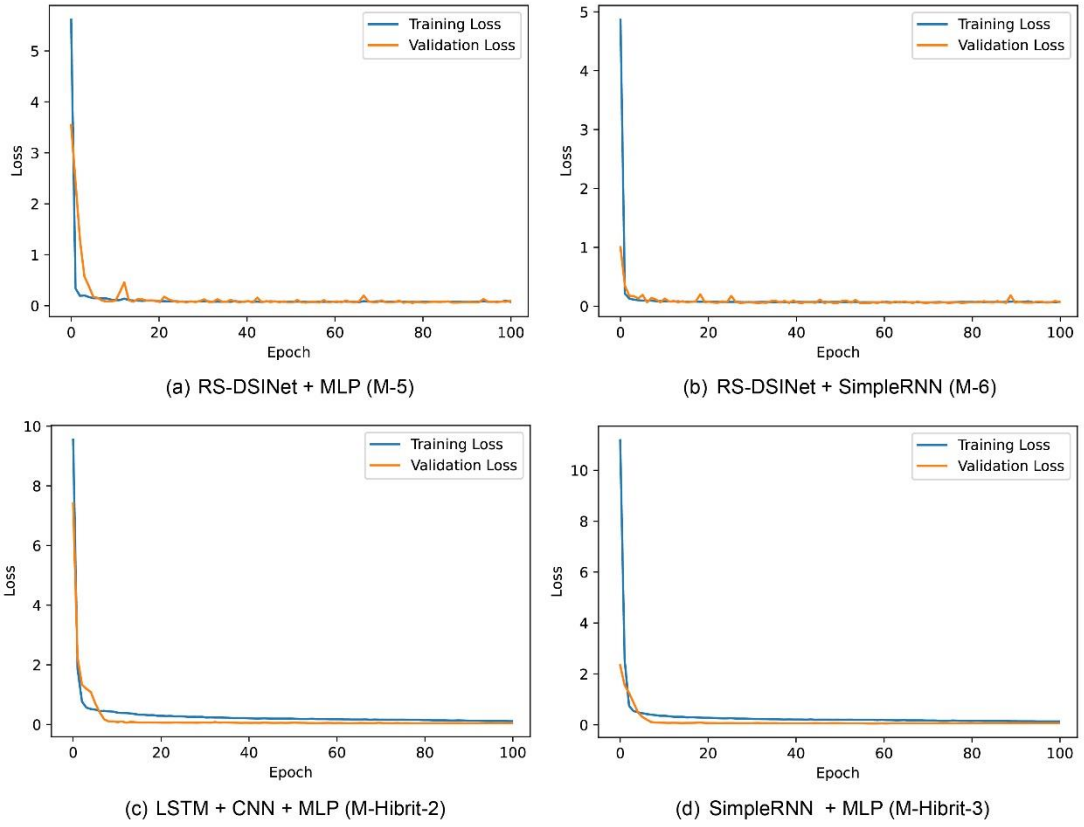
Kitap tavsiye uygulamalarında sıklıkla kullanılan Goodbooks-10K veri kümesi ile derin öğrenme modelleri ve makine öğrenme yöntemleri kullanılarak tavsiye sistemi derecelendirme uygulamaları geliştirilmiştir. Bu bölümde ampirik sonuçlar verilmiştir.

4.2.1. Derin öğrenme

Goodbooks-10K veri kümesi kullanılarak M (1-6) ve hibrit yöntemlerden M-Hibrit-1, M-Hibrit-2, M-Hibrit-3 derin sinir ağları ile geliştirilen kitap tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.9.'da verilmiştir. Derin sinir ağ mimarilerin eğitim ve test sürecinde veri kümesi 9(:1):1 (eğitim/doğrulama/test) oranında ayrılmıştır. M-6 modeli doğrulama veri kümesi ile gerçekleştirilen analiz çalışmasında 0.1927 MAE hata değeri üretirken, test veri kümesi gerçekleştirilen analiz çalışmasında 0.1939 MAE hata değeri üretmiştir. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, M-Hibrit-3 hibrit modeli 0.1916 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.2717 MAE metrik değerine göre en düşük başarıya sahip model olmuştur (Çizelge 4.9.). Modellerin eğitim sürecindeki ÇZ incelendiğinde M-5 modelinin eğitimi 1 dk 34 sn, M-2 modelinin eğitimi 17 dk 15 sn'de tamamlanmıştır (Çizelge 4.9.). Goodbooks-10K veri kümesi ile eğitilen derin sinir ağlarının eğitim sürecindeki kayıp (training loss) ve doğrulama kaybı (validation loss) grafikleri Şekil 4.7.'de, MAE ve validation MAE grafikleri Şekil 4.8.'de verilmiştir. Derin sinir ağları ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.9.'da verilmiştir.

Çizelge 4.9. Goodbooks-10K veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama ve test veri kümeleri ile gerçekleştirilen çalışmanın ampirik sonuçları (eğitim-doğrulama-test yaklaşımıyla)

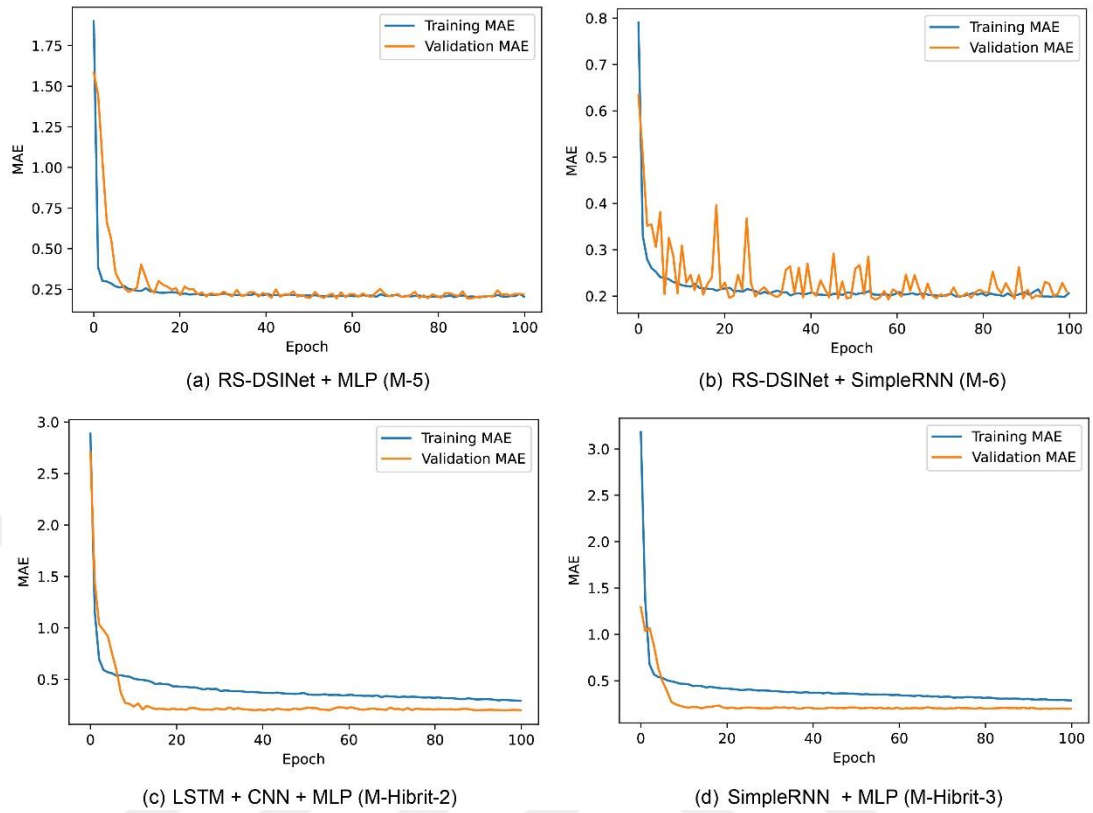
Model	Doğrulama MAE	MAE	MSE	RMSE	MdAE	ÇZ
M-1	0.2276	0.2334	0.0849	0.2913	0.1997	14 dk 1 sn
M-2	0.2076	0.2119	0.0720	0.2683	0.1761	17 dk 15 sn
M-3	0.2843	0.2717	0.1680	0.4098	0.2260	7 dk 2sn
M-4	0.1964	0.2001	0.0653	0.2555	0.1623	12 dk 51 sn
M-5	0.1965	0.1941	0.0638	0.2525	0.1533	1 dk 34 sn
M-6	**0.1927**	0.1939	0.0632	0.2513	0.1484	7 dk 4 sn
M-Hibrit-1	0.1959	0.1956	0.0630	0.2510	0.1599	5 dk 58 sn
M-Hibrit-2	0.1977	0.1935	0.0631	0.2513	0.1585	3 dk 35 sn
M-Hibrit-3	0.1948	**0.1916**	**0.0626**	**0.2503**	**0.1531**	1 dk 47 sn



Şekil 4.7. Goodbooks-10K eğitim veri setinde (a) M-5, (b) M-6, (c) M-Hibrit-2, (d) M-Hibrit-3 derin sinir ağ mimarilerinin training loss ve validation loss grafikleri

Şekil 4.7. (a)'ya göre M-5 modelinin en düşük loss değeri 0.0657, en yüksek loss değeri 5.6171, en düşük validation loss değeri 0.0626, en yüksek validation loss değeri

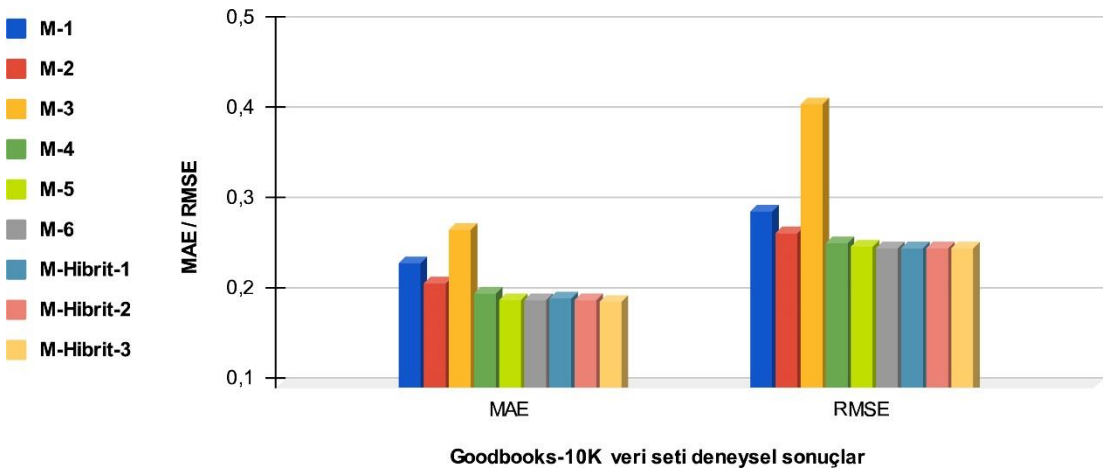
3.5478'dir. Loss değerinde 12. döngüye kadar genellikle azalma gözlemlenirken, 13. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 84. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 9. döngüye kadar önemli bir azalma gözlemlenirken, 10. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 78. döngüde elde etmiştir. Şekil 4.7. (b)'ye göre M-6 modelinin en düşük loss değeri 0.0651, en yüksek loss değeri 4.8705, en düşük validation loss değeri 0.0620, en yüksek validation loss değeri 1.0044'tür. Loss değerinde 13. döngüye kadar önemli bir azalma gözlemlenmiştir. 14. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 94. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 3. döngüye kadar önemli bir azalma gözlemlenirken, 4. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 56. döngüde elde etmiştir. Şekil 4.7. (c)'ye göre M-Hibrit-2 modelinin en düşük loss değeri 0.1334, en yüksek loss değeri 9.5511, en düşük validation loss değeri 0.0648, en yüksek validation loss değeri 7.4190'dır. Loss değerinde 7. döngüye kadar önemli bir azalma gözlemlenmiştir. 8. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlense de model genellikle azalış eğilimindedir. 100. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 11. döngüye kadar önemli bir azalma gözlemlenirken, 12. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation loss değerini 95. döngüde elde etmiştir. Şekil 4.7. (d)'ye göre M-Hibrit-3 modelinin en düşük loss değeri 0.1334, en yüksek loss değeri 11.1787, en düşük validation loss değeri 0.0635, en yüksek validation loss değeri 2.3442'dir. Loss değerinde 16. döngüye kadar önemli bir azalma gözlemlenmiştir. 17. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlense de model genellikle azalış eğilimindedir. 100. döngüde model en düşük loss değerini üretmiştir. Validation loss değerinde 12. döngüye kadar önemli bir azalma gözlemlenirken, 13. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Ayrıca 50. döngüden sonra validation loss grafiğinde önemli bir değişiklik gözlemlenmemiştir. Model en düşük validation loss değerini 93. döngüde elde etmiştir.



Şekil 4.8. Goodbooks-10K eğitim veri setinde (a) M-5, (b) M-6, (c) M-Hibrit-2, (d) M-Hibrit-3 derin sinir ağ mimarilerinin training MAE ve validation MAE grafikleri

Şekil 4.8. (a)'ya göre M-5 modelinin en düşük mae hata değeri 0.1996, en yüksek mae hata değeri 1.9004, en düşük validation mae hata değeri 0.1961, en yüksek validation mae hata değeri 1.5838'dir. MAE değerinde 7. döngüye kadar önemli bir azalma gözlemlenmiştir. 8. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 84. döngüde model en düşük MAE hata değerini üretmiştir. Validation mae değerinde 9. döngüye kadar önemli bir azalma gözlemlenirken, 10. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 88. döngüde elde etmiştir. Şekil 4.8. (b)'ye göre M-6 modelinin en düşük mae değeri 0.1980, en yüksek MAE değeri 0.7906, en düşük validation mae değeri 0.1927, en yüksek validation MAE değeri 0.6347'dir. MAE değerinde 13. döngüye kadar önemli bir azalma gözlemlenmiştir. 14. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. 99. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 3. döngüye kadar önemli bir azalma

gözlemlenirken, 4 döngüden sonra fonksiyon grafiğinde farklı artış ve azalış oranları gözlemlenmiştir. Model en düşük validation MAE hata değerini 56. döngüde elde etmiştir. Şekil 4.8. (c)'ye göre M-Hibrit-2 modelinin en düşük MAE değeri 0.2887, en yüksek Mae değeri 2.8880, en düşük validation MAE değeri 0.1977, en yüksek validation MAE değeri 2.7074'dir. MAE değerinde 7. döngüye kadar önemli bir azalma gözlemlenmiştir. 8. döngüden fonksiyon grafiğinde farklı oranlarda artış ve azalışlar gözlemlense de model genellikle azalış eğilimindedir. 100. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 11. döngüye kadar önemli bir azalma gözlemlenirken, 12. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış ve azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 100. döngüde elde etmiştir. Şekil 4.8. (d)'ye göre M-Hibrit-3 modelinin en düşük MAE değeri 0.2899, en yüksek MAE değeri 3.1817, en düşük validation MAE değeri 0.1948, en yüksek validation MAE değeri 1.2917'dir. MAE değerinde 13. döngüye kadar önemli bir azalma gözlemlenmiştir. 14. döngüden sonra fonksiyon grafiğinde artış ve azalışlar gözlemlenmiştir. 100. döngüde model en düşük MAE hata değerini üretmiştir. Validation MAE değerinde 13. döngüye kadar önemli bir azalma gözlemlenirken, 14. döngüden sonra fonksiyon grafiğinde farklı oranlarda artış veya azalışlar gözlemlenmiştir. Model en düşük validation MAE hata değerini 93. döngüde elde etmiştir.



Şekil 4.9. Derin sinir ağlarının test sonuçları (MAE ve RMSE hata metrikleriyle)

Şekil 4.9.'a göre Goodbooks-10K veri seti geliştirilen uygulamada MAE metrik değerine göre M-Hibrit-3 en başarılı model olurken M-3 en başarısız model olmuştur.

RMSE metrik değerine göre M-Hibrit-3 en başarılı model olurken M-3 en başarısız model olmuştur.

4.2.1.1 5 katlı çapraz doğrulama

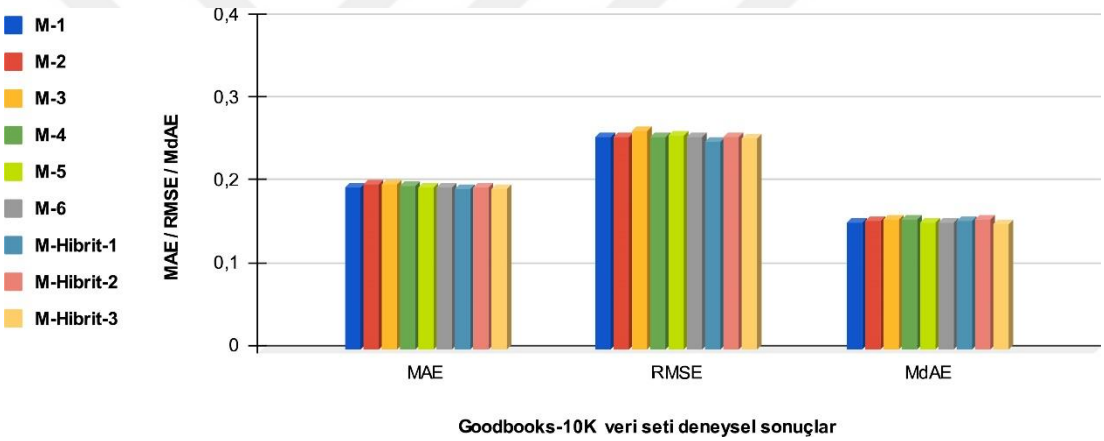
Kitap tavsiye sistemi derecelendirme probleminde önerilen M (1-6) derin sinir ağlarının ve M-Hibrit-1, M-Hibrit-2, M-Hibrit-3 hibrit yöntemlerinin detaylı performans analizi için 5-fold CV yöntemi uygulanmıştır. Deneysel sonuçlar ve ÇZ Çizelge 4.10.'da verilmiştir. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, MAE metriğinin ortalama (ORT) değerine göre M-Hibrit-3 en başarılı model olmuştur. M-Hibrit-3 Fold 1'de 0.1933, Fold 2'de 0.1945, Fold 3'de 0.1953, Fold 4'de 0.1946 ve Fold 5'de 0.1945 MAE hata değeri üretmiştir. M-Hibrit-3 modelinin ORT MAE hata değeri 0.1944, MAE metrik değerine göre standart sapması (SS) 0.0007'dir. M-3 modeli 0.1996 ORT MAE metrik değeriyle en düşük başarıya sahip model olmuştur. Fold 1'de 0.2098, Fold 2'de 0.1999, Fold 3'de 0.1929, Fold 4'de 0.1931 ve Fold 5'de 0.2023 MAE hata değeri üretmiştir. M-3 modelinin MAE metrik değerine göre standart sapması (SS) 0.0063'tür. Modellerin eğitim sürecindeki ÇZ incelendiğinde M-5 modelinin eğitimi 8 dk 35 sn, M-2 modelinin eğitimi 85 dk 40 sn'de tamamlanmıştır. Derin öğrenme modelleri ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE, RMSE, MdAE hata metrik değerlerinin sütun grafiği Şekil 4.10.'da verilmiştir.

Çizelge 4.10. 5-fold cross validation yöntemiyle Goodbooks-10K veri kümesi kullanılarak önerilen M (1-6), M-Hibrit (1-3) modellerin performansı

Model	Metrik	F1	F2	F3	F4	F5	Ort	SS	ÇZ
M-1	MAE	0.1953	0.1954	0.1991	**0.1944**	0.2000	0.1968	0.0023	75 dk 29 sn
	RMSE	**0.2530**	0.2532	0.2631	0.2576	0.2567	0.2567	0.0037	
	MdAE	0.1512	0.1587	0.1531	**0.1474**	0.1593	0.1540	0.0045	
M-2	MAE	0.2056	**0.1937**	0.2008	0.1954	0.2016	0.1994	0.0043	85 dk 40 sn
	RMSE	0.2638	**0.2531**	0.2611	0.2540	0.2578	0.2579	0.0041	
	MdAE	0.1612	**0.1483**	0.1576	0.1488	0.1609	0.1554	0.0057	
M-3	MAE	0.2098	0.1999	**0.1929**	0.1931	0.2023	0.1996	0.0063	26 dk 5 sn
	RMSE	0.2653	0.2716	**0.2502**	0.2522	0.2854	0.2649	0.0130	
	MdAE	0.1794	0.1525	0.1550	0.1555	**0.1482**	0.1581	0.0109	
M-4	MAE	**0.1946**	0.2026	0.1976	0.1988	0.1993	0.1986	0.0026	57 dk 54 sn
	RMSE	**0.2516**	0.2595	0.2585	0.2591	0.2595	0.2576	0.0030	
	MdAE	0.1592	0.1627	**0.1525**	0.1550	0.1580	0.1575	0.0035	

Çizelge 4.10. (**devam**)

Model	Metrik	F1	F2	F3	F4	F5	Ort	SS	ÇZ
M-5	MAE	**0.1926**	0.1988	0.2008	0.1943	0.1965	0.1966	0.0030	8 dk 35 sn
	RMSE	**0.2501**	0.2643	0.2694	0.2544	0.2585	0.2593	0.0069	
	MdAE	0.1528	**0.1518**	0.1545	0.1542	0.1522	0.1531	0.0011	
M-6	MAE	0.1967	0.1957	0.2033	0.1938	**0.1936**	0.1966	0.0035	32 dk 32 sn
	RMSE	0.2536	0.2546	0.2710	**0.2501**	**0.2532**	0.2565	0.0074	
	MdAE	**0.1510**	0.1513	0.1557	0.1519	0.1549	0.1530	0.0020	
M-Hibrit-1	MAE	0.1950	0.1942	0.1944	**0.1934**	0.1962	0.1947	0.0009	28 dk 10 sn
	RMSE	**0.2507**	0.2516	0.2531	0.2531	0.2536	**0.2524**	0.0011	
	MdAE	0.1603	**0.1520**	0.1566	0.1542	0.1562	0.1558	0.0028	
M-Hibrit-2	MAE	0.1978	0.1965	0.1974	**0.1954**	0.1963	0.1967	0.0009	9 dk 56 sn
	RMSE	0.2557	**0.2544**	0.2592	0.2566	0.2559	0.2564	0.0016	
	MdAE	0.1605	0.1576	0.1556	**0.1536**	0.1576	0.1570	0.0023	
M-Hibrit-3	MAE	**0.1933**	0.1945	0.1953	0.1946	0.1945	**0.1944**	0.0007	8 dk 50 sn
	RMSE	0.2541	0.2566	0.2580	**0.2521**	0.2555	0.2553	0.0020	
	MdAE	0.1527	0.1520	**0.1493**	0.1558	0.1520	**0.1524**	0.0021	



Şekil 4.10. Derin sinir ağ mimarilerinin test sonuçları (MAE, RMSE ve MdAE hata metriklerinin ortalama değerleriyle)

Şekil 4.10.'a göre Goodbooks-10K veri seti geliştirilen 5-fold CV uygulamasında MAE metrik değerine göre M-Hibrit-3 en başarılı model olurken M-3 en başarısız model olmuştur. RMSE metrik değerine göre M-Hibrit-1 en başarılı model olurken M-3 en başarısız model olmuştur.

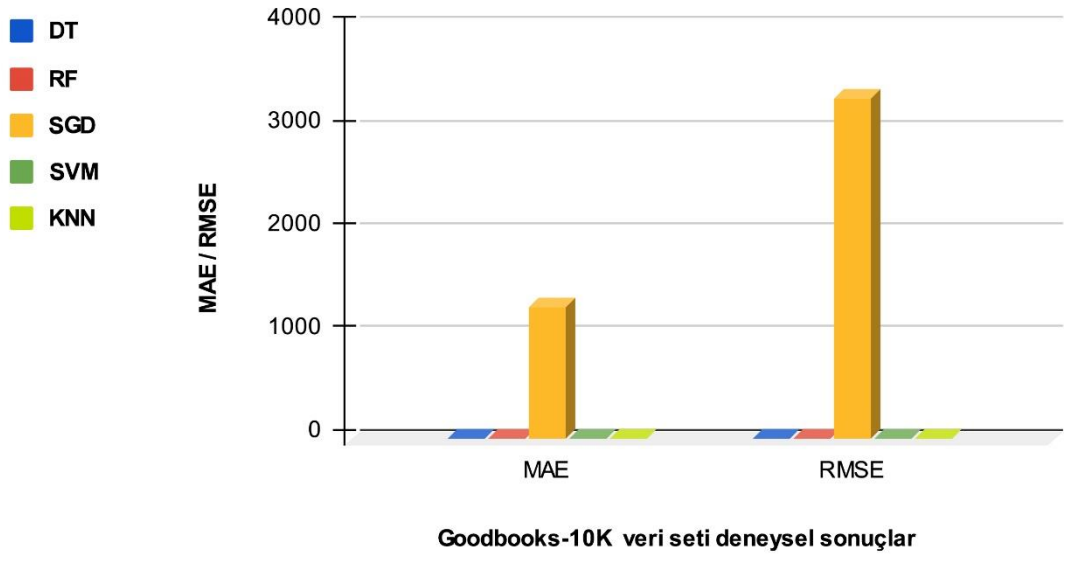
4.2.2. Makine öğrenmesi

Goodbooks-10K veri kümesi kullanılarak DT, RF, SGD, SVM, KNN geleneksel makine öğrenme algoritmaları ile geliştirilen kitap tavsiye sistemi derecelendirme

uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.11.'de verilmiştir. Makine öğrenme algoritmaları ile geliştirilen uygulamada eğitim ve test sürecinde veri kümesi 9:1 (eğitim/test) oranında ayrılmıştır. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, RF 0.1781 MAE hata metrik değerine göre en başarılı model olurken, SGD 1285.5319 MAE metrik değeri ile tez kapsamında en başarısız sonucu üretmiştir (Çizelge 4.11.). Geleneksel makine öğrenme algoritmalarının eğitim sürecindeki ÇZ incelendiğinde KNN 0.01 sn, RF 2 dk 33 sn'de eğitimi tamamlamıştır (Çizelge 4.11.). Makine öğrenme algoritmaları ile geliştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.11.'de gösterilmiştir.

Çizelge 4.11. Goodbooks-10K veri kümesiyle geliştirilen derecelendirme uygulamasında makine öğrenme algoritmalarının performansı

Model	MAE	MSE	RMSE	MdAE	ÇZ
DT	0.2448	0.1037	0.3220	0.1900	2sn
RF	**0.1781**	**0.0539**	**0.2322**	**0.1406**	2 dk 33 sn
SGD	1285.5319	1084.0000	3293.2626	5633.8306	0.06 sn
SVM	0.1981	0.0656	0.2561	0.1597	4 sn
KNN	0.2019	0.0673	0.2595	0.1637	0.01 sn



Şekil 4.11. Makine öğrenme algoritmalarının test sonuçları (MAE ve RMSE hata metrikleri)

Şekil 4.11.'e göre Goodbooks-10K veri seti ve makine öğrenme algoritmaları ile geliştirilen uygulamada MAE metrik değerine göre SVM en başarılı model olurken

SGD en başarısız model olmuştur. RMSE metrik değerine göre RF en başarılı model olurken SGD en başarısız model olmuştur.

4.3. FilmTV Movies Veri Seti

Film tavsiye uygulamalarında sıklıkla kullanılan veri setlerinden FilmTV Movies veri kümesi ile derin sinir ağları ve makine öğrenme yöntemleri kullanılarak tavsiye sistemi derecelendirme uygulamaları geliştirilmiştir. Bu bölümde ampirik sonuçlar verilmiştir.

4.3.1. Derin öğrenme

FilmTV Movies veri kümesi kullanılarak M (1-6) derin sinir ağları ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.12.'de verilmiştir. Derin öğrenme modellerinin eğitim ve test sürecinde veri kümesi 9(:1):1 (eğitim/doğrulama/test) oranında ayrılmıştır. M-1 modeli doğrulama veri kümesinde 0.6807 MAE, test veri kümesinde 0.7584 MAE hata değeri üretmiştir. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, M-2 modeli 0.7135 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.8134 MAE metrik değerine göre en düşük başarıya sahip model olmuştur (Çizelge 4.12.). Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-5 modeli 5 dk 12 sn, M-6 modeli 32 dk 18 sn'de eğitim sürecini tamamlamıştır (Çizelge 4.12.).

Çizelge 4.12. FilmTV Movies veri kümesiyle gerçekleştirilen derecelendirme uygulamasında derin öğrenme modellerinin performansı (eğitim-doğrulama-test yaklaşımıyla)

Model	Doğrulama MAE	Test MAE	Test MSE	Test RMSE	Test MdAE	ÇZ
M-1	**0.6807**	0.7584	0.9110	0.9545	0.6292	8 dk 33 sn
M-2	0.6849	**0.7135**	**0.8424**	**0.9178**	0.6166	16 dk 54 sn
M-3	0.8142	0.8134	1.0580	1.0286	0.7089	27 dk 24 sn
M-4	0.7259	0.7222	0.8477	0.9207	0.5869	8 dk 15 sn
M-5	0.7075	0.7278	0.8690	0.9322	**0.5859**	5 dk 12 sn
M-6	0.7712	0.7488	0.8802	0.9382	0.6460	32 dk 18 sn

4.3.2. Makine öğrenmesi

FilmTV Movies veri kümesi kullanılarak DT, RF, SGD, SVM, KNN geleneksel makine öğrenme algoritmaları ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.13.'de verilmiştir. Geleneksel makine öğrenme algoritmaları ile geliştirilen uygulamada eğitim ve test sürecinde veri kümesi 9:1 (eğitim/test) oranında ayrılmıştır. Eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, SVM 0.7232 MAE hata metrik değerine göre en başarılı model olurken, DT 1.3720 MAE metrik değerine göre en düşük başarıya sahip model olmuştur (Çizelge 4.13.). Geleneksel makine öğrenme algoritmalarının eğitim sürecindeki çalışma zamanları incelendiğinde KNN 0.04 sn, RF 5 dk 55 sn'de eğitimi tamamlamıştır (Çizelge 4.13.).

Çizelge 4.13. FilmTV Movies veri kümesiyle geliştirilen derecelendirme uygulamasında makine öğrenme algoritmalarının performansı

Model	MAE	MSE	RMSE	MdAE	ÇZ
DT	1.3720	3.0284	1.7402	1.2000	4.40 sn
RF	1.0692	1.7424	1.3200	0.9450	4 dk 55 sn
SGD	0.7308	**0.8261**	**0.9089**	0.6417	9 sn
SVM	**0.7232**	0.8399	0.9164	**0.5936**	1 dk 56 sn
KNN	0.8160	1.0238	1.0118	0.7097	0.04 sn

4.4. IMDb Movies Extensive Veri Seti

IMDb Movies Extensive veri kümesi kullanılarak derin sinir ağları ve makine öğrenme, algoritmaları ile tavsiye sistemi derecelendirme uygulamaları geliştirilmiştir. Bu bölümde ampirik sonuçlar verilmiştir.

4.4.1. Derin öğrenme

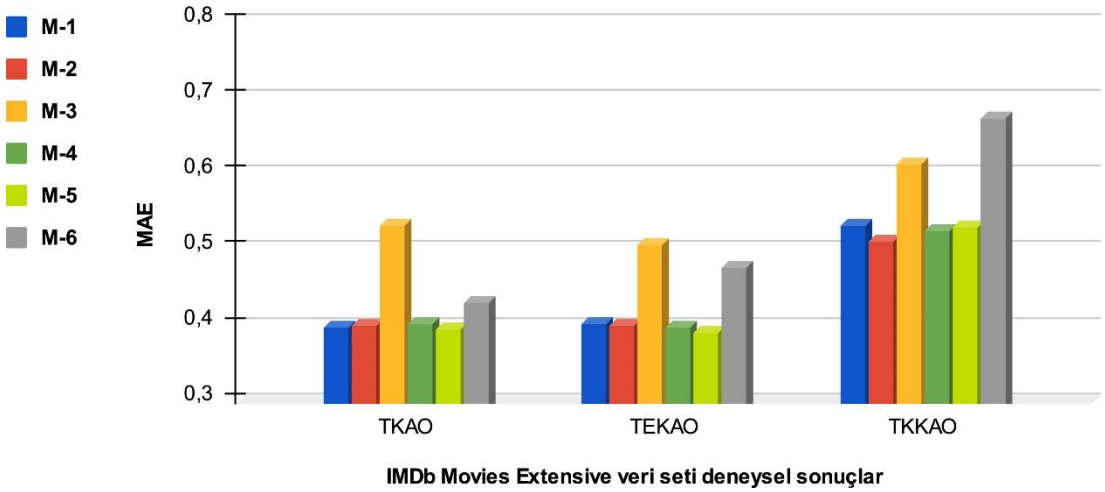
IMDb Movies Extensive veri kümesindeki tüm kullanıcıların ağırlıklı oy ortalamaları, tüm erkek/kadın kullanıcıların ağırlıklı oy ortalamaları, 18, 30, 45 yaş grubundaki tüm kullanıcılar tarafından filmlere verilen oyların ortalaması ve son olarak her yaş grubundaki erkek/kadın kullanıcıları tarafından filmlere verilen oyların

ortalaması kullanılarak M (1-6) derin sinir ağları ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları bu bölümde verilmiştir. Derin sinir ağ mimarilerin eğitim ve test sürecinde veri kümesi 9(:1):1 (eğitim/doğrulama/test) oranında ayrılmıştır.

Tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO), tüm erkek kullanıcıların ağırlıklı oy (TEKAO) ortalamaları ve tüm kadın kullanıcıların ağırlıklı oy (TKKAO) ortalamaları kullanılarak geliştirilen uygulamanın deneysel sonuçları Çizelge 4.14.'de verilmiştir. Mimarilerin eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, TKAO ortalamasıyla geliştirilen uygulamada M-5 modeli 0.3929 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.5307 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TEKAO ortalamasıyla geliştirilen uygulamada M-5 modeli 0.3883 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.5037 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TKKAO ortalamasıyla geliştirilen uygulamada M-2 modeli 0.5093 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.6710 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Derin sinir ağları ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.12.'de verilmiştir.

Çizelge 4.14. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

Model	**Test MAE**			**Test RMSE**		
	TKAO	**TEKAO**	**TKKAO**	**TKAO**	**TEKAO**	**TKKAO**
M-1	0.3967	0.3994	0.5311	0.5128	0.5114	0.6830
M-2	0.3978	0.3982	**0.5093**	0.5088	0.5101	**0.6613**
M-3	0.5307	0.5037	0.6112	0.6817	0.6565	0.7941
M-4	0.3993	0.3949	0.5238	0.5112	0.5022	0.6778
M-5	**0.3929**	**0.3883**	0.5286	**0.5068**	**0.4996**	0.6712
M-6	0.4276	0.4743	0.6710	0.5364	0.5923	0.8263



Şekil 4.12. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle)

Şekil 4.12.'e göre TKAO kullanılarak geliştirilen uygulamada MAE metrik değerine göre M-5 modeli en başarılı model olurken M-3 en başarısız model olmuştur. TEKAO kullanılarak geliştirilen uygulamada MAE metrik değerine göre M-5 en başarılı model olurken M-3 en başarısız model olmuştur. TKKAO kullanılarak geliştirilen uygulamada MAE metrik değerine göre M-2 en başarılı model olurken M-6 en başarısız model olmuştur.

Tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO), tüm erkek kullanıcıların ağırlıklı oy (TEKAO) ortalamaları ve tüm kadın kullanıcıların ağırlıklı oy (TKKAO) ortalamaları kullanılarak geliştirilen uygulamada derin öğrenme modellerinin eğitim sürecindeki doğrulama sonuçları ve ÇZ Çizelge 4.15.'de verilmiştir. TKAO ortalamasıyla geliştirilen uygulamada M-2 modeli 0.3856 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.4374 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 18 dk 6 sn, M-3 modelinin eğitimi 27 dk 5 sn'de tamamlanmıştır. TEKAO ortalamasıyla geliştirilen uygulamada M-2 modeli 0.3806 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.4634 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 17 dk 24 sn, M-3 modelinin eğitimi 27 dk 31 sn'de tamamlanmıştır. TKKAO ortalamasıyla geliştirilen uygulamada M-2 modeli 0.4628 MAE hata metrik değerine

göre en başarılı model olurken, M-6 modeli 0.5612 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 16 dk 54 sn, M-6 modelinin eğitimi 27 dk 33 sn'de tamamlanmıştır.

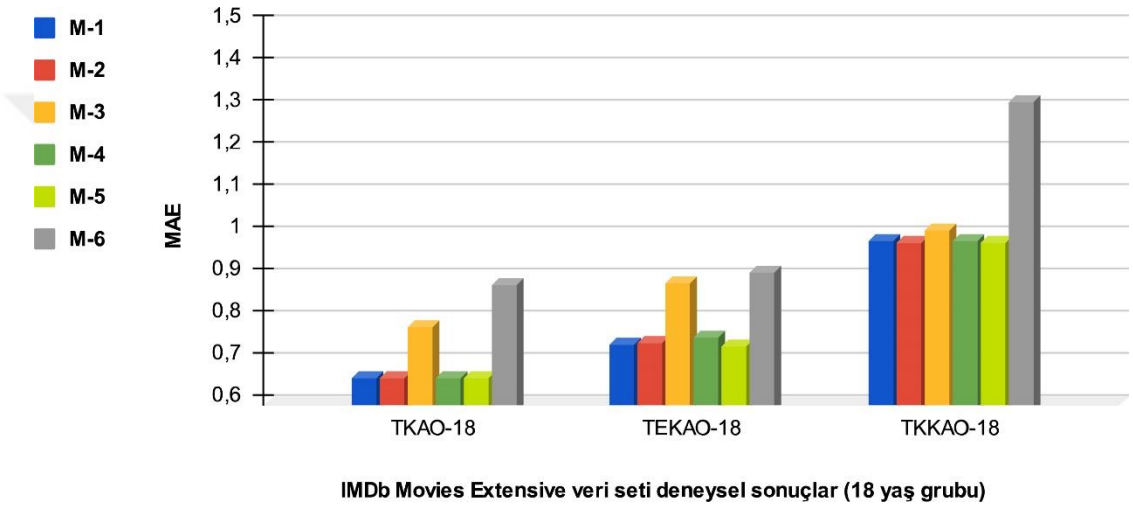
Çizelge 4.15. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama (eğitim sırasındaki test verileri) veri kümeleri ile yapılan test çalışmasının deneysel sonuçları ve çalışma zamanı (tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

Model	Doğrulama MAE			Çalışma Zamanı		
	TKAO	TEKAO	TKKAO	TKAO	TEKAO	TKKAO
M-1	0.3870	0.3917	0.4804	9 dk 2 sn	8 dk 50 sn	9 dk 3 sn
M-2	**0.3856**	**0.3806**	**0.4628**	18 dk 6 sn	17 dk 24 sn	16 dk 54 sn
M-3	0.4374	0.4634	0.5298	27 dk 5 sn	27 dk 31 sn	27 dk 26 sn
M-4	0.3924	0.3926	0.4889	8 dk 14 sn	8 dk 16 sn	8 dk 18 sn
M-5	0.4046	0.3845	0.4892	4 dk 1 sn	3 dk 58 sn	3 dk 59 sn
M-6	0.4033	0.4251	0.5612	29 dk 43 sn	27 dk 17 sn	27 dk 33 sn

18 yaş grubu tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO-18), 18 yaş grubu erkek kullanıcıların ağırlıklı oy (TEKAO-18) ortalamaları ve 18 yaş grubu kadın kullanıcıların ağırlıklı oy (TKKAO-18) ortalamaları kullanılarak derin öğrenme mimarileri ile geliştirilen uygulamanın deneysel sonuçları Çizelge 4.16.'da verilmiştir. Önerilen mimarilerin eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, TKAO-18 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.6551 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.8782 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TEKAO-18 ortalamasıyla geliştirilen uygulamada M-5 modeli 0.7305 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.9066 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TKKAO-18 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.9784 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 1.3118 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Derin sinir ağları ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.13.'de verilmiştir.

Çizelge 4.16. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (18 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

Model	Test MAE			Test RMSE		
	TKAO-18	TEKAO-18	TKKAO-18	TKAO-18	TEKAO-18	TKKAO-18
M-1	0.6563	0.7346	0.9819	0.8696	1.0114	1.3107
M-2	**0.6551**	0.7391	**0.9784**	0.8696	1.0208	**1.3020**
M-3	0.7743	0.8787	1.0055	1.0053	1.1548	1.3318
M-4	0.6567	0.7512	0.9808	0.8693	1.0317	1.3200
M-5	0.6571	**0.7305**	0.9786	**0.8654**	**1.0061**	1.3074
M-6	0.8782	0.9066	1.3118	1.0936	1.1712	1.6552



Şekil 4.13. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle)

Şekil 4.13.'e göre TKAO-18 geliştirilen uygulamada MAE metrik değerine göre M-1 en başarılı model olurken M-6 en başarısız model olmuştur. TEKAO-18 geliştirilen uygulamada MAE metrik değerine göre M-5 en başarılı model olurken M-6 en başarısız model olmuştur. TKKAO-18 geliştirilen uygulamada MAE metrik değerine göre M-5 en başarılı model olurken M-6 en başarısız model olmuştur.

18 yaş grubu tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO-18), 18 yaş grubu erkek kullanıcıların ağırlıklı oy (TEKAO-18) ortalamaları ve 18 yaş grubu kadın kullanıcıların ağırlıklı oy (TKKAO-18) ortalamaları kullanılarak geliştirilen uygulamanın doğrulama sonuçları ve ÇZ Çizelge 4.17.'de verilmiştir. TKAO-18 ortalamasıyla geliştirilen uygulamada M-4 modeli 0.6247 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.7569 MAE metrik değerine göre en

düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-4 modelinin eğitimi 8 dk 15 sn, M-6 modelinin eğitimi 25 dk 18 sn'de tamamlanmıştır. TEKAO-18 ortalamasıyla geliştirilen uygulamada M-1 modeli 0.6564 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.7785 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-1 modelinin eğitimi 10 dk 2 sn, M-6 modelinin eğitimi 28 dk 57 sn'de tamamlanmıştır. TKKAO-18 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.9316 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 1.0916 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 17 dk 1 sn, M-6 modelinin eğitimi 30 dk 42 sn'de tamamlanmıştır.

Çizelge 4.17. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama sonuçları (eğitim sırasındaki test verileri) ve çalışma zamanı (18 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

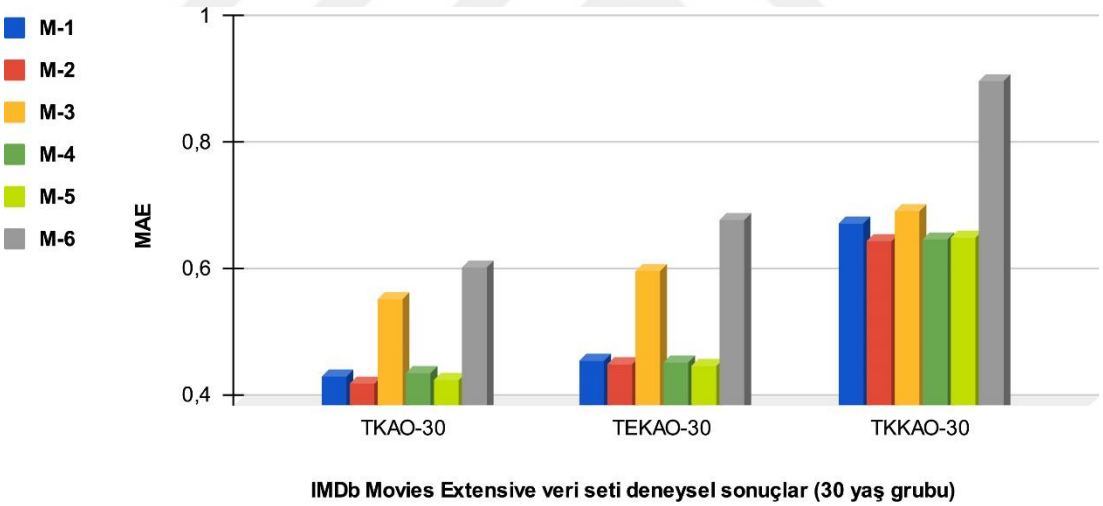
Model	Doğrulama MAE			Çalışma Zamanı		
	TKAO-18	TEKAO-18	TKKAO-18	TKAO-18	TEKAO-18	TKKAO-18
M-1	0.6296	**0.6564**	0.9370	10 dk 1 sn	10 dk 2 sn	10 dk 7 sn
M-2	0.6286	0.6599	**0.9316**	17 dk 17 sn	16 dk 56 sn	17 dk 1 sn
M-3	0.6871	0.7554	0.9958	27 dk 20 sn	27 dk 11 sn	35 dk 30 sn
M-4	**0.6247**	0.6633	0.9538	8 dk 15 sn	8 dk 20 sn	8 dk 8 sn
M-5	0.6412	0.6641	0.9387	1 dk 45 sn	1 dk 43 sn	1 dk 44 sn
M-6	0.7569	0.7785	1.0916	25 dk 18 sn	28 dk 57 sn	30 dk 42 sn

30 yaş grubu tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO-30), 30 yaş grubu erkek kullanıcıların ağırlıklı oy (TEKAO-30) ortalamaları ve 30 yaş grubu kadın kullanıcıların ağırlıklı oy (TKKAO-30) ortalamaları kullanılarak geliştirilen uygulamanın deneysel sonuçları Çizelge 4.18.'de verilmiştir. Mimarilerin eğitim sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, TKAO-30 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.4296 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.6127 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TEKAO-30 ortalamasıyla geliştirilen uygulamada M-5 modeli 0.4558 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.6869 MAE metrik değerine göre en düşük başarıya sahip model

olmuştur. TKKAO-30 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.6551 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.9081 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Derin öğrenme modelleri ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.14.'de verilmiştir.

Çizelge 4.18. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (30 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

Model	Test MAE			Test RMSE		
	TKAO-30	TEKAO-30	TKKAO-30	TKAO-30	TEKAO-30	TKKAO-30
M-1	0.4409	0.4655	0.6824	0.5718	0.6038	0.9322
M-2	**0.4296**	0.4587	**0.6551**	0.5610	0.6017	0.8996
M-3	0.5612	0.6071	0.7018	0.7505	0.7695	0.9705
M-4	0.4460	0.4623	0.6565	0.5716	0.5966	**0.9007**
M-5	0.4326	**0.4558**	0.6582	**0.5568**	**0.5852**	0.9059
M-6	0.6127	0.6869	0.9081	0.7547	0.8299	1.1482



Şekil 4.14. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle)

Şekil 4.14.'e göre TKAO-30 geliştirilen uygulamada MAE metrik değerine göre M-2 en başarılı model olurken M-6 en başarısız model olmuştur. TEKAO-30 geliştirilen uygulamada MAE metrik değerine göre M-5 en başarılı model olurken M-6 en başarısız model olmuştur. TKKAO-30 geliştirilen uygulamada MAE metrik değerine göre M-2 en başarılı model olurken M-6 en başarısız model olmuştur.

30 yaş grubu tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO-30), 30 yaş grubu erkek kullanıcıların ağırlıklı oy (TEKAO-30) ortalamaları ve 30 yaş grubu kadın kullanıcıların ağırlıklı oy (TKKAO-30) ortalamaları kullanılarak geliştirilen uygulamanın doğrulama sonuçları ve ÇZ Çizelge 4.19.'da verilmiştir. TKAO-30 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.4330 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.5569 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 17 dk 11 sn, M-6 modelinin eğitimi 26 dk 45 sn'de tamamlanmıştır. TEKAO-30 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.4389 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.5801 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 17 dk 10 sn, M-3 modelinin eğitimi 25 dk 46 sn'de tamamlanmıştır. TKKAO-30 ortalamasıyla geliştirilen uygulamada M-1 modeli 0.6057 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.7654 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-1 modelinin eğitimi 8 dk 59 sn, M-6 modelinin eğitimi 27 dk 25 sn'de tamamlanmıştır.

Çizelge 4.19. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama sonuçları (eğitim sırasındaki test verileri) ve çalışma zamanı (30 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

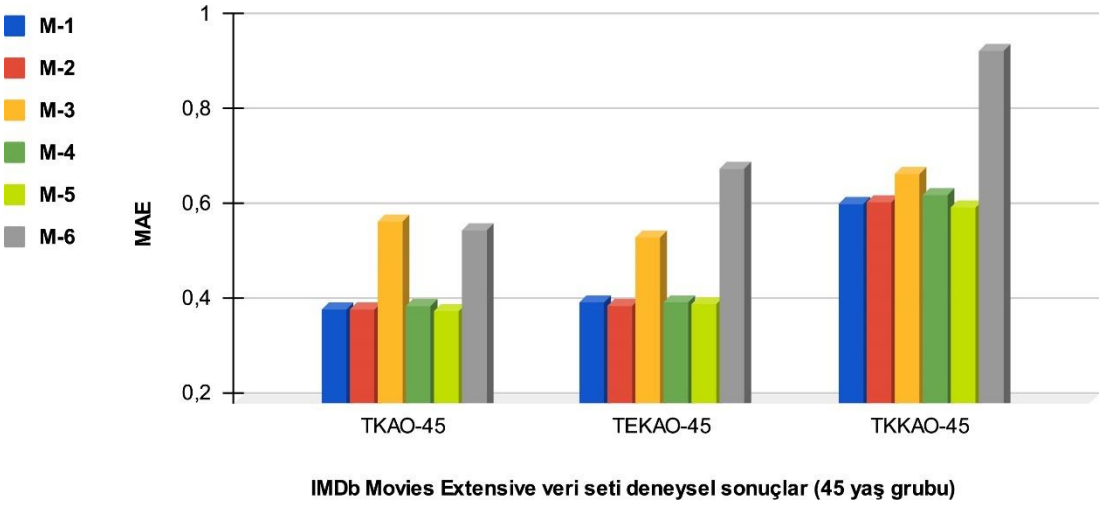
Model	Doğrulama MAE			Çalışma Zamanı		
	TKAO-30	TEKAO-30	TKKAO-30	TKAO-30	TEKAO-30	TKKAO-30
M-1	0.4479	0.4665	**0.6057**	9 dk 17 sn	9 dk 14 sn	8 dk 59 sn
M-2	**0.4330**	**0.4389**	0.6136	17 dk 11 sn	17 dk 10 sn	16 dk 43 sn
M-3	0.5437	0.5801	0.6460	34 dk 16 sn	25 dk 46 sn	25 dk 56 sn
M-4	0.4537	0.4618	0.6182	8 dk 20 sn	8 dk 15 sn	8 dk 12 sn
M-5	0.4612	0.4754	0.6130	4 dk 2 sn	4 dk 3 sn	3 dk 57 sn
M-6	0.5569	0.5702	0.7654	26 dk 45 sn	27 dk 11 sn	27 dk 25 sn

45 yaş grubu tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO-45), 45 yaş grubu erkek kullanıcıların ağırlıklı oy (TEKAO-45) ortalamaları ve 45 yaş grubu kadın kullanıcıların ağırlıklı oy (TKKAO-45) ortalamaları kullanılarak geliştirilen uygulamanın deneysel sonuçları Çizelge 4.20.'de verilmiştir. Mimarilerin eğitim

sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, TKAO-45 ortalamasıyla geliştirilen uygulamada M-5 modeli 0.3880 MAE hata metrik değerine göre en başarılı model olurken, M-3 modeli 0.5747 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TEKAO-45 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.3985 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.6876 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. TKKAO-45 ortalamasıyla geliştirilen uygulamada M-5 modeli 0.6036 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.9350 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Derin sinir ağları ile gerçekleştirilen uygulamada deneysel süreç sonucunda elde edilen MAE ve RMSE hata metrik değerlerinin sütun grafiği Şekil 4.15.'de verilmiştir.

Çizelge 4.20. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının ampirik sonuçları (45 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

Model	MAE			RMSE		
	TKAO-45	TEKAO-45	TKKAO-45	TKAO-45	TEKAO-45	TKKAO-45
M-1	0.3909	0.4044	0.6138	0.5151	0.5239	0.8372
M-2	0.3910	**0.3985**	0.6161	0.5183	**0.5178**	0.8445
M-3	0.5747	0.5406	0.6763	0.7452	0.6995	0.8980
M-4	0.3975	0.4055	0.6305	0.5190	0.5259	0.8549
M-5	**0.3880**	0.4016	**0.6036**	**0.5088**	0.5230	**0.8336**
M-6	0.5556	0.6876	0.9350	0.6928	0.8203	1.1440



Şekil 4.15. Derin sinir ağ mimarilerinin test sonuçları (MAE metriğiyle)

Şekil 4.15.'e göre TKAO-45 geliştirilen uygulamada MAE metrik değerine göre M-5 en başarılı model olurken M-3 en başarısız model olmuştur. TEKAO-45 geliştirilen uygulamada MAE metrik değerine göre M-2 en başarılı model olurken M-6 en başarısız model olmuştur. TKKAO-45 geliştirilen uygulamada MAE metrik değerine göre M-5 en başarılı model olurken M-6 en başarısız model olmuştur.

45 yaş grubu tüm kullanıcıların ağırlıklı oy ortalamaları (TKAO-45), 45 yaş grubu erkek kullanıcıların ağırlıklı oy (TEKAO-45) ortalamaları ve 45 yaş grubu kadın kullanıcıların ağırlıklı oy (TKKAO-45) ortalamaları kullanılarak geliştirilen uygulamanın doğrulama sonuçları ve ÇZ Çizelge 4.21.'de verilmiştir. TKAO-45 ortalamasıyla geliştirilen uygulamada M-2 modeli 0.3928 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.4923 MAE metrik değeriyle en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-2 modelinin eğitimi 17 dk 42 sn, M-6 modelinin eğitimi 27 dk 23 sn'de tamamlanmıştır. TEKAO-45 ortalamasıyla geliştirilen uygulamada M-4 modeli 0.3859 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.5892 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-4 modelinin eğitimi 8 dk 27 sn, M-6 modelinin eğitimi 27 dk 19 sn'de tamamlanmıştır. TKKAO-45 ortalamasıyla geliştirilen uygulamada M-1 modeli 0.5826 MAE hata metrik değerine göre en başarılı model olurken, M-6 modeli 0.7885 MAE metrik değerine göre en düşük başarıya sahip model olmuştur. Modellerin eğitim sürecindeki çalışma zamanları incelendiğinde M-1 modelinin eğitimi 9 dk 57 sn, M-6 modelinin eğitimi 28 dk 48 sn'de tamamlanmıştır.

Çizelge 4.21. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında derin sinir ağlarının doğrulama sonuçları (eğitim sırasındaki test verileri) ve çalışma zamanı (45 yaş grubu tüm, erkek ve kadın kullanıcıların ağırlık ortalaması ile)

Model	Doğrulama MAE			Çalışma Zamanı		
	TKAO-45	TEKAO-45	TKKAO-45	TKAO-45	TEKAO-45	TKKAO-45
M-1	0.3947	0.3895	**0.5826**	10 dk 7 sn	10 dk	9 dk 57 sn
M-2	**0.3928**	0.3960	0.5856	17 dk 47 sn	17 dk 6sn	17 dk 16 sn
M-3	0.4664	0.4969	0.6687	28 dk 55 sn	28 dk 46 sn	33 dk 33 sn
M-4	0.4121	**0.3859**	0.5864	8 dk 30 sn	8 dk 27 sn	8 dk 21 sn
M-5	0.4141	0.4036	0.6017	1 dk 14 sn	4 dk 1 sn	8 dk 4 sn
M-6	0.4923	0.5892	0.7885	27 dk 23 sn	27 dk 19 sn	28 dk 48 sn

4.4.2. Makine öğrenmesi

IMDb Movies Extensive veri kümesindeki tüm kullanıcıların ağırlıklı oy ortalamaları kullanılarak DT, RF, SGD, SVM, KNN geleneksel makine öğrenme algoritmaları ile geliştirilen film tavsiye sistemi derecelendirme uygulamasının ampirik sonuçları ve ÇZ Çizelge 4.22.'de verilmiştir. Geleneksel makine öğrenme algoritmaları ile geliştirilen uygulamada eğitim ve test sürecinde veri kümesi 9:1 (eğitim/test) oranında ayrılmıştır. Geleneksel makine öğrenme algoritmalarının eğitimi sonucunda test veri kümeleriyle gerçekleştirilen performans analizinde, tüm kullanıcıların ağırlıklı oy ortalamaları kullanılarak geliştirilen uygulamada SGD 0.3817 MAE hata metrik değerine göre en başarılı model olurken, DT 0.6872 MAE metrik değerine göre en düşük başarıya sahip model olmuştur (Çizelge 4.22.). Geleneksel makine öğrenme algoritmalarının eğitim sürecindeki ÇZ incelendiğinde KNN algoritması 0.03 sn, RF algoritması 2 dk 42 sn'de eğitimi tamamlamıştır (Çizelge 4.22.).

Çizelge 4.22. IMDb Movies Extensive veri kümesiyle geliştirilen derecelendirme uygulamasında geleneksel makine öğrenmesi algoritmalarının ampirik sonuçları (tüm kullanıcıların ağırlık ortalaması ile)

Model	MAE	MSE	RMSE	MdAE	Çalışma Zamanı
DT	0.6872	0.7656	0.8750	0.6000	3.69 sn
RF	0.4899	0.4149	0.6441	0.4036	2 dk 42 sn
SGD	**0.3817**	**0.2521**	**0.5021**	**0.3113**	10.56 sn
SVM	0.4373	0.3167	0.5627	0.3508	1 dk 58 sn
KNN	0.4377	0.3426	0.5853	0.3306	0.03 sn

4.5. Literatür Karşılaştırılması

Bu bölümde tavsiye sistemi derecelendirme problemlerinin çözümü için literatürde önerilen yöntemlerin deneysel sonuçları ve tez kapsamında önerilen RS-DSINet mimarileri ve önerilen hibrit modellerin deneysel sonuçları karşılaştırılarak verilmiştir. Tez kapsamında önerilen derin öğrenme modellerin ampirik sonuçları ve literatür kapsamında deneysel sonuçlar sonucunda elde edilen ampirik sonuçlar Çizelge 4.23.'de verilmiştir.

Çizelge 4.23. Önerilen derin öğrenme modelinin literatürdeki yöntemlerle karşılaştırılması

Literatür	Model	Makine Öğrenme Teknikleri	Veri Seti	MAE
Xia ve ark., 2020	AGAMF-N	✓	MovieLens 10M	0.5784
Hammou ve ark., 2019	DPMF	✓	MovieLens 10M	0.6053
Yue ve ark., 2018	I-AutoRec	✓	MovieLens 10M	0.6044
Li ve ark., 2021	DSAP-AL	✓	MovieLens 25M	0.8215
Li ve ark., 2022	EKTDCR	✓	MovieLens 25M	0.7347
Ullah ve ark., 2020	Deep Edu	✓	Goodbooks-10K	0.6056
Wu ve ark., 2022	GLFA	✓	Goodbooks-10K	0.6800
Sanwal ve ark., 2021	ANN	✓	IMDb movies extensive	0.6700
Önerilen Yöntem	M-Hibrit-2 (BiLSTM + CNN + FCCMLP)	✓	MovieLens 10M	**0.3629**
	M-6 (RS-DSINet + SimpleRNN)	✓	MovieLens 25M	**0.5112**
	M-Hibrit-3 (SimpleRNN + FCCMLP)	✓	Goodbooks-10K	**0.1916**
	M-5 (RS-DSINet + FCCMLP)	✓	IMDb movies extensive	**0.3929**

Çizelge 4.23.'de MovieLens 10M, MovieLens 25M, Goodbooks-10K, IMDb movies extensive veri kümesi ile literatür kapsamında ve tez kapsamında gerçekleştirilen deneysel sonuçlar verilmiştir. Çizelge 4.23.'e göre tez kapsamında önerilen RS-DSINet ve hibrit yöntemlerin literatür kapsamında yapılan çalışmalara göre başarılı sonuçlar ürettiği gözlemlenmiştir. MovieLens 10M veri kümesi kullanılarak gerçekleştirilen çalışmalarda Xia ve ark. (2020) önerdiği AGAMF-N yöntemi 0.5784 MAE hata değeri üretirken, tez kapsamında önerilen M-Hibrit-2 (BiLSTM + CNN + FCCMLP) hibrit modeli 0.3629 MAE hata değeri ile başarılı bir sonuç üretmiştir. MovieLens 25M veri kümesi kullanılarak gerçekleştirilen çalışmalarda Li ve ark. (2022) önerdiği EKTDCR yöntemi 0.7347 MAE hata değeri üretirken, tez kapsamında önerilen M-6 (RS-DSINet + SimpleRNN) modeli 0.5112 MAE hata değeri ile başarılı bir sonuç üretmiştir. Goodbooks-10K veri kümesi

kullanılarak gerçekleştirilen çalışmalarda Ullah ve ark. (2020) önerdiği Deep Edu yöntemi 0.6056 MAE hata değeri üretirken, tez kapsamında önerilen M-Hibrit-3 (SimpleRNN + FCCMLP) hibrit modeli 0.1916 MAE hata değeri ile başarılı bir sonuç üretmiştir. IMDb movies extensive veri kümesi kullanılarak gerçekleştirilen çalışmalarda Sanwal ve ark. (2021) önerdiği ANN yöntemi 0.6700 MAE hata değeri üretirken, tez kapsamında önerilen M-5 (RS-DSINet + FCCMLP) modeli 0.3929 MAE hata değeri ile başarılı bir sonuç üretmiştir.

5. SONUÇLAR ve ÖNERİLER

5.1. Sonuçlar

Bu çalışmada film ve kitap derecelendirme probleminin çözümünde yapay zeka teknolojilerinden derin öğrenme modelleri, geleneksel makine öğrenmesi algoritmaları ve geleneksel tavsiye sistemleri yöntemlerden içerik tabanlı filtreleme ve işbirlikçi filtreleme teknikleri önerilmiştir.

Çalışmada önerilen mimariler MovieLens 10M, MovieLens 20M, Goodbooks-10K, FilmTV movies, IMDb movies extensive veri setleri ile eğitip test edilmiştir. Tez kapsamında ön işlem adımlarından girdi vektörlerin belirlenmesi sürecinde birinci aşamada veri kümelerinde yer alan sayısal olmayan özniteliklerin kullanımı için one hot encoding yöntemi uygulanırken, sayısal öznitelikler için 0-1 normalizasyon işlemi uygulanmıştır. İkinci aşamada elde edilen büyük boyutlu vektörlerin işlem hacmini azaltmak amacıyla PCA yöntemi uygulanmıştır. Denetimli öğrenme süreci için gerekli etiketlerin belirlenmesi sürecinde MovieLens 10M ve MovieLens 20M veri kümeleri için ortalama puanlar hesaplanmıştır. Goodbooks-10K, FilmTV movies veri kümelerinde veri kümeleri öz niteliklerinden ağırlıklı puanlar kullanılmıştır. IMDb veri setinde tüm kullanıcıların ağırlıklı oy ortalamaları ile birlikte tüm erkek/kadın kullanıcıların ağırlıklı oy ortalamaları, 18, 30, 45 yaş grubundaki erkek/kadın kullanıcıların ağırlıklı oyların ortalaması kullanılmıştır. Çalışmada, filmlerin/kitapların içerik bilgilerinden yararlanılarak içerik bilgileri ile derecelendirmeler arasındaki örtük ilişkilerin keşfine odaklanılmıştır. Deneysel sonuçların yorumlanmasında loss grafikleri, MAE hata ölçüm metrik grafikleri, MAE, MSE, RMSE, MdAE hata ölçüm metrikleri kullanılmıştır.

Yapay zeka teknolojilerinden derin sinir ağları ile önerilen uygulamalarda LSTM, BiLSTM, CNN, GRU, FCCMLP, SimpleRNN katmanları ile birlikte, artık ağlar, ek özellik vektörleri, batch normalizasyon, ReLU aktivasyon fonksiyonu ile geliştirilen RS-DSINet modeli ve LSTM, BiLSTM, CNN, GRU, FCCMLP,

SimpleRNN ağları ile geliştirilen hibrit yöntemler uygulanmıştır. Derin sinir ağları geliştirilen derecelendirme uygulamasının birinci aşamasında MovieLens (MovieLens 10M, MovieLens 25M), Goodbooks-10K, FilmTV movies, IMDb movies extensive veri kümeleri ile eğitim-doğrulama-test yaklaşımı, ikinci aşamada MovieLens 10M, Goodbooks-10K veri kümeleri ile 5 katlı çapraz doğrulama tekniği uygulanmıştır. Araştırma bulgularımıza göre RS-DSINet ve hibrit mimarileri ile önerilen derin öğrenme modellerinin derecelendirme çalışmalarında başarılı sonuçlar ürettiğini göstermiştir.

MovieLens 10M veri kümesi ile önerilen film derecelendirme uygulamasında; derin öğrenme modellerinden M-Hibrit-2 (BiLSTM + CNN + FCCMLP) hibrit modeli 0.3629 MAE hata değeriyle, geleneksel makine öğrenme algoritmalarında SVM 0.3662 MAE hata değeriyle ve geleneksel tavsiye sistemi yöntemlerinden CBF 0.7563 MAE hata değeriyle en başarılı yöntemler olmuştur. MovieLens 25M veri kümesi ile önerilen film derecelendirme uygulamasında; derin öğrenme modellerinden M-6 0.5112 MAE hata değeriyle en başarılı yöntem olmuştur. Goodbooks-10K veri kümesi ile önerilen kitap derecelendirme uygulamasında; derin öğrenme modellerinden M-Hibrit-3 (SimpleRNN + FCCMLP) hibrit modeli 0.1916 MAE hata değeriyle, geleneksel makine öğrenme algoritmalarından RF 0.1781 MAE hata değeriyle en başarılı yöntemler olmuştur. FilmTV Movies veri kümesi ile önerilen film derecelendirme uygulamasında; derin öğrenme algoritmalarından M-2 0.7135 MAE hata değeriyle, geleneksel makine öğrenme algoritmalarında SVM 0.7232 MAE hata değeriyle en başarılı yöntemler olmuştur. IMDb Movies Extensive veri kümesi ile önerilen film derecelendirme uygulamasında tüm kullanıcıların ağırlıklı oy ortalamaları kullanılarak geliştirilen uygulamada, derin öğrenme modellerinden M-5 0.3929 MAE hata değeriyle, geleneksel makine öğrenme algoritmalarından SGD 0.3817 MAE hata değeriyle en başarılı yöntemler olmuştur.

5.2. Öneriler

İçerik ve işbirlikçi filtreleme tekniklerinin başarısını iyileştirebilmek için yapay zeka teknolojileri ile hibrit yöntemler geliştirilebilir. Çalışma kapsamında önerilen

derecelendirme uygulaması filmlerin/kitapların içerik bilgilerinden yararlanılarak geliştirilmiştir. Burada, yapay zeka teknolojilerinden derin sinir ağlarının başarısını iyileştirmek için veri setlerinde yer alan kullanıcı bilgileri derecelendirme uygulamalarına dahil edilebilir, sinir ağlarının derinliği artırılabilir. Makine öğrenme algoritmaları ile önerilen derecelendirme uygulamasında algoritmalarının başarısını iyileştirmek için optimal hiper-parametrelerin belirlenmesi sürecinde parçacık sürü optimizasyonu, balina optimizasyon algoritması, gri kurt optimizasyon, genetik algoritma uygulanabilir. Son olarak makine öğrenme teknikleri ile önerilen yöntemlerde kümeleme yöntemleri dahil edilerek hibrit yöntemler geliştirilebilir. Burada makine öğrenme teknolojileri ile özellikler çıkartılabilir, elde edilen öz niteliklerden hangi özelliklerin baskın oldu kümeleme yöntemleri ile tespit edilip regresyon problemlerine uygulanabilir.

KAYNAKLAR

AFOUDI, Y., LAZAAR, M., and AL ACHHAB, M., 2021. Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. Simulation Modelling Practice and Theory, 113: 102375. https://doi.org/10.1016/j.simpat.2021.102375.

AHA, D. W., KIBLER, D., and ALBERT, M. K., 1991. Instance-based learning algorithms. Machine learning, 6(1): 37-66. https://doi.org/10.1007/BF00153759.

AIREN, S., and AGRAWAL, J., 2022. Movie recommender system using k-nearest neighbors variants. National Academy Science Letters, 45(1): 75-82. https://doi.org/10.1007/s40009-021-01051-0.

ALJUNID, M. F., and DH, M., 2020. An efficient deep learning approach for collaborative filtering recommender system. Procedia Computer Science, 171: 829-836. https://doi.org/10.1016/j.procs.2020.04.090.

ANWAAR, F., ILTAF, N., AFZAL, H., and NAWAZ, R., 2018. HRS-CE: A hybrid framework to integrate content embeddings in recommender systems for cold start items. Journal of computational science, 29: 9-18. https://doi.org/10.1016/j.jocs.2018.09.008.

ANWAR, T., and UMA, V., 2021. Comparative study of recommender system approaches and movie recommendation using collaborative filtering. International Journal of System Assurance Engineering and Management, 12(3): 426-436. https://doi.org/10.1007/s13198-021-01087-x.

ANWAR, T., UMA, V., HUSSAIN, M., and PANTULA, M., 2022. Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: A comparative analysis. Multimedia Tools and Applications, 1-19. https://doi.org/10.1007/s11042-021-11883-z.

ASHARF, J., MOUSTAFA, N., KHURSHID, H., DEBIE, E., HAIDER, W., and WAHAB, A., 2020. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. Electronics, 9(7): 1177. https://doi.org/10.3390/electronics9071177.

AZEROUAL, O., and KOLTAY, T., 2022. RecSys Pertaining to Research Information with Collaborative Filtering Methods: Characteristics and Challenges. Publications, 10(2): 17. https://doi.org/10.3390/publications10020017.

BATMAZ, Z., YUREKLI, A., BILGE, A., and KALELI, C., 2019. A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review, 52(1): 1-37. https://doi.org/10.1007/s10462-018-9654-y.

BEHERA, G., and NAIN, N., 2022. Handling data sparsity via item metadata embedding into deep collaborative recommender system. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2021.12.021.

BRE, F., GIMENEZ, J. M., and FACHINOTTI, V. D., 2018. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. Energy and Buildings, 158: 1429-1441. https://doi.org/10.1016/j.enbuild.2017.11.045.

BREIMAN, L., 2001. Random forests. Machine learning, 45(1): 5-32. https://doi.org/10.1023/A:1010933404324.

CERVANTES, J., GARCIA-LAMONT, F., RODRÍGUEZ-MAZAHUA, L., and

LOPEZ, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408: 189-215. https://doi.org/10.1016/j.neucom.2019.10.118.

CHEN, Y., ZHENG, W., LI, W., and HUANG, Y., 2021a. Large group activity security risk assessment and risk early warning based on random forest algorithm. Pattern Recognition Letters, 144: 1-5. https://doi.org/10.1016/j.patrec.2021.01.008.

CHEN, Z., ZHAO, W., and WANG, S., 2021b. Kernel meets recommender systems: A multi-kernel interpolation for matrix completion. Expert Systems with Applications, 168: 114436. https://doi.org/10.1016/j.eswa.2020.114436.

CHICAIZA, J., and VALDIVIEZO-DIAZ, P., 2021. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. Information, 12(6): 232. https://doi.org/10.3390/info12060232.

CHO, K., VAN MERRIËNBOER, B., GULCEHRE, C., BAHDANAU, D., BOUGARES, F., SCHWENK, H., and BENGIO, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. https://doi.org/10.48550/arXiv.1406.1078.

CHOUDHURY, S. S., MOHANTY, S. N., and JAGADEV, A. K., 2021. Multimodal trust based recommender system with machine learning approaches for movie recommendation. International Journal of Information Technology, 13(2): 475-482. https://doi.org/10.1007/s41870-020-00553-2.

CUI, Z., KE, R., PU, Z., and WANG, Y., 2020. Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values. Transportation Research Part C: Emerging Technologies, 118: 102674. https://doi.org/10.1016/j.trc.2020.102674.

DE DIOS ROJAS OLVERA, J., GÓMEZ-VARGAS, I., and VÁZQUEZ, J. A., 2022. Observational Cosmology with Artificial Neural Networks. Universe, 8(2): 120. https://doi.org/10.3390/universe8020120.

DOMO., 2021. Data Never Sleeps 9.0. https://www.domo.com/learn/infographic/data-never-sleeps-9.

DUONG, T. N., VUONG, T. A., NGUYEN, D. M., and DANG, Q. H., 2020. Utilizing an Autoencoder-Generated Item Representation in Hybrid Recommendation System. IEEE Access, 8: 75094-75104. https://doi.org/10.1109/ACCESS.2020.2989408.

ES-SABERY, F., HAIR, A., QADIR, J., SAINZ-DE-ABAJO, B., GARCÍA-ZAPIRAIN, B., and DE LA TORRE-DÍEZ, I., 2021. Sentence-level classification using parallel fuzzy deep learning classifier. IEEE Access, 9: 17943-17985. https://doi.org/10.1109/ACCESS.2021.3053917.

FAN, Y., SHEN, Y., and MAI, J., 2008. Study of the model of e-commerce personalized recommendation system based on data mining. In 2008 International Symposium on Electronic Commerce and Security, 03-05 August, Guangzhou, China, s.647-651. https://doi.org/10.1109/ISECS.2008.106.

FANG, X., 2021. Making recommendations using transfer learning. Neural Computing and Applications, 33(15): 9663-9676. https://doi.org/10.1007/s00521-021-05730-3.

FATIMA, A., and PETHE, A., 2022. Implementation of RRAM based Swish Activation Function and its Derivative on 28nm FD-SOI. In 2022 International Electrical Engineering Congress, 09-11 March, Khon Kaen, Thailand, s.1-4.

https://doi.org/10.1109/iEECON53204.2022.9741701.
FKIH, F., 2021. Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2021.09.014.
GRAVES, A., and SCHMIDHUBER, J., 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6): 602-610. https://doi.org/10.1016/j.neunet.2005.06.042.
GROENSFELDER, T., GIEBELER, F., GEUPEL, M., SCHNEIDER, D., and JAEGER, R., 2020. Application of machine learning procedures for mechanical system modelling: capabilities and caveats to prediction-accuracy. Advanced Modeling and Simulation in Engineering Sciences, 7(1): 1-29. https://doi.org/10.1186/s40323-020-00163-4.
GRONAUER, S., and DIEPOLD, K., 2022. Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review, 55(2): 895-943. https://doi.org/10.1007/s10462-021-09996-w.
GUO, P., MENG, W., XU, M., LI, V. C., and BAO, Y., 2021. Predicting mechanical properties of high-performance fiber-reinforced cementitious composites by integrating micromechanics and machine learning. Materials, 14(12): 3143. https://doi.org/10.3390/ma14123143.
GUO, Q., ZHUANG, F., QIN, C., ZHU, H., XIE, X., XIONG, H., and HE, Q., 2020. A survey on knowledge graph-based recommender systems. IEEE Transactions on Knowledge and Data Engineering, 34(8): 3549-3568. https://doi.org/10.1109/TKDE.2020.3028705.
HAMMOU, B. A., LAHCEN, A. A., and MOULINE, S., 2019. An effective distributed predictive model with Matrix factorization and random forest for Big Data recommendation systems. Expert Systems with Applications 137: 253-265. https://doi.org/10.1016/j.eswa.2019.06.046.
HARSHVARDHAN, G. M., GOURISARIA, M. K., RAUTARAY, S. S., and PANDEY, M., 2021. UBMTR: Unsupervised Boltzmann machine-based time-aware recommendation system. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2021.01.017.
HE, K., ZHANG, X., REN, S., and SUN, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 27-30 June, Las Vegas, NV, USA, s.770-778. https://doi.org/10.1109/CVPR.2016.90.
HOCHREITER, S., and SCHMIDHUBER, J., 1997. Long short-term memory. Neural computation, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735.
HUANG, G., LIU, Z., VAN DER MAATEN, L., and WEINBERGER, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 21-26 July, Honolulu, HI, USA, s.4700-4708. https://doi.org/10.1109/CVPR.2017.243.
JAHAN, I., AHMED, M. F., ALI, M. O., and JANG, Y. M., 2022. Self-gated rectified linear unit for performance improvement of deep neural networks. ICT Express. https://doi.org/10.1016/j.icte.2021.12.012.
JESSE, M., and JANNACH, D., 2021. Digital nudging with recommender systems: Survey and future directions. Computers in Human Behavior Reports, 3: 100052. https://doi.org/10.1016/j.chbr.2020.100052.

JOY, J., and PILLAI, R. V. G., 2021. Review and classification of content recommenders in E-learning environment. Journal of King Saud University-Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2021.06.009.

KANDEL, I., and CASTELLI, M., 2020. Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review. Applied Sciences, 10(6): 2021. https://doi.org/10.3390/app10062021.

KE, G., DU, H. L., and CHEN, Y. C., 2021. Cross-platform dynamic goods recommendation system based on reinforcement learning and social networks. Applied Soft Computing, 104: 107213. https://doi.org/10.1016/j.asoc.2021.107213.

KHALIFA, Y., MANDIC, D., and SEJDIĆ, E., 2021. A review of Hidden Markov models and Recurrent Neural Networks for event detection and localization in biomedical signals. Information Fusion, 69: 52-72. https://doi.org/10.1016/j.inffus.2020.11.008.

KHANAL, S. S., PRASAD, P. W. C., ALSADOON, A., and MAAG, A., 2020. A systematic review: machine learning based recommendation systems for e-learning. Education and Information Technologies, 25(4): 2635-2664. https://doi.org/10.1007/s10639-019-10063-9.

KINGMA, D. P., and BA, J., 2014. Adam: A method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980.

KOWALD, D., and LACIC, E., 2022. Popularity Bias in Collaborative Filtering-Based Multimedia Recommender Systems. https://doi.org/10.48550/arXiv.2203.00376.

LE, X. H., HO, H. V., LEE, G., and JUNG, S., 2019. Application of long short-term memory (LSTM) neural network for flood forecasting. Water, 11(7): 1387. https://doi.org/10.3390/w11071387.

LECUN, Y., BENGIO, Y., and HINTON, G., 2015. Deep learning. nature, 521(7553): 436-444. https://doi.org/10.1038/nature14539.

LECUN, Y., JACKEL, L. D., BOSER, B., DENKER, J. S., GRAF, H. P., GUYON, I., HENDERSON, D., HOWARD, R. E., and HUBBARD, W., 1989. Handwritten digit recognition: Applications of neural network chips and automatic learning. IEEE Communications Magazine, 27(11): 41-46.

LI, Y., REN, J., LIU, J., and CHANG, Y., 2021. Deep sparse autoencoder prediction model based on adversarial learning for cross-domain recommendations. Knowledge-Based Systems, 220: 106948. https://doi.org/10.1016/j.knosys.2021.106948.

LI, Y., WU, Q., HOU, L., and LI, J., 2022. Entity knowledge transfer-oriented dual-target cross-domain recommendations. Expert Systems with Applications, 195: 116591. https://doi.org/10.1016/j.eswa.2022.116591.

LIU, H., and LANG, B., 2019. Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20): 4396. https://doi.org/10.3390/app9204396.

LIPTON, Z. C., BERKOWITZ, J., and ELKAN, C., 2015. A critical review of recurrent neural networks for sequence learning. https://doi.org/10.48550/arXiv.1506.00019.

MASTROMICHALAKIS, S., 2020. ALReLU: A different approach on Leaky ReLU activation function to improve Neural Networks Performance. https://doi.org/10.48550/arXiv.2012.07564

MOAYEDI, H., and MOSAVI, A., 2021. Suggesting a stochastic fractal search paradigm in combination with artificial neural network for early prediction of cooling load in residential buildings. Energies, 14(6): 1649. https://doi.org/10.3390/en14061649.

MUSTO, C., DE GEMMIS, M., LOPS, P., and SEMERARO, G., 2021. Generating post hoc review-based natural language justifications for recommender systems. User Modeling and User-Adapted Interaction, 31(3): 629-673. https://doi.org/10.1007/s11257-020-09270-8.

MURTAGH, F., 1991. Multilayer perceptrons for classification and regression. Neurocomputing, 2(5-6): 183-197. https://doi.org/10.1016/0925-2312(91)90023-5.

NÁPOLES, G., GRAU, I., and SALGUEIRO, Y., 2020. Recommender system using long-term cognitive networks. Knowledge-Based Systems, 206: 106372. https://doi.org/10.1016/j.knosys.2020.106372.

NEU, G., 2021. Information-theoretic generalization bounds for stochastic gradient descent. In Conference on Learning Theory, 15-19 August, Boulder, Colorado, USA, 134: 3526-3545.

NGUYEN, L. V., HONG, M. S., JUNG, J. J., and SOHN, B. S., 2020. Cognitive similarity-based collaborative filtering recommendation system. Applied Sciences, 10(12): 4183. https://doi.org/10.3390/app10124183.

NING, X., YAC, L., WANG, X., BENATALLAH, B., DONG, M., and ZHANG, S., 2020. Rating prediction via generative convolutional neural networks based regression. Pattern Recognition Letters, 132: 12-20. https://doi.org/10.1016/j.patrec.2018.07.028.

OMUYA, E. O., OKEYO, G. O., and KIMWELE, M. W., 2021. Feature selection for classification using principal component analysis and information gain. Expert Systems with Applications, 174: 114765. https://doi.org/10.1016/j.eswa.2021.114765.

PAPADAKIS, H., PAPAGRIGORIOU, A., PANAGIOTAKIS, C., KOSMAS, E., and FRAGOPOULOU, P., 2022. Collaborative filtering recommender systems taxonomy. Knowledge and Information Systems, 64: 35-74. https://doi.org/10.1007/s10115-021-01628-7.

PEŠKA, L., TASHU, T. M., and HORVÁTH, T., 2019. Swarm intelligence techniques in recommender systems-A review of recent research. Swarm and Evolutionary Computation, 48: 201-219. https://doi.org/10.1016/j.swevo.2019.04.003.

PHUNG, V. H., and RHEE, E. J., 2019. A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. Applied Sciences, 9(21): 4500. https://doi.org/10.3390/app9214500.

PORTUGAL, I., ALENCAR, P., and COWAN, D., 2018. The use of machine learning algorithms in recommender systems: A systematic review. Expert Systems with Applications, 97: 205-227. https://doi.org/10.1016/j.eswa.2017.12.020.

QIU, W., WEN, G., and LIU, H., 2021. A back-propagation neural network model based on genetic algorithm for prediction of build-up rate in drilling process. Arabian Journal for Science and Engineering, 1-11. https://doi.org/10.1007/s13369-021-05634-3.

QUE, Y., ZHONG, S., and CHEN, K., 2022. A Score-Guided Regularization Strategy-Based Unsupervised Structural Damage Detection Method. Applied Sciences, 12(10): 4887. https://doi.org/10.3390/app12104887.

QUINLAN, J. R., 1986. Induction of decision trees. Machine learning, 1(1): 81-106. https://doi.org/10.1007/BF00116251.

RAHMAN, M. M., SHAKERI, M., TIONG, S. K., KHATUN, F., AMIN, N., PASUPULETI, J., and HASAN, M. K., 2021. Prospective methodologies in hybrid renewable energy systems for energy prediction using artificial neural networks. Sustainability, 13(4): 2393. https://doi.org/10.3390/su13042393.

RAJENDRAN, D. P. D., and SUNDARRAJ, R. P., 2021. Using topic models with browsing history in hybrid collaborative filtering recommender system: Experiments with user ratings. International Journal of Information Management Data Insights, 1(2): 100027. https://doi.org/10.1016/j.jjimei.2021.100027.

RAMA, K., KUMAR, P., and BHASKER, B., 2021. Deep autoencoders for feature learning with embeddings for recommendations: a novel recommender system solution. Neural Computing and Applications, 33(21): 14167-14177. https://doi.org/10.1007/s00521-021-06065-9.

RAMACHANDRAN, P., ZOPH, B., and LE, Q. V., 2017. Searching for activation functions. https://doi.org/10.48550/arXiv.1710.05941.

RAY, B., GARAIN, A., and SARKAR, R., 2021. An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews. Applied Soft Computing, 98: 106935. https://doi.org/10.1016/j.asoc.2020.106935.

ROY, D., and DING, C., 2021. Multi-source based movie recommendation with ratings and the side information. Social Network Analysis and Mining, 11(1): 1-20. https://doi.org/10.1007/s13278-021-00785-5.

ROY, D., and DUTTA, M., 2022. A systematic review and research perspective on recommender systems. Journal of Big Data, 9(1): 1-36. https://doi.org/10.1186/s40537-022-00592-5.

SANWAL, M., CALISKAN, C., 2021. A hybrid movie recommender system and rating prediction model. International Journal of Information Technology and Applied Sciences (IJITAS), 3(3): 161-168. https://doi.org/10.52502/ijitas.v3i3.128

SARASWAT, M., CHAKRAVERTY, S., and KALA, A., 2020. Analyzing emotion based movie recommender system using fuzzy emotion features. International Journal of Information Technology, 12(2): 467-472. https://doi.org/10.1007/s41870-020-00431-x.

SARKER, I. H., KAYES, A. S. M., and WATTERS, P., 2019. Effectiveness analysis of machine learning classification models for predicting personalized context-aware smartphone usage. Journal of Big Data, 6(1): 1-28. https://doi.org/10.1186/s40537-019-0219-y.

SEJWAL, V. K., and ABULAISH, M., 2020. CRecSys: A Context-Based Recommender System Using Collaborative Filtering and LOD. IEEE Access, 8: 158432-158448. https://doi.org/10.1109/ACCESS.2020.3020005.

SEJWAL, V. K., and ABULAISH, M., 2021. CAMO: A context-aware movie ontology generated from LOD and movie databases. Multimedia Tools and Applications, 80(5): 7247-7269. https://doi.org/10.1007/s11042-020-10076-4.

SEWAK, M., SAHAY, S. K., and RATHORE, H., 2020. An overview of deep learning architecture of deep neural networks and autoencoders. Journal of Computational and Theoretical Nanoscience, 17(1): 182-188. https://doi.org/10.1166/jctn.2020.8648.

SHAMBOUR, Q., 2021. A deep learning based algorithm for multi-criteria recommender systems. Knowledge-Based Systems, 211: 106545. https://doi.org/10.1016/j.knosys.2020.106545.

SHARMA, B., HASHMI, A., GUPTA, C., KHALAF, O. I., ABDULSAHIB, G. M., and ITANI, M. M., 2022. Hybrid Sparrow Clustered (HSC) Algorithm for Top-N Recommendation System. Symmetry, 14(4): 793. https://doi.org/10.3390/sym14040793.

SINHA, B. B., and DHANALAKSHMI, R., 2022. DNN-MF: Deep neural network matrix factorization approach for filtering information in multi-criteria recommender systems. Neural Computing and Applications, 34: 10807–10821. https://doi.org/10.1007/s00521-022-07012-y.

SUJITHRA ALIAS KANMANI, R., SURENDIRAN, B., and IBRAHIM, S. P., 2021. Recency augmented hybrid collaborative movie recommendation system. International Journal of Information Technology, 13(5): 1829-1836. https://doi.org/10.1007/s41870-021-00769-w.

SOLANES, A., and RADUA, J., 2022. Advances in Using MRI to Estimate the Risk of Future Outcomes in Mental Health-Are We Getting There?. Frontiers in Psychiatry, 13. https://doi.org/10.3389/fpsyt.2022.826111.

SUN, X., GONG, L., HAN, Z., ZHAO, P., YU, J., and WANG, S., 2022. Neural Metric Factorization for Recommendation. Mathematics, 10(3): 503. https://doi.org/10.3390/math10030503.

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., and RABINOVICH, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7-12 June, Boston, MA, USA, s.1-9. https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298594.

TAHMASEBI, H., RAVANMEHR, R., and MOHAMADREZAEI, R., 2021. Social movie recommender system based on deep autoencoder network using Twitter data. Neural Computing and Applications, 33(5): 1607-1623. https://doi.org/10.1007/s00521-020-05085-1.

THAKKER, U., PATEL, R., and SHAH, M., 2021. A comprehensive analysis on movie recommendation system employing collaborative filtering. Multimedia Tools and Applications, 80: 28647–28672. https://doi.org/10.1007/s11042-021-10965-2.

TOLEDO-PÉREZ, D. C., RODRÍGUEZ-RESÉNDİZ, J., GÓMEZ-LOENZO, R. A., and JAUREGUİ-CORREA, J. C., 2019. Support vector machine-based EMG signal classification techniques: A review. Applied Sciences, 9(20): 4402. https://doi.org/10.3390/app9204402.

ULLAH, F., ZHANG, B., KHAN, R. U., CHUNG, T. S., ATTIQUE, M., KHAN, K., KHEDIRI S. E., and JAN, S., 2020. Deep edu: a deep neural collaborative filtering for educational services recommendation. IEEE Access, 8: 110915-110928. https://doi.org/10.1109/ACCESS.2020.3002544.

VAPNIK, V., 1999. The nature of statistical learning theory. Springer science and business media. https://doi.org/10.1007/978-1-4757-2440-0.

WANG, H., ZUO, Y., LI, H., and WU, J., 2021a. Cross-domain recommendation with user personality. Knowledge-Based Systems, 213: 106664. https://doi.org/10.1016/j.knosys.2020.106664.

WANG, M., ZHANG, H., DAI, H., and SHEN, L., 2022. A deep learning-aided

seismic fragility analysis method for bridges. In Structures, 40: 1056-1064. https://doi.org/10.1016/j.istruc.2022.04.058.

WANG, X., WANG, Y., GUO, L., XU, L., GAO, B., LIU, F., and LI, W., 2021b. Exploring Clustering-Based Reinforcement Learning for Personalized Book Recommendation in Digital Library. Information, 12(5): 198. https://doi.org/10.3390/info12050198.

WU, D., HE, Y., LUO, and X., 2022. Graph-incorporated Latent Factor Analysis for High-dimensional and Sparse Matrices. https://doi.org/10.48550/arXiv.2204.07818.

XIA, H., LI, J. J., LIU, Y., 2020. Collaborative filtering recommendation algorithm based on attention GRU and adversarial learning. IEEE Access 8: 208149-208157. https://doi.org/10.1109/ACCESS.2020.3038770.

YALCIN, E., and BILGE, A., 2022. Treating adverse effects of blockbuster bias on beyond-accuracy quality of personalized recommendations. Engineering Science and Technology, an International Journal, 33: 101083. https://doi.org/10.1016/j.jestch.2021.101083.

YASSINE, A. F. O. U. D. I., MOHAMED, L. A. Z. A. A. R., and AL ACHHAB, M., 2021. Intelligent recommender system based on unsupervised machine learning and demographic attributes. Simulation Modelling Practice and Theory, 107: 102198. https://doi.org/10.1016/j.simpat.2020.102198.

YUE, L., Sun, X. X., GAO, W. Z., FENG, G. Z., and ZHANG, B. Z., 2018. Multiple auxiliary information based deep model for collaborative filtering. Journal of Computer Science and Technology 33 (4): 668-681. https://doi.org/10.1007/s11390-018-1848-x.

ZHANG, X., CHEN, L., WANG, Y., and WANG, G., 2021. Improving incremental nonnegative matrix factorization method for recommendations based on three-way decision making. Cognitive Computation, 1-19. https://doi.org/10.1007/s12559-021-09897-8.

ZHANG, Y., ZUO, W., SHI, Z., and ADHIKARI, B. K., 2022. Integrating reviews and ratings into graph neural networks for rating prediction. Journal of Ambient Intelligence and Humanized Computing, 1-21. https://doi.org/10.1007/s12652-021-03626-7.

ZHAO, H., CHEN, Z., JIANG, H., JING, W., SUN, L., and FENG, M., 2019. Evaluation of three deep learning models for early crop classification using sentinel-1A imagery time series—A case study in Zhanjiang, China. Remote Sensing, 11(22): 2673. https://doi.org/10.3390/rs11222673.

ZHU, X., REHMAN, K. U., BO, W., SHAHZAD, M., and HASSAN, A., 2021a. Data-driven soft sensor model based on deep learning for quality prediction of industrial processes. SN Computer Science, 2(1): 1-10. https://doi.org/10.1007/s42979-020-00440-4.

ZHU, Y., WU, X., QIANG, J., YUAN, Y., and LI, Y., 2021b. Representation learning with collaborative autoencoder for personalized recommendation. Expert Systems with Applications, 186: 115825. https://doi.org/10.1016/j.eswa.2021.115825.