T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

DOĞAL DİL İŞLEME VE DERİN ÖĞRENME TEKNİKLERİ İLE DUYGU ANALİZİ: TÜRKÇE METİNLER ÜZERİNE BİR ÇALIŞMA

Yüksek Lisans Tezi

Tezi Hazırlayan
Zeynep Sena PEKEL

İstanbul, 2024

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

DOĞAL DİL İŞLEME VE DERİN ÖĞRENME TEKNİKLERİ İLE DUYGU ANALİZİ: TÜRKÇE METİNLER ÜZERİNE BİR ÇALIŞMA

Yüksek Lisans Tezi

Tezi Hazırlayan
Zeynep Sena PEKEL

Öğrenci No
2120003038

ORCID ID
0000-0002-6362-6428

Danışman
Doç. Dr. Atınç YILMAZ

İstanbul, 2024

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum **"Doğal Dil İşleme ve Derin Öğrenme Teknikleri ile Duygu Analizi: Türkçe Metinler Üzerine Bir Çalışma"** başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun bir şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 07.06.2024

Zeynep Sena PEKEL

T.C.
İSTANBUL BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ
TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI

07/06/2024

Enstitümüz ***Bilgisayar Mühendisliği*** Anabilim Dalı ***Bilgisayar Mühendisliği*** Programı yüksek lisans öğrencilerinden ***2120003038*** numaralı ***Zeynep Sena PEKEL***'in *"İstanbul Beykent Üniversitesi Lisansüstü Eğitim – Öğretim Yönetmeliği"*nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği ***"Doğal Dil İşleme ve Derin Öğrenme Teknikleri İle Duygu Analizi: Türkçe Metinler Üzerine Bir Çalışma"*** konulu tezini, Yönetim Kurulumuzun 28/05/2024 tarih ve 2024/24 sayılı toplantısında seçilen ve On-Line toplanan biz jüri üyeleri huzurunda, İstanbul Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 29. maddesinin 3. fıkrası gereğince 45 dakika süre ile Zoom programı aracılığıyla on-line olarak aday tarafından savunulmuş ve sonuçta adayın tezi hakkında ***"OYBİRLİĞİ"*** ile ***"KABUL"*** kararı verilmiştir.

İşbu tutanak, 2 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.

DANIŞMAN
Doç. Dr. At*** YI***
(İstanbul Beykent Üniversitesi)

ÜYE
Dr. Öğr. Üyesi Ta*** Fİ***
(İstanbul Beykent Üniversitesi)

ÜYE
Dr. Öğr. Üyesi So*** SE***
(Kütahya Dumlupınar Üniversitesi)

Adı Soyadı : Zeynep Sena PEKEL

Danışmanı : Doç. Dr. Atınç YILMAZ

Derecesi ve Tarihi : Yüksek Lisans (Tezli), 2024

Alanı : Bilgisayar Mühendisliği

Anahtar Kelimeler : Doğal Dil İşleme, Derin Öğrenme, Duygu Analizi

ÖZ

DOĞAL DİL İŞLEME VE DERİN ÖĞRENME TEKNİKLERİ İLE DUYGU ANALİZİ: TÜRKÇE METİNLER ÜZERİNE BİR ÇALIŞMA

E-ticaret platformları, günümüzde tüketicilerin ürün tercihlerini belirlemede hızlı, kolay, temassız bir alışveriş deneyimi sunmasıyla pek çok insan tarafından tercih edilmektedir. Müşterilerin çevrimiçi alışverişleri esnasında satın alma işlemi öncesi inceledikleri ürün hakkında platformda yer alan olumlu ve olumsuz yorumlar, müşterilerin gerçekçi fikir edinmeleri ve işletmelerin ürün/hizmet kalitelerini arttırmak amacı ile bilgi edinmesi için kritik bir rol oynamaktadır. Bu yorumların işletmeler veya müşteriler için tek tek okunarak analiz edilmesi oldukça zor ve zaman alıcı bir süreçtir. Analiz sürecinin zorluğu, beraberinde insan hatası riski de getirebilmektedir. Bu sebeple işletme ve müşterilerin hızlı ve doğru analiz yapabilmesi için çeşitli derin öğrenme modelleri ile duygu analizi çalışmaları yapılmaktadır. Bu çalışmada, belirli ürün gruplarına yönelik e-ticaret sitelerinde yapılan yorumlar üzerinde derin öğrenme algoritmaları olan, Uzun kısa vadeli bellek modeli (LSTM), Evrişimsel sinir ağları algoritması (CNN), Geçitli tekrarlayan birim (GRU), Transformatörlerin çift yönlü kodlayıcı gösterimleri (BERT) algoritmasıyla oluşturulan tek algoritmalı ve hibrit modeller ile duygu analizi gerçekleştirilmiştir. Bu çalışmada e-ticaret platformlarında yapılan yorumların derin öğrenme algoritmaları ile analiz edilmesiyle elde edilen sonuçların, e-ticaret platformları üzerinde müşteri memnuniyetini arttırması ve buna bağlı olarak pazarlama stratejileri gibi alanlarda katkı sağlanması öngörülmektedir.

Name Surname : Zeynep Sena PEKEL

Advisor : Assoc. Prof. Dr. Atınç YILMAZ

Type and History : Master (Thesis), 2024

Major : Computer Engineering

Keywords : Natural Language Processing, Deep Learning, Sentiment Analysis

ABSTRACT

SENTIMENT ANALYSIS WITH NATURAL LANGUAGE PROCESSING AND DEEP LEARNING TECHNIQUES: A STUDY ON TURKISH TEXTS

E-commerce platforms are widely preferred by many individuals today due to their ability to provide a fast, easy, and contactless shopping experience, aiding consumers in determining their product preferences. During online shopping experiences, the positive and negative reviews about products available on the platform play a crucial role in enabling customers to form realistic opinions and for businesses to enhance the quality of their products/services. However, conducting analysis of these reviews individually can be a complex and time-intensive process, potentially introducing human error risks. Therefore, various deep learning models are being explored for sentiment analysis to enable businesses and customers to conduct rapid and accurate analyses. This study focuses on sentiment analysis of reviews posted on e-commerce sites for specific product categories using deep learning algorithms such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), and Bidirectional Encoder Representations from Transformers (BERT). Single algorithmic and hybrid models are developed to perform sentiment analysis. The results obtained from analyzing e-commerce platform reviews using deep learning algorithms are expected to enhance customer satisfaction on e-commerce platforms and contribute to areas such as marketing strategies.

İÇİNDEKİLER

Sayfa No.

ÖZ

ABSTRACT

TABLOLAR LİSTESİ iii

ŞEKİLLER LİSTESİ iv

KISALTMALAR vi

SÖZLÜK vii

GİRİŞ 1

1. DUYGU ANALİZİ VE DERİN ÖĞRENME KAVRAMLARI

1.1. Duygu Analizi 5

1.2. Doğal Dil İşleme 8

1.3. Derin Öğrenme 13

2. ÇALIŞMADA KULLANILAN ALGORİTMALAR VE TEKNİKLER

2.1. Derin Öğrenmede LSTM (Long ShortTerm Memory) 24

2.2. Geçitli Tekrarlayan Birim (GRU-Gated Recurrent Unit-) Tekniği 27

2.3. Evrişimsel Sinir Ağları Algoritması (CNN) 31

2.4. CNN, GRU VE LSTM Kullanılarak Yapılan Çalışmalar Dair Literatür 36

2.5. Çift Yönlü Kodlayıcı Gösterimleri (Bidirectional Encoder Representations from Transformers) 39

3. UYGULAMA YÖNTEMİ

3.1. Verilerin Toplanması ve Düzenlenmesi 43

3.2. Deneysel Çalışma Ortamı 51

3.3. Analiz Yöntemleri 51

4. ÇALIŞMA BULGULARI

4.1. BERT Modeli 53

4.2. LTSM Modeli 55

4.3. GRU Modeli 60

4.4. CNN Modeli 65

4.5. LSTM ve CNN Hibrit Modeli.....70
4.6. LSTM ve GRU Hibrit Modeli.....74
4.7. GRU ve CNN Hibrit Modeli 78
SONUÇ 85
KAYNAKÇA.....87

TABLOLAR LİSTESİ

Sayfa No.

Tablo 1. Derin öğrenme algoritmaları 19
Tablo 2. Nesne takibi için kullanılan yöntemler 32
Tablo 3. Veri Setinde En Sık Geçen Kelimeler 46
Tablo 4. LSTM Modeli Sonuçları 59
Tablo 5. GRU Modeline Ait Sonuçlar 64
Tablo 6. CNN Modeli Sonuçları 69
Tablo 7. LSTM-CNN Hibrit Modeli Sonuçları 73
Tablo 8. LSTM-GRU Hibrit Modeli Sonuçları 77
Tablo 9. GRU-CNN Hibrit Modeli Sonuçları 83
Tablo 10. Modellerin Performans Sonuçları 84

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 1. Veri Ön İzleme 9
Şekil 2. Doğal Dil İşleme Aşamaları 12
Şekil 3. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme Kronolojisi 14
Şekil 4. Makine Öğrenmesi Alanları Sınıflandırması 17
Şekil 5. LeNet mimarisi 20
Şekil 6. AlexNet Mimarisi 21
Şekil 7. ZF Net Mimarisi 21
Şekil 8. GoogleNet Mimarisi 22
Şekil 9. Resnet Mimarisi 23
Şekil 10. LSTM Mimarisi 25
Şekil 11. Geçitli Tekrarlayan Birim Hücre Mimarisi 28
Şekil 12. CNN Ağ Yapısı 33
Şekil 13. BERT Çalışma Yapısı 40
Şekil 14. Veri Setine Ait Kategorik Dağılımlar 44
Şekil 15. Veri Setinin İlk 19 Verisi 45
Şekil 16. Kelime Frekansı Grafiği 47
Şekil 17. İkili Kelime Grubu Frekans Grafiği 48
Şekil 18. Bütün Yorumlar için WordCloud Görseli 49
Şekil 19. Olumlu Yorumlar için WordCloud Görseli 50
Şekil 20. Olumsuz Yorumlar için WordCloud Görseli 50
Şekil 21. BERTurk Modelinin Çağırılması 53
Şekil 22. Verilerin Birimlere Ayrıştırılması 54
Şekil 23. Birimlerin Modele Sunulması 54
Şekil 24. Eğitim ve Doğrulama Verilerinin Oranı Tanımlanması 55
Şekil 25. Eğitim ve Doğrulama Test Verisinin Biçimi 55
Şekil 26. LSTM Modeli Oluşturulması ve Katmanları 56
Şekil 27. Modelin İlk 24 Iterasyon (Epoch) Sonuçları 57
Şekil 28. Modelin Son 24 Iterasyon (Epoch) sonuçları 58
Şekil 29. Modelin En yüksek Doğruluk Sonucu 58

Şekil 30. LSTM Modelinin Iterasyon Doğruluğu 59
Şekil 31. Modelin Karmaşıklık Matrisi 60
Şekil 32. GRU Modeli Oluşturulması ve Model Özeti 61
Şekil 33. GRU Modelinin İlk 24 İterasyon İlerlemesi..... 62
Şekil 34. GRU Modelinin Son 14 İterasyon İlerlemesi..... 63
Şekil 35. GRU Modeli Eğitim Doğruluğu..... 63
Şekil 36. GRU Modeli Karmaşıklık Matrisi Sonucu..... 65
Şekil 37. CNN Modeli Oluşturulması ve Katmanları..... 66
Şekil 38. CNN Modeli Eğitim Doğruluğu..... 67
Şekil 39. CNN Modelinin İlk 24 İterasyon İlerlemesi..... 68
Şekil 40. CNN Modelinin Son 11 İterasyon İlerlemesi..... 68
Şekil 41. CNN Modeli için Karmaşıklık Matrisi..... 70
Şekil 42. LSTM ve CNN Modelinin Oluşturulması ve Katmanları 71
Şekil 43. LSTM ve CNN Modelinin İlk 36 Iterasyon Sonuçları..... 72
Şekil 44. LSTM ve CNN Modelinin Son 5 Iterasyon Sonuçları 72
Şekil 45. LSTM ve CNN Modelinin Eğitim Doğruluğu 73
Şekil 46. LSTM ve CNN Modeli Karmaşıklık Matrisi 74
Şekil 47. LSTM ve GRU Hibrit Modeline Ait Katman Bilgileri 75
Şekil 48. LSTM ve GRU Hibrit Modeli İlk 24 İterasyon Sonuçları 76
Şekil 49. LSTM ve GRU Hibrit Modelinin Son 26 İterasyon Sonuçları 77
Şekil 50. LSTM ve GRU Modeline Ait Karmaşıklık Matrisi Sonuçları 78
Şekil 51. GRU ve CNN Modelinin Oluşturulduğu Kod Satırları..... 80
Şekil 52. GRU ve CNN Modelinin İlk 24 İterasyon Sonuçları 81
Şekil 53. GRU ve CNN Hibrit Modelinin Son 10 Iterasyon Sonuçları..... 81
Şekil 54. GRU ve CNN Hibrit Modelinin Iterasyon Doğruluk Grafiği 82
Şekil 55. GRU-CNN Hibrit Modeli Karmaşıklık Matrisi 83

KISALTMALAR

AI: Yapay Zekâ (Artificial Intelligence)

BERT: Transformatörlerin Çift Yönlü Kodlayıcı Gösterimleri (Bidirectional Encoder Representations from Transformers)

CNN: Evrişimli Sinir Ağları (Convolutional Neural Networks)

GRU: Geçitli Tekrarlayan Birim (Gated Recurrent Unit)

LSTM: Uzun Kısa Süreli Bellek ağları (Long Short-Term Memory)

ML: Makine Öğrenimi (Machine Learning)

MLM: Maskelenmiş Dil Modelleme (Masked Language Modeling)

NLP: Doğal Dil İşleme (Natural Language Processing)

NSP: Sonraki Cümle Tahmini (Next Sentence Prediction)

SÖZLÜK

Çift Yönlü Kodlayıcı Gösterimleri: (BERT), Google tarafından 2019 yılında geliştirilen derin öğrenme ve dikkat mekanizması kullanılarak oluşturulan doğal dil işleme (NLP) tekniğidir.

Derin Öğrenme: İnsanların öğrenme sürecini taklit ederek, sinir hücreleri ve hücreler arasındaki öğrenme bağlantısını yapay olarak oluşturarak makinelere insan beyninin öğrenme işlemine benzer şekilde öğretmeyi hedefler.

Doğal Dil İşleme: Bilgisayar ve makinelerin insan dilini anlayıp, yorumlaması ve insan gibi öğrenerek üretmesiyle ilgilenen yapay zekâ dalıdır.

Duygu Analizi: metinlerdeki duygusal içeriği tanımlamayı hedefleyen bir doğal dil işleme tekniğidir.

Gçitli Tekrarlayan Birim (Gru): (Gated Recurrent Unit), bir tür sinir ağı (RNN) hücresidir. GRU, özellikle doğal dil işleme alanında zaman serisi tahmini, örüntüleri tanıma gibi konularda kullanılabilmektedir.

Uzun Kısa Süreli Bellek ağları (LSTM): Uzun dönemli bağlılıkları öğrenebilmeyi amaçlayan bir tür tekrarlayan sinir ağıdır.

GİRİŞ

Duygu analizi, metinlerdeki duygusal içeriği tanımlamayı hedefleyen bir doğal dil işleme tekniğidir. Biz insan olarak duyduğumuz, okuduğumuz ve gördüğümüz verileri analiz eder ve bu analiz sayesinde dışarıdan aldığımız verileri kategorize ederiz. Bu alandaki temel amaç, tıpkı biz insanlar gibi bilgisayarın aldığı metin verisini analiz ederek kategorize edebilmesidir. Bu kategoriler, olumlu, olumsuz, nötr olarak bir konuya olan tepki ve yaklaşımı ifade etmektedir. Doğal dil işleme; bilgisayarların da insan dilini anlayabilmesi, yorumlaması ve üretebilmesiyle ilgilenen yapay zekâ dalıdır. Derin öğrenme, yapay zekâ modellerini kullanarak karmaşık ilişkileri ve desenleri otomatik olarak öğrenerek verilere dayalı tahminlerde bulunabilen bir makine öğrenme tekniğidir. Duygu analizinde karar verme süreçlerini geliştirmek için kullanılan makine öğrenmesi yaklaşımlarından biridir. Son zamanlarda duygu analizine olan ilgi kullanıcıların duygularını analiz edebilmek amacıyla oldukça artmıştır. Bu sebeple kolay ulaşabilen Facebook, Twitter ve Instagram gibi sıklıkla kullanılan sosyal medya platformları, çeşitli konularda analizler yapmayı amaçlayan şirketler için önemli bir veri kaynağı olmuştur.

Günümüzde sosyal medya bizlere duygularımızı belli şekillerde ifade etmemizi sağlamaktadır. Milyonlarca sosyal medya kullanıcısı duygularını metin, görsel, video, ses gibi verileri topluluğa açık bir şekilde paylaşmaktadır. Özellikle internet ve sosyal medya üzerinden paylaşılan milyonlarca metin, kullanıcıların duygusal durumları, düşünceleri ve geri bildirimleri hakkında zengin bir kaynak sağlamaktadır. Bu sebeple dijital ortamlarda veri çeşitliliği ve hacmi artmaktadır. Sosyal medya üzerinden veri analizinde, anket gibi geleneksel veri setlerinden farklı olarak sosyal medyadaki metinler dağınık bir yapıda olup, yazım yanlışları, dillere özgü çeşitlilik ve varyasyonlar içerebilen karmaşık bir yapıda olması analizi zorlaştırmaktadır. Analizin etkili olabilmesi için bu büyük verinin temizlenmesi gerekmektedir. Her ne kadar zorluklar bulunsa da duygu analizi müşteri ilişkileri yönetimi, pazarlama gibi birçok alanda kullanıldığında şirketlere fırsatlar sağlamaktadır.

Örneğin bir firma ürünü hakkındaki geri bildirimlerini analiz edebilmekte, ürün veya hizmetlerini geliştirerek marka itibarını kontrol edebilmektedir. Bunun dışında risk ve fırsatları tespit ederek karar verme süreçlerinde gerekli bilgiyi sağlayabilmektedir.

E-ticaret sektörü, hızla büyüyen ve değişen dinamiklere sahip bir alan olarak gün geçtikçe daha da önem kazanmaktadır. Bu alandaki rekabetin artmasıyla birlikte, müşteri memnuniyeti ve deneyimi giderek öne çıkmaktadır. Müşterilerin ürünler hakkında yaptığı yorumlar, satın alma kararlarını etkileyen önemli faktörlerden biridir. Dolayısıyla, e-ticaret sitelerindeki ürün grupları için yapılan yorumların duygu analiziyle incelenmesi, işletmelerin müşteri geri bildirimlerini daha etkili bir şekilde değerlendirmelerini sağlayabilir. Bu doğrultuda duygu analizi çalışmaları, İngilizce metinler üzerine yoğunlaştırılmış ve bu dildeki metinler üzerinde birçok araştırma yapılmıştır. Bu çalışmada e-ticaret sitelerinde belli ürün grupları için yapılan yorumlar üzerinde duygu analizi çalışması yapılıp kullanılan deri öğrenme ve doğal dil işleme tekniklerinin Türkçe metinler üzerindeki etkinliğinin incelenmesi amaçlanmıştır. Böylece, işletmeler müşteri memnuniyetini artırmak için daha hassas stratejiler geliştirebilecek araştırma bulgularına ulaşacak, ürün ve hizmet kalitesini iyileştirebilecek adımlar atabileceklerdir. Ayrıca, bu çalışma akademik literatüre katkı sağlayarak, Türkçe metinler üzerinde duygu analizi alanındaki bilgiyi arttıracak bulgular sunacak ve bu alanda daha ileri araştırmaların yapılmasına olanak tanıyacaktır. Eğer duygu analizi çalışmaları yapılmazsa öncelikle, müşteri geri bildirimlerinin değerlendirilmesinde eksiklikler ortaya çıkabilir. Müşterilerin ürünler hakkındaki duygusal tepkileri göz ardı edilirse, işletmeler müşteri memnuniyetini artırmak veya hizmetlerini iyileştirmek için gerekli adımları atmada eksik kalabilirler. Ayrıca, duygu analizi yapılmayan yorumların işletmeler için anlamlı bir şekilde sınıflandırılması ve analiz edilmesi daha zor olabilir. Bu durumda, müşteri geri bildirimlerinden elde edilecek değerli içgörüler kaybolabilir ve işletmelerin rekabet avantajı sağlaması güçleşebilir. Bununla birlikte, duygu analizi yapılmaması işletmelerin marka itibarı üzerinde olumsuz etkiler de yaratabilir.

Negatif yorumlar veya memnuniyetsizlikler fark edilmediği takdirde, işletmeler müşteri şikayetlerine veya endişelerine uygun şekilde yanıt veremezler ve bu da marka itibarının zarar görmesine yol açabilir. Sonuç olarak, duygu analizi çalışmalarının yapılmaması e-ticaret işletmeleri için müşteri memnuniyeti, rekabet avantajı ve marka itibarı açısından ciddi riskler oluşturabilir. Bu nedenle, müşteri geri bildirimlerinin etkili bir şekilde değerlendirilmesi için duygu analizine dayalı yöntemlerin kullanılması son derece önemlidir. Uzun Kısa Süreli Bellek Ağları (LSTM) algoritmaları, metin sınıflandırma problemlerine getirdikleri çözümlerle dikkat çekmektedir. Bu nedenle, tez çalışması sürecinde bu algoritmalardan LSTM kullanılmıştır. Bununla beraber çalışmada diğer yöntemler olan geçitli tekrarlayan birim (GRU) ve CNN olarak bilinen Convolitional Neural Network kullanılmıştır. Evrişimli Sinir Ağları (CNN), özellikle görüntü işleme ve desen tanıma gibi alanlarda geniş bir kullanım alanına sahiptir. Girdi veriler üzerinde lokal özellikleri vurgulayarak çalışır ve bu da özellikle görüntü verileri için çok etkilidir. Paralel işlemlere olanak tanır, bu da GPU'larda verimli bir şekilde çalışmalarına olanak sağlar. Ancak, daha karmaşık yapılarından dolayı eğitim ve bellek kullanımı açısından maliyetlidir. Gated Recurrent Unit (GRU), özellikle seriler ve zaman serileri gibi zamanla değişen verilerin işlenmesinde kullanılır. Basit bir yapıya sahiptir ve daha az parametreye sahiptir, bu da eğitim süresini azaltabilir. Ancak, LSTM kadar uzun vadeli bağımlılıkları işleyemez. Long Short-Term Memory (LSTM), sıklıkla doğal dil işleme ile yapılan analiz çalışmalarında ve zaman serisi analizi gibi alanlarda kullanılmaktadır. Diğer tekniklerden daha karmaşık yapıya sahiptir ve hafıza hücresi, unutma kapısı, giriş kapısı ve çıkış kapısı gibi bileşenleri içerir. Bu nedenle, uzun vadeli bağlılıkları daha etkili bir şekilde öğrenebilir ve işleyebilir. Ancak, daha fazla parametreye sahip olması ve eğitiminin daha uzun sürmesi gibi dezavantajları vardır.

Çalışma beş bölümden oluşmaktadır.

Birinci bölümde duygu analizi ve doğal dil işlemenin ardından derin öğrenme teorik yönleriyle ele alınmıştır. İkinci bölüm ise bu çalışmada kullanılan algoritmalar ve tekniklere yer verilmiştir. Bu bölümde duygu analizi kapsamında belirli unsurların ve yaklaşımların hem güçlü hem de kısıtlamalarına yer verilmiştir.

Üçüncü bölümde, uygulama yöntemi yer almaktadır. Bu bölümde çalışmanın amacı, önemi, çalışmanın sınırları, çalışmanın yöntemi ve deseni, bu çalışmadaki veri toplama yöntemi ve verilerin analizi gibi konulara yer verilmiştir. Dördüncü bölümde çalışmanın bulguları sunulmuştur. Son bölümde ise çalışma sonuçları doğrultusunda önerilere yer verilmiştir.

1. DUYGU ANALİZİ VE DERİN ÖĞRENME KAVRAMLARI

Çalışmanın bu bölümünde duygu analizi, derin öğrenme, kullanılan derin öğrenme teknikleri konularına teorik olarak ve bu konularda yapılan çalışmaların özetine yer verilmiştir.

1.1. Duygu Analizi

Duygu analizi, bir metin işleme tekniğidir. İşlenen verilerin metinlerden oluşması ile bir metin sınıflandırma problemi olarak değerlendirilir. Bir kelime, cümle veya metnin ifade ettiği anlamı analiz etmek için kullanılır. Genellikle, olumlu (pozitif), olumsuz (negatif) ve nötr (tarafsız) olmak üzere üç farklı sonuç elde edilir. Duygu analizinde pozitif ve negatif sınıflar yaygın olarak kullanılsa da bazı çalışmalarda düşüncenin belirtildiği kelime bir duyguyu pozitif veya negatif olarak belirtemeyebileceğinden, bu gibi kelimeler için nötr olması düşünülerek bu üç sınıf üzerinden analiz çalışmaları yapılmıştır. Bu yöntem, metinlerin duygusal içeriğini anlamak ve insanların duygusal tepkilerini ölçmek için yaygın olarak kullanılmaktadır (Şeker, 2016, s. 22). Duygu analizi, metin madenciliğinin önemli bir parçasıdır ve içinde altı çizili duyguları, görüşleri ve öznelliği analiz etme sürecidir. Bu yöntem, metinlerdeki duygusal içeriği anlamak ve insan davranışını etkileyen faktörleri belirlemek için kullanılır (Taboada vd., 2011, s. 267). Duygu analizi, bir diğer adıyla fikir madenciliği metinlerdeki duygusal içeriğin anlamlandırılması ve sınıflandırılması için kullanılan bir doğal dil işleme tekniği olarak karşımıza çıkmaktadır. Bu kategorideki metinler hem araştırmacılar ve analistler hem de kurumlar için büyük önem taşımaktadır. Çünkü bu metinler, insan davranışını anlamada ve pazarlama stratejilerini geliştirmede son derece değerli bilgiler sunar (Pang ve Lee, 2008, s. 9898).

Metin verileri genellikle cümleler halinde bulunur ancak birden fazla cümlenin bir araya gelerek bir konuya veya olguya olan görüşü içermesi durumunda kimi zaman paragrafın analiz edilmesi gerekmektedir.

Bu durumda incelenmesi gereken metin verisine göre duygu analizi belirli seviyelerde yapılabilmektedir. En çok kullanılan seviyeler belge düzeyi, konu odaklı, cümle düzeyi olmaktadır.

Belge düzeyinde duygu analizi: Doküman içerisindeki bütün verilerin veya bölümleri oluşturan paragrafların incelendiği düzeydir. Dokümanın kendisi belli bir konuda tek bir duygu öğesi olmaktadır. Genellikle bir belge veya metnin büyük ölçüdeki duygusal tonunu yansıtmaktadır. Ancak, bazen metinlerin içerdiği daha ince detaylar veya farklı duygusal yönler dikkate alınmadığından, daha spesifik ve detaylı bir analiz gerektiren durumlarda yetersiz kalabilmektedir. Bu tür durumlarda, belirli konular veya özelliklerle ilgili duyguları belirlemek için daha ince ayrıntılar üzerinde çalışabilen konu odaklı duygu analizi veya cümle düzeyinde duygu analizi teknikleri kullanılabilmektedir. Konu odaklı duygu analizi geleneksel belge düzeyinde yapılan duygu analizinden daha spesifik bir inceleme sunmaktadır.

Konu odaklı duygu analizi: Bir metinde belirli özelliklerle ilişkilendirilmiş duyguları tanımlamayı amaçlayan bir analiz tekniğidir. Özellik ve ilişkilere odaklanmasından, metinlerdeki duygusal içeriğin daha ayrıntılı ele alınabilmesini sağlamaktadır. Metin öğesi için duygu unsurlarını belirlemek amaçlanır, unsurlar tek bir duygu unsuru veya birbirileri ile bağlantılı olan birden fazla unsuru içerebilmektedir.

Cümle düzeyinde duygu analizi: Bir metin verisindeki her bir cümlenin duygusal içeriğin belirlenmesi ve sınıflandırılmasını amaçlayan tekniktir. Sınıflandırmada cümleyi bir öğe olarak alınır ve içerisindeki kelimeler özellik olarak ayrıştırılmaktadır.

2000'li yılların başlarından itibaren duygu analizi internetin kullanımının artmasıyla birlikte doğal dil işleme alanında en çok araştırılan konulardan biri olmuştur.

İnsanların fikirler, duyguları, değerlendirmeleri, bir konuya olan bakış açıları ve o konu hakkındaki kabul görme yaklaşımı gibi alanlarda analiz yapılabileceğinden problem alanı oldukça geniştir. Doğal dil işleme, duygu analizinden farklı olarak araştırma kapsamı 2000'lerden önce başlanmıştır. O zamanlardan itibaren çok etkin bir araştırma alanı olmaktadır.

Bu yıllardan itibaren çalışmalar duygu analizi özelinde yapılmaya başlanmış ve günümüzde yaygınlaşmıştır. Bu doğrultuda, araştırma alanının geniş olmasıyla birlikte müşteri ilişkileri yönetimi, reklamcılık gibi alanlarda fırsatlar firmalar için zenginleşmesi, araştırmaların yapılması için yüksek bir motivasyon sağlamaktadır (Liu. 2012, s. 7).

Günümüzde teknolojik gelişmelerin hızıyla birlikte, yeni problemler de ortaya çıkmakta ve bu problemlere çözüm olması için yeni ihtiyaçlar doğmaktadır. Bu durum, yeni araştırma alanlarının ve bilgisayar bilimlerinin oluşmasına yol açmaktadır. Bu yeni alanlar arasında sosyal medya ön plana çıkmaktadır (Çoban, 2026, s. 5). Dijital medyanın ve internet kullanımının artmasıyla birlikte, insanlar büyük miktarda metin verisi üretmekte ve paylaşmaktadır. Sosyal medya platformları, bloglar, ürün incelemeleri ve haber siteleri gibi çeşitli kaynaklarda kullanıcılar kendi fikirleri, görüşleri ve duygularını özgürce ifade edebilmektedirler. Sosyal medya platformları ve web siteleri gibi çeşitli kaynaklardan gelen geri bildirimler, genellikle büyük miktarda veri içerir. Ancak, sadece konuşulanları bilmek yeterli değildir. Bununla birlikte, kullanıcıların hissettiklerini anlamak da son derece önemlidir. Duygu analizi, bu geri bildirimlerdeki duyguları ve hisleri ortaya çıkarmak için kullanılan bir yöntemdir (Liu, 2012).

Sosyal medya, günümüzde yalnızca iletişim için bir araç olmaktan çıkıp çeşitli alanlarda etkili olan ve gelişen olaylara yön verebilen güçlü bir araç haline gelmiştir. Günümüzde birçok sosyal medya kullanıcısı, Facebook, Twitter gibi gündemdeki sosyal medya platformlarında herhangi bir durum ya da konu hakkında duygularını, düşüncelerini ve deneyimlerini özgürce ifade etmektedir. Bu nedenle, sosyal medya; ekonomi, pazarlama, siyaset ve fikir madenciliği gibi çeşitli alanlarda zengin birer veri kaynağı olarak hizmet etmektedir.

Ancak, hızla artışta olan veri hacmi ile büyük boyuttaki veriden analizler için anlamlı olacak bilgiyi çıkarabilmek için çeşitli otomatik, süreç kolaylaştırıcı teknik ve yöntemlere ihtiyaç duyulmaktadır. Duygu analizi, bu durumda önemli katkılar sunmaktadır. Duygu analizi işletmeler için önemli bilgiler üretmektedir. Satış ve pazarlama alanı, duygu analizinin en yaygın olarak kullanıldığı yerlerden biridir.

Bu bakımdan işletmeler, online alışveriş siteleri ve sosyal medya platformları üzerinde tüketici yorumlarının duygularını (pozitif, negatif veya nötr şekilde gruplara ayrılması) belirlemek için sıklıkla duygu analizine başvururlar. Duygu analizi, müşterilerin ürün, hizmet ve marka hakkında neyi sevdiklerini ve ne düşündüklerini anlamalarına yardımcı olur (Güran vd., 2014, s. 87). Özellikle bir ürünün değerlendirilmesinde, tüketicilerin memnuniyetini veya memnuniyetsizliğini belirlemek için kullanılabilir. Ayrıca, bir şirket, çıkardığı yeni bir ürün veya yaptığı reklam kampanyası ile ilgili sosyal medya hesaplarında yapılan yorumları ve bu yorumların diğer okuyucular üzerindeki etkisini ölçmek için duygu analizini kullanabilir. Şirketler, sundukları satış sonrası hizmetlerin müşterilerinin memnun kalıp kalmadığını değerlendirebilirler. Politikacılar seçim kampanyalarına yönelik seçmen tepkilerini ölçmek için duygu analizini seçim zamanında kullanabilirler (Demiroğlu, 2021, s. 8).

1.2. Doğal Dil İşleme

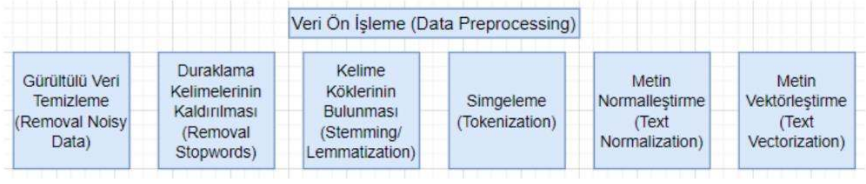
Doğal dil işleme (NLP), bilgisayarlar ve makinelerin insan dilini yorumlama ve anlamasını ve insanlarla etkileşimde bulunmasını sağlayan bir alandır. Doğal dil işleme, dilbilim ve bilgisayar biliminin birleşiminden doğan bir disiplindir. Bu disiplin, insanların kullandığı dilleri bilgisayarlar tarafından işlenebilir bir forma dönüştürmeyi amaçlar. Doğal dil işlemenin temel hedefi, doğal dilde yazılmış metinlerin bilgisayarlar tarafından anlaşılmasını, yorumlanmasını ve yanıtlanmasını sağlamaktır. Bu hedefe ulaşabilmek için belirli adımlar izlenmelidir. İlk olarak, verilerin toplanması ve doğal dilde yazılmış metinlerin elektronik ortamda bilgisayara aktarılması gerekmektedir. Ardından, dilbilim yöntemleriyle, matematiksel ve istatistiksel modeller kullanılarak bu metinlerin anlamı anlaşılmaya çalışılır.

Bu teknikler, metindeki kelimelerin anlamlarını, yapılarını, bağlamlarını ve anlamsal ilişkilerini çıkarmak için kullanılır. Doğal dil işleme, insanlarla bilgisayarlar arasındaki etkileşimi artırmayı ve özellikle büyük veri kümeleri üzerinde çalışan kurumlar ve araştırmacılar için doğal dil verilerini analiz etmeyi ve anlamayı kolaylaştırmayı hedefleyen geniş bir uygulama alanına sahiptir. Bu alanda birçok pratik kullanım örneği mevcuttur. Örneğin, arama motorları, kullanıcıların aradıkları bilgileri anlayarak en uygun sonuçları sunmayı amaçlar.

Akıllı sanal asistanlar, kullanıcıların doğal dilde iletişim kurduklarında istedikleri görevleri gerçekleştirir ve günlük yaşamda sıkça kullanılır. Otomatik çeviri sistemleri, farklı diller arasında anlık çeviri yapar. Otomatik metin özetleme ise uzun metinleri öz olarak sunar. Sosyal medya izleme, kullanıcılara uygun reklamların gösterilmesini sağlar. Bu örnekler, doğal dil işlemenin sağladığı geniş uygulama alanlarından sadece birkaçıdır (Yönet, 2023, s. 25-26).

Doğal Dil İşleme, doğal dillerin yapılarını çözümleyerek anlaşılmasını veya yeniden üretilmesini amaçlar. Bu çözümleme, bir dizi kolaylık sağlar; yazılı belgelerin otomatik olarak istenilen dile çevirmek, soru-cevap sistemleri, otomatik cevaplama ve komut anlama, konuşma sentezi, konuşma üretimi, otomatik metin özeti çıkarabilme, bilgiye ulaştırma gibi birçok alanı kapsar.

Bilgisayar teknolojilerinin kullanımının yaygınlaşması, bu tür alanlar üzerine geliştirilen yazılımların günlük yaşama her durumda dahil olmasını sağlamıştır. Örnek olarak, birçok kelime işlemci, yazılan metni dilbilgisi kurallarına göre düzenleyen bir imla denetimi aracı içerir. Bu araçlar asıl olarak doğal dil işleme yazılımlarıdır, çünkü yazılan metni analiz eder ve dilbilgisi kurallarını kontrol eder. Doğal Dil İşlemede, veri ilk olarak doğal dilde yazılmış bir metindir ve bilgisayarlar tarafından anlaşılabilir makine diline dönüştürülmesi gerekir. Bu dönüşüm işlemleri genellikle veri ön işleme olarak adlandırılır ve makine öğrenimi alanında önemli bir adımdır. Veri ön işleme ile ilgili olarak bir örnek aşağıdaki gibi Şekil 1'de gösterilmiştir (Ballı, 2021, s. 5).



Şekil 1. Veri Ön İzleme

Kaynak: Yazar tarafından oluşturulmuştur.

Gürültülü veri temizleme metnin içinden anlam ifade etmeyen, model eğitilirken gerekli olmayan ve modelin başarısını doğrudan kötü yönde etkileyecek verilerin çıkarılması işlemidir. Emojiler, özel karakterler veya noktalama işaretleri gürültü veri olarak örnek verilebilir.

Duraklama anlamı taşıyan kelimeler, cümlenin içerisinde tam olarak cümlenin bütün anlamını değiştirmeyen fakat dile özgü yapı dolayı kullanılan kelimelerdir. Türkçe dili için bağlaçlar bu kelimelere örnek verilebilir. Kelimelerin köklerine ayrıştırma, yani kök haline getirilme sürecidir. Bu süreç, doğal dil öğrenme uygulanan dile ve dilin yapısına göre değişebilir ve bu durum özelinde farklı teknikler ile algoritmaları bulunmaktadır. Simgeleme işlemi, cümlenin çeşitli kurallara göre daha küçük ve anlam ifade eden birimlere bölünmesi işlemidir. Cümleler kelimelere, deyimlere ve söz öbeklerine ayrılır. Bu işlem ayrı olarak, metin bölümleme (text segmentation) veya sözcük analizi (lexical analysis) olarak da adlandırılmaktadır. Metin normalleştirme, daha basit olarak ifade edildiğinde metnin standart bir form haline dönüştürülmesidir. Örnek olarak, Türkçe kullanılarak yazılmış bir tweet içerisinde "tmm", "tamaaammmm" kelimelerinin, kanonik hali olan "tamam" kelimesine dönüştürülmesidir.

Son olarak, metin vektörleştirme, doğal dil işleme alanında çalışma gerçekleştirilirken, metin halindeki bilgisayarın anlayamayacağı doğal olan dili bilgisayarın anlayabileceği girdi haline getirip yorumlaması ve makine öğrenmesi algoritmalarını çalıştırmak için sayısal temsillerine yani metin verilerini girdi vektörleri haline getirme işlemine denmektedir (Ballı, 2021, s. 6-9).

Teknolojinin ilerlemesiyle birlikte, doğal dil işleme tarafından geliştirilmiş birçok sistem, günlük yaşamımızın vazgeçilmez bir parçası haline gelmiştir. Bu teknolojiler, sıklıkla kullandığımız birçok cihaz ve uygulama aracılığıyla hayatımıza entegre olmuştur. Doğal Dil işleme birçok farklı alanda kullanılır. Bu alanlardan bazıları şu şekildedir (Yönet, 2023, s. 27-29):

Metin Sınıflandırma ve Etiketleme: Haber sınıflandırma, duygu analizi, spam filtreleme gibi metinleri belirli kategorilere veya etiketlere ayırma işlemlerinde kullanılır.

Bilgi Çıkarma: Metinlerden belirli bilgileri çıkarma işlemleri için kullanılır. Örneğin, bir metindeki önemli tarihleri, kişileri veya yerleri tanımlama.

Dil Çevirisi: Bir dildeki metinleri başka bir dile çevirme işlemleri için kullanılır. Google çeviri (translate) gibi dil çeviri araçları bir örneğidir.

Metin Üretimi: Otomatik yazılım içeriği oluşturma, makale yazma, şiir veya hikâye oluşturma gibi alanlarda kullanılır. GPT-3 gibi büyük dil modelleri, metin üretimi için kullanılan örneklerdir.

Konuşma Tanıma ve Konuşma Sentezi: Sesli komutları metin formatına dönüştürme ve metin tabanlı komutları sesli komutlara dönüştürme işlemlerinde kullanılır.

Metin Özetleme: Uzun metinleri özetleme işlemleri için kullanılır. Örneğin, makale özetleri oluşturma veya toplantı notlarını özetleme gibi.

Soru-Cevap Sistemleri: Metin verileri üzerindeki soruları anlama ve doğru cevapları bulma işlemlerinde kullanılır. Bu, birçok kullanım alanı bulan bir uygulamadır, örneğin, sanal asistanlar veya bilgi tabanlı sistemler.

Tıbbi Alan: Tıbbi raporları analiz etme, hastalık tanısı koyma, tıbbi belgeleri düzenleme gibi alanlarda kullanılır.

Eğitim ve Öğrenme: Dil öğrenme uygulamaları, otomatik dil öğretimi, yazım denetimi ve dil bilgisi öğrenimi gibi alanlarda kullanılır.

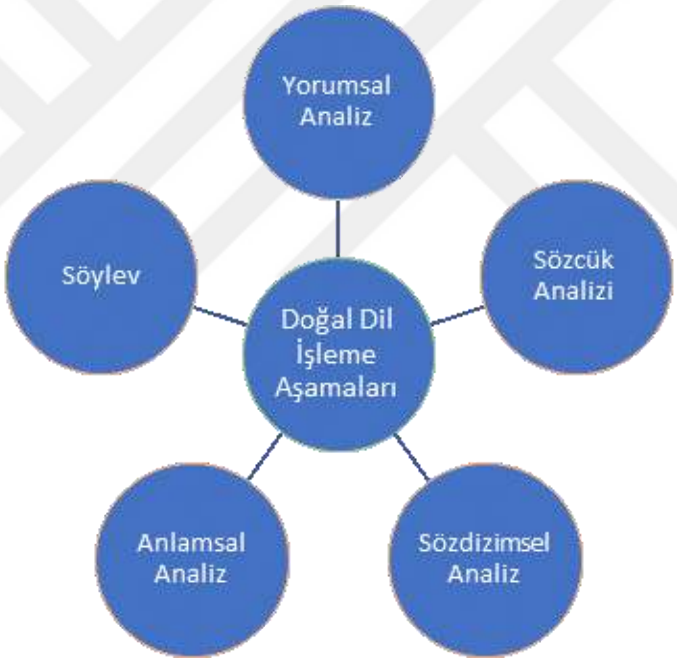
Sosyal Medya Analizi: Sosyal medya platformlarındaki metinleri analiz etme, trendleri belirleme, duygu analizi yapma gibi alanlarda kullanılır.

Doğal Dil İşleme, gün geçtikçe daha fazla alanda kullanılmaya başlanmaktadır ve bu alanlardan sadece birkaçıdır.

1980'lerin sonuna kadar, doğal dil işleme sistemleri karmaşık kurallarla oluşturuluyordu. Ancak, hesaplamanın gücündeki sürekli olan artış ve Makine Öğrenmesi algoritmalarının ortaya çıkmasıyla, Doğal Dil İşleme alanında büyük bir ilerleme kaydedilmiştir.

Bu dönemde, doğal dil işleme alanında kullanılan sistemlerin çoğu, İstatiksel Doğal Dil Öğrenme başlığı altında yer aldı. İstatistiksel yöntemlerin kullanımıyla ilgili önemli bir dönüm noktası, Bahl ve arkadaşlarının çalışmasıdır. (Bahl vd., 1983). Bu makale, konuşma tanıma işlemi için en yüksek olasılıkla kod çözme formülasyonunu tanımlamıştır. Çalışmada, konuşma tanıma görevinde kullanılmak üzere çeşitli istatistiksel modeller açıklandı. 1990'lara ait önemli bir başka istatistiksel çalışma ise Brown ve diğerlerinin makalesidir (Brown vd., 1990). Bu çalışmada, Fransızca'dan İngilizce'ye istatistiksel bir çeviri yaklaşımı sunulmuştur, Bayes Teoremi temel alınarak yapılan bu yaklaşım, doğal dil işleme alanında önemli bir araştırma olarak kabul edilir (Ballı, 2021, s. 29).

Doğal dil işlemenin aşamaları aşağıdaki Şekil 2'de gösterildiği gibidir.



Şekil 2. Doğal Dil İşleme Aşamaları

Kaynak: Yazar tarafından oluşturulmuştur.

Şekilde görüldüğü gibi doğal dil işleme temelde beş aşamadan oluşmaktadır. Bunlar sözcük analizi, sözdizimsel analiz, anlamsal analiz, söylev ve yorumsal analiz şeklinde sıralanmaktadır. Bu açıklamalardan sonra doğal dil işlemenin sağlamış olduğu faydaları açıklamak yerinde olacaktır.

Doğal dil işlemenin potansiyel faydaları şu şekilde sıralanır:

Doğal dil işleme, bireylerin bilgisayarlarla doğal dili kullanarak iletişim kurmalarına destek sağlar.

Makine öğrenme algoritmalarıyla birlikte kullanılarak büyük ölçekli verilerin analizine katkı sağlayabilir

Sunmuş olduğu çeşitli çözümler ve kolaylıklar sayesinde, zaman ve iş gücü tasarrufu sağlamaktadır.

Bu avantajlarının yanı sıra doğal dil öğrenme bazı potansiyel dezavantajlar da taşımaktadır. Aşağıda dezavantajlara değinilmiştir;

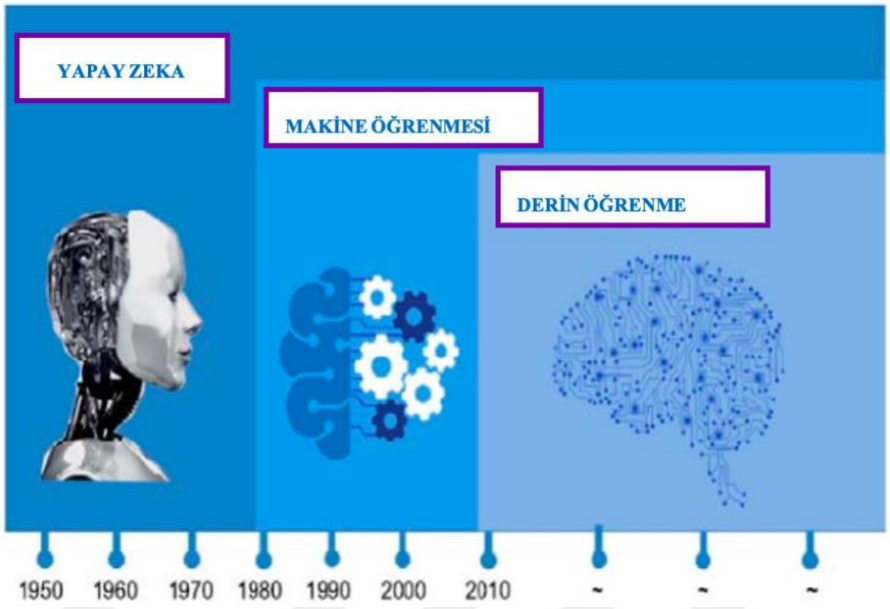
Sondan eklemeli dillerde (Türkçe) çalışmak zorlu olabilir.

Dilin kurallara uyulmaması ve anlamsız cümlelerin kullanılması, doğal dil işleme çalışmalarını zorlaştırabilir.

Dilbilgisi kurallarına ve noktalama işaretlerine uyulmaması, doğal dil işleme çalışmalarını zorlaştırabilir (Yönet, 2023, s. 33).

1.3. Derin Öğrenme

Derin öğrenme, insanların öğrenme sürecini taklit ederek, sinir hücreleri ve hücreler arasındaki öğrenme bağlantısını yapay olarak oluşturarak makinelere insan beyninin öğrenme işlemine benzer şekilde öğretmeyi hedefler. Derin öğrenme, yapay zekânın bir alt dalı olan makine öğreniminin bir alt dalıdır ve bu süreçte veriler, birbirini takip eden katmanlarda işlenir (Aydın, 2023, s. 17). Günümüzde yapay zekânın sıklıkla çalışmalarda kullanıldığı alanlardan bazıları ise doğal dil işleme, konuşma ve ses tanıma, sanal asistanlar (müşteri hizmetleri, akıllı ev sistemi), makine öğrenmesi platformları (tahmin ve sınıflandırma uygulamaları), yapay zekâ destekli optimizasyon uygulamaları, karar yönetimi, derin öğrenme platformları, biyometrikler ve robotik süreç otomasyonları, metin analizi gibi alanlardır (Tan vd., 2019). Yapay zekâ, makine öğrenmesi ve derin öğrenme kronolojisi aşağıda Şekil 3'te gösterilmiştir.



Şekil 3. Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme Kronolojisi

Kaynak: Kayaalp, K., ve Süzen, A. A. (2018). *Derin öğrenme ve Türkiye'deki uygulamaları.* IKSAD International Publishing House.

Derin öğrenme konusunu daha detaylı olarak açıklamadan önce yapay zekâ ve makine öğrenmesi konuları bu bölümde açıklanmıştır.

Yapay Zekâ

Yapay zekâ, bilgisayarların insanlar gibi düşünmelerini, öğrenmelerini, problemleri çözmelerini ve karar vermelerini hedefleyen bir sistemdir. Temel amacı, makineleri daha verimli ve etkin hale getirmektir. Yapay zekâ, bilgisayar bilimleri, matematik, istatistik, psikoloji ve beyin bilimleri gibi çeşitli disiplinlerin birleşimiyle oluşmuş geniş bir alandır. Yapay zekâ kavramı, Alan Turing'in 1950'de yayınladığı makalesiyle modern anlamda tanımlanmıştır. Bu makalede, akıllı makinelerin nasıl oluşturulacağı ve özellikle zekalarının nasıl test edileceği tartışılmıştır. İnsanların bir makineyle etkileşime geçtiklerinde, eğer makine insan gibi algılanıyorsa, bu durumda makinenin akıllı olduğu söylenebilir. Alan Turing, bilgisayarlardaki akıllı olan davranış biçimini, bilişsel olan görevler için insan seviyesinde performans elde edebilme yeteneği olarak tanımlamıştır. Bu tanım, sonrasında daha popüler hale gelen Turing testi olarak adlandırılmıştır (Aydın, 2023, s. 10; Turing, 1950, s. 491).

Yapay zekâ sistemleri, yirmi birinci yüzyılın başlarından itibaren toplumun her kesiminde yaygın bir şekilde kullanılmaktadır. İş yerlerindeki bilgisayar tabanlı sistemlerden evlerde kullanılan elektrikli süpürgelere, araç navigasyonundan banka hesaplarının yönetimine kadar birçok alanda yapay zekâ teknolojilerinden faydalanılmaktadır (Özgeldi, 2019, s. 200).

Yapay zekâ çalışmaları, insan zekasını taklit etme çabalarıdır ve genellikle makinelerin insanların öğrenme süreçlerini modellemesiyle ilgilidir. İnsanların öğrenme süreçleri genellikle beyinde gerçekleşir ve bu nedenle beyin yapısının incelenerek makineler üzerinde benzer yapıların oluşturulması mümkündür. İnsan öğrenme süreci, nöron adı verilen beyin hücrelerinin etkileşimi yoluyla gerçekleşir. Bu nedenle, bilgisayarlar aracılığıyla yapay sinir ağları oluşturularak öğrenim süreci simüle edilmiştir (Yılmaz ve Kaya, 2021).

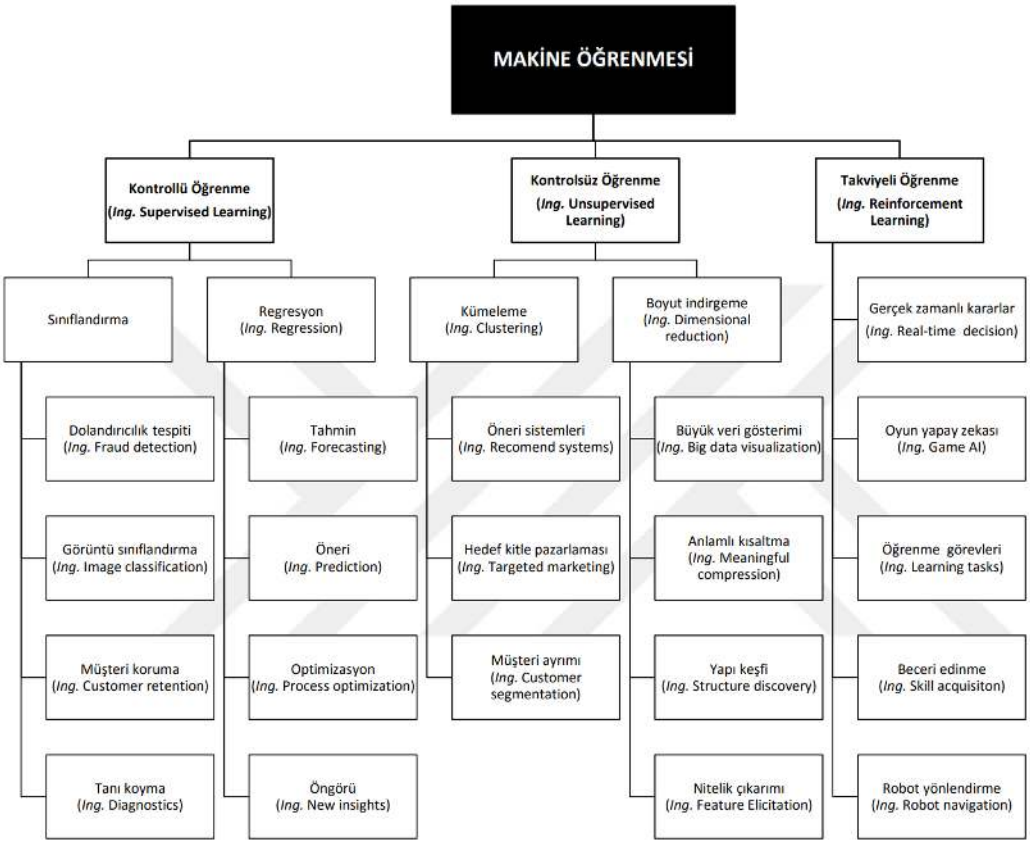
Yapay zekâ sistemleri, geçmişte insanlar için zor olan birçok görevi günümüz insanı için kolaylaştırarak hayal dahi edilemeyecek avantajlar sağlamıştır. Örneğin, en uygun rotayı bulmak için kullanılan yapay zekâ sistemleri, alışveriş deneyimlerinde kişisel öneriler sunan en tercih edilebilecek ürünleri belirlemek için kullanılan uygulamalar, otoyollarda kaza riskini en az seviyeye indirmeyi amaçlayan yapay zekâ destekli otonom araçlar gibi birçok yapay zekâ uygulaması, insanlara zaman kazanımı, maliyet azalması ve iş gücünün azalması açısından önemli avantajlar ve katkılar sağlamaktadır. Ancak, yapay zekâ fikrinin ilk ortaya çıktığı günden beri, bu sistemlerin kontrolsüz bir şekilde gelişerek insanlığın sonunu getirebileceği endişesi de var olmuştur. Yapay zekanın sınırlarının neler olabileceği ve insanlık adına bir tehdit oluşturup oluşturmayacağı sürekli olarak gündemde tartışılmaktadır. Ancak, bu endişelere rağmen, bu tür bir durumun gerçekleşmesi için bir neden bulunmamaktadır. Çünkü bilgisayar bilimi genel olarak "ne" ve "nasıl" soruları ile modellenmeye dayanmaktadır. Yapay zekanın kendiliğinden insan kontrolünden çıkması, bilgisayarın kendi başına "niçin" sorusunu da modellemesi gerekir. Ayrıca, insan yalnızca zekadan ibaret değildir; insanı diğer canlılardan ayıran özellikleri olan bilinç ve irade gibi özellikleri vardır. Bu nedenle, yapay zekâ sistemlerinin insan beyniyle rekabet edebilmesi için yapay bir bilinç ve iradeye sahip olmaları gerekmektedir (Sarıel, 2017, s. 22; Köroğlu, 2017; Coşkun ve Gülleroğlu, 2021, s. 954).

Yapay zekâ (AI), farklı strateji ve tekniklerle makineleri insan benzeri hale getirmek için çalışmalar yapılan kapsayıcı bir alandır.

Yapay zekâ, Alexa, Siri olmak üzere, akıllı asistanlardan robot süpürgelere ve otonom otomobillere kadar geniş bir yelpazeyi kapsamaktadır. Makine öğrenimi (ML), yapay zekanın bir alt dalıdır. ML, bilgisayar sistemlerinin karmaşık görev ve işlemleri açık talimatlar verilmeden gerçekleştirmek adına kullanacakları algoritmaları ve istatistiksel modelleri geliştirme bilimidir. Bu sistemler, açık talimatları uygulayarak problemi çözmek yerine örüntüler, örüntelere göre tahmin ve çıkarımlar üzerine çalışır. Bilgisayar sistemleri, büyük hacimde olan geçmiş verileri işlemek ve veri örüntülerini tanımak için makine öğrenimi algoritmaları kullanmaktadır. Makine öğrenimi, yapay zekâ alanının bir parçası olmaktadır ancak tüm yapay zekâ faaliyetleri makine öğrenimiyle sınırlı kalmamaktadır.

Makine Öğrenmesi

Makine öğrenimi, günlük hayatımızın her yerinde, birçoğumuzun belki de farkında olmadığı bir şekilde kullanılmaktadır. Örneğin, nesneleri tanıma, konuşmayı metne çeviren yazılımlar veya kullanıcıların ilgi alanlarına ve geçmişte inceledikleri ürünlere dayanarak ürün önerileri sunan elektronik ticaret siteleri gibi temel örnekler verilebilir. Makine öğrenimi, denetimli-kontrollü, denetimsiz-kontrolsüz ve takviyeli-pekiştirmeli öğrenme olmak üzere üç ayrı kategoriye ayrılır ve bu kullanım alanları aşağıdaki Şekil 4'te gösterilmiştir.



Şekil 4. Makine Öğrenmesi Alanları Sınıflandırması

Kaynak: Vural, 2021

Denetimli öğrenme, önceden sınıflandırılmış verilere dayanarak tahminler yapmak için eğitilen bir makine öğrenimi türüdür. Bu öğrenme yöntemi etiketlenmiş verilere ihtiyaç duyar. Denetimli öğrenme sürecinde, modele örnekler sunulmaktadır. Her bir örnek hem girdi verilerini hem de bu verilere karşılık gelen etiketleri içermektedir. Model, tahminler yapmak için girdi verilerini tanıma amacıyla kullanır ve bu tahminleri doğruluğunu kontrol etmek için verilerin etiketleriyle karşılaştırmaktadır. Denetimli öğrenmenin amacı, eğitim verilerinden öğrenerek yeni ve bilinmeyen verilerle ilgili doğru tahminler yapabilmektir. Denetimli öğrenme genellikle sınıflandırma ve regresyon gibi uygulamalarda kullanılır. Denetimsiz öğrenme, verilerde etiketlenmiş bilgiye ihtiyaç duymayan bir makine öğrenimi türüdür. Bu yöntemde modele birçok örnek sunulur. Ancak, bu örneklerde yalnızca girdi verileri bulunur; etiketler bulunmaz.

Denetimsiz öğrenme, veri setlerinin çok büyük olması ve bu verilerin etiketlenmesinin zor veya mümkün olmaması durumlarında kullanılabilir. Son olarak pekiştirmeli öğrenme sürecinde, önceden belirlenmiş herhangi bir çıktı değeri girdilere dayanarak verilmez. Bunun yerine, elde edilen çıktılar doğru ve yanlış olarak işaretlenir. Yanlış sonuçlar tespit edildiğinde, program güncellenir ve öğrenme süreci yeniden gerçekleştirilir (Kim, 2017, s. 151).

Yapay zekâ ve makine öğrenmesi kavramlarının açıklanmasından sonra yeniden derin öğrenme konusu daha detaylandırılmıştır. Derin öğrenme algoritması, makine öğrenmesi algoritmasının bir alt dalı olmaktadır. Derin öğrenme, verilerin kullanım işleminde makine öğrenmesi algoritması gibi direkt olarak değil, katman olarak isimlendirilen özel bir türü temsil etmektedir. Derin öğrenme algoritmaları, yapay sinir ağlarına dayanan ve enerji tabanlı modellerden oluşmaktadır. Bu algoritmalar mimari olarak genellikle birçok katman ve gizli değişkenden oluşmaktadır. Derin öğrenme, birçok gizli katmanın varlığından dolayı parametre sayısı oldukça fazla olan bir alandır. Her bir katman, kendinden sonra yerleştirilen birden fazla eğitilebilir katmandan oluşur. Derin öğrenme modelleri, temel olarak sinir ağı katmanlarıyla donatılmıştır. Katmanların birbirine bağlanma yapısı, katman sayısı gibi birçok özellik yeni derin öğrenme modellerinin çıkmasını sağlamıştır (Özcan, 2014).

Gradyan temelli Geleneksel (Konvolüsyonel) Sinir Ağları, el yazısıyla yazılmış 0'dan 9'a kadar olan sayıları öğrenip sınıflandırmak için kullanılan ilk temel derin öğrenme ağı olan LeNet ile başlamaktadır. 2006'da derin öğrenme yapısının gösterilmesine rağmen, derin öğrenme, daha çok ImageNet yarışmasıyla popüler hale gelmiştir. ImageNet yarışması, görüntü tanıma için çeşitli algoritmaların sunulduğu bir platformdur. Bu algoritmalar içerisinde daha çok derin öğrenme mimarileri ön plana çıkmaktadır. Derin öğrenme mimarileri, nesne tanıma alanında başarılı sonuçlar ortaya koymuştur. Bunun yanı sıra derin öğrenme mimarileri, örüntü tanıma, tespit, sınıflandırma gibi alanlarda yanı sıra gelecek tahmini, ilaç üretimi, sözlük oluşturma, sinyal işleme, tıbbi uygulamalar ve finansal analiz gibi pek çok alanda kullanılmaktadır.

Elde edilen sonuçlara göre, derin öğrenme yapılarının bilinen pek çok yöntemden daha iyi sonuçlar verdiği açıkça görülmektedir (Hinton vd., 2006, s. 1527).

Aşağıda yer alan Tablo 1'de derin öğrenme algoritmalarının ortaya çıkış tarihleri, parametre sayıları ve hata oranları yer almaktadır (Doğan ve Türkoğlu, 2019, s. 425).

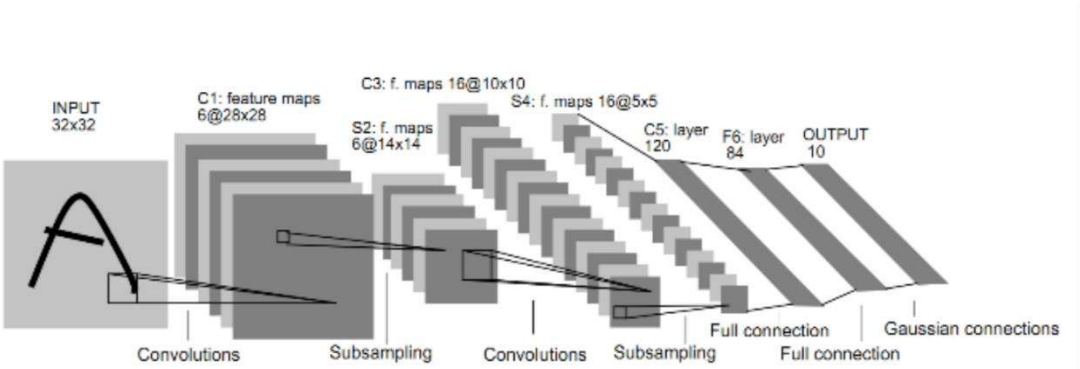
Tablo 1.Derin öğrenme algoritmaları

Yıl	**Derin Öğrenme Algoritması**	**Geliştirici**	**Hata Oranı**	**Parametre Sayısı**
1998	LeNet	Yann LeCun ve ark		60 Bin.
2012	AlexNEt	Alex Krizhevsky, Geoffrey Hinton, Ilya Sutskever	%15.3	60 Mil.
2013	ZFNet	Matthew Zeiler ve Rob Fergus	%14.8	
2014	GoogLeNet	Google	%6.67	4 Mil
2014	VGG Net	Simonyan, Zisserman	%7.3	138 Mil
2015	ResNet	Kaiming He	%3.6	

Kaynak: Yazar tarafından oluşturulmuştur.

LeNet: LeCun ve arkadaşları tarafından geliştirilen ilk konvolüsyonel yapay sinir ağı olarak bilinir. LeCun, banka çeklerindeki sayıların tanımlanabilmesi için geliştirdiği bu ağı LeNet olarak adlandırmıştır. Bu ağ, 0-9 arasındaki sayıları sınıflandırmış olduğu için 10 sınıflı bir yumuşatma katmanı kullanmaktadır. LeNet mimarisi içinde ortalama havuzlama (Average pooling) kullanılmıştır.

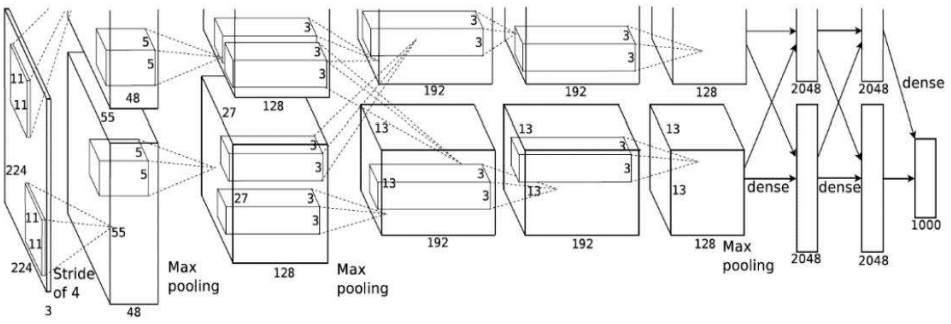
LeNet mimarisi, tam bağlı bir katman (dense), ReLU (aktivasyon) katmanı, havuzlama katmanı ve yumuşak bağlı katmanlardan oluşur. Aşağıdaki Şekil 5'te LeNet mimarisinin yapısı gösterilmiştir. Bu, bir dijit verisi sınıflandırmak için kullanılan bir mimari örneğini gösterir. Giriş verisi 32x32 piksel boyutundaki bir görüntüyü içerir (LeCun vd., 1998; Doğan ve Türkoğlu, 2019, s. 420).



Şekil 5. LeNet mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

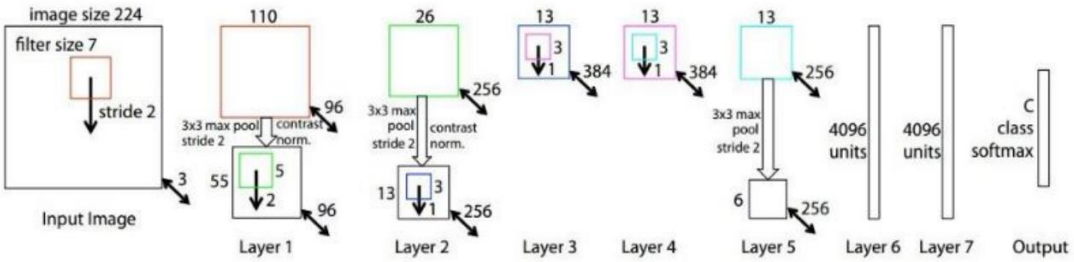
AlexNet: 2012'de gerçekleştirilen 2012 ILSVRC ImageNet Büyük Ölçekli Görsel Tanıma yarışmasında, AlexNet adlı derin öğrenme mimarisi birinci olmuştur. Bu mimari, örüntü tanıma için hata oranını %26'lardan %15'lere kadar düşürmeyi başarmıştır. Bu başarı, AlexNet'in en tanınmış derin öğrenme mimarilerinden biri haline gelmesini sağlamıştır. 25 katmandan oluşmakta olan AlexNet derin öğrenme ağı içinde 5 temel konvolüsyon katmanı yer almaktadır. Genellikle her konvolüsyon katmanının ardından bir aktivasyon katmanı olan relu katmanı kullanılmaktadır. Bununla birlikte giriş katmanı, normalizasyon katmanı, havuzlama(pooling) katmanı, dropout kamanı, tam bağlı (full-connected) katman, yumuşak bağlı (SoftMax) katman ve çıkış katmanı bulunmaktadır. AlexNet mimarisi aşağıdaki Şekil 6'da gösterilmiştir. Bu çok katmanlı mimaride, her katmanın kendi işlemini yapmasından sonra bir sonraki katmana verileri girdileri olarak iletmesi gerekmektedir. Giriş verileri, ağ içerisinde katmanlar arası iletilirken veri miktarları oldukça fazladır (Doğan ve Türkoğlu, 2019, s. 420).



Şekil 6. AlexNet Mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

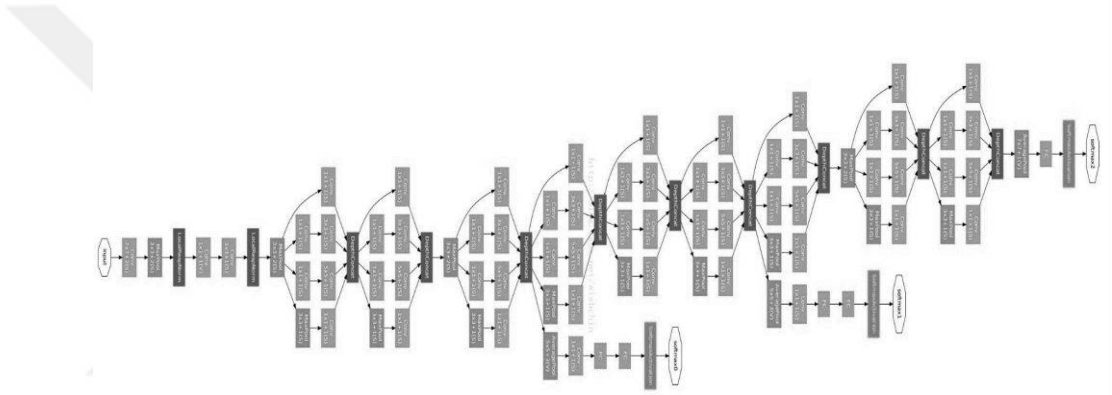
ZF Net: AlexNet'in popüler hale gelmesiyle birlikte derin öğrenme mimarileri 2012'den itibaren çeşitlenmeye başladı. Matthew Zeiler ve Rob Fergus, 2013 ILSVRC ImageNet yarışması için ZFNet adında bir mimari geliştirdiler ve bu mimari ile %11.2 hata oranı sonucuyla yarışmayı kazanmışlardır. Aşağıdaki şekilde AlexNet mimarisine bazı değişiklikler eklenerek ortaya çıkan bir mimari yapısı bulunmaktadır. AlexNet'te giriş katmanında 11x11 filtre kullanılırken, ZFNet'te kullanılan filtre 7x7 matrislidir. Aktivasyon için RELU fonksiyonu, eğitim için gradyan iniş ve hata kaybı için çapraz entropi kaybı kullanılmıştır. ZFNet mimarisinde, görselleştirme tekniği olarak Deconvolutional Network geliştirilmiştir. Bu teknik Şekil 7'de sunulmuştur. Teknik mimariye farklı v etkili bir boyut kazandırmış ve derin öğrenme mimarisini çok daha ileri bir noktaya taşımıştır (Doğan ve Türkoğlu, 2019, s. 422).



Şekil 7. ZF Net Mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

GoogleNet: 2014 yılında düzenlenen ILSVRC yarışmasını kazanan GoogleNet (Inception) olmuştur. GoogleNet, karmaşık bir yapıya sahiptir ve %5.7 gibi düşük bir hata oranıyla yüksek bir başarı elde etmiştir. 22 katman derinliğine sahip GoogleNet, 144 katman içeren bir yapıdan oluşmaktadır. Inception modülü, farklı boyutlar üzerinde filtreleme yaparak daha önce görülmeyen derin öğrenme mimarilerinden farklı bir yapı sunmuştur. Diğer derin öğrenme mimarilerinden farklı bir şekilde, art arda yer alan katmanlardan ziyade derinlikli bir yapı kullanılmıştır. Bu derinliği, modüler bir filtreleme mantığı oluşturmaktadır. GoogleNet mimarisinin yapısı, aşağıdaki Şekil 8'de gösterilmiştir (Szegedy vd., 2015; Doğan ve Türkoğlu, 2019, s. 423).



Şekil 8. GoogleNet Mimarisi

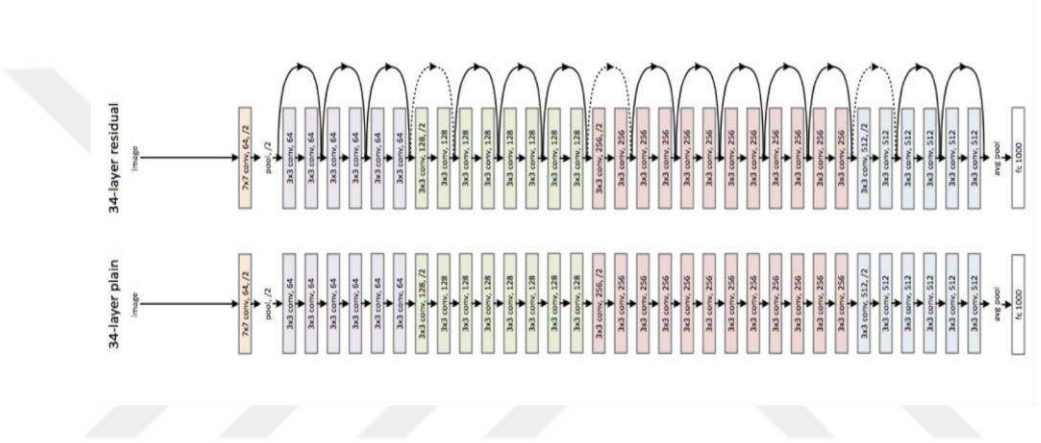
Kaynak: Yazar tarafından oluşturulmuştur.

VggNet: 2014 yılında geliştirilen bir derin öğrenme modeli olan bu mimari, ImageNet 2014 yarışmasında %7.3 hata oranı elde ederek başarılı bir performans göstermiştir. Simonyan ve Zisserman tarafından Oxford Üniversitesi'nde tasarlanmış olan mimari, 6 farklı modeli içermektedir. Bu 6 farklı model, sırasıyla 11, 13, 16 ve 19 konvolüsyon katmanından oluşur (Simonyan ve Zisserman, 2015; Doğan ve Türkoğlu, 2019, s. 421).

ResNet: Microsoft ResNet, 2015 yılında gerçekleştirilmiş ILSVRC ImageNet yarışmasının galibi olmuştur.

Bu yarışmada %3,6 gibi bir hata oranı elde etmiştir. İnsanlar ortalama olarak %5-%10 hata oranıyla görüntü sınıflandırma yaparken, %3,6'lık bir hata oranını elde etmek görsel tanıma alanında insanlardan daha iyi bir performans sergilediğini göstermektedir.

Daha önceki mimarilerden daha derinleştirilmiş bir yapıya sahip olan bu mimarinin katmanlarının sayısı, diğer derin öğrenme mimarilerine kıyasla daha fazladır. Resnet 152 katman içeren bir mimaridir. Aşağıdaki Şekil 9'da örnek Resnet mimarisi görülmektedir (Doğan ve Türkoğlu, 2019, s. 425).



Şekil 9. Resnet Mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

2. ÇALIŞMADA KULLANILAN ALGORİTMALAR VE TEKNİKLER

Literatürde derin öğrenme kullanılarak yapılan duygu analizi çalışmaları, İngilizce metinler üzerine yoğunlaştırılmış ve bu dildeki metinler üzerinde birçok araştırma yapılmıştır. Bu çalışmada e-ticaret sitelerinde belli ürün grupları için yapılan yorumlar üzerinde duygu analizi çalışması yapılıp kullanılan derin öğrenme ve doğal dil işleme tekniklerinin Türkçe metinler üzerindeki etkinliğinin incelenmesi amaçlanmıştır. Çalışmanın bu bölümünde bu çalışmada kullanılan algoritmalar ve tekniklere yer verilmiştir. Bu bölümde duygu analizi kapsamında belirli unsurların ve yaklaşımların hem güçlü hem de kısıtlamalarına açıklanmıştır. Uzun Kısa Süreli Bellek Ağları (LSTM- Long ShortTerm Memory) algoritmaları, öğrenme problemlerine getirdikleri çözümlerle dikkat çekmektedir. Bu nedenle, tez çalışması sürecinde bu algoritmalardan LSTM kullanılmıştır. Bununla beraber çalışmada diğer bir yöntem olan geçitli tekrarlayan birim (GRU) ve CNN olarak bilinen Convolitional Neural Network bu bölümde açıklanmıştır. Sözü geçen modeller, derin sinir ağı mimarilerine dayanmaktadır.

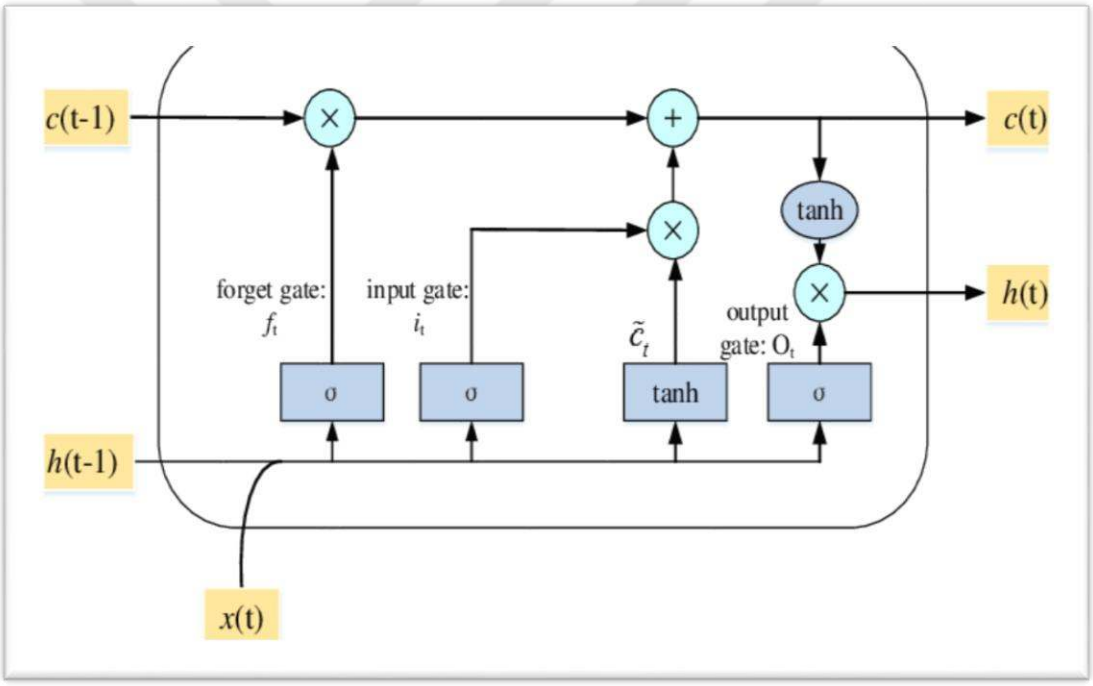
2.1. Derin Öğrenmede LSTM (Long ShortTerm Memory)

Uzun Kısa Süreli Bellek Ağları (LSTM), özellikle "LSTM" olarak bilinen, uzun süreli bağlılıkları öğrenebilmekte olan özel bir tür tekrarlayan sinir ağıdır. Bu ağlar, tekrarlayan sinir ağları (RNN) gibi tekrarlayan bağlantılardan oluşmaktadır. LSTM, Hochreiter ve Schmidhuber tarafından tanıtılmış ve sonraki gerçekleştirilen çalışmalarda birçok kişi tarafından eklemeler yapılarak popüler hale getirilmiştir. Bilgileri uzun süre hatırlamak için tasarlanmış olan Uzun Kısa Süreli Bellek ağları (LSTM), diğer RNN türlerinden farklı olarak, ağırlık güncelleme prosedüründe karşılaşılan sorunları ele alır. Bu sorunlar, ağırlık değişikliklerinin çok küçük (gradyan kaybı) veya çok büyük (patlayan gradyanlar) olması durumunda ortaya çıkar.

Bir girdinin gizli katmanlar ve ağ çıktısı üzerinde uyguladığı etki, ağın tekrarlayan bağlantıları etrafında dönerek azalabilir veya artabilir. Bu durum literatürde gradyan kaybı problemi olarak bilinir.

LSTM, bu gradyan kaybı sorununun çözümü için özel olarak tasarlanmış bir RNN mimarisi olmaktadır. LSTM hücreleri, kapılar ve ağırlıklar tarafından oluşturulur. Bir LSTM katmanı, bellek bloklarından oluşan bir dizi bloğa sahiptir. Her blok, bir veya daha fazla tekrar eden bağlı bellek hücresi ve bu hücreler için devamlı yazma, okuma ve sıfırlama işlemlerini gerçekleştiren üç çarpımsal birim olan giriş, çıkış ve unutma kapıları içermektedir. Ağ, hücrelerle ancak kapıları aracılığıyla etkileşime geçebilmektedir (Rothman, 2018, s. 376; Gavcar ve Metin, 2021, s. 5).

x_t, h_t ve c_t'nin t anında sırasıyla girdi, gizli durum ve hücre durumu olarak tanımlanan LSTM mimarisi aşağıdaki Şekil 10'da sunulmuştur.



Şekil 10. LSTM Mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

LSTM ağı, t=1 zaman diliminden T'ye kadar aşağıdaki formülleri kullanarak ağ birimi etkinleştirmelerini değerlendirerek $x = (x1, \dots, xT)$ giriş serisinden $y = (y1, \dots, yT)$ çıkış serisine bir aktarımı hesaplar:

$forgetgate = \sigma(Wixxt + Wimmt{-}1 + Wicct{-}1 + bi)$ (2.1)

$inputgate = \sigma(Wfxxt + Wfmmt{-}1 + Wfcct{-}1 + bf)$ (2.2)

$outputgate = \sigma(Woxxt + Wommt{-}1 + Wocct{-}1 + bo)$ (2.3)

$ct = ft \odot ct{-}1 + it \odot g(Wcxxt + Wcmmt{-}1 + bc)$ (2.4)

$mt = ot \odot h(ct)$ (2.5)

$yt = \phi\ (Wymmt + by)$ (2.6)

W terimleri ağırlık matris anlamını taşımaktadır (örneğin, Wix, ilk giriş kapısından bir sonraki girişe kadar olan ağırlık matrisidir), Wic, Wfc, Woc, gözetleme bağlantıları için ağırlıkların köşeden matrisleri olmaktadır. b terimleri sapma vektörlerini belirtmektedir (örneğin, bi, giriş kapısının sapma vektörüdür), σ sigmoid fonksiyonunu belirtmektedir, i, f, o ve c sırasıyla giriş kapısı, unutma kapısı, çıkış kapısı ve hücre aktivasyon vektörleri olmaktadır, bunların tümü hücre çıkış aktivasyon vektörüyle aynı boyutta sahiptir. m, $\odot$ vektörlerin çarpımını belirtmektedir, g ve h, hücre girişi ve hücre çıkışı aktivasyon fonksiyonlarını belirtmektedir (Çifcibası, 2021, s. 12-13).

Derin öğrenme alanında, Uzun Kısa Süreli Bellek ağlarının (LSTM) bir dizi avantajı bulunmaktadır:

Uzun Süreli Bağımlılıkları Öğrenebilme: LSTM'ler, uzun süreli bağımlılıkları öğrenme yeteneği ile öne çıkar. Bu özellik, zaman serileri gibi uzun vadeli bağımlılıkları olan veri setlerinde etkili bir şekilde çalışmalarını sağlar.

Gradyan Problemlerini Çözme: LSTM'ler, kaybolan veya patlayan gradyan gibi sorunları çözmek için özel olarak tasarlanmıştır. Bu sayede, derin sinir ağlarının daha uzun süreli eğitimlerde daha istikrarlı bir şekilde çalışmasını sağlar.

Çok Katmanlı Modelleme: LSTM'ler, derin öğrenme modellerinin çok katmanlı yapısına kolayca entegre edilebilir. Bu da daha karmaşık problemlerin çözümü için daha fazla esneklik sağlar.

Duyarlılık ve Esneklik: LSTM'ler, girdi verilerinin zaman veya sıra bilgisini koruyarak işleyebilme yeteneği ile özellikle zamansal verilerde etkilidir. Bu, metin

işleme, konuşma tanıma, video analizi gibi birçok alanda kullanılabilirliğini artırır.

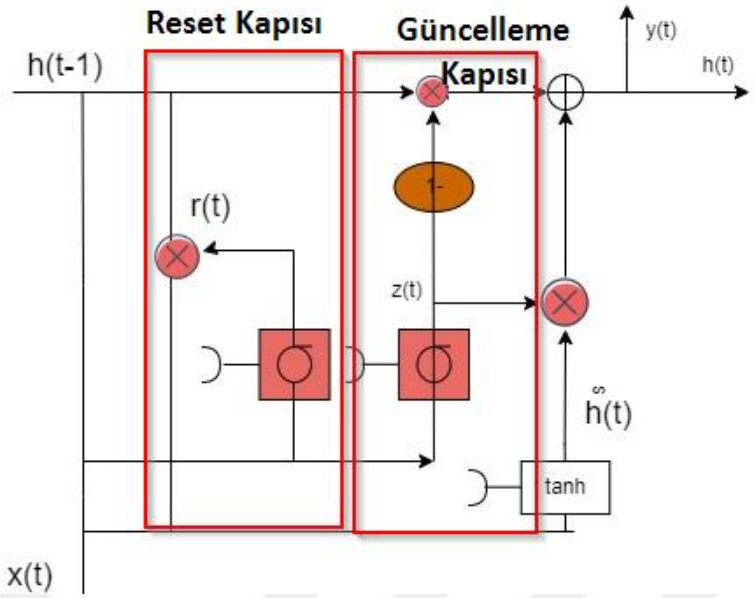
Hafıza Kapasitesi: LSTM'ler, girdi verilerindeki önemli bilgileri uzun süre saklayabilme yeteneğine sahiptir. Bu özellik, özellikle bellek gerektiren görevlerde avantaj sağlar.

Paralel İşleme Yeteneği: LSTM'ler, bazı varyasyonlarında paralel işleme yeteneğine sahiptir, bu da eğitim ve çıkarım süreçlerinde hız artışı sağlar.

Bu avantajlar, LSTM'leri derin öğrenme alanında popüler ve etkili bir seçenek haline getirmiştir.

2.2. Geçitli Tekrarlayan Birim (GRU-Gated Recurrent Unit-) Tekniği

Geçitli tekrarlayan birim (GRU), klasik yapay sinir ağlarının uzun dönem bağımlılık sorununa çözüm üreten LSTM modelinin bir varyasyonudur. GRU, özellikle zaman serisi verileri veya doğal dil işleme gibi sıralı verilerin işlenmesinde kullanılır. GRU, önceki zaman adımlarından bilgiyi taşıyan gizli bir durum vektörüne sahiptir ve bu vektör, modelin geçmiş bilgiyi hatırlamasına ve gelecekteki tahminlerde kullanmasına olanak tanır. GRU, LSTM (Uzun Kısa Süreli Bellek) gibi bir başka popüler RNN varyantıyla benzer özelliklere sahiptir, ancak daha basit bir yapıya sahiptir. GRU, LSTM'ye kıyasla daha az parametre gerektirir ve eğitimi daha hızlıdır, bu nedenle bazı durumlarda tercih edilir. Ancak, her iki modelin de başarımı problem bağlamına ve veri setine bağlı olarak değişebilir. GRU'nun temel amacı, sıralı verilerdeki zaman ve bağımlılıkları modellemek ve bu veriler üzerinde etkili bir şekilde işlem yapmaktır. Şekil 11 GRU hücre detayını göstermektedir.



Şekil 11. Geçitli Tekrarlayan Birim Hücre Mimarisi

Kaynak: Yazar tarafından oluşturulmuştur.

Yeni GRU mimarisi, LSTM'den esinlenerek geliştirilmiş ve temel avantajlarını korurken mimariyi basitleştirmeyi amaçlar. GRU'lar, LSTM'e kıyasla daha basit bir yapıya sahiptir çünkü daha az kapı kullanır ve gizli durumlar ile bellek hücreleri arasındaki ayrımı ortadan kaldırır. Yalnızca iki kapı bulunur: sıfırlama ve güncelleme kapıları. Sıfırlama kapısı, önceki gizli durumun hangi kısmının muhafaza edileceğine karar verirken, güncelleme kapısı, yeni girişin ne kadarının gizli duruma dahil edileceğini belirler. Gizli durum aynı zamanda hücre durumu ve çıkış işlevini de yerine getirir, bu yüzden ayrı bir çıkış kapısı yoktur. Bu nedenle GRU'nun uygulanması daha basittir ve LSTM'den daha az parametre gerektirir (Tunç, 2024: s. 47-48).

Geçitli Tekrarlayan Birim (GRU), geleneksel RNN modelinin bir varyasyonu olarak öne sürülen bir tür tekrarlayan sinir ağı (RNN) mimarisi olmaktadır. GRU içerisinde bulunan kapılar, önceki verinin ne kadarının saklanacağını ve yeni oluşan girdinin mevcut duruma ne kadarının entegre olacağını belirler. Güncelleme kapısı, önceki verinin korunma düzeyini kontrol eder ve ağın önemli bilgileri zaman içinde kaybedilmesini engeller.

Sıfırlama kapısı ise mevcut durumun hesaplanmasında geçmiş verinin ve durumun ne kadarının dikkate alınması gerektiğini belirleyerek ağın daha yeni bilgilere odaklı olmasını sağlamaktadır. GRU'nun maruz kalınan durumunun seviyesini kontrol edebilecek herhangi bir mekanizma olmadığı için, her seferinde bütün durumu açığa çıkarmaktadır.

GRU'nun matematiksel formülleri aşağıdaki gibidir.

1. Kapıyı Sıfırla

   $z_t = \sigma(W_z x_t + V z h_{t-1} + b_z)$

2. Güncelleme Kapısı

   $r_t = \sigma(W_r x_t + V r h_{t-1} + b_r)$

3. Aday Aktivasyonu

   $\tilde{h}_t = \tanh(W_c x_t + V c (r_t \otimes h_{t-1}))$

4. Gizli Durum Güncellemesi yukarıdaki denklemlerde:

   $\tilde{h}_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes h_t$

Formüllerde;

$\tilde{h_t}$, t zaman adımındaki gizli durumu temsil eder.

• x_t , t zaman adımındaki girişi temsil eder.

• $\otimes$ eleman bazında çarpmayı belirtir.

• W_r , W_z ve W_c , antrenman sırasında öğrenilen ağırlık matrisleridir.

Sıfırlama kapısı (r_t) önceki gizli duruma ait (h_{t-1}) ne kadarının unutulmuş olması gerektiğini belirlerken, güncelleme kapısı (z_t) güncellemek üzere yeni aday aktivasyonunun ($\tilde{h_t}$) ne kadarının güncelleme için kullanılacağını belirlemektedir . z_t, bir sonraki durum hücresine ne kadarlık bilgi iletilmesi gerektiğini belirleyen güncelleme kapısını temsil etmektedir.

Güncelleme kapısı için daha büyük olan bir veri, bir sonraki durum hücresine iletilen daha fazla bilgi olduğu anlamına gelmektedir. rt, önceki verilerin ne kadarlık bir kısmının göz ardı edilmesinin gerektiğini belirleyen sıfırlama kapısını temsil etmektedir. Sıfırlama kapısı için daha büyük bir veri, önceki hücreden daha fazla bilginin göz ardı edilebileceği anlamına gelmektedir (Tung, 2024, s. 33-34).

Özetlemek gerekirse DRU uygulamada hem avantaj hem de dezavantajlara sahiptir.

Avantajlar:

- Daha az parametre gerektirir: GRU, LSTM'e göre daha az parametre kullanır, bu da modelin daha hızlı eğitilmesini ve daha az bellek tüketmesini sağlar.
- Daha basit yapısı: GRU, sadece sıfırlama ve güncelleme kapılarına sahip olduğu için daha basit bir yapıya sahiptir, bu da daha hızlı eğitim süreleri ve daha az hesaplama gücü gerektirir.
- Daha az aşırı uyum: GRU, daha az kapıya sahip olduğu için daha az eğilimlidir aşırı uyuma, özellikle daha küçük veri setlerinde daha iyi performans gösterebilir.

Dezavantajlar:

- Daha az esneklik: GRU'nun daha az kapıya sahip olması, bazı karmaşık ilişkileri modellemekte LSTM kadar esnek olmadığı anlamına gelebilir.
- Daha az bellek kapasitesi: GRU'nun daha az kapıya sahip olması, uzun vadeli bağımlılıkları modellemek için LSTM'den daha az bellek kapasitesine sahip olduğu anlamına gelebilir.
- Performansın veri setine bağlı olarak değişmesi: GRU'nun LSTM'den daha iyi veya daha kötü performans gösterip göstermediği, kullanılan veri seti ve problem alanına bağlı olarak değişebilir.

2.3. Evrişimsel Sinir Ağları Algoritması (CNN)

Derin öğrenme ile nesne tespiti ve takibi alanında yapılan çalışmalar incelendiğinde genel olarak dört adım uygulanmaktadır. Bunlar (Tan ve ark., 2021):

Ön işleme aşaması; Veri setlerinde belirlenen kategorilere göre oluşturulan verilerin işlenmeye hazır hale getirilmesi için yapılan ön işlemlerdir. Bu ön işlemler, veriyi sonraki analizler ve adımlar için hazırlayan işlemleri içerir ve genellikle sistematik hataları düzeltmeyi veya telafi etmeyi amaçlar. Örneğin, dijital görüntüler geometrik veya atmosferik düzeltmelere tabi tutulabilir. Ancak, tüm bu düzeltmeler her veri seti için gerekmeyebilir. Bu hatalar genellikle sistematik olduğundan, verinin kullanıma sunulmadan önce giderilmesi mümkündür.

Nesnenin tespiti; Nesne tespiti aşaması genellikle iki adımdan oluşur. İlk adımda, nesne belirgin hale getirilir ve ikinci adımda nesne arka plandan ayrılır. Bu aşamada, genellikle nesnenin görünüm ve şekline ilişkin özellikler kullanılır ve bu özellikler genellikle nesne tespit algoritmalarının temelini oluşturur. Nesne tespiti aşaması, nesnenin izlenmesi için uygun bir algoritmanın seçilmesini belirler ve sonraki aşamaların başarısını büyük ölçüde etkiler, bu yüzden önemlidir. Nesnenin tespiti ve takibinde kullanılan bazı yöntemler şunlardır:

- R-CNN (Bölgeye Dayalı Evrişimli Ağlar)
- YOLO (You Only Look Once)
- SSD (Tek Çekim Çok Kutulu Tespit)
- Mask R-CNN
- Fast R-CNN
- Faster R-CNN

Nesnenin sınıflandırılması; Nesneleri özelliklerine göre gruplandırmak için genellikle bir sınıflandırma işlemi uygulanır. Bu işlemde, nesnenin kendine özgü özellikleri kullanılır.

Örneğin, doku, eğim, zaman, renk ve hareket gibi özellikler, nesne sınıflandırmasında en yaygın olarak kullanılan özniteliklerdir. Bu özellikler, nesneleri birbirinden ayırt etmek ve belirli kategorilere veya sınıflara atamak için kullanılır. Sınıflandırma işlemi, genellikle önceden eğitilmiş bir model veya sınıflandırma algoritması kullanılarak gerçekleştirilir ve nesnelerin belirli sınıflara ait olup olmadığını belirlemek için bu özniteliklerin kombinasyonunu değerlendirir.

Nesnenin takibi; Nesneleri izlemek için çeşitli algoritmalar mevcuttur. Bazıları yalnızca tek bir nesnenin takibini yaparken, diğerleri birden fazla nesnenin izlenmesi için kullanılabilir. Bu algoritmalar genellikle geniş kategorilere ayrılabilir. Nesne izlemek için yaygın olarak kullanılan yöntemler genellikle üç gruba ayrılabilir ve bu yöntemlerin özellikleri Tablo 2'de gösterilmiştir (Hanbay ve Üzen, 2017, s. 45).

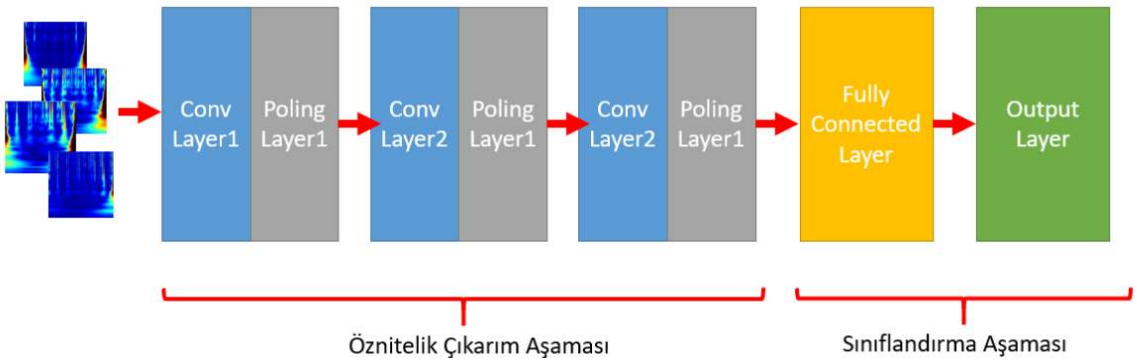
Tablo 2. Nesne takibi için kullanılan yöntemler

Nesne Takip Yöntemi	Kullanımı	İşlem Zamanı	Başarı
Nokta tabanlı	Deterministik model Olasılık modeli	Düşük-Orta Orta-Yüksek	Orta Orta
Çekirdek tabanlı	Çoklu görünüme dayalı Şablon tabanlı	Düşük-Orta Orta	Düşük OrtaYüksek
Siluet tabanlı	Kontur gelişimi Şekil eşleştirme	Orta Yüksek	OrtaYüksek Yüksek

Kaynak: Yazar tarafından oluşturulmuştur.

Nesne tespiti, sınıflandırılması problemini çözmek üzere özelleşmiş bir derin öğrenme mimarisi olan evrişimli sinir ağları CNN olarak kısaltılmaktadır (Fukushima, 1980, s. 194). Evrişimli Sinir Ağı (CNN), özellikle görüntü analizi alanında etkili olan ve genellikle derin öğrenme sinir ağları arasında yaygın olarak kullanılan bir mimaridir. CNN, geleneksel yapay sinir ağlarına benzer bir temel yapıya sahiptir, ancak özel olarak geliştirilmiş evrişim işlemlerini kullanarak girdi olarak görüntüleri işler.

Bu evrişim işlemleri, belirli bir görüntü ile önceden tanımlanmış filtreler arasındaki eleman bazında çarpımı içerir ve bu şekilde farklı öznitelikleri vurgular. Bu süreç, görüntülerden otomatik olarak öznitelikler çıkarmak için uygulanır. Geleneksel makine öğrenme yöntemlerinde öznitelikler genellikle manuel olarak belirlenirken, CNN ve benzeri derin öğrenme yöntemleri, bu öznitelikleri otomatik olarak evrişim gibi işlemlerle elde eder. Evrişimli Sinir Ağı (CNN) mimarileri genellikle iki temel aşamadan oluşur. İlk aşama, görüntülerden özniteliklerin çıkarıldığı evrişim aşamasını içerir. Bu aşamada, özel işlemler uygulanarak çeşitli öznitelikler elde edilir. Sınıflandırma aşaması ise bu öznitelikleri kullanarak görüntünün sınıfını belirler ve genellikle tam bağlantılı katmanları içerir. Bir CNN, evrişim katmanları, havuzlama katmanları ve tam bağlantılı (FC) katmanları gibi katmanlardan oluşur ve bu katmanlar bir araya getirilerek CNN ağı oluşturulur. CNN ağlarının önemli bileşenleri arasında bırakma (dropOut) katmanları ve aktivasyon fonksiyonları da bulunur. Aşağıda Şekil 12'de gösterilen yapı, özellikle görüntü işleme ve sınıflandırma görevleri için tasarlanmış güçlü bir derin öğrenme aracını ifade eder.



Şekil 12. CNN Ağ Yapısı

Kaynak: Yazar tarafından oluşturulmuştur.

CNN modelleri, çeşitli görüntü işleme tekniklerinde başarılı bir şekilde kullanılmıştır ve CNN (Evrişimli Sinir Ağı) modellerinin temel mimarisini temsil etmektedir.

Modelin başarısını etkileyen temel faktörler arasında, özellikleri belirlemek için evrişimli katman, doğrusal olmayan katmanlarla sistemin doğrusal olmamasını

sağlamak, ağırlık sayısını azaltmak ve uyumu kontrol etmek için havuzlama katmanı, veriyi düzleştirerek hazırlamak için düzleştirme katmanı yer almaktadır. Klasik sinir ağı için veri ve sınıflandırma kullanan standart sinir ağı, bu beş katmandan oluşmaktadır (Alaca, 2023).

CNN'ler, temelde insan görsel sisteminden ilham alarak, görüntülerdeki öznitelikleri çıkarmak ve anlamak için tasarlanmıştır. CNN'ler genellikle üç temel bileşen içerir:

Evrişim Katmanları (Convolutional Layers): Bu katmanlar, girdi görüntü üzerinde bir dizi evrişim işlemi uygular. Her bir evrişim işlemi, bir filtre (kernel) kullanılarak görüntünün farklı bölgelerindeki özellikleri vurgular. Örneğin, bir kenar, bir köşe veya bir desen gibi.

Havuzlama Katmanları (Pooling Layers): Bu katmanlar, evrişim sonrası elde edilen öznitelik haritasının boyutunu azaltmak ve özniteliklerin translasyon ve ölçek değişikliklerine karşı daha güçlü olmasını sağlamak için kullanılır. Havuzlama işlemi genellikle maksimum havuzlama veya ortalama havuzlama gibi yöntemlerle gerçekleştirilir.

Tam Bağlantılı Katmanlar (Fully Connected Layers): Bu katmanlar, öznitelikleri sınıflara atamak için kullanılır. Son havuzlama katmanından elde edilen öznitelikler, tam bağlantılı katmanlara beslenir ve ardından sınıflandırma yapmak için kullanılır.

CNN'ler, genellikle veri setleri üzerinde eğitilir ve ardından yeni görüntülerdeki nesneleri tanımak veya sınıflandırmak için kullanılır. Özellikle bilgisayarlı görü ile ilgili problemlerde oldukça başarılı sonuçlar vermektedirler.

Bu açıklamalardan sonra çalışmada kullanılan CNN, GRU ve LSTM modelleri kısaca karşılaştırılmalı olarak özetlenmiştir.

CNN (Evrişimli Sinir Ağı)

- Temelde görüntü işleme ve desen tanıma gibi görevlerde kullanılır.
- Evrişim katmanları ve havuzlama katmanlarından oluşur.
- Girdi veri üzerinde lokal özelliklerin vurgulanmasını sağlar.
- Yapıları paralel işlemlere olanak tanır, bu da GPU'larda verimli bir şekilde çalışmalarına olanak sağlar.
- Bellek kullanımı ve hesaplama gücü bakımından daha verimli olabilir.

GRU (Gated Recurrent Unit)

- Seri veri işleme ve zaman serisi tahmini gibi görevlerde kullanılır.
- LSTM'nin bir türevidir ve daha basit bir yapıya sahiptir.
- Daha az parametreye sahiptir ve eğitimi daha hızlı olabilir.
- Bir hafıza hücresi ve güncelleme ve unutma kapılarından oluşur.
- Uzun zaman aralıklarındaki bağımlılıkları daha iyi ele alabilir.

LSTM (Long Short-Term Memory)

- Seri veri işleme, doğal dil işleme ve zaman serisi analizi gibi görevlerde kullanılır.
- Geleneksel RNN'lerin (Recurrent Neural Network) problemlerini gidermek için tasarlanmıştır.
- Daha karmaşık bir yapıya sahiptir ve hafıza hücresi, unutma kapısı, giriş kapısı ve çıkış kapısı gibi bileşenleri içerir.
- Uzun vadeli bağımlılıkları daha etkili bir şekilde öğrenebilir ve işleyebilir.
- Daha fazla parametreye sahiptir ve eğitimi daha uzun sürebilir.

2.4. CNN, GRU VE LSTM Kullanılarak Yapılan Çalışmalar Dair Literatür

CNN, GRU ve LSTM gibi modellerin kullanıldığı çalışmalar birçok farklı alanda gerçekleştirilir. Bu üç modelin sıkça kullanıldığı bazı alanlar:

Görüntü İşleme ve Tanıma: CNN, özellikle nesne tanıma, yüz tanıma, görüntü sınıflandırma ve segmentasyon gibi görüntü işleme ve tanıma görevlerinde yaygın olarak kullanılır.

Doğal Dil İşleme (NLP): LSTM ve GRU gibi sinir ağı modelleri, metinlerin anlamını çıkarmak, dil modelleri oluşturmak, metin sınıflandırma, metin üretimi (doğal dil üretimi) gibi doğal dil işleme görevlerinde kullanılır.

Zaman Serisi Analizi: GRU ve LSTM, zaman serilerinin tahmini, zaman serisi sınıflandırma, finansal veri analizi, hava durumu tahmini gibi zamanla değişen verilerin analizi için kullanılır.

Hareket Tanıma: CNN, video analizi ve hareket tanıma gibi görevlerde kullanılabilir. Örneğin, otomatik sürücülerde, güvenlik kameralarında ve oyun kontrollerinde kullanılabilir.

Ses İşleme: Ses tanıma, konuşma tanıma ve ses sentezi gibi ses işleme görevlerinde LSTM ve GRU gibi modeller kullanılabilir.

Tıbbi Görüntüleme: CNN, tıbbi görüntüleme alanında tümör tespiti, patoloji tanıma ve radyoloji gibi görevlerde kullanılabilir.

Bu modeller, genellikle karmaşık yapılarının ve öğrenme yeteneklerinin yanı sıra çeşitli alanlarda verimlilikleri nedeniyle tercih edilirler. Bu bölümde literatürde farklı alanlarda yapılan araştırma sonuçlarına yer verilmiştir.

Çakmak ve Selvi yapmış oldukları araştırmada CNN, RNN, LSTM, GRU kullanarak protein ikincil yapı tahmini yapmışlardır. Yapılan araştırmada dört derin öğrenme yöntemi olan CNN, RNN, LSTM ve GRU kullanılarak geliştirilen modellerin performanslarının bir karşılaştırılmıştır.

CNN, RNN, LSTM ve GRU modellerinin doğruluk oranları sırasıyla %82,54, %82,06, %81,1 ve %81,48 olarak bulunmuştur (Çakmak ve Selvi, 2022, s. 44).

Toprak, 2023'te Türkiye, Rusya ve Katar'daki kritik petrol endeksi ve hisse senedi fiyatlarını içeren bir veri setini kullanarak fiyat tahmini çalışması gerçekleştirdi. Araştırma, Rassal Orman Regresyonu (RFR), LSTM ve CNN+LSTM yöntemlerini kullanarak sonuçları kıyasladı. Karşılaştırmalar, LSTM'in en iyi sonuçları verdiğini ortaya koymuştur (Toprak, 2023). Yılmaz ve Orman, 2021'de, Twitter'daki Covid-19 verileri üzerine duygu analizi çalışması yürütmüştür.

Yapılan çalışmada, olumlu ve olumsuz düşünceler sınıflandırıldı ve derin öğrenme yöntemlerinden biri olan LSTM modeli kullanılarak maksimum %97 doğruluk başarısı elde edilmiştir (Yılmaz ve Orman, 2021). Dönmez ve Becerikli ise yapmış oldukları çalışmalarında, Türkçe tweetler üzerinde duygu analizi yapılarak üç farklı derin öğrenme algoritması- LSTM (Long Short-Term Memory, Uzun Kısa Vadeli Bellek), BERT (Bidirectional Encoder Representations from Transformers, Dönüştürücülerden Çift Yönlü Kodlayıcı Temsilleri) ve CNN (Convolutional Neural Networks, Evrişimli Sinir Ağı)- karşılaştırılmıştır. Üç model eğitim veri setleri üzerinde eğitilerek test veri setleri üzerinde değerlendirilmiştir. Ulaşılan sonuçlara göre modellerin doğruluk oranları karşılaştırılmıştır. Sonuç olarak LSTM ve CNN-LSTM modelleri her ne kadar çok başarılı olsa da BERT modelinde elde edilen doğruluk oranının daha başarılı olduğu görülmüştür (Dönmez ve Becerikli, 2023, s. 136).

Çelik ve Odabaş, gerçekleştirdikleri araştırmada, işitme engelli bireylerin kameralar karşısında yaptıkları hareketlerin algılanıp işaret diline çevrilmesinde herhangi bir sensör kullanmadılar. Bunun yerine, Konvolüsyonel Yapay Sinir Ağları (CNN) ve Uzun Kısa Süreli Bellek (LSTM) gibi derin öğrenme tekniklerini kullandılar. 10 rakam ve 29 harfin işaret dili hareketleriyle eğitilen CNN + LSTM modelleri, tahminlemede %97 başarı oranı elde etti. Bu sonuçlar, işitme engelli bireylerin kameralar karşısında yaptıkları hareketlerin algılanıp metne dönüştürülmesinde derin öğrenme yöntemlerinin kullanılabileceğini göstermektedir (Çelik ve Odabaş, 2020, s. 930).

Yapılan bir başka araştırmada Salur ve Aydın'ın (2020), GSM operatörlerine dair tweetlerin derin öğrenme algoritmalarıyla duygu analizi yapılmıştır. Dört farklı derin öğrenme algoritması kullanılarak sınıflandırma başarısı karşılaştırılmıştır. En yüksek başarı oranı, Evrişimli Sinir Ağları (CNN) ve Uzun Kısa Süreli Hafıza (LSTM) modeliyle elde edilen yaklaşık %82'lik başarı olmuştur (Salur ve Aydın, 2020).

Yapılan bir başka araştırmada ise iki farklı veri seti üzerinde kiraz yaprak hastalığı tespiti için farklı derin öğrenme modellerinin performansı incelenmiştir. Ortalama sonuçlara göre, MobileNet-V2 modeli %98.68 doğruluk ve %98.83 hassasiyet ile yüksek bir performans sergilemiş, Inception-V3 modeli %94.17 F1-Skor ile tatmin edici sonuçlar vermiştir. Önerilen CNN modeli %95.83 F1- Skor ile dikkat çekmiş, önerilen CNN + LSTM modeli %96.33 geri çağırma değeri ile tatmin edici sonuçlar sunmuş ve önerilen CNN + BiLSTM modeli %96.33 F1-Skor ile yüksek bir performans göstermiştir (Bozcu, 2024, s. 2).

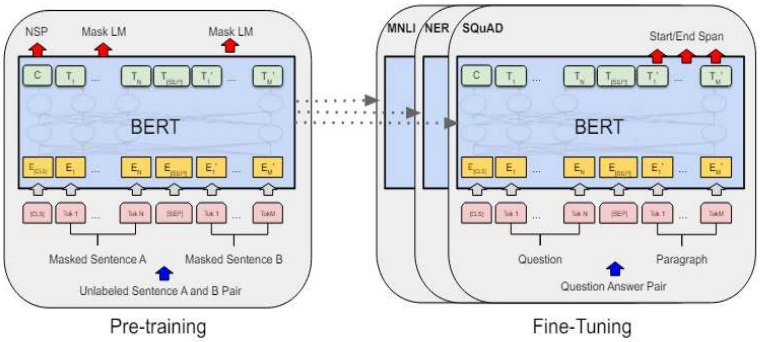
Yapılan bir başka araştırmada ise, Derin Sinir Ağları (Deep Neural Network - DNN) veya Çok Katmanlı Algılayıcı (Multi-Layer Perceptron - MLP), Evrişimsel Sinir Ağı (Convolutional Neural Network - CNN) ve Uzun Kısa Süreli Bellek (Long Short Term Memory - LSTM) adı verilen özel Reccurent Sinir Ağı (Recurrent Neural Network - RNN) olmak üzere üç Derin Öğrenme algoritması kullanarak MNIST, FASHION-MNIST, CIFAR-10 ve ARAPÇA veri setleri aracılığıyla el yazı karaktere tanıma ve resim sınıflandırma problemlere uygulandıktan sonra, ilk aşamada modellerimizin sonuçları karşılaştırılmıştır. Kıyaslamadan el yazı karakter ve resim sınıflandırmada, CNN modeli en iyi olduğunu kaydedilmiştir (Salouhou, 2019). Kesim ve arkadaşları tarafından çalışmada, farklı sekans modellerinin çok kipli duygu tanıma performansları karşılaştırıldı. Ses ve görüntüler önce çok katmanlı CNN modelleriyle işlendi ve ardından bu çıktılar çeşitli sekans modellerine iletilerek değerlendirildi. GRU, Transformer, LSTM ve Maksimum Havuzlama gibi modeller çalışmada kullanılmıştır. Deneylerde çok kipli CREMA-D veri seti kullanılmış ve CREMA-D veri seti üzerinde yapılan karşılaştırmada, en yüksek F1 skoru 0.640 ile GRU bazlı mimari tarafından elde edilirken, kesinlik metriği açısından en iyi sonuç 0.699 ile LSTM bazlı mimari olarak belirlenmiştir (Kesim vd., 2023).

Görüldüğü üzere CNN (Convolutional Neural Networks), LSTM (Long Short-Term Memory) ve GRU (Gated Recurrent Unit) gibi derin öğrenme yöntemleri, çeşitli alanlarda başarıyla uygulanmıştır. Ancak, e-ticaret alanında bu yöntemlerin kullanımına yönelik yapılmış özgün bir çalışma henüz bulunmamaktadır. E-ticaret, günümüzde hızla büyüyen ve giderek daha fazla insanın tercih ettiği bir alandır.

Ancak, müşteri deneyimini geliştirmek ve satışları artırmak için duygu analizi gibi yöntemlerin kullanımı oldukça sınırlıdır. E-ticaret platformları, milyonlarca kullanıcının etkileşime girdiği karmaşık bir ortam sunar. Bu platformlardaki ürün incelemeleri, sosyal medya etkileşimleri ve müşteri geri bildirimleri, müşteri duygularının anlaşılması için zengin bir veri kaynağı oluşturur. CNN, LSTM ve GRU gibi derin öğrenme modelleri, bu verileri analiz etmek ve müşteri duygularını belirlemek için potansiyel olarak çok değerli araçlar olabilir. E-ticarette duygu analizine yönelik bir araştırma eksikliği, bu alanda derin öğrenme yöntemlerinin potansiyelini henüz tam olarak keşfedemediğimizi göstermektedir. Bu alandaki bir çalışma, e-ticaret şirketlerine müşteri memnuniyetini artırma, ürün ve hizmet kalitesini iyileştirme ve pazarlama stratejilerini optimize etme konusunda yeni bir bakış açısı sunabilir. Bu nedenle, e-ticaret alanında duygu analizi üzerine yapılmış özgün bir araştırmanın önemi oldukça büyüktür. Mevcut literatür ve bu gerekçelerle b çalışmada e-ticaret sitelerinde belli ürün grupları için yapılan yorumlar üzerinde duygu analizi çalışması yapılıp kullanılan deri öğrenme ve doğal dil işleme tekniklerinin Türkçe metinler üzerindeki etkinliğinin incelenmesi amaçlanmıştır.

2.5. Çift Yönlü Kodlayıcı Gösterimleri (Bidirectional Encoder Representations from Transformers)

BERT (Bidirectional Encoder Representations from Transformers) Google firmasında çalışan Jacob Devlin ve ekip arkadaşları tarafından 2019'da geliştirilen bir doğal dil işleme (NLP) algoritmasıdır. BERT metin verilerini işlerken, bağlamlarını anlayabilmektedir. Bir kelimenin anlamının, cümle içindeki diğer kelimelerle etkileşimini dikkate alarak belirlenmesini cümleleri hem soldan hem de sağdan okumasıyla sağlayabilmektedir (Devlin vd., 2019). BERT dil modelinin çalışma yapısı şekil 13'te gösterilmiştir.



Şekil 13. BERT Çalışma Yapısı

Kaynak: Devlin, J., Chang, M.-W., Lee, K., ve Toutanova, K. (2019, 2-7 Haziran). *Bert: Pre-training of deep bidirectional transformers for language understanding* [Bildiri Sunumu]. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis. https://doi.org/10.18653/v1/N19-1423

BERT dil modeli, Masked Language Modeling (MLM), ve Next Sentence Prediction (NSP) adı verilen iki temel yaklaşımla eğitilerek oluşturulmuştur.

MLM yaklaşımında, maskelenmiş bir kelimenin ne olduğunu çevresindeki kelimelerden aldığı bağlam bilgisine göre maskelenmiş kelimelerin orijinal hallerini tahmin edebilmek amaçlanmaktadır. Bu yaklaşımda modelden esas olarak, maskelenmiş kelimeleri tahmin etmesiyle genel bir dil anlayışı geliştirmesi beklenmektedir. NSP yaklaşımı, iki ardışık cümle çiftini ele alarak, bu cümle çiftinin birinin ikinci cümlesinin, birinci cümleyi takip edip etmediğini tahmin etmeye, cümleler arasında ilişkiyi anlamaya odaklıdır. Çift yönlü analiz yeteneği ile BERT daha etkili öğrenim süreci sağlamaktadır.

BERT ile yapılan çalışmalar, ilk yıllardan itibaren dünyada en çok kullanılan diller üzerine yaygınlaşmıştır ancak ilerleyen zamanlarda birçok dil eklenmesiyle BERT modeli geniş bir dil yelpazesinde etkinlik göstermiştir. Türkçe diline yönelik geliştirilen BERT modeli, *BERTurk* olarak adlandırılmıştır. Bu çalışmada BERTurk modeli kullanılmıştır.

Veri ön işleme adımları modelin doğruluğunu etkileyen önemli faktörlerden biridir. BERTurk modeli önceden büyük veriler üzerine eğitilmiş bir model olmasıyla, bu çalışmada oluşturulan modellere girdi olarak verilen veriler için bir birim ayrıştırıcı (tokenizer) olarak dahil edilerek modellerin doğru sonuç verme etkinliğini arttırması, veri ön işleme adımlarının daha kolaylaştırılması sağlamak amaçlanmıştır.

3. UYGULAMA YÖNTEMİ

Üçüncü bölüm, uygulama yöntemini oluşturmaktadır. Bu bölümde çalışmanın amacı ve önemi, çalışmanın sınırları, çalışmanın yöntemi ve deseni, bu çalışmadaki veri toplama tekniği ve verilerin analizine yer verilmiştir.

Bu çalışmada e-ticaret sitelerinde belli ürün grupları için yapılan yorumlar üzerinde duygu analizi çalışması yapılıp kullanılan deri öğrenme ve doğal dil işleme tekniklerinin Türkçe metinler üzerindeki etkinliğinin incelenmesi amaçlanmıştır. Oluşturulan modeller, metin verilerinin duygusal içeriğini sınıflandırmak amacıyla tasarlanmıştır.

E-ticaret siteleri günümüzde tüketicilerin ürün tercihlerini belirlemede önemli bir rol oynamaktadır. Bu çalışmada, e-ticaret sitelerindeki belirli ürün grupları için yapılan yorumlar üzerinde duygu analizi çalışması gerçekleştirilmiştir. Amacımız, bu yorumların derinlemesine incelenerek, kullanılan derin öğrenme ve doğal dil işleme tekniklerinin Türkçe metinler üzerindeki etkinliğini değerlendirmektir. Çünkü tüketicilerin ürün hakkındaki duygusal tepkilerini anlamak, pazarlama stratejilerinin oluşturulmasında ve ürün geliştirme süreçlerinde önemli bir rol oynamaktadır. Bu çalışma, e-ticaret platformlarında yapılan yorumların analiz edilmesiyle, Türkçe metinler üzerinde duygu analizi alanında yeni görüşler sunmayı ve bu alandaki tekniklerin geliştirilmesine katkıda bulunmayı amaçlamaktadır.

Çalışmanın yöntemi olarak “LSTM" olarak bilinen, uzun süreli bağımlılıkları öğrenebilen özel bir tür tekrarlayan sinir ağı kullanılarak bir model oluşturulmuştur. Ardından ikinci model olarak “GRU” olarak bilinen, zamanla değişen verileri işlemek için kullanılabilen sinir ağı kullanılarak model oluşturulmuştur. İkinci modelin oluşturulmasından sonra, üçüncü modelde “CNN” olarak bilinen ve desen tanıma ile metin veya görsellerin desenini oluşturma ile işlem gücünü azaltma özelliğiyle kullanılabilen sinir ağı kullanılmıştır.

Bu modellerin oluşturulmasıyla elde edilen sonuçlara istinaden, bu sinir ağlarının birleşimi ile hibrit modeller oluşturulmuştur. Çalışmada, hibrit modelin verimliliğine odaklanılmıştır.

Çalışmada kullanılan veri seti Kaggle veri tabanından temin edilmiştir. Python kodlama dili veri ön hazırlığı, verilerin analizi için grafiklerin oluşturulması, verilerin işlemesi ve model eğitimi için kullanılmıştır.

Modeller, vektör girdisi oluşturmak amacıyla gömme (embedding) katmanı ile başlamaktadır. Bu katman, verilen tek boyutlu sıralı metin girdi dizisini, belirli bir boyutlu vektörlerle temsil eden matematiksel bir dönüşüm gerçekleştirmiştir. CNN (evrişimli sinir ağları) ve hibrit modellerde, sıralı veriler üzerinde çalışmak için Conv1D (1 boyutlu evrişimli) katmanı kullanılmaktadır. Conv1D katmanları, belirli bir boyutlu pencereyi veri dizisi boyunca kaydırarak özellik haritaları oluşturmuştur. Ardından sonuç boyutunu azaltmak için 1 Boyutlu Maksimum Havuzlama (MacPool1D) katmanı kullanılmıştır. Bu işlemde, modelin daha genel ve özgün özellikler öğrenmesi amaçlanmıştır.

LSTM katmanı, uzun süreli bağımlılıkları öğrenmek için kullanılan özel kapılara sahip bir tür tekrarlayan sinir ağır (RNN) katmanıdır. Hibrit modellerde LSTM katmanı, özellikle uzun vadeli bağımlılıkları yakalamak için tercih edilmiştir.

GRU katmanı ise, LSTM'e göre daha hızlı ve daha az hesaplama gerektiren bir yapıya sahip olmasıyla tercih edilmiştir.

3.1. Verilerin Toplanması ve Düzenlenmesi

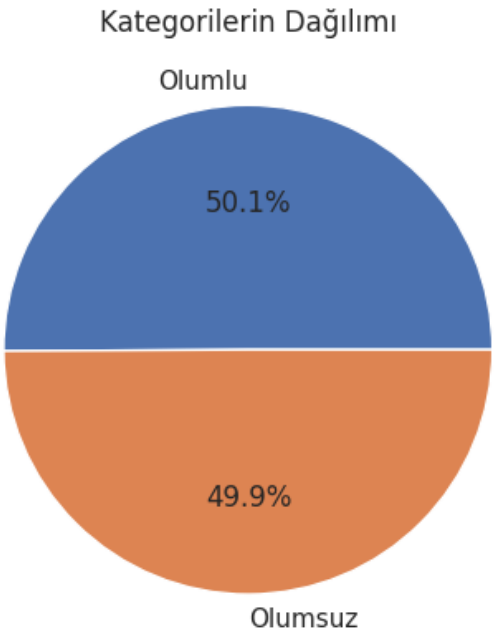
Büyük veri kümelerinin analizi işletmeler ve diğer kuruluşlar için önem taşımaktadır. İnternetin yaygın kullanımıyla birlikte veri hacmi oldukça büyümüştür. Veriler analiz edilirken anlamsız bir verinin olması, performans ve doğruluk açısından olumsuz etki doğurmaktadır. Bu sebeple veriler ön işlemeden geçmeli ve analize uygun hale getirilmelidir. Veri setinin doğru seçilmesi, modelin doğruluğu üzerinde büyük önem taşımaktadır.

E-ticaret sitelerinde belli ürün grupları için yapılan yorumlar üzerinde duygu analizi çalışması ile derin öğrenme ve doğal dil işleme tekniklerinin Türkçe metinler üzerindeki etkinliğinin incelenmesi amacıyla yapılan bu çalışmada veriler Kaggle veri tabanından temin edilmiştir. Temin edilen iki veri setinin birleştirilmesiyle 16 bin satır

932 satırlık metin verisi ve olumlu, olumsuz olarak iki sınıflık ürün yorumları üzerine çalışılmıştır. Her bir model, aynı veri seti üzerinde denenmiş ve sonuçlar karşılaştırılmıştır.

Çalışmada Python kodlama dili veri ön hazırlığı, verilerin analizi için grafiklerin oluşturulması, verilerin işlemesi ve model eğitimi için kullanılmıştır. Pandas kütüphanesi veri setini veri çerçevesi olarak yükleyerek işlemek adına indirilmiştir. CSV formatı ile Python dilinde veriler .csv dosyasından okunmuştur. Veri seti Pandas kütüphanesinin DataFrame "pandas.core.frame.DataFrame" sınıfı kullanılarak veri bilgisi 16.978 satır ve iki sütun olarak görülmüştür. Veri setine ait kategorik dağılımlar oluşturulmuş ve olumlu ve olumsuz olarak 2 ayrı kategori saptanmıştır.

Özelleştirilmiş bu veri setinde, olumlu veri satır sayısı 8.504, olumsuz veri satır sayısı 8 bin 474'tür. Olumlu yorumlar 0, olumsuz yorumlar 1 olarak etiketlendirilmiştir. Verilerin kategorik dağılımlarına ilişkin grafik görseli aşağıda Şekil 14'teki gibidir.



Şekil 14. Veri Setine Ait Kategorik Dağılımlar

Kaynak: Yazar tarafından oluşturulmuştur.

Kategori dağılımına bakıldığında yaklaşık olarak iki eşit kategori dağılımı görülmektedir. Aşağıdaki Şekil 15'te veri setine ait ilk 19 veri gösterilmiştir.

	text	category
1	Ürün gayet güzel boyutları dört kişilik aile için yeterli 7 günde teslim ettiler	Olumlu
2	Ürünü kullandım gayet memnunum	Olumlu
3	Hızlı kargo	Olumlu
4	8 yıl önce lg 42 lm serisi almıştım ve tv den gayet memnundum ancak ekran kırılmas	Olumsuz
5	ürün fena değil ancak gövdenin plastik olması beni tedirgin etti. en azından ürün bilgilerine bu belirtilmeliymiş	Olumlu
6	çok kötü hiç bir işe yaramıyor	Olumsuz
7	Çok geç ısıtıyor ekmeği kızartmıyor yani düşük performans ??????????	Olumsuz
8	good	Olumlu
9	gayet güzel sadece kare tepsi kenarlara sürterek giriyor tam ayarını tutturmak zorluyor :) 2 tepsiyi aynı anda pişirmesi de cok iyi, yorumlardan acaba cizik gelir mi diye korkmustum ancak hiç bir sorunu olmadan geldi.	Olumlu
10	Fiyatına göre iyi gibi ama ince halıları kaldırıyor yerden gücünden dolayı. Süpürge ol	Olumsuz
11	İşi yapan Antalya Üçay mühendislik işi alana kadar sonrası adam yerine koymuyorla	Olumsuz
12	Çok kötü. Hiç şarz yok. Kalitesiz. Kullanamadık	Olumsuz
13	Ürün ses deneyimi baslar tizler her konuda güzel.300 liraya en güzel 5+1 lerden. Kar	Olumsuz
14	Kesinlikle rezalet bu kadar rezillik olmaz. Ikinci el ürün geldi. Herşeyiylr ikinci el olduğu belli. Kablo orjinal değil ayna kısmı parmak izi dolu, gövdede yoğun çizikler var çok ayıp gerçekten...	Olumsuz
15	Küçük ve ince kılları bile çok zor alıyor. Çin malı, seyyarda satılan traş makineleri bil	Olumsuz
16	Basketbol antrenmanlarımda sürekli olarak kulağımda gevşedi, tam anlamıyla rahat	Olumsuz
17	Tasarımı güzel ve kullanışlı. Benzer modellere göre oldukça hafif ve çekiş gücü yüks	Olumlu
18	Hocam canaksiz diyorsunuz wi-fi internet olmadan nasıl izliyoruz anlamadim	Olumsuz
19		

Şekil 15. Veri Setinin İlk 19 Verisi

Kaynak: Yazar tarafından oluşturulmuştur.

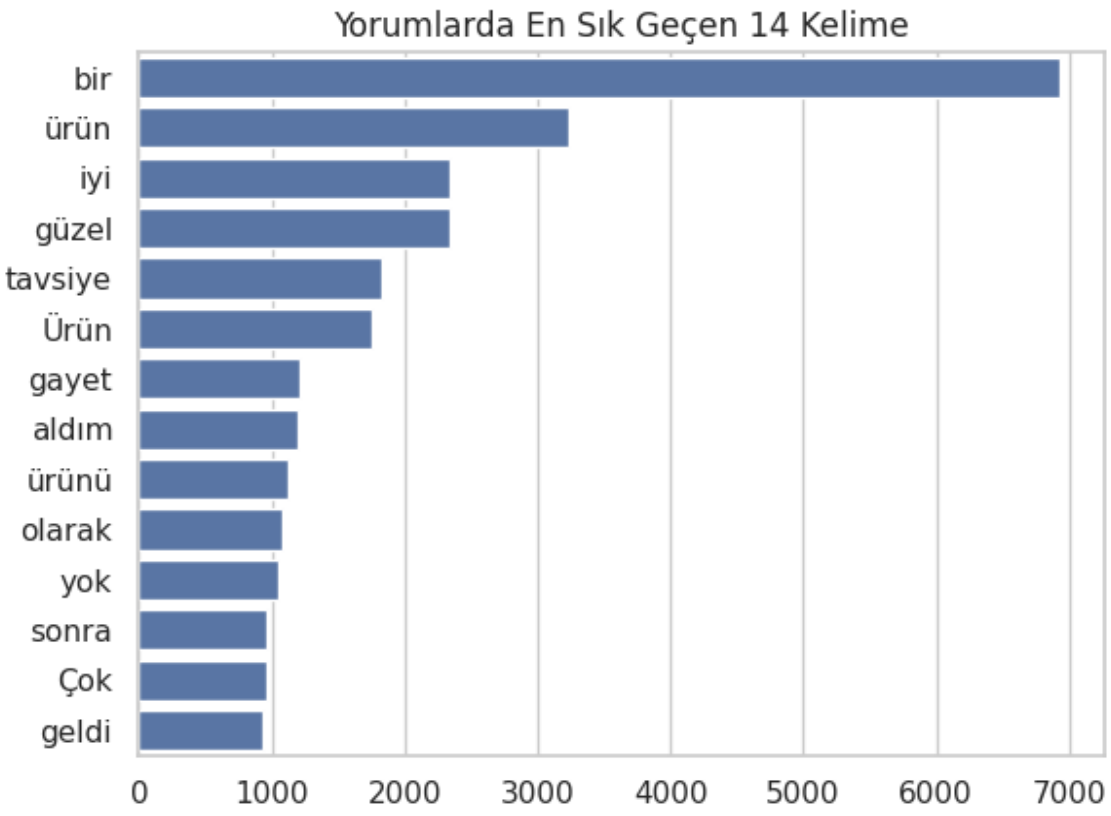
En sık geçen kelimeler ve ikili kelime gruplarını incelemek, veri setindeki belli bir deseni anlamaya yardımcı olabilmektedir. Bu sebeple en sık geçen kelimeler veri setinde incelenmiştir. En sık geçen 10 kelime aşağıdaki gibi Tablo 3'te gösterilmiştir;

Tablo 3. Veri Setinde En Sık Geçen Kelimeler

Kelime	Kelimenin kullanılma sayısı
“Bir”	6.924
“Çok”	6.106
“Ve”	5.566
“Ürün”	3.228
“Bu”	2.786
“Ama”	2.646
“İçin”	2.552
“İyi”	2.338
“Güzel”	2.334
“Daha”	2.142

Kaynak: Yazar tarafından oluşturulmuştur.

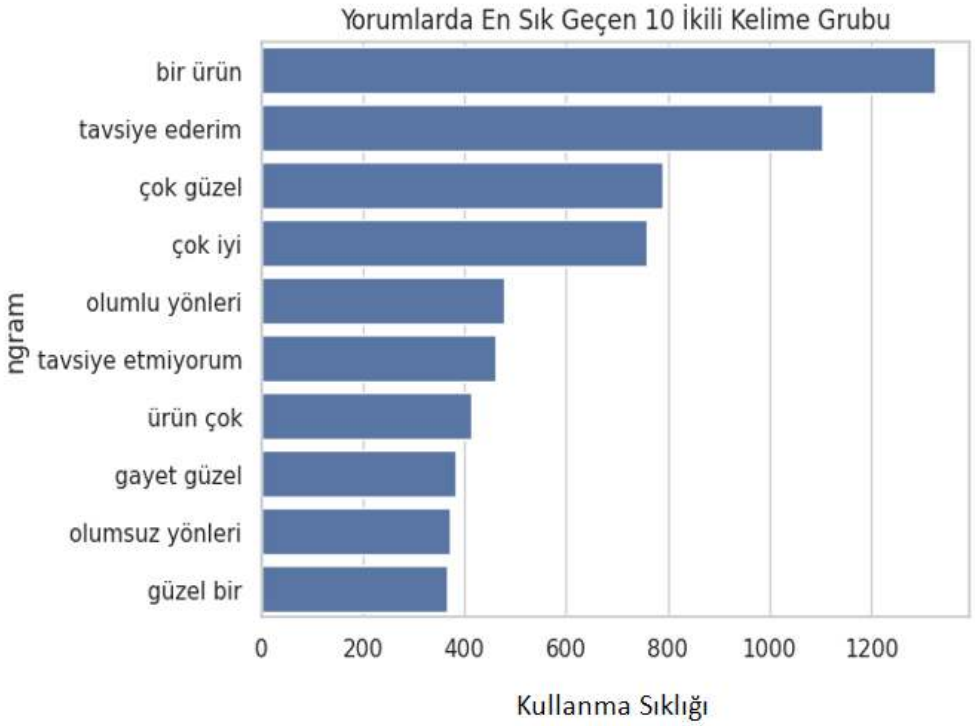
“Ve”, “de” gibi kelimeler Türkçe ’de çok sık kullanıldığı için bu kelimeler dışındaki “güzel”, “iyi” gibi kelimeler duygu anlamı içerdiğinden analiz için önem taşıyabilmektedir. 10 kelimelik analizden sonra en sık geçen 14 kelime için frekans grafiği Şekil 16’deki gibidir.



Şekil 16. Kelime Frekansı Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

Tek kelime üzerine yapılan frekans analizi sonrası en çok geçen 10 ikili kelime grupları Şekil 17'daki gibidir.



Şekil 17. İkili Kelime Grubu Frekans Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

Görüldüğü üzere, "çok iyi", "gayet güzel", "çok güzel", "tavsiye ederim" gibi kelimeler olumlu anlamlar taşımaktadır. "Tavsiye etmiyorum" gibi ifadeler ise olumsuz anlam taşımaktadır.

Bütün yorumlar üzerine Word Cloud oluşturulup ve görselleştirilmiştir, ardından olumlu veri seti ve olumsuz veri seti için ayrı WordCloud oluşturulmuştur. Veri setindeki her bir metin alınmış ve metinlerin içerdiği kelimeleri işlenmiştir. Öncelikle, her bir metindeki kelimeler tokenize edilmiş ve stop word listesinde bulunmayan kelimeler seçilmiştir. Daha sonra, seçilen kelimelerin dil bilgisi kurallarına göre kök haline getirilmesi için üzerinde lematizasyon (lemmatization) işlemi uygulanmıştır. Kök haline gelmiş kelimelerden oluşan bir metin koleksiyonu (corpus) oluşturulmuştur. Bu metin koleksiyonu, WordCloud kütüphanesine beslenmiş ve görselleştirilmiştir. WordCloud, metinlerde en sık geçen kelimelerin görsel bir temsilini sağlamaktadır.

Kelimelerin görsel olarak büyüklüğü, o kelimenin metinlerde ne kadar sık geçtiğini temsil etmektedir. Veri setindeki bütün yorumlar için oluşturulan WordCloud görsel aşağıda Şekil 18'de gösterilmiştir.

Şekil 18. Bütün Yorumlar için WordCloud Görseli

Kaynak: Yazar tarafından oluşturulmuştur.

Bütün yorumlar için WordCloud görseline bakıldığında "tavsiye ederim", "güzel", "kötü", "iyi" gibi duygu ifade eden kelimelerin diğer kelimelere istinaden daha fazla kullanıldığı görseldeki boyutlarına göre anlaşılabilmektedir. Bütün yorumlar için dikkate alındığında ise genel anlamda "tavsiye ederim" cümlesi neredeyse en sık kullanılan olarak, olumlu olduğunu ifade ettiği görüntülenmektedir. Olumlu yorumlar üzerine yapılan WordCloud görseli aşağıda Şekil 19, olumsuz yorumlar için oluşturulan WordCloud görseli Şekil 20'de gösterildiği gibidir.

Şekil 19. Olumlu Yorumlar için WordCloud Görseli

Kaynak: Yazar tarafından oluşturulmuştur.

Şekil 20. Olumsuz Yorumlar için WordCloud Görseli

Kaynak: Yazar tarafından oluşturulmuştur.

3.2. Deneysel Çalışma Ortamı

Duygu analizi veri setinde yer alan cümlelerin olumlu ya da olumsuz olduğunu belirler. Önerilen modeller, Keras kütüphanesi kullanılarak Python'da hazırlanmıştır. Python, Guido Van Rossom tarafından oluşturulan bir nesne yönelimli ve yüksek seviye programlama dili, çok sayıda işletim sistemi ile uyumlu ve kolay sözdizimi ile birçok kişi tarafından tercih edilebilmektedir.

Önerilen modelin arka ucu olarak Google tarafından geliştirilen ve yaygın olarak kullanılan açık kaynaklı bir makine öğrenimi platformu olan TensorFlow kullanılmıştır. Veri setinde her oluşturulan modelde BERT simgeleştirme (tokenizer) kullanılarak, LSTM, GRU, CNN modelleri oluşturulmuş ve sonuçlar gösterilmiştir. Bu sonuçlara istinaden hibrit modeller üzerine odaklanılmış ve BERT simgeleştirme (tokenizer) kullanılmış veri girdisi ile LSTM ve GRU, LSTM ve CNN, GRU ve CNN hibrit modelleri oluşturularak sonuçlar gösterilmiştir.

3.3. Analiz Yöntemleri

Bu çalışmada modellerin başarısını ölçmek için doğruluk (accuracy), duyarlılık (recall), kesinlik (precision), F1-Skor (F1-Score) ve karmaşıklık matrisi (confusion matrix) gibi performans metrikleri kullanılmıştır. Metriklere ait açıklamaları aşağıda yer almaktadır.

Doğruluk: bir sınıflandırma modelinin doğru tahmin oranını gösteren performans metriğidir. Doğruluk, doğru tahmin edilen örneklerin toplam örnek sayısına oranı olarak hesaplanmaktadır.

Duyarlılık: bir sınıflandırma modelinin bir sınıfı ne kadar başarılı şekilde tanımladığını ölçen performans metriğidir. Duyarlılık, gerçek pozitif (actual positive) olan örneklerin doğru şekilde tanımlanmış pozitif (predicted positive) örneklerine oranını ifade eder.

Kesinlik: bir sınıflandırma modelinin pozitif olarak tahmin ettiği örneklerin gerçekte ne kadarının pozitif olduğunu gösteren bir performans metriğidir. Kesinlik, doğru pozitif (true positive) olan örneklerin toplam tahmin edilen pozitif (predicted positive) örneklerine oranını ifade eder.

F1-Skoru: kesinlik ve duyarlılığı dengeli bir şekilde birleştiren bir performans metriğidir. F1 skoru, kesinlik ve duyarlılık arasındaki harmonik ortalama olarak hesaplanır. F1 skoru, modelin hem kesinlik hem de duyarlılık açısından iyi performans gösterip göstermediğini anlamak için kullanılır.

Karmaşıklık Matrisi: modelin gerçek ve tahmin edilen sınıfları ne kadar doğru veya yanlış tahmin ettiğini görsel bir şekilde göstermektedir. Karmaşıklık matrisi dört ana bileşenden oluşmaktadır. Ana bileşenlerin detaylarına aşağıda yer verilmiştir.

Gerçek Pozitifler (True Positives); gerçek pozitif sınıfların doğru bir şekilde tahmin edilmiş olduğu sonuçtur.

Yanlış Pozitifler (False Positives); Negatif sınıfların pozitif olarak yanlış tahmin edildiği sonuçtur.

Gerçek Negatifler (True Negatives); Gerçek negatif sınıfların doğru tahmin edildiği sonuçtur.

Yanlış Negatifler (False Negatives); Pozitif sınıfların negatif olarak yanlış tahmin edildiği sonuçtur.

4. ÇALIŞMA BULGULARI

Dördüncü bölümde çalışmanın bulguları verilerin model eğitimi öncesi temizlenerek ve ardından LTSM, GRU, CNN, LSTM ve GRU, LSTM ve CNN, GRU ve CNN modelleri eğitilmesi ile veri setindeki duygu analizinin etkinliğine ait bulguları sunulmaktadır.

4.1. BERT Modeli

Bu çalışmada oluşturulan modellerde simgeleştirme (tokenizer) olarak BERTurk modeli kullanılmıştır. Kullanılan modeli kullanılmıştır. Metin verileri, modelin veriyi işleyebilmesi için modele girdi olarak verilecek sayısal temsillere dönüştürülmelidir. BERT simgeleştiricisi (tokenizer), her kelime veya alt kelimeyi benzersiz bir sayısal değere dönüştürür. Bu sayede, model metin verileri üzerinde işlem yapabilir. Metin verilerinin model tarafından doğru işlenmesi için, belirli bir biçimde olması gerekmektedir. BertTurk modeline ait simgeleştirici (tokenizer) metin verilerini modelin işleyebileceği şekilde düzenlemektedir. Simgeleştirici (tokenizer), Türkçe metin verilerini parçalara ayırarak ve her bir parça bir birimi (token) temsil ederek modele uygun bir girdiye, sayısal vektörler temsillerine dönüştürülmüştür.

Çalışmanın ilk aşamasında gerekli kütüphaneler ve veri seti yüklenmiştir. Sonrasında verinin yapısı incelenmiş ve metinler birimlere ayrılıp modele girdi olarak verilmiştir. Bu süreç, modelin duygusal eğilimi öğrenmesi ve yeni ifadeleri sınıflandırmasını sağlamaktadır. Birim (token) BERT modelinin çağırılması aşağıda Şekil 21'deki gibidir.

```
[ ]  tokenizer = BertTokenizer.from_pretrained("dbmdz/bert-base-turkish-cased")
     #model = BertModel.from_pretrained("dbmdz/bert-base-turkish-cased")
```

Şekil 21. BERTurk Modelinin Çağırılması

Kaynak: Yazar tarafından oluşturulmuştur.

BERTurk modelinin çağırılması işleminden sonra aşağıdaki kod bloğunda Pytorch tensörleri NumPy dizilerine dönüştürerek ve metin veri kümesini kategorik hedef değişkenini ikili sınıf dizisine dönüştürerek bir veri hazırlama süreci gerçekleştirilmiştir. Şekil 22 ve 23'te bu sürecin yapıldığı kod blokları gösterilmiştir.

```
[ ]  encoding = tokenizer.batch_encode_plus(
         veri["text"],
         max_length=100,
         padding="max_length",
         truncation=True,
         return_tensors="pt",
         add_special_tokens=True)

     input_ids = encoding["input_ids"]
     attention_mask = encoding["attention_mask"]
```

Şekil 22. Verilerin Birimlere Ayrıştırılması

Kaynak: Yazar tarafından oluşturulmuştur.

```
[ ]  tensor = input_ids
     X = tensor.detach().cpu().numpy()
     y = pd.get_dummies(veri["category"]).values.astype(int)
```

Şekil 23. Birimlerin Modele Sunulması

Kaynak: Yazar tarafından oluşturulmuştur.

Modellerde kullanılan metin veri seti 0,3 oranında test verisi, 0,5 oranda ise eğitim verisi olarak ikiye ayrılmıştır. Aşağıda eğitim ve doğrulama veri oranların verilmesi ve veri biçimleri işlemine ait kod satırları şekil 24 ve 25'te gösterilmiştir.

```
[ ] X_train, X_val_test, y_train, y_val_test = train_test_split(X, y, test_size=0.30, random_state=40)
    X_val, X_test, y_val, y_test = train_test_split(X_val_test, y_val_test, test_size=0.50, random_state=40)
```

Şekil 24. Eğitim ve Doğrulama Verilerinin Oranı Tanımlanması

Kaynak: Yazar tarafından oluşturulmuştur.

```
x_train shape (11884, 100)
x_test shape (2547, 100)
x_val shape (2547, 100)
y_train shape (11884, 2)
y_test shape (2547, 2)
y_val shape (2547, 2)
```

Şekil 25. Eğitim ve Doğrulama Test Verisinin Biçimi

Kaynak: Yazar tarafından oluşturulmuştur.

4.2. LTSM Modeli

Model eğitimi, Keras kütüphanesi kullanılarak gerçekleştirilmiştir. Modelin mimarisi, ardışık bir sinir ağı modeli olarak tasarlanmıştır. Bu bölümde eğitim ve doğrulama (validation) veri setleri üzerinde yapılan iterasyonlar (epoch) ile kayıp (loss) ve doğruluk (accuracy), duyarlılık (recall), kesinlik (precision), F1 skoru (F1 score) metrikleri aktarılmıştır. Aşağıda, kullanılan katmanlar ve parametreler ayrıntılı olarak açıklanmıştır. Şekil 26'da LSTM modeli oluşturulmasına ait kod satırları ve modelin katmanları gösterilmiştir.

```python
# LSTM modeli oluşturuluyor.
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(4000, 128, input_length=100),
    tf.keras.layers.LSTM(64),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(loss='binary_crossentropy', metrics=["accuracy"], optimizer=Adam())
print(model.summary())
```

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 100, 128)          512000

 lstm_2 (LSTM)               (None, 64)                49408

 dense_6 (Dense)             (None, 64)                4160

 dense_7 (Dense)             (None, 2)                 130

=================================================================
Total params: 565698 (2.16 MB)
Trainable params: 565698 (2.16 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Şekil 26. LSTM Modeli Oluşturulması ve Katmanları

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin giriş katmanı gömme katmanıdır, kelime gömme matrisini oluşturmaktadır. Giriş metin verisini belirli bir boyutta gömmek için kullanılmıştır. Bu katmanda her kelime 128 boyutunda bir vektörle temsil edilmektedir. Giriş uzunluğu 100 birim, çıkış boyutu "None, 100, 128" olarak belirtilmiştir. Çift yönlü (bidirectional) LSTM katmanı, LSTM hücrelerinden oluşan çift yönlü bir katmandır. Giriş boyutu gömme katmanının çıkış boyutu olarak "None, 100, 128" olmaktadır. LSTM katmanından sonra, altıncı tam bağlı katman gelmektedir. Modelin gizli katmanı olarak görev yapmaktadır. Girdi boyutu LSTM katmanının çıktı boyutu olan 64 birim olmaktadır. ReLU aktivasyon fonksiyonu ile çıktı üreterek yedinci tam bağlı katmana girdi iletmektedir. Yedinci tam bağlı katman, altıncı tam bağlı katmanın çıkış verilerini alarak 64 girdi boyutuna sahiptir. Softmax aktivasyonu kullanılarak duygusal sınıflandırma yapılmıştır.

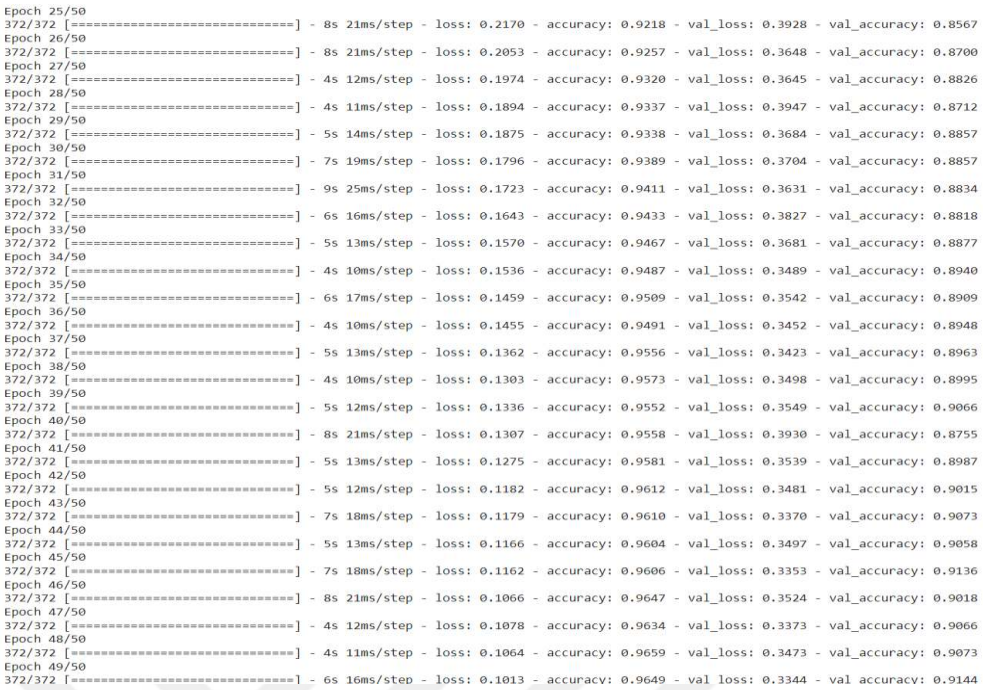
Modelin derlenmesi sırasında, kayıp fonksiyonu olarak kategorik çapraz entropi (categorical crossentropy) kullanılmıştır. Doğruluk (accuracy) metriği kullanılarak modelin performansı değerlendirilmiştir. Optimizasyon algoritması olarak "Adam" optimizasyon algoritması kullanılmıştır. Oluşturulan LSTM modelinin toplam parametre sayısı 565.698'dir ve tüm parametreler eğitilebilirdir.

Modelin eğitimi, veri setinin 0.3 oranında eğitim için ve 0.7 oranını doğrulama için ayırmak üzere gerçekleştirilmiştir. Eğitim süreci 50 iterasyon (epoch) boyunca yapılması sınırlandırılmıştır. Eğitim sırasında aşırı uyum (overfitting) durumunu önlemek, için geri çağırma yöntemi olarak, erken durdurma (early stopping) kullanılmıştır. Erken durdurma, kayıp (loss) fonksiyonu oranı belli bir süre sabit bir durumda devam ettiğinde ve iyileşme göstermediğinde model eğitimini olarak durdurmak için kullanılmıştır. İterasyon 50 iterasyon boyunca devam etmiştir. İterasyonlarda incelendiğinde en yüksek doğruluk oranı test verisi için %97, kayıp (loss) değeri ise %9.15 görülmüştür. Aşağıda Şekil 27 'de ilk 24 iterasyon (epoch) Şekil 28'de son 24 iterasyon (epoch) sonuçları, Şekil 29'da ise en yüksek doğruluk sonucu gösterilmiştir.

```
Epoch 1/50
372/372 [==============================] - 47s 120ms/step - loss: 0.6882 - accuracy: 0.5157 - val_loss: 0.6851 - val_accuracy: 0.5077
Epoch 2/50
372/372 [==============================] - 19s 51ms/step - loss: 0.6814 - accuracy: 0.5243 - val_loss: 0.6932 - val_accuracy: 0.5065
Epoch 3/50
372/372 [==============================] - 14s 38ms/step - loss: 0.6911 - accuracy: 0.5163 - val_loss: 0.6710 - val_accuracy: 0.6580
Epoch 4/50
372/372 [==============================] - 13s 35ms/step - loss: 0.6789 - accuracy: 0.5417 - val_loss: 0.6906 - val_accuracy: 0.5092
Epoch 5/50
372/372 [==============================] - 10s 26ms/step - loss: 0.6819 - accuracy: 0.5610 - val_loss: 0.6895 - val_accuracy: 0.5104
Epoch 6/50
372/372 [==============================] - 13s 34ms/step - loss: 0.6848 - accuracy: 0.5239 - val_loss: 0.6860 - val_accuracy: 0.5344
Epoch 7/50
372/372 [==============================] - 11s 30ms/step - loss: 0.6830 - accuracy: 0.5286 - val_loss: 0.6860 - val_accuracy: 0.5253
Epoch 8/50
372/372 [==============================] - 9s 24ms/step - loss: 0.6372 - accuracy: 0.6022 - val_loss: 0.6162 - val_accuracy: 0.6643
Epoch 9/50
372/372 [==============================] - 6s 17ms/step - loss: 0.5926 - accuracy: 0.6805 - val_loss: 0.6240 - val_accuracy: 0.6663
Epoch 10/50
372/372 [==============================] - 6s 17ms/step - loss: 0.5669 - accuracy: 0.6866 - val_loss: 0.6047 - val_accuracy: 0.6745
Epoch 11/50
372/372 [==============================] - 7s 19ms/step - loss: 0.5666 - accuracy: 0.6929 - val_loss: 0.5825 - val_accuracy: 0.6910
Epoch 12/50
372/372 [==============================] - 5s 13ms/step - loss: 0.5674 - accuracy: 0.6982 - val_loss: 0.6099 - val_accuracy: 0.6745
Epoch 13/50
372/372 [==============================] - 5s 14ms/step - loss: 0.5969 - accuracy: 0.6888 - val_loss: 0.5619 - val_accuracy: 0.7287
Epoch 14/50
372/372 [==============================] - 8s 22ms/step - loss: 0.5251 - accuracy: 0.7561 - val_loss: 0.5140 - val_accuracy: 0.7664
Epoch 15/50
372/372 [==============================] - 6s 15ms/step - loss: 0.4444 - accuracy: 0.8094 - val_loss: 0.4708 - val_accuracy: 0.8045
Epoch 16/50
372/372 [==============================] - 8s 21ms/step - loss: 0.3931 - accuracy: 0.8400 - val_loss: 0.4697 - val_accuracy: 0.8068
Epoch 17/50
372/372 [==============================] - 4s 11ms/step - loss: 0.3416 - accuracy: 0.8667 - val_loss: 0.4750 - val_accuracy: 0.8182
Epoch 18/50
372/372 [==============================] - 6s 16ms/step - loss: 0.3130 - accuracy: 0.8812 - val_loss: 0.3966 - val_accuracy: 0.8465
Epoch 19/50
372/372 [==============================] - 6s 15ms/step - loss: 0.2933 - accuracy: 0.8905 - val_loss: 0.3909 - val_accuracy: 0.8508
Epoch 20/50
372/372 [==============================] - 5s 15ms/step - loss: 0.2683 - accuracy: 0.9021 - val_loss: 0.3797 - val_accuracy: 0.8630
Epoch 21/50
372/372 [==============================] - 7s 20ms/step - loss: 0.2531 - accuracy: 0.9079 - val_loss: 0.3677 - val_accuracy: 0.8712
Epoch 22/50
372/372 [==============================] - 5s 15ms/step - loss: 0.2564 - accuracy: 0.9050 - val_loss: 0.3727 - val_accuracy: 0.8673
Epoch 23/50
372/372 [==============================] - 6s 16ms/step - loss: 0.2359 - accuracy: 0.9132 - val_loss: 0.3689 - val_accuracy: 0.8677
Epoch 24/50
372/372 [==============================] - 5s 14ms/step - loss: 0.2289 - accuracy: 0.9168 - val_loss: 0.3612 - val_accuracy: 0.8724
```

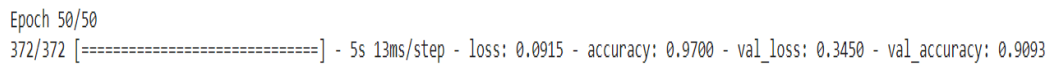
Şekil 27. Modelin İlk 24 Iterasyon (Epoch) Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 28. Modelin Son 24 Iterasyon (Epoch) sonuçları

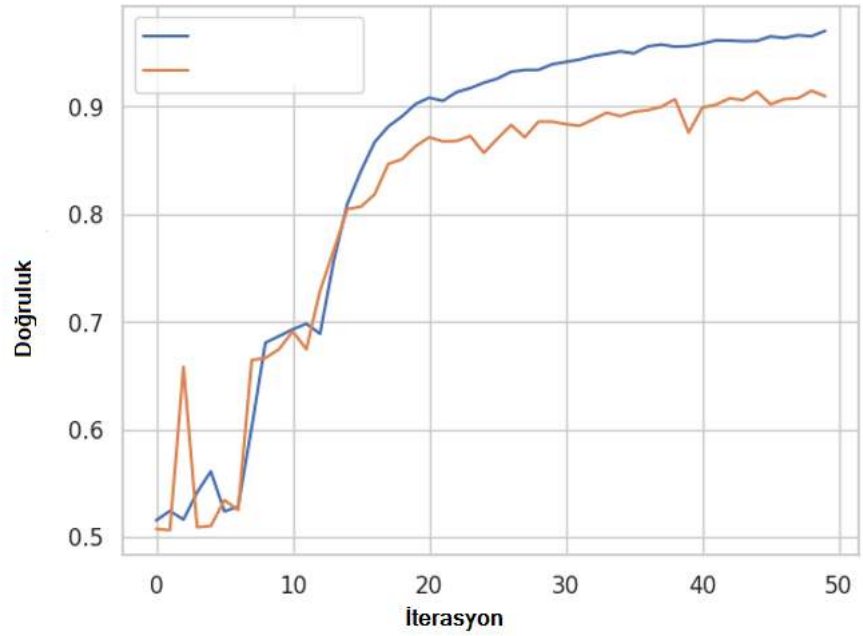
Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 29. Modelin En yüksek Doğruluk Sonucu

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin doğruluğunu göstermek için oluşturulan model doğruluğu grafiği Şekil 30'deki gibidir.



Şekil 30. LSTM Modelinin Iterasyon Doğruluğu

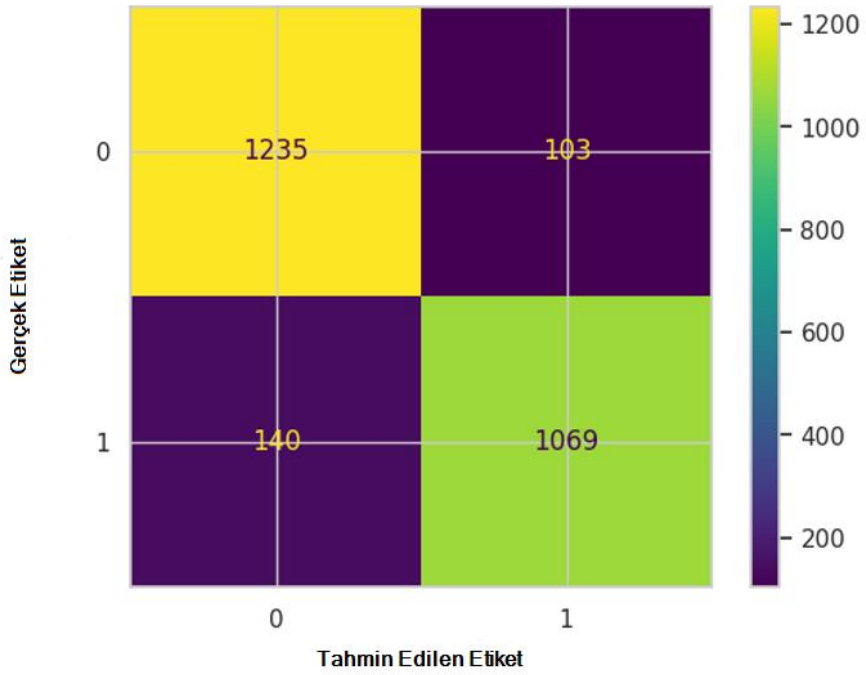
Kaynak: Yazar tarafından oluşturulmuştur.

Doğruluk değerleri haricinde modelin kesinlik (precision), duyarlılık (recall), F1 skoru (F1 Score), karmaşıklık matrisi (confusion matrix) sonuçları da verilmiştir. Modele ait bütün metrik sonuçlar ve karmaşıklık matrisi aşağıda Tablo 4'te gösterilmiştir. Modele ait karmaşıklık matrisi (confusion matrix) grafiği Şekil 31'te gösterilmiştir.

Tablo 4. LSTM Modeli Sonuçları

Sütun1	Kesinlik (precision)	Duyarlılık (recall)	F1 Skoru (F1 Score)	Gerçek Negatif	Gerçek Pozitif	Yanlış Negatif	Yanlış Pozitif
Pozitif	0.90	0.92	0.91	1069	1235	140	103
Negatif	0.91	0.88	0.90				
Doğruluk	0.91						

Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 31. Modelin Karmaşıklık Matrisi

Kaynak: Yazar tarafından oluşturulmuştur.

Kurulan model sonucunda elde edilen sonuçlar incelendiğinde, kesinlik pozitif olarak etiketlenen sınıf 0 için için %90 olarak hesaplanmıştır. Kesinlik, negatif etiketlenen 1 sınıfı için %91 olarak hesaplanmıştır. Modelin duyarlılık sonuçları pozitif sınıf için %92, negatif için %88 hesaplanmıştır. Modelin F1 skoru pozitif sınıf için %91, negatif sınıf için %90 olarak hesaplanmıştır.

4.3. GRU Modeli

Oluşturulan GRU modeli gömme katmanı ile başlamaktadır. Gömme katmanında giriş dizisi belli bir boyutta vektör uzayına dönüştürülür. Bu katmandan çıkan veri, kelimelerin dağıtımsal temsillerini oluşturmaktadır. Çıkış boyutu bir sonraki katman olan çift yönlü (bidirectional) GRU katmanının giriş birimi 64'tür.

Bu katman, giriş dizisini işlerken her bir zaman adımında hem ileri hem de geri yönlü GRU hücreleri kullanır. Zaman serisi verilerindeki bağlamsal bilgiyi daha iyi anlamaya katkı sağlamaktadır.

Bir sonraki katman olan tam bağlı katmanın giriş boyu 64 birimdir. Bu katman gizli katmandan gelen özellikleri daha yüksek seviyeli özelliklere dönüştürmektedir. ReLU aktivasyon fonksiyonu kullanılmıştır. Bir sonraki tam bağlı çıkış katmanının birim sayısı 2 olmaktadır. Softmax aktivasyon fonksiyonu kullanılarak sınıf olasıkları elde edilmesi amaçlanmıştır.

Model, ikili çapraz entropi (binary crossentropy) kayıp fonksiyonu kullanılarak ve Adam optimizer ile derlenmiştir. Eğitim sırasında doğruluk (accuracy) metriği izlenilmiştir. Modelin oluşturulması ve özet halinin çıkarılması için yazılan kod satırları aşağıda şekil 32'te gösterilmiştir.

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(4000, 128, input_length=100),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(64)),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(loss='binary_crossentropy', metrics="accuracy", optimizer=Adam())
print(model.summary())
```

```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_4 (Embedding)     (None, 100, 128)          512000

 bidirectional_1 (Bidirecti  (None, 128)               74496
 onal)

 dense_8 (Dense)             (None, 64)                8256

 dense_9 (Dense)             (None, 2)                 130

=================================================================
Total params: 594882 (2.27 MB)
Trainable params: 594882 (2.27 MB)
Non-trainable params: 0 (0.00 Byte)
```

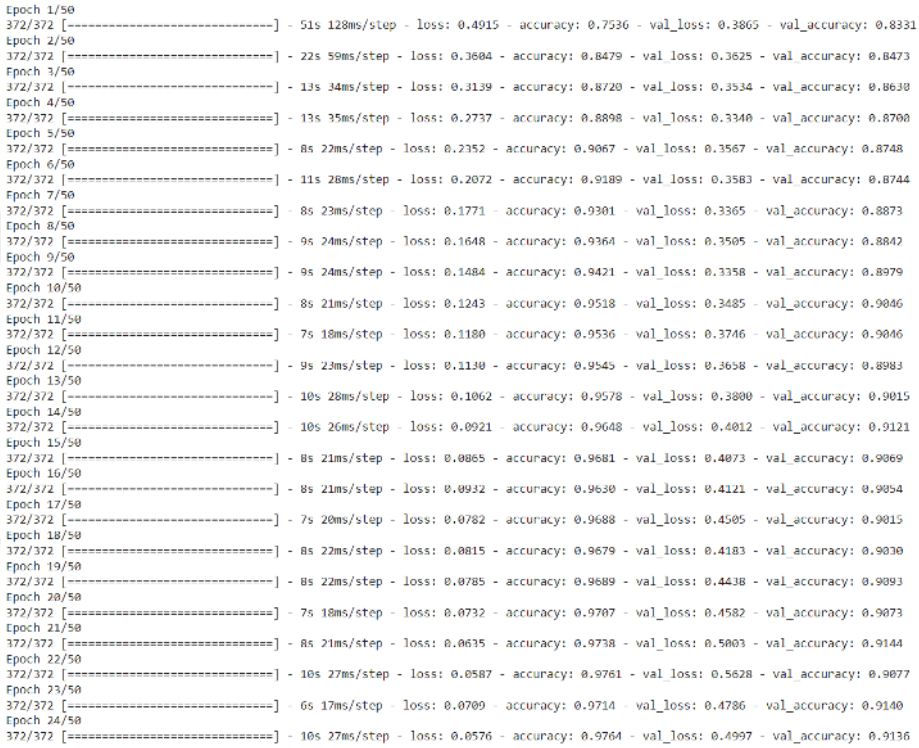
Şekil 32. GRU Modeli Oluşturulması ve Model Özeti

Kaynak: Yazar tarafından oluşturulmuştur.

GRU modeline ait sonuçlar doğruluk (accuracy) haricinde, duyarlılık (recall), kesinlik (precision), F1 skoru (F1 Score) ve karmaşıklık matrisi (confusion matrix) ile değerlendirilmiştir.

Eğitim iterasyonu (epoch) 50 olacak şekilde sınırlandırılıp eğitim sırasında aşırı uyum (over fitting) durumunun engellenmesi amacı ile erken durdurma (early stopping) geri çağırma (callback) yöntemi kullanılmıştır.

Eğitim iterasyonu 38 iterasyon boyunca devam etmiştir. Eğitim toplamda 7 dakika sürmüştür. GRU modelinin şekil 33'te ilk 24 ve şekil 34'te son 14 eğitim iterasyonuna ait doğruluk ve kayıp oranlarının sonuçları gösterilmiştir.



Şekil 33. GRU Modelinin İlk 24 İterasyon İlerlemesi

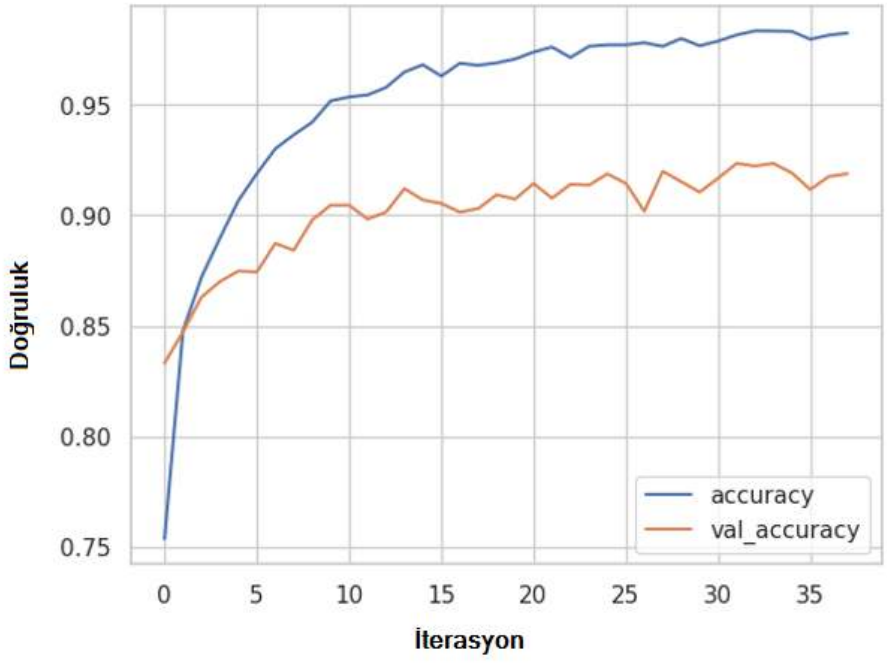
Kaynak: Yazar tarafından oluşturulmuştur.

```
Epoch 25/50
372/372 [==============================] - 8s 22ms/step - loss: 0.0573 - accuracy: 0.9771 - val_loss: 0.5165 - val_accuracy: 0.9187
Epoch 26/50
372/372 [==============================] - 9s 25ms/step - loss: 0.0559 - accuracy: 0.9771 - val_loss: 0.5396 - val_accuracy: 0.9144
Epoch 27/50
372/372 [==============================] - 7s 20ms/step - loss: 0.0547 - accuracy: 0.9780 - val_loss: 0.5380 - val_accuracy: 0.9018
Epoch 28/50
372/372 [==============================] - 7s 19ms/step - loss: 0.0606 - accuracy: 0.9764 - val_loss: 0.4717 - val_accuracy: 0.9199
Epoch 29/50
372/372 [==============================] - 8s 23ms/step - loss: 0.0524 - accuracy: 0.9800 - val_loss: 0.5053 - val_accuracy: 0.9152
Epoch 30/50
372/372 [==============================] - 9s 24ms/step - loss: 0.0565 - accuracy: 0.9767 - val_loss: 0.5235 - val_accuracy: 0.9105
Epoch 31/50
372/372 [==============================] - 7s 19ms/step - loss: 0.0546 - accuracy: 0.9788 - val_loss: 0.4954 - val_accuracy: 0.9168
Epoch 32/50
372/372 [==============================] - 8s 23ms/step - loss: 0.0433 - accuracy: 0.9816 - val_loss: 0.4955 - val_accuracy: 0.9234
Epoch 33/50
372/372 [==============================] - 8s 23ms/step - loss: 0.0400 - accuracy: 0.9834 - val_loss: 0.5618 - val_accuracy: 0.9223
Epoch 34/50
372/372 [==============================] - 6s 17ms/step - loss: 0.0376 - accuracy: 0.9833 - val_loss: 0.6138 - val_accuracy: 0.9234
Epoch 35/50
372/372 [==============================] - 10s 26ms/step - loss: 0.0432 - accuracy: 0.9832 - val_loss: 0.5981 - val_accuracy: 0.9191
Epoch 36/50
372/372 [==============================] - 8s 21ms/step - loss: 0.0508 - accuracy: 0.9796 - val_loss: 0.5782 - val_accuracy: 0.9117
Epoch 37/50
372/372 [==============================] - 8s 21ms/step - loss: 0.0457 - accuracy: 0.9815 - val_loss: 0.5461 - val_accuracy: 0.9176
Epoch 38/50
372/372 [==============================] - 9s 23ms/step - loss: 0.0418 - accuracy: 0.9824 - val_loss: 0.5588 - val_accuracy: 0.9187
```

Şekil 34. GRU Modelinin Son 14 İterasyon İlerlemesi

Kaynak: Yazar tarafından oluşturulmuştur.

GRU modelinin eğitim iterasyonu sırasında doğruluk (accuracy) metriği izlenmiştir. Modelin eğitim iterasyonları (epoch) sırasındaki doğruluk (accuracy) değerleri Şekil 35'deki gibidir.



Şekil 35. GRU Modeli Eğitim Doğruluğu

Kaynak: Yazar tarafından oluşturulmuştur.

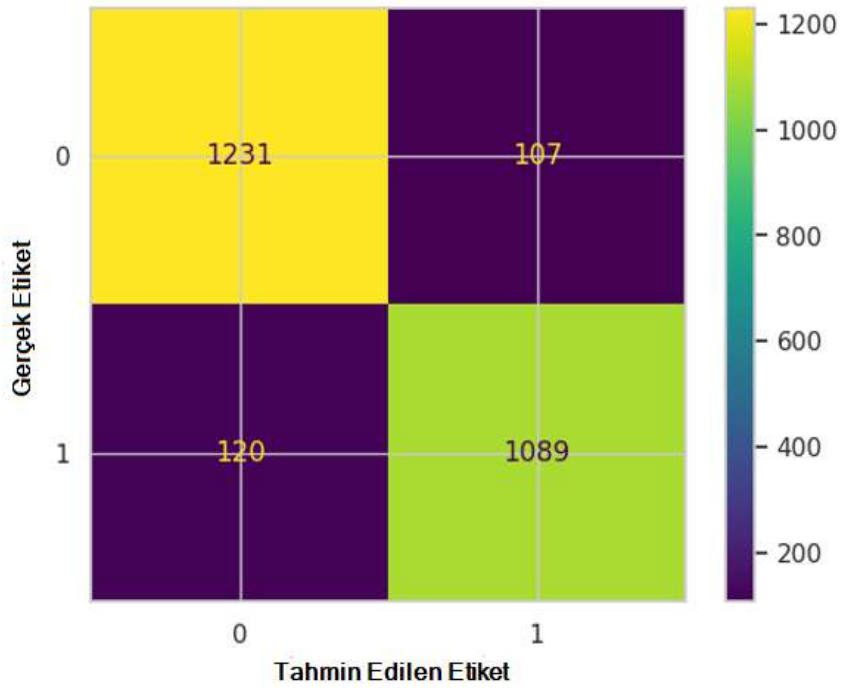
Doğruluk metriği dışında, duyarlılık (recall), kesinlik (precision), F1 Skoru (F1 score) ve karmaşıklık matrisi (confusion matrix) sonuçları da verilmiştir. Tablo 5'te GRU modeline ait bütün sonuçlar gösterilmiştir.

Tablo 5. GRU Modeline Ait Sonuçlar

Sütun1	Kesinlik (precision)	Duyarlılık (recall)	F1 Skoru (F1 Score)	Gerçek Negatif	Gerçek Pozitif	Yanlış Negatif	Yanlış Pozitif
Pozitif	0.91	0.92	0.92	1089	1231	120	107
Negatif	0.91	0.90	0.91				
Doğruluk	0.91						

Kaynak: Yazar tarafından oluşturulmuştur.

Genel doğruluk %91 bulunmuştur. Duyarlılık 0 etiketli pozitif sınıf için %92 ve 1 olarak etiketlenen negatif sınıf için %90 bulunmuştur. Kesinlik (precision) pozitif sınıf için %91, negatif sınıf için %91 olarak bulunmuştur. F1 skoru, pozitif sınıf için %92, negatif sınıf için %91 bulunmuştur. Karmaşıklık matrisi (confusion matrix) verileri, 120 negatif yanlış tahmin değeri ve 107 pozitif yanlış tahmin olduğu görülmüştür. Şekil 36'da karmaşıklık matrisi grafiği gösterilmiştir.



Şekil 36. GRU Modeli Karmaşıklık Matrisi Sonucu

Kaynak: Yazar tarafından oluşturulmuştur.

4.4. CNN Modeli

Bu çalışmada oluşturulan CNN modeli, birkaç temel katman içermektedir. İlk olarak, giriş katmanı, 4000 boyutlu bir kelime dağarcığından gelen kelimeleri 128 boyutlu gömme vektörlerine dönüştürmek için kullanılmıştır. Daha sonra, bir evrişimli katman, 64 filtre ile bir boyutlu evrişim uygulayarak metin özelliklerini öğrenmek için kullanılmıştır. Bu katmanın ardından, bir tam bağlı katman, 64 nöron içermektedir ve ReLU aktivasyon fonksiyonu ile aktive edilmiştir. Havuzlama katmanı, max pooling yöntemi kullanılarak özellikleri özetlemek için kullanılmıştır.

Düzleştirme katmanı, çıkışı düzleştirerek vektörleştirme işlemi gerçekleştirmiştir. Son olarak, çıkış katmanı, ikili sınıflandırma yapmak için softmax aktivasyon fonksiyonu ile donatılmıştır.

Önerilen model, ikili çapraz entropi kaybı kullanılarak eğitilmiş ve Adam optimizer kullanılarak optimize edilmiştir. Modelin toplam parametre sayısı 530,818'dir. Modelin oluşturulması ve modelin katman özelliklerinin yazdırılması için yazılan kod satırları Şekil 37'de gösterilmiştir.

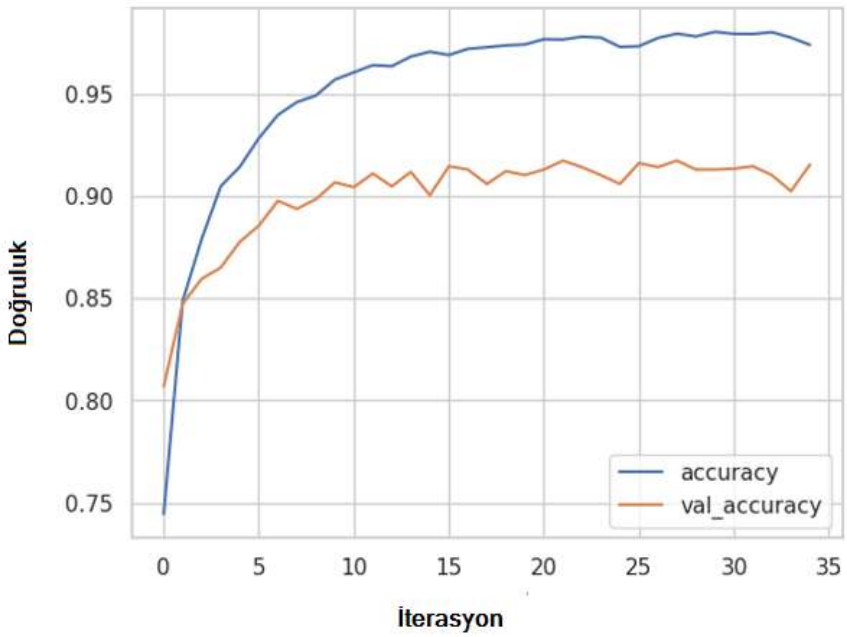
```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(4000, 128, input_length=100),
    tf.keras.layers.Conv1D(64, 1, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.MaxPooling1D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(loss='binary_crossentropy', metrics=["accuracy"], optimizer=Adam())
print(model.summary())
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_5 (Embedding)     (None, 100, 128)          512000

 conv1d_3 (Conv1D)           (None, 100, 64)           8256

 dense_10 (Dense)            (None, 100, 64)           4160

 max_pooling1d_3 (MaxPoolin  (None, 50, 64)            0
 g1D)

 flatten_2 (Flatten)         (None, 3200)              0

 dense_11 (Dense)            (None, 2)                 6402

=================================================================
Total params: 530818 (2.02 MB)
Trainable params: 530818 (2.02 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Şekil 37. CNN Modeli Oluşturulması ve Katmanları

Kaynak: Yazar tarafından oluşturulmuştur.

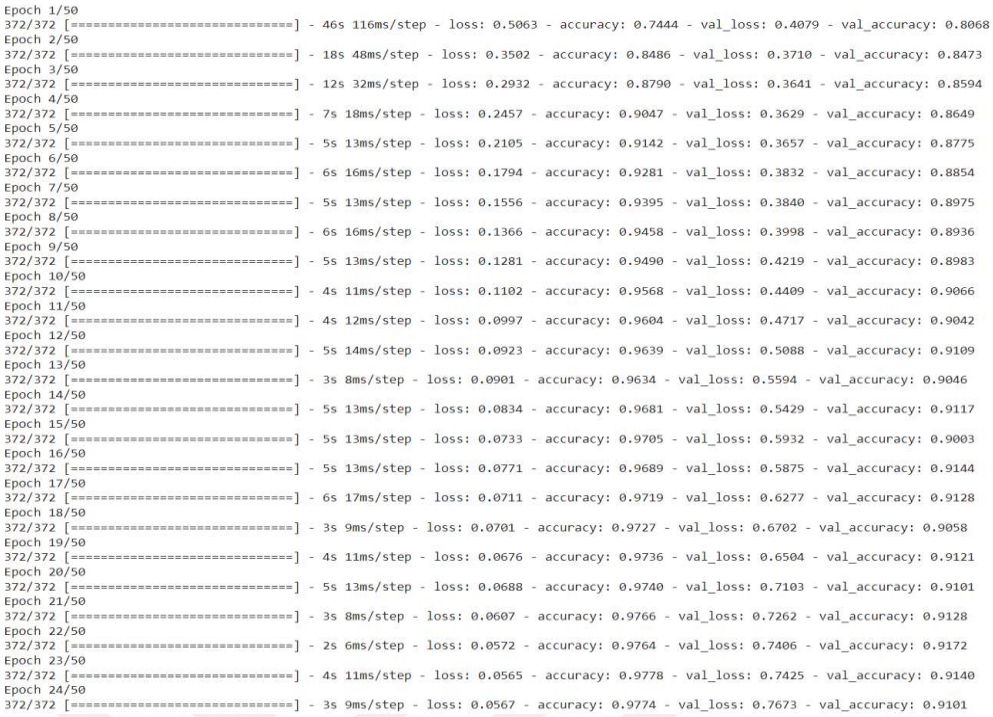
Model, eğitim sürecinde 50 epoch olarak sınırlandırılmıştır. Modelin performansını izlemek ve aşırı uyumu (overfitting) önlemek için erken durdurma (early stopping) yöntemi kullanılmıştır. Eğitim 35 iterasyon boyunca devam etmiştir. Eğitim toplamda 7 dakika sürmüştür. Eğitim sırasında, modelin doğruluk değerleri ve kayıpları her iterasyon (epoch) sonunda kaydedilmiştir. Eğitim süresi boyunca izlenen doğruluk değerleri Şekil 38'deki gibidir.



Şekil 38. CNN Modeli Eğitim Doğruluğu

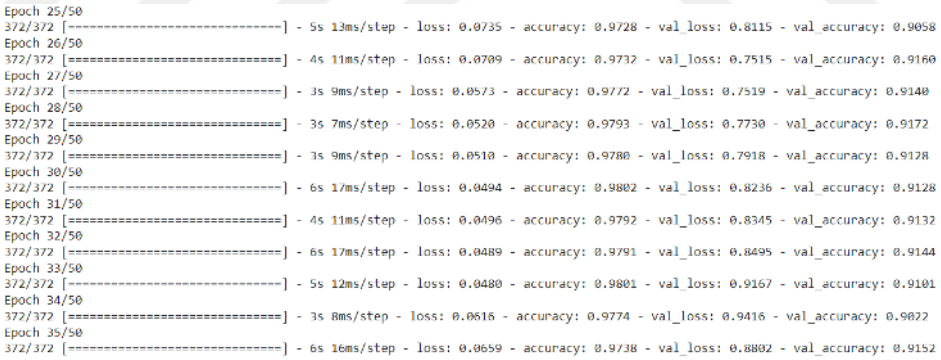
Kaynak: Yazar tarafından oluşturulmuştur.

Bu süreç, modelin doğrulama veri kümesi üzerindeki performansını izlemek ve eğitim sırasında aşırı uyumu önlemek için gerçekleştirilmiştir. Modelin iterasyon (epoch) aşamaları Şekil 39 ve 40'ta gösterilmiştir.



Şekil 39. CNN Modelinin İlk 24 İterasyon İlerlemesi

Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 40. CNN Modelinin Son 11 İterasyon İlerlemesi

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin doğruluk oranı %90 olarak belirlenmiştir, bu da modelin genel olarak iyi bir performans sergilediğini göstermektedir. Precision (kesinlik) ve recall (geri çağırma) değerleri sırasıyla pozitif sınıf için %91 ve %90, negatif sınıf için ise %89 ve %91 olarak bulunmuştur. Bu sonuçlar, modelin her iki sınıf için de yüksek doğrulukla sınıflandırma yaptığını ve dengeli bir performans sergilediğini göstermektedir. Ayrıca,

F1-score (F1 puanı) değerleri sınıf 0 için %91 ve sınıf 1 için %90 olarak hesaplanmıştır, bu da modelin kesinlik ve geri çağırma arasında iyi bir denge sağladığını göstermektedir.

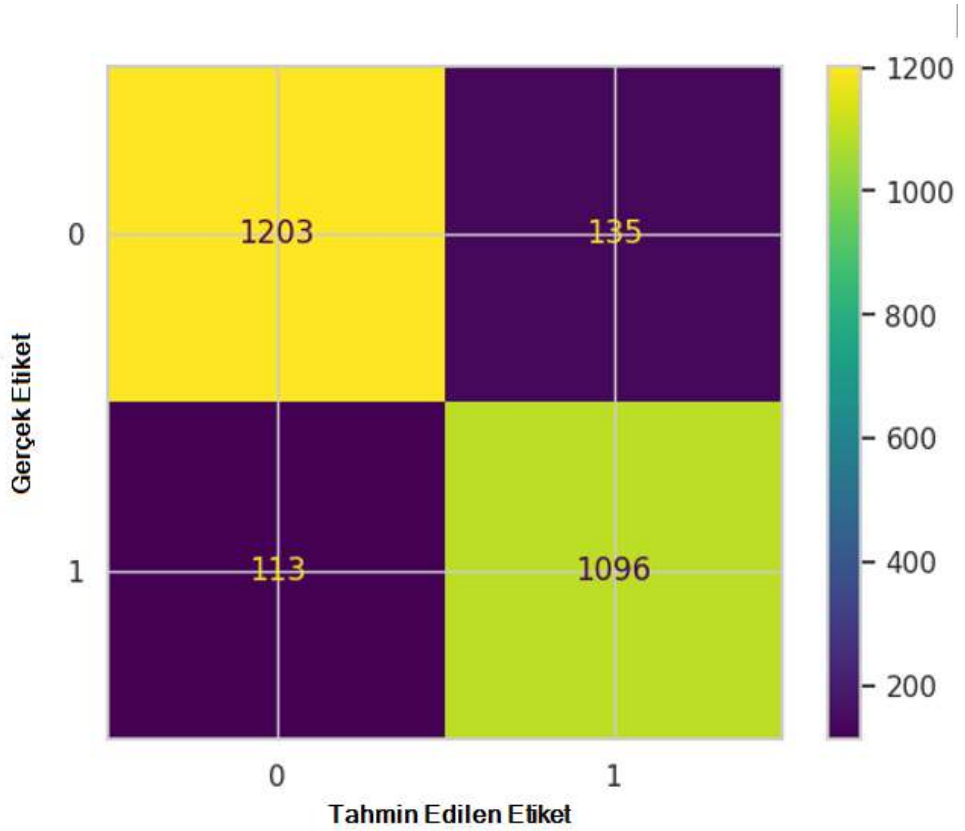
Destek (support) değerleri, pozitif sınıf için 1338 ve negatif sınıf için 1209 olarak belirlenmiştir, bu da her iki sınıf için de yeterli sayıda örnek olduğunu göstermektedir. Modelin bütün sonuçları Tablo 6'da gösterilmiştir.

Tablo 6. CNN Modeli Sonuçları

Sütun1	Kesinlik (precision)	Duyarlılık (recall)	F1 Skoru (F1 Score)	Gerçek Negatif	Gerçek Pozitif	Yanlış Negatif	Yanlış Pozitif
Pozitif	0.91	0.90	0.91	1096	1203	113	135
Negatif	0.89	0.91	0.90				
Doğruluk	0.90						

Kaynak: Yazar tarafından oluşturulmuştur.

Karmaşık ortalama (macro avg) ve ağırlıklı ortalama (weighted avg) F1 skorları, modelin tüm sınıflar için dengeli bir performans sergilediğini göstermektedir. Karmaşıklık matrisi (confusion matrix), modelin sınıflar arasında ne kadar doğru ve yanlış sınıflandırmalar yaptığını gösterir. Modelin sınıf 0'ı (pozitif sınıf) doğru sınıflandırma oranı 1203 ve yanlış sınıflandırma oranı 135 olarak belirlenmiştir. Benzer şekilde, modelin sınıf 1'i (negatif sınıf) doğru sınıflandırma oranı 1096 ve yanlış sınıflandırma oranı 113 olarak hesaplanmıştır. Modelin karmaşıklık matrisi (confusion matrix) grafiği Şekil 41'de gösterilmiştir.



Şekil 41. CNN Modeli için Karmaşıklık Matrisi

Kaynak: Yazar tarafından oluşturulmuştur.

4.5. LSTM ve CNN Hibrit Modeli

Bu çalışmada, LSTM (Uzun Kısa Süreli Bellek) ve Evrişimli Sinir Ağı (CNN) birleşimi ile hibrit model oluşturulmuştur. Oluşturulan model, metin verilerini işlemek için birkaç temel katmana sahiptir. İlk katman olarak, 4000 boyutlu bir kelime dağarcığından gelen kelimeleri 128 boyutlu gömme vektörlerine dönüştürmek için bir gömme katmanı kullanılmıştır. Ardından, bir LSTM katmanı, metin verilerinin zamansal ilişkilerini modellemek ve öğrenmek için kullanılmıştır. Bu katman, 64 nöron içerir ve return_sequences parametresiyle dönüş dizilerinin tüm zaman adımlarında çıktı üretir. Daha sonra, bir evrişimli katman, 32 filtre ile bir boyutlu evrişim uygulayarak metin özelliklerini öğrenmek için kullanılmıştır. Aktivasyon fonksiyonu olarak ReLU kullanılmıştır. Bu katmanın çıktıları, bir havuzlama katmanı aracılığıyla özetlenir.

Sonrasında gelen düzleştirme katmanı, düzleştirme katmanı çıktıları vektörleştirir. Düzleştirme katmanından sonra gelen tam bağlı katman, 32 nöron içermekte ve ReLU aktivasyon fonksiyonu ile aktive edilmiştir. Son katman, ikili sınıflandırma yapmak için softmax aktivasyon fonksiyonu ile donatılmıştır. Model, ikili çapraz entropi (binary_crossentropy) kaybı kullanılarak eğitilmiş ve Adam en iyileyici (optimizer) kullanılmıştır. Modelin oluşturulduğu kod satırları Şekil 42'de gösterilmiştir.

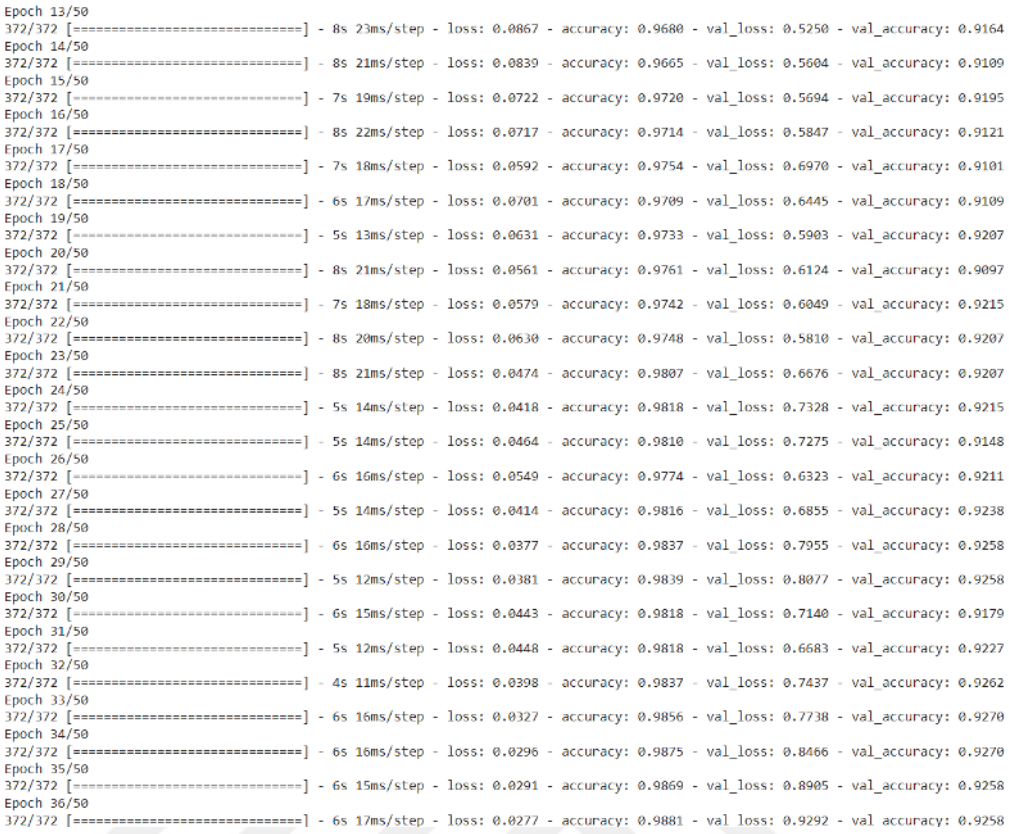
```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(4000, 128, input_length=100),
    tf.keras.layers.LSTM(64, return_sequences=True),
    tf.keras.layers.Conv1D(32, 1, activation='relu'),
    tf.keras.layers.MaxPooling1D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(loss='binary_crossentropy', metrics=["accuracy"], optimizer=Adam())
print(model.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 100, 128)          512000

 lstm_1 (LSTM)               (None, 100, 64)           49408

 conv1d_2 (Conv1D)           (None, 100, 32)           2080

 max_pooling1d_2 (MaxPoolin  (None, 50, 32)            0
 g1D)

 flatten_1 (Flatten)         (None, 1600)              0

 dense_4 (Dense)             (None, 32)                51232

 dense_5 (Dense)             (None, 2)                 66

=================================================================
Total params: 614786 (2.35 MB)
Trainable params: 614786 (2.35 MB)
Non-trainable params: 0 (0.00 Byte)
```

Şekil 42. LSTM ve CNN Modelinin Oluşturulması ve Katmanları

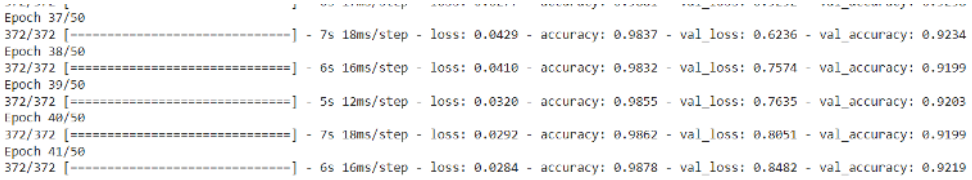
Kaynak: Yazar tarafından oluşturulmuştur.

Eğitim iterasyonu(epoch) 50 olarak sınırlandırılmıştır. Erken durdurma (early stopping) özelliği ile eğitim 41 iterasyon olarak toplamda 5 dakika sürmüştür. Şekil 43 ve 44'te iterasyonlara ait kayıp ve doğruluk oranları ile saniye süreleri gösterilmiştir.



Şekil 43. LSTM ve CNN Modelinin İlk 36 Iterasyon Sonuçları

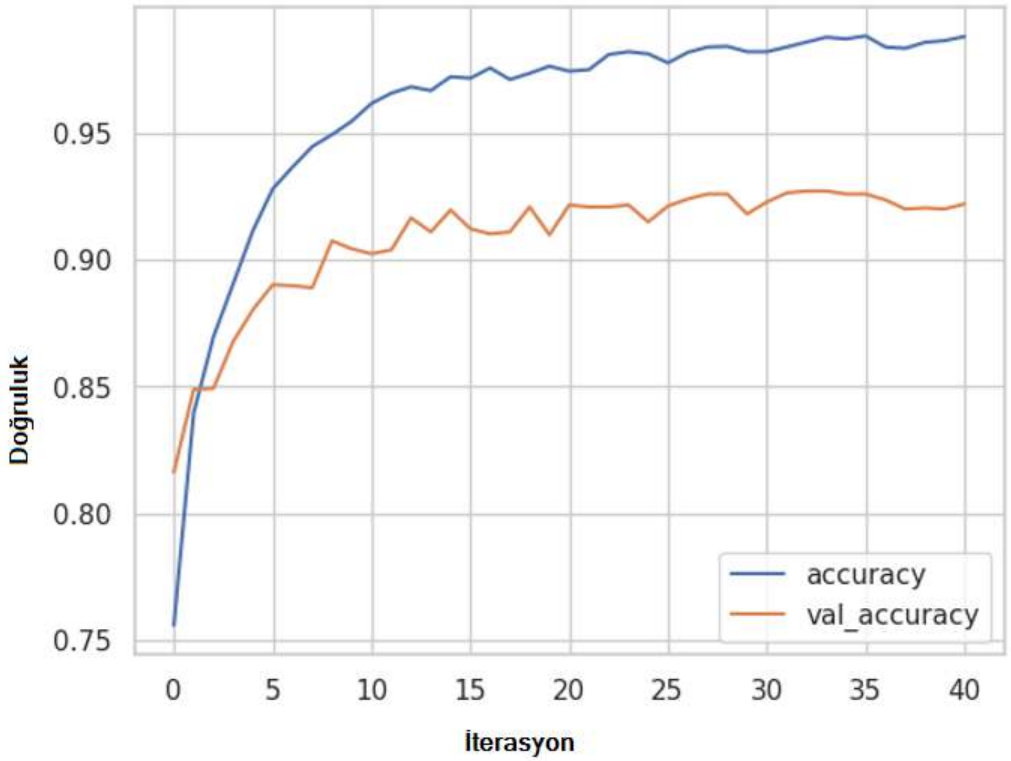
Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 44. LSTM ve CNN Modelinin Son 5 Iterasyon Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin iterasyonları boyunca doğruluk sonuçları Şekil 45‘te gösterilmiştir.



Şekil 45. LSTM ve CNN Modelinin Eğitim Doğruluğu

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin pozitif sınıf için hesaplanan kesinlik (precision), duyarlılık (recall) ve F1 skoru (F1 Score) değerleri sırasıyla %93, %91 ve %92'tür. Negatif sınıf için bu değerler %90, %92 ve %91 olarak belirlenmiştir. Modelin bütün sonuçları aşağıda Tablo 7 'deki gibidir.

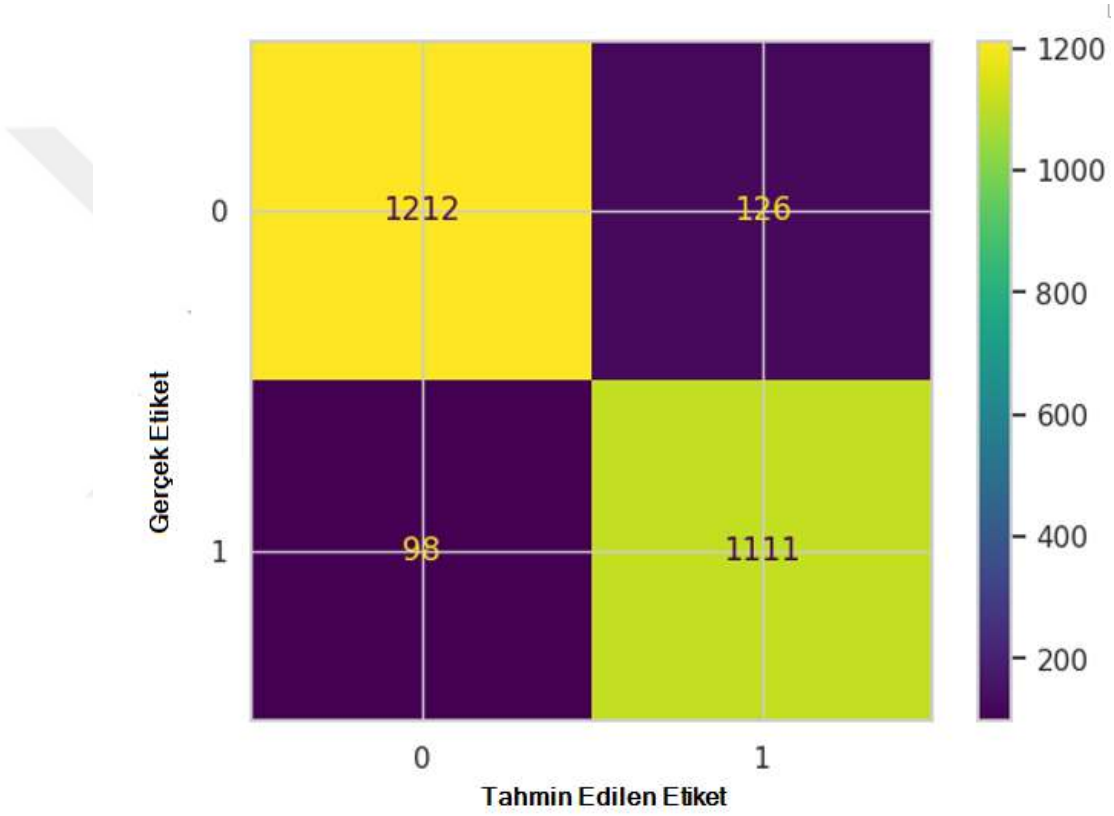
Sütun1	Kesinlik (precision)	Duyarlılık (recall)	F1 Skoru (F1 Score)	Gerçek Negatif	Gerçek Pozitif	Yanlış Negatif	Yanlış Pozitif
Pozitif	0.93	0.91	0.92	1111	1212	98	126
Negatif	0.90	0.92	0.91				
Doğruluk	0.91						

Tablo 7. LSTM-CNN Hibrit Modeli Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin toplam doğruluk oranı %91 olarak hesaplanmıştır. Karmaşıklık matrisine göre, modelin pozitif sınıfı 1212 doğru tahminle ve 126 yanlış tahminle, negatif sınıfı ise 1111 doğru tahminle ve 98 yanlış tahminle sınıflandırdığı görülmektedir.

Bu sonuçlar, modelin genel olarak başarılı olduğunu ancak pozitif sınıfı negatif sınıfa göre biraz daha iyi tahmin ettiğini göstermektedir. Karmaşıklık matrisi (confusion matrix) grafiği Şekil 46'deki gibidir.



Şekil 46. LSTM ve CNN Modeli Karmaşıklık Matrisi

Kaynak: Yazar tarafından oluşturulmuştur.

4.6. LSTM ve GRU Hibrit Modeli

Bu çalışmada oluşturulan LSTM ve GRU hibrit modeli, ilk katmanı kelime gömme (embedding) katmanı ile oluşturulmuştur. Bu katman, metin verilerini vektör uzayında temsil etmektedir.

Gömme katmanının ardından, bir dizi dikkatli biçimde tasarlanmış hücre katmanı gelir. Bu katmanlar, uzun-kısa süreli bellek (LSTM) ve güncelleme kapılarına sahip olan geçitli tekrarlayan birim (GRU) hücrelerini içerir. Son olarak, modelin çıktısını oluşturan bir dizi yoğun (dense) katman gelir. Bu katmanlar, sinir ağı çıktısını üretmek için kullanılmıştır.

Modelin performansını değerlendirmek için, ikili çapraz entropi (binary_crossentropy) kayıp fonksiyonu ve doğruluk (accuracy) metriği kullanılarak derleme işlemi gerçekleştirilir. Aktivasyon fonksiyonları olarak "relu" ve “softmax” kullanılmıştır.

Son olarak, modelin mimarisi ve parametreleri özetlenerek, modelin yapısı hakkında genel bir bilgi oluşturulması istenmiştir. Şekil 47’de modelin oluşturulması ve modelin katman bilgileri gösterilmiştir.

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(4000, 128, input_length=100),
    tf.keras.layers.LSTM(64, return_sequences=True),
    tf.keras.layers.GRU(32),
    tf.keras.layers.Dense(32, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(loss='binary_crossentropy', metrics=["accuracy"], optimizer=Adam())
print(model.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 100, 128)          512000

 lstm (LSTM)                 (None, 100, 64)           49408

 gru_1 (GRU)                 (None, 32)                9408

 dense_2 (Dense)             (None, 32)                1056

 dense_3 (Dense)             (None, 2)                 66

=================================================================
Total params: 571938 (2.18 MB)
Trainable params: 571938 (2.18 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Şekil 47. LSTM ve GRU Hibrit Modeline Ait Katman Bilgileri

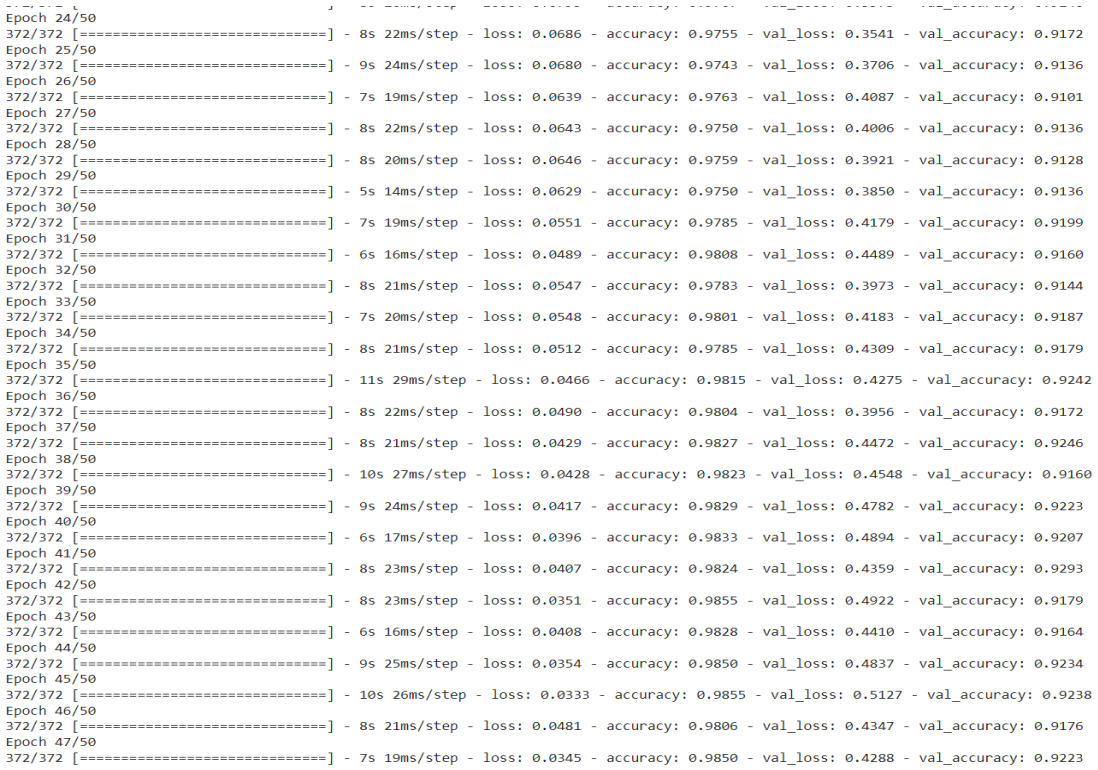
Kaynak: Yazar tarafından oluşturulmuştur.

Modelin eğitim iterasyonu (epoch) 50 ile sınırlandırılmıştır. Ardından modelin aşırı uyum (overfitting) durumuna geçmemesi amacıyla erken durdurma (early stopping) kullanılarak eğitim iterasyonu izlenmiştir. Eğitim 47 iterasyon boyunca devam etmiştir. Toplamda eğitim 7 dakika 40 saniye sürmüştür. İterasyon (epoch) esnasında doğruluk (accuracy) verileri izlenmiştir. Şekil 48 ve 49'da iterasyonlar boyunca doğruluk (accuracy), kayıp (loss) ilerlemesi gösterilmiştir.

```
Epoch 1/50
372/372 [==============================] - 54s 132ms/step - loss: 0.6594 - accuracy: 0.6072 - val_loss: 0.6501 - val_accuracy: 0.6592
Epoch 2/50
372/372 [==============================] - 27s 71ms/step - loss: 0.6231 - accuracy: 0.6754 - val_loss: 0.6142 - val_accuracy: 0.7236
Epoch 3/50
372/372 [==============================] - 11s 29ms/step - loss: 0.5996 - accuracy: 0.6965 - val_loss: 0.5365 - val_accuracy: 0.7558
Epoch 4/50
372/372 [==============================] - 12s 31ms/step - loss: 0.4418 - accuracy: 0.8071 - val_loss: 0.3775 - val_accuracy: 0.8437
Epoch 5/50
372/372 [==============================] - 11s 29ms/step - loss: 0.3263 - accuracy: 0.8662 - val_loss: 0.3499 - val_accuracy: 0.8594
Epoch 6/50
372/372 [==============================] - 11s 31ms/step - loss: 0.2659 - accuracy: 0.8967 - val_loss: 0.3302 - val_accuracy: 0.8732
Epoch 7/50
372/372 [==============================] - 10s 27ms/step - loss: 0.2242 - accuracy: 0.9154 - val_loss: 0.3335 - val_accuracy: 0.8744
Epoch 8/50
372/372 [==============================] - 10s 26ms/step - loss: 0.1902 - accuracy: 0.9304 - val_loss: 0.3320 - val_accuracy: 0.8889
Epoch 9/50
372/372 [==============================] - 8s 22ms/step - loss: 0.1637 - accuracy: 0.9387 - val_loss: 0.3430 - val_accuracy: 0.8885
Epoch 10/50
372/372 [==============================] - 9s 24ms/step - loss: 0.1489 - accuracy: 0.9467 - val_loss: 0.3290 - val_accuracy: 0.8854
Epoch 11/50
372/372 [==============================] - 9s 23ms/step - loss: 0.1337 - accuracy: 0.9510 - val_loss: 0.3368 - val_accuracy: 0.8960
Epoch 12/50
372/372 [==============================] - 6s 16ms/step - loss: 0.1286 - accuracy: 0.9530 - val_loss: 0.3066 - val_accuracy: 0.9069
Epoch 13/50
372/372 [==============================] - 8s 21ms/step - loss: 0.1112 - accuracy: 0.9601 - val_loss: 0.3269 - val_accuracy: 0.9081
Epoch 14/50
372/372 [==============================] - 7s 18ms/step - loss: 0.1100 - accuracy: 0.9603 - val_loss: 0.3117 - val_accuracy: 0.9054
Epoch 15/50
372/372 [==============================] - 8s 22ms/step - loss: 0.0962 - accuracy: 0.9665 - val_loss: 0.3242 - val_accuracy: 0.9011
Epoch 16/50
372/372 [==============================] - 7s 20ms/step - loss: 0.0971 - accuracy: 0.9659 - val_loss: 0.3229 - val_accuracy: 0.9085
Epoch 17/50
372/372 [==============================] - 10s 27ms/step - loss: 0.0986 - accuracy: 0.9650 - val_loss: 0.3204 - val_accuracy: 0.9124
Epoch 18/50
372/372 [==============================] - 7s 19ms/step - loss: 0.0944 - accuracy: 0.9666 - val_loss: 0.3077 - val_accuracy: 0.9136
Epoch 19/50
372/372 [==============================] - 9s 25ms/step - loss: 0.0843 - accuracy: 0.9697 - val_loss: 0.3298 - val_accuracy: 0.9179
Epoch 20/50
372/372 [==============================] - 11s 31ms/step - loss: 0.0779 - accuracy: 0.9717 - val_loss: 0.3384 - val_accuracy: 0.9160
Epoch 21/50
372/372 [==============================] - 9s 24ms/step - loss: 0.0845 - accuracy: 0.9697 - val_loss: 0.3563 - val_accuracy: 0.9093
Epoch 22/50
372/372 [==============================] - 9s 24ms/step - loss: 0.0768 - accuracy: 0.9719 - val_loss: 0.3617 - val_accuracy: 0.9156
Epoch 23/50
372/372 [==============================] - 8s 20ms/step - loss: 0.0795 - accuracy: 0.9707 - val_loss: 0.3573 - val_accuracy: 0.9140
Epoch 24/50
372/372 [==============================] - 8s 22ms/step - loss: 0.0686 - accuracy: 0.9755 - val_loss: 0.3541 - val_accuracy: 0.9172
```

Şekil 48. LSTM ve GRU Hibrit Modeli İlk 24 İterasyon Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.



Şekil 49. LSTM ve GRU Hibrit Modelinin Son 26 İterasyon Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.

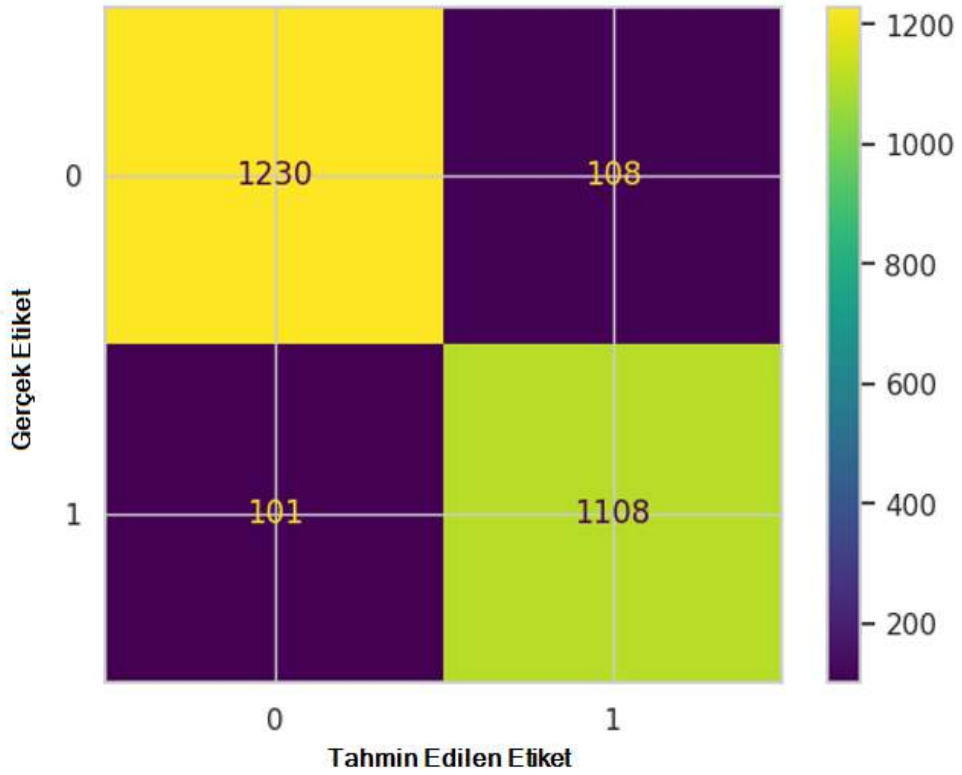
Analiz sonuçlarına göre, modelin pozitif ve negaitf sınıflarını oldukça başarılı bir şekilde tahmin ettiği görülmektedir. Özellikle, pozitif sınıf için %92 kesinlik (precision) ve %92 duyarlılık (recall) sonucu elde edilmiştir. Benzer şekilde, negatif sınıf için %91 kesinlik (precision) ve %92 duyarlılık (recall) sonucu elde edilmiştir. Bu sonuçlar, modelin her iki sınıfta da dengeli bir performans sergilediğini göstermektedir. Tablo 8'de modelin bütün sonuçları gösterilmiştir.

Tablo 8. LSTM-GRU Hibrit Modeli Sonuçları

Sütun1	Kesinlik (precision)	Duyarlılık (recall)	F1 Skoru (F1 Score)	Gerçek Negatif	Gerçek Pozitif	Yanlış Negatif	Yanlış Pozitif
Pozitif	0.92	0.92	0.92	1108	1230	101	108
Negatif	0.91	0.92	0.91				
Doğruluk	0.92						

Kaynak: Yazar tarafından oluşturulmuştur.

Karışıklık matrisi dengeli görülen sonuca göre bir yapıya sahiptir. Her iki sınıf için de benzer sayıda doğru tahminler yapılmıştır. Ancak, matrisin çapraz hücrelerine baktığımızda, bazı yanlış sınıflandırmaların olduğunu görmekteyiz. 101 hatalı pozitif sınıfında ve 108 hatalı negatif sınıfına ait tahmin yapıldığı Şekil 50'de görülmektedir.



Şekil 50. LSTM ve GRU Modeline Ait Karmaşıklık Matrisi Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.

4.7. GRU ve CNN Hibrit Modeli

İlk katman, bir giriş katmanı olarak gömme katmanını oluşturmaktadır. Bu katman, metin verilerini bir vektör uzayına dönüştürür. 6000 kelime için bir kelime dağarcığı boyutuna ve her kelimeyi temsil etmek için 128 boyutlu vektörlere sahiptir. Gömme katmanının ardından, Evrişimli Sinir Ağı (CNN) katmanları gelmektedir. Modelde iki adet Conv1D katmanı bulunmaktadır. Her biri, metin verilerindeki özellikleri yakalamak için tasarlanmış 64 filtre içerir.

ReLU aktivasyon fonksiyonu kullanılarak çıkışları doğrusallaştırılır. Evrişimli katmanlardan sonra, en büyük havuzlama katmanları gelir. Maksimum Havuzlama (MaxPooling1D), her evrişimli katmandan sonra kullanılarak özellik haritalarının boyutunu azaltmaktadır ve modelin daha az sayıda parametreyle daha fazla özellik çıkarımı yapmasına olanak tanımaktadır.

Daha sonra, bir çift yönlü (bidirectional) GRU (Gated Recurrent Unit) katmanı gelir. Bu katman, modelin metin verilerini hem ileri hem de geri yönde işlemesine olanak tanır. Bu, zaman serisini daha iyi kavramak için bilgilerin önden ve sondan alınmasını sağlar. Düzleştirme katmanı, modelin çıktısını düzleştirerek bir vektör haline getirir. Bu, tam bağlantılı katmanlara girebilmek için gereklidir. Son olarak, iki adet tam bağlantılı (Dense) katman gelir. Her biri, 64 nörona sahiptir ve ReLU aktivasyon fonksiyonu kullanılarak çıkışları doğrusallaştırılmıştır. Modelin çıkış katmanı, iki sınıflı sınıflandırma problemlerine uygun olarak tasarlanmıştır. Softmax aktivasyon fonksiyonu, modelin her bir sınıfa ait olasılık dağılımını üretmektedir.

Son olarak, model, eğitim için Adam optimizer ve ikili çapraz entropi (binary crossentropy) kayıp fonksiyonu kullanılarak derlenmiştir. Adam iyileştirici (optimizer) kullanılmıştır. Modelin oluşturulduğu kod satırları ve modelin katmanları şekil 51 ‘de gösterildiği gibidir.

```
model = tf.keras.Sequential([
    tf.keras.layers.Embedding(6000, 128, input_length=100),
    tf.keras.layers.Conv1D(64, 1, activation='relu'),
    tf.keras.layers.MaxPooling1D(),
    tf.keras.layers.Conv1D(64, 1, activation='relu'),
    tf.keras.layers.MaxPooling1D(),
    tf.keras.layers.Bidirectional(tf.keras.layers.GRU(64)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
model.compile(loss='binary_crossentropy', metrics=["accuracy"], optimizer=Adam())
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 100, 128)          768000

 conv1d (Conv1D)             (None, 100, 64)           8256

 max_pooling1d (MaxPooling1  (None, 50, 64)            0
 D)

 conv1d_1 (Conv1D)           (None, 50, 64)            4160

 max_pooling1d_1 (MaxPoolin  (None, 25, 64)            0
 g1D)

 bidirectional (Bidirection  (None, 128)               49920
 al)

 flatten (Flatten)           (None, 128)               0

 dense (Dense)               (None, 64)                8256

 dense_1 (Dense)             (None, 2)                 130

=================================================================
Total params: 838722 (3.20 MB)
Trainable params: 838722 (3.20 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
```

Şekil 51. GRU ve CNN Modelinin Oluşturulduğu Kod Satırları

Kaynak: Yazar tarafından oluşturulmuştur.

Model 50 iterasyon (epoch) ile sınırlandırılmıştır. Erken durdurma (early stopping) ile modelin eğitiminin belli bir doğruluk artışı olmadığında duracak şekilde çalışması sağlanmıştır. Eğitim toplamda 34 iterasyon (epoch) boyunca devam etmiştir. Toplam eğitim süresi 5 dakika 15 saniye sürmüştür. Modelin eğitimi boyunca doğruluk (accuracy) değerleri Şekil 52 ve 53'te gösterildiği gibi izlenmiştir.

```
Epoch 1/50
372/372 [==============================] - 60s 141ms/step - loss: 0.4367 - accuracy: 0.7878 - val_loss: 0.3242 - val_accuracy: 0.8665
Epoch 2/50
372/372 [==============================] - 18s 48ms/step - loss: 0.2519 - accuracy: 0.9021 - val_loss: 0.2727 - val_accuracy: 0.8924
Epoch 3/50
372/372 [==============================] - 16s 42ms/step - loss: 0.1962 - accuracy: 0.9281 - val_loss: 0.2557 - val_accuracy: 0.9077
Epoch 4/50
372/372 [==============================] - 9s 25ms/step - loss: 0.1608 - accuracy: 0.9435 - val_loss: 0.2528 - val_accuracy: 0.9105
Epoch 5/50
372/372 [==============================] - 9s 23ms/step - loss: 0.1370 - accuracy: 0.9516 - val_loss: 0.2374 - val_accuracy: 0.9179
Epoch 6/50
372/372 [==============================] - 7s 20ms/step - loss: 0.1169 - accuracy: 0.9596 - val_loss: 0.2550 - val_accuracy: 0.9203
Epoch 7/50
372/372 [==============================] - 7s 19ms/step - loss: 0.0960 - accuracy: 0.9687 - val_loss: 0.2631 - val_accuracy: 0.9238
Epoch 8/50
372/372 [==============================] - 9s 24ms/step - loss: 0.0812 - accuracy: 0.9742 - val_loss: 0.2627 - val_accuracy: 0.9313
Epoch 9/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0764 - accuracy: 0.9737 - val_loss: 0.2405 - val_accuracy: 0.9344
Epoch 10/50
372/372 [==============================] - 9s 23ms/step - loss: 0.0628 - accuracy: 0.9775 - val_loss: 0.2612 - val_accuracy: 0.9384
Epoch 11/50
372/372 [==============================] - 8s 21ms/step - loss: 0.0532 - accuracy: 0.9818 - val_loss: 0.2865 - val_accuracy: 0.9403
Epoch 12/50
372/372 [==============================] - 7s 20ms/step - loss: 0.0445 - accuracy: 0.9854 - val_loss: 0.3016 - val_accuracy: 0.9403
Epoch 13/50
372/372 [==============================] - 9s 23ms/step - loss: 0.0395 - accuracy: 0.9870 - val_loss: 0.2794 - val_accuracy: 0.9411
Epoch 14/50
372/372 [==============================] - 8s 21ms/step - loss: 0.0454 - accuracy: 0.9842 - val_loss: 0.2901 - val_accuracy: 0.9384
Epoch 15/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0354 - accuracy: 0.9881 - val_loss: 0.2897 - val_accuracy: 0.9415
Epoch 16/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0385 - accuracy: 0.9865 - val_loss: 0.3147 - val_accuracy: 0.9435
Epoch 17/50
372/372 [==============================] - 5s 14ms/step - loss: 0.0254 - accuracy: 0.9919 - val_loss: 0.4011 - val_accuracy: 0.9364
Epoch 18/50
372/372 [==============================] - 8s 22ms/step - loss: 0.0390 - accuracy: 0.9862 - val_loss: 0.3610 - val_accuracy: 0.9352
Epoch 19/50
372/372 [==============================] - 6s 16ms/step - loss: 0.0305 - accuracy: 0.9889 - val_loss: 0.3387 - val_accuracy: 0.9415
Epoch 20/50
372/372 [==============================] - 5s 14ms/step - loss: 0.0247 - accuracy: 0.9916 - val_loss: 0.3828 - val_accuracy: 0.9360
Epoch 21/50
372/372 [==============================] - 7s 19ms/step - loss: 0.0215 - accuracy: 0.9922 - val_loss: 0.4485 - val_accuracy: 0.9399
Epoch 22/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0337 - accuracy: 0.9871 - val_loss: 0.3460 - val_accuracy: 0.9435
Epoch 23/50
372/372 [==============================] - 5s 13ms/step - loss: 0.0227 - accuracy: 0.9920 - val_loss: 0.3771 - val_accuracy: 0.9329
Epoch 24/50
372/372 [==============================] - 6s 15ms/step - loss: 0.0213 - accuracy: 0.9921 - val_loss: 0.3740 - val_accuracy: 0.9407
```

Şekil 52. GRU ve CNN Modelinin İlk 24 İterasyon Sonuçları

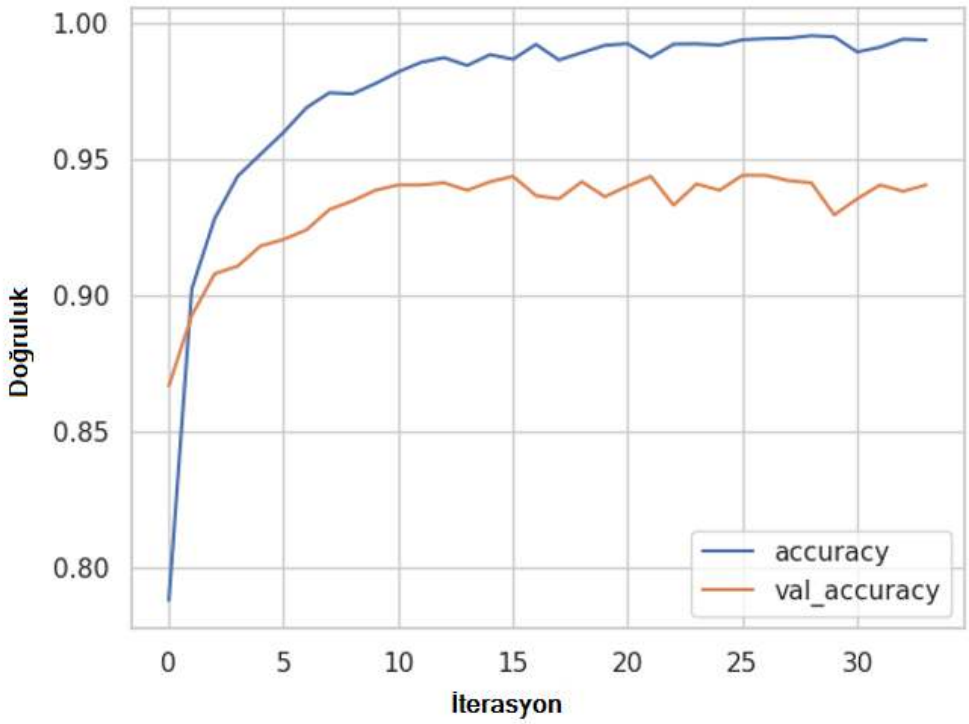
Kaynak: Yazar tarafından oluşturulmuştur.

```
Epoch 25/50
372/372 [==============================] - 6s 17ms/step - loss: 0.0223 - accuracy: 0.9917 - val_loss: 0.3485 - val_accuracy: 0.9384
Epoch 26/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0166 - accuracy: 0.9936 - val_loss: 0.4024 - val_accuracy: 0.9439
Epoch 27/50
372/372 [==============================] - 5s 14ms/step - loss: 0.0146 - accuracy: 0.9940 - val_loss: 0.4688 - val_accuracy: 0.9439
Epoch 28/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0133 - accuracy: 0.9942 - val_loss: 0.4484 - val_accuracy: 0.9419
Epoch 29/50
372/372 [==============================] - 6s 16ms/step - loss: 0.0121 - accuracy: 0.9951 - val_loss: 0.5151 - val_accuracy: 0.9411
Epoch 30/50
372/372 [==============================] - 6s 16ms/step - loss: 0.0127 - accuracy: 0.9947 - val_loss: 0.5111 - val_accuracy: 0.9293
Epoch 31/50
372/372 [==============================] - 6s 16ms/step - loss: 0.0285 - accuracy: 0.9891 - val_loss: 0.3908 - val_accuracy: 0.9352
Epoch 32/50
372/372 [==============================] - 7s 19ms/step - loss: 0.0228 - accuracy: 0.9909 - val_loss: 0.4550 - val_accuracy: 0.9403
Epoch 33/50
372/372 [==============================] - 7s 18ms/step - loss: 0.0161 - accuracy: 0.9939 - val_loss: 0.4238 - val_accuracy: 0.9380
Epoch 34/50
372/372 [==============================] - 6s 17ms/step - loss: 0.0136 - accuracy: 0.9935 - val_loss: 0.5067 - val_accuracy: 0.9403
```

Şekil 53. GRU ve CNN Hibrit Modelinin Son 10 Iterasyon Sonuçları

Kaynak: Yazar tarafından oluşturulmuştur.

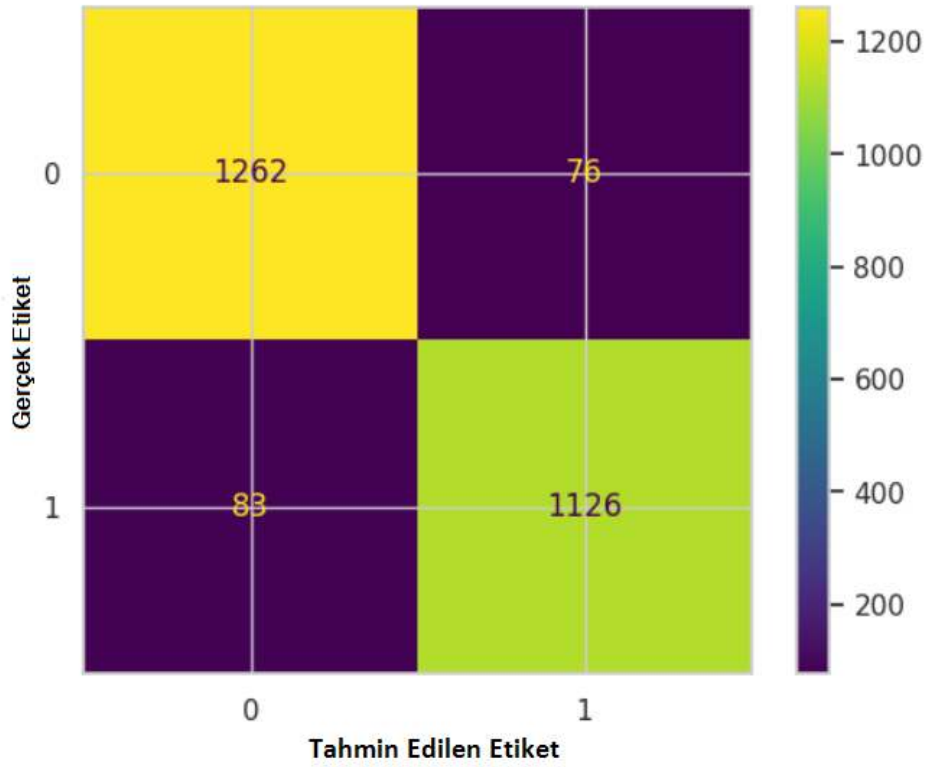
Iterasyon sonuçları izlendiğinde modelin en yüksek doğruluk değerine ulaştığı epok 34'tür. Bu noktada, model eğitim setindeki örüntüleri oldukça iyi öğrenmiş ve doğrulama setindeki verileri başarıyla sınıflandırmıştır. Modelin val_accuracy değeri 0.9403'tür, bu da doğrulama setindeki verilerin yaklaşık %94 doğrulukla sınıflandırıldığını göstermektedir. Şekil 54'de modelin eğitim iterasyonları sırasındaki doğruluk (accuracy) değerleri grafiği gösterilmiştir.



Şekil 54. GRU ve CNN Hibrit Modelinin Iterasyon Doğruluk Grafiği

Kaynak: Yazar tarafından oluşturulmuştur.

Modelin sınıflandırma performansı, doğruluk (accuracy), duyarlılık (recall), kesinlik (precision) ve F1-skoru (F1-score) gibi metrikler kullanılarak analiz edilmiştir. Elde edilen sonuçlara göre, modelin test veri setinde %94 doğruluk oranı elde ettiği gözlemlenmiştir. Duyarlılık (recall) metriği 0 ve 1 etiketli sınıf için sırasıyla %94 ve %93 olarak bulunmuştur. Ayrıca, F1-skoru %94 olarak hesaplanmıştır. Karmaşıklık matrisi (confusion matrix) incelendiğinde, pozitif sınıf için 1262 doğru ve 76 yanlış tahmin, negatif sınıf için ise 1126 doğru ve 83 yanlış tahmin yapıldığı belirlenmiştir. Karmaşıklık matrisi Şekil 55'te gösterildiği gibidir.



Şekil 55. GRU-CNN Hibrit Modeli Karmaşıklık Matrisi

Kaynak: Yazar tarafından oluşturulmuştur.

Bu sonuçlar, GRU ve CNN hibrit modelinin sınıflandırma görevlerinde yüksek performans sergileyebilme potansiyelini ortaya koymaktadır. Modelin doğruluk (accuracy) ile değerlendirildiği bütün performans metriklerinin sonucu Şekil 62'de gösterilmiştir.

Tablo 9. GRU-CNN Hibrit Modeli Sonuçları

Sütun1	Kesinlik (precision)	Duyarlılık (recall)	F1 Skoru (F1 Score)	Gerçek Negatif	Gerçek Pozitif	Yanlış Negatif	Yanlış Pozitif
Pozitif	0.94	0.94	0.94	1126	1262	83	76
Negatif	0.94	0.93	0.93				
Doğruluk	0.94						

Kaynak: Yazar tarafından oluşturulmuştur.

Bütün modellerin sonuçları incelendiğinde, sonuçlar her bir model için birbirine çok yakın olduğu görülmektedir. Her bir modele ait karşılaştırmalı sonuçlar Tablo 4'te gösterildiği gibidir.

Tablo 10. Modellerin Performans Sonuçları

Performans Metrikleri	GRU - CNN Hibrit Modeli	LSTM - GRU Hibrit Modeli	LSTM - CNN Modeli	LSTM Modeli	GRU Modeli	CNN Modeli
Pozitif Sınıf Doğruluk Sonucu	%94	%92	%91	%90	%91	%90
Negatif Sınıf Doğruluk Sonucu	%94	%92	%91	%90	%91	%90
Pozitif Sınıf Duyarlılık Sonucu	%94	%92	%91	%92	%92	%90
Negatif Sınıf Duyarlılık Sonucu	%93	%92	%92	%88	%90	%91
Pozitif Sınıf Kesinlik Sonucu	%94	%92	%93	%90	%91	%91
Negatif Sınıf Kesinlik Sonucu	%94	%91	%90	%91	%91	%89
Pozitif Sınıf F1-Skoru	%94	%92	%92	%91	%92	%91
Negatif Sınıf F1-Skoru	%93	%91	%91	%90	%91	%90

Kaynak: Yazar tarafından oluşturulmuştur.

SONUÇ

Günümüz teknolojisinde büyük veri ve derin öğrenmenin hızla geliştiği çağda, duygu analizi ve metin sınıflandırma konusunda önemli çalışmalar yapılmaktadır. Bu çalışmada, e-ticaret sitelerindeki belirli ürün grupları için yapılan yorumlar üzerinde derin öğrenme kullanılarak geliştirilen modeller ile duygu analizi çalışması gerçekleştirilmiştir. Kaggle veri tabanından temin edilen 16 bin 982 satır ve 2 kategorili ürün yorumları veri seti üzerinde çalışılmıştır. BERTurk modeli kullanılarak birimlere ayrıştırılan veri setinin derin öğrenme modellerine girdi olarak verilmesiyle LSTM, GRU ve CNN algoritmaları ile modeller oluşturularak duygu analizi yapılmıştır. Bu algoritmalar ile oluşturulan hibrit modeller ile duygu analizi sonuçları karşılaştırılmıştır. Veri analizi ve hazırlığı sürecinde Python programlama dili kullanılmıştır.

Derin öğrenme algoritmasıyla gerçekleştirilen model eğitiminden önce BERTurk modeli kullanılarak veriler modellere girdi olarak vermek üzere birimlere ayrıştırılmıştır. Modelin performansı incelendiğinde, e-ticaret verilerinde duygu analizi sonuçlarının eklendiği GRU-CNN hibrit modelinin başarılı olduğu ve yorumlar üzerinde etkili tahminler yaptığı gözlemlenmiştir. Bu sonuçlar, sınıflandırma problemleri için derin öğrenme algoritmalarının hibrit bir biçimde kullanılmasının etkinliğini öne çıkarmıştır.

Modelin eğitim sonuçlarına göre, her bir modelin doğruluk (accuracy) başta olmak üzere, bütün modellerin sonuçları birbirlerine çok yakın bulunmuştur. En düşük doğruluk sonucu %90 ile LSTM ve CNN algoritmaları ile oluşturulmuş iki ayrı modelin verdiği görülmüştür. En yüksek doğruluk (accuracy) sonucu %94 ile GRU-CNN hibrit modelinin sonucu olarak bulunmuştur. Bu sonuçlar, e-ticaret verilerinin duygu analizi için kullanılan hibrit derin öğrenme modelinin güvenilirliğini ve etkililiğini doğrulamaktadır.

Bu alanda gelecekte yapılacak olan çalışmalar için şunlar önerilmektedir:

Daha Büyük Veri Setleri ile Çalışma: Daha geniş kapsamlı ve çeşitli veri setlerinin kullanılması, modelin daha genel ve güvenilir sonuçlar üretmesine olanak sağlayabilir.

Daha Gelişmiş Derin Öğrenme Algoritmaları ile Oluşturulan Hibrit Modellerin Kullanımı: GRU ve CNN hibrit modelinin yanı sıra, daha gelişmiş derin öğrenme tekniklerinin kullanıldığı hibrit modeller oluşturulması duygu analizi performansını artırabilir.

Duygu Analizinin İçerik Kapsamının Genişletilmesi: Metinlerdeki duyguların yanı sıra, ürün özelliklerine veya hizmet kalitesine ilişkin yorumların da analiz edilmesi, müşteri memnuniyeti ve ürün/hizmet kalitesi hakkında daha kapsamlı bilgiler sağlayabilir.

Çoklu Dil Desteği: E-ticaret sitelerinde kullanılan çok dilli verilerin duygu analiziyle incelenmesi, küresel pazarlarda daha etkili bir şekilde bilgi sağlayabilir.

Anlamsal Analiz Tekniklerinin Entegrasyonu: Metinlerdeki duygusal ifadelerin yanı sıra, anlamsal analiz tekniklerinin (örneğin, duyguların nedenleri ve sonuçları hakkında bilgi sağlayabilecek durum analizi) de kullanılması, daha derinlemesine içgörüler elde edebilir.

Makine Çevirisi ile Duygu Analizi Entegrasyonu: Farklı dillerdeki yorumların makine çevirisi ile analiz edilmesi ve ardından duygu analizinin yapılması, küresel ölçekte daha kapsamlı bir anlayış sağlayabilir.

Gerçek Zamanlı Analiz İmkanlarının Geliştirilmesi: E-ticaret platformlarında gerçek zamanlı olarak yorumların analiz edilmesi ve bu analiz sonuçlarının anlık olarak müşterilere veya işletmelere iletilmesi, hızlı karar alma süreçlerini destekleyebilir.

Kullanıcı Profili ve Tercihlerinin Entegrasyonu: Müşteri profilleri ve geçmiş tercihleri gibi kişiselleştirilmiş bilgilerin duygu analizi sürecine entegre edilmesi, daha kişisel ve etkili bir alışveriş deneyimi sağlayabilir.

Bu öneriler, e-ticaret sitelerindeki duygu analizi çalışmalarını daha ileriye götürmek ve daha değerli içgörüler elde etmek için potansiyel yöntemler sunmaktadır.

KAYNAKÇA

Alaca, Y. (2023). *Graf ve karekod yöntemleriyle dönüştürülmüş log kayıtları üzerinde derin öğrenme tabanlı siber saldırı tespiti* [Doktora Tezi]. Karabük Üniversitesi.

Bahl, LR., Jelinek, F., Mercer, R.L. (1983) A maximum likelihood approach to continuous speech recognition. *IEEE Trans Pattern Anal Mach Intell*. *5*(2),179-190. https://doi.org/10.1109/TPAMI.1983.4767370

Ballı, Ç. (2021). *Doğal dil işleme ile türkçe içerikli paylaşımlardan sosyal medya kullanıcılarının duygu analizi* [Yüksek Lisans Tezi]. Ankara Üniversitesi.

Bozcu, H. (2024*). Derin öğrenme tabanlı kiraz yapraklarında hasar tespiti* [Yüksek Lisans Tezi]. Bilecik Şeyh Edebali Üniversitesi.

Brown, P.F., Cocke, J., Pietra, S.D., Pietra, V.J., Jelinek, F., Lafferty, J.D., Mercer, R.L., ve Roossin, P.S. (1990). A statistical approach to machine translation. *Computational Linguistics*, *16*(2), 79-85.

Çakmak, E., ve Selvi, İ. H. (2022). Derin öğrenme (CNN, RNN, LSTM, GRU) kullanarak protein ikincil yapı tahmini. *Acta Infologica*, *6*(1), 43-52. DOI: 10.26650/acin.1008075

Çelik, Ö. ve Odabaş, A. (2020). Sign2Text: Konvolüsyonel sinir ağları kullanarak Türk işaret dili tanıma. *Avrupa Bilim ve Teknoloji Dergisi*, (19), 923-934. https://doi.org/10.31590/ejosat.747231

Çifcibası, T. (2021). *Duygu analizi ve derin öğrenme kullanılarak borsaya kote ulaştırma şirketlerine ait endekslerin tahmin edilmesi* [Yüksek Lisans Tezi]. Marmara Üniversitesi.

Demiroğlu, E. (2021). *Sosyal medyada online tüketici yorumlarının performansını öngörme* [Yüksek Lisans Tezi]. Gebze Teknik Üniversitesi.

Devlin, J., Chang, M.-W., Lee, K., ve Toutanova, K. (2019, 2-7 Haziran). *Bert: Pre-training of deep bidirectional transformers for language understanding* [Bildiri Sunumu]. In Proceedings of the 2019 Conference of the North

American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis. https://doi.org/10.18653/v1/N19-1423

Doğan, F., ve Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *DÜMF Mühendislik Dergisi*, *10*(2), 409–445. https://doi.org/10.24012/dumf.411130.

Dönmez, H. G. ve Becerikli Y. (2023). Türkçe Twitter verileri üzerinde duygu analizi: LSTM, CNNLSTM, BERT algoritmalarının karşılaştırılması. *International Journal of Advanced Natural Sciences and Engineering Researches, 7*(6), 132-138. https://doi.org/10.59287/ijanser.1146

Fukushima, K. (1980). A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193-202. https://doi.org/10.1007/BF00344251

Gavcar, E., ve Metin, H. M. (2021). Hisse senedi değerlerinin makine öğrenimi (derin öğrenme) ile tahmini. *Ekonomi ve Yönetim Araştırmaları Dergisi, 10*(2), 1-11.

Güran, A., Uysal, M., ve Doğrusöz, Ö. (2014). Destek vektör makineleri parametre optimizasyonunun duygu analizi üzerindeki etkisi. *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Bilimleri Dergisi*, *16*(48), 86-93.

Hanbay, K., ve Üzen, H. (2017). Nesne tespit ve takip metotları: Kapsamlı bir derleme. *Türk Doğa ve Fen Dergisi, 6*(2), 40-49.

Hinton, G. E., Osindero, S., ve Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527-1554.

Kayaalp, K., ve Süzen, A. A. (2018). *Derin öğrenme ve Türkiye'deki uygulamaları.* IKSAD International Publishing House.

Kesim, E., Helli, S.S., ve Cavsak, S.N. (2023). *Çok kipli duygu tanıma için zaman bazlı modellerin karşılaştırılması.* Computer Science, Machine Learning https://arxiv.org/pdf/2306.13076

Kim, P. (2017). *Matlab deep learning: With machine learning, neural networks and artificial intelligence.* Apress. https://doi.org/10.1007/978-1-4842-2845-6

LeCun, Y., Bottou, L., Bengio, Y., ve Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324. https://doi.org/10.1109/5.726791

Liu, B. (2012). *Sentiment analysis and opinion mining: Synthesis lectures on human language technologies.* Morgan ve Claypool Publishers. https://doi.org/10.1007/978-3-031-02145-9

Özcan, H. (2014). *Çok düşük çözünürlüklü yüz imgelerinde derin öğrenme uygulamaları* [Yüksek Lisans Tezi]. Deniz Harp Okulu.

Özgeldi, M. (2019). Yapay zekâ ve insan kaynakları. In G. Telli (Ed.), *Yapay zekâ ve gelecek içinde* (s. 198-222). Doğu Kitapevi.

Pang, B., ve Lee, L. (2008). Opinion mining and sentiment analysis. *World Journal of Gastroenterology, 22*(45), 9898–9908. https://doi.org/10.3748/wjg.v22.i45.9898

Rothman, D. (2018). *Artificial intelligence by example*. Packt.

Salouhou, A. (2019). *El yazısı karakter tanıma ve resim sınıflandırmada derin ögrenme yaklaşımları* [Yüksek Lisans Tezi]. Fatih Sultan Mehmet Vakıf Üniversitesi.

Salur, M. U., ve Aydin, I. (2020). A novel hybrid deep learning model for sentiment classification. *IEEE Access*, 8, 58080-58093. https://doi.org/10.1109/ACCESS.2020.2982538

Sarıel, S. (2017). Günümüzde yapay zekâ. In M. Karaca (Ed.), *İnsanlaşan makineler ve yapay zekâ içinde* (s. 21-25). İstanbul Teknik Üniversitesi Vakfı Dergisi.

Simonyan, K., ve Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*. https://doi.org/10.48550/arXiv.1409.1556

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... ve Rabinovich, A. (2015, 7 Haziran). *Going deeper with convolutions*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston.

Şeker, S. E. (2016). Duygu analizi (sentimental analysis). *YBS Ansiklopedi, 3*(3), 21-36.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., ve Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*(2), 267–307. https://doi.org/10.1162/COLI_a_00049

Tan, F. G., Yüksel, A. S., Aydemir, E., Ersoy, M. (2021). Derin öğrenme teknikleri ile nesne tespiti ve takibi üzerine bir inceleme. *Avrupa Bilim ve Teknoloji Dergisi* (25), 159-171.

Tan, K. R., Villarıno, M. L., ve Maderazo, C. (2019, 8-9 Kasım). *Automatic music mood recognition using Russell's twodimensional valence-arousal space from audio and lyrical data as classified using SVM and Naïve Bayes.* In IOP Conference Series: Materials Science and Engineering (Vol. 482, No. 1, p. 012019). IOP Publishing.

Toprak, Ş. (2023). *Makine öğrenmesi yöntemlerini kullanarak bir petrokimya firmasının hisse senedi fiyat tahmini* [Yüksek Lisans Tezi]. Sakarya Üniversitesi.

Tunç, T. (2024). *Akıllı batarya kapasitesinin derin öğrenme yöntemleriyle tahmini* [Yüksek Lisans Tezi]. Başkent Üniversitesi.

Tung, H. (2024). *GRU ile bölgesel tüketim modelleme ve tahmin: derin öğrenme ile piyasa davranışlarını anlama* [Yüksek Lisans Tezi]. Batman Üniversitesi.

Turing, A. M. (1950). The word problem in semi-groups with cancellation. *Journal of Symbolic Logic. 117*(1), 74-76. http://dx.doi.org/10.2307/2267491

Vural, A. (2021). *Görüntü işlemede derin öğrenme tabanlı süper çözünürlük uygulamaları* [Yüksek Lisans Tezi]. İstanbul Teknik Üniversitesi.

Yılmaz, A., ve Kaya, U. (2021). *Derin öğrenme*. Kodlab.

Yılmaz, MC., ve Orman, Z. (2021). LSTM Derin öğrenme yaklaşımı ile covid-19 pandemi sürecinde Twitter verilerinden duygu analizi. *Acta Infologica, 5*(2), 359-372.

Yönet, E. (2023). *Şiir kategorisinin doğal dil işleme yöntemleri kullanılarak tahmin edilmesi* [Yüksek Lisans Tezi]. Konya Teknik Üniversitesi.

İnternet Kaynağı:

Kaggle (2021, 27 Nisan), *Duygu analizi veri seti (olumlu/olumsuz),* 27 Nisan 2024 tarihinde https://www.kaggle.com/datasets/burhanbilenn/turkish-customer-reviews-for-binary-classification adresinden edinilmiştir.

Kaggle (2021, 27 Nisan), *Duygu analizi veri seti (olumlu, olumsuz, tarafsız),* 27 Nisan 2024 tarihinde https://www.kaggle.com/datasets/burhanbilenn/duygu-analizi-icin-urun-yorumlari adresinden edinilmiştir.