

**SOSYAL AĞLARDA DUYGU ANALİZİ İÇİN HİBRİT BİR
YÖNTEM GELİŞTİRİLMESİ**

MUHAMMET SİNAN BAŞARSLAN

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
DR. ÖĞR. ÜYESİ FATİH KAYAALP**

DÜZCE, 2023

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SOSYAL AĞLARDA DUYGU ANALİZİ İÇİN HİBRİT BİR
YÖNTEM GELİŞTİRİLMESİ

Muhammet Sinan BAŞARSLAN tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Dr. Öğr. Üyesi Fatih KAYAALP

Düzce Üniversitesi

Jüri Üyeleri

Dr. Öğr. Üyesi Fatih KAYAALP

Düzce Üniversitesi

Doç. Dr. Abdullah Talha KABAKUŞ

Düzce Üniversitesi

Dr. Öğr. Üyesi Şafak KAYIKÇI

Bolu Abant İzzet Baysal Üniversitesi

Prof. Dr. Pakize ERDOĞMUŞ

Düzce Üniversitesi

Prof. Dr. Kemal POLAT

Bolu Abant İzzet Baysal Üniversitesi

Tez Savunma Tarihi: 12/06/2023

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

12 Haziran 2023

(İmza)

Muhammet Sinan BAŞARSLAN

TEŐEKKÜR

Doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Dr. Öğr. Üyesi Fatih KAYAALP'e en içten dileklerle teşekkür ederim. Lisans eğitimimden bu yana eğitim başta olmak üzere her zaman desteklerini esirgemeyen Prof. Dr. Pakize ERDOĞMUŐ ve Prof. Dr. Resul KARA hocalarıma teşekkür ederim. Tez izleme Kurul üyelerim Doç. Dr. Abdullah Talha KABAKUŐ ve Dr. Öğr. Üyesi Şafak KAYIKÇI hocalarıma da emekleri için teşekkür ederim.

Lisansüstü eğitim süreçlerinde her zaman destekleriyle yanımda olan Doç. Dr. Ümit AĞBULUT ve Dr. Öğr. Üyesi Hüseyin BAKIR hocalarıma da teşekkür ederim.

Yaşadığım sağlık problemlerim sebebiyle, Tez çalışma sürecime geri dönmemi sağlayan Prof. Dr. Necmettin PAMİR, Prof. Dr. Kazım Meriç ŐENGÖZ ve Prof. Dr. Koray ÖZDUMAN hocalarıma sonsuz teşekkürlerimi sunarım.

Her zaman desteklerini benden esirgemeyen, hep yanımda olan aileme ve çalışma arkadaşlarıma Őükranlarımı sunarım.

12 Haziran 2023

Muhammet Sinan BAŐARSLAN

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ.....	viii
ÇİZELGE LİSTESİ.....	x
KISALTMALAR.....	xii
SİMGELER	xiv
ÖZET	xv
ABSTRACT	xvi
EXTENDED ABSTRACT.....	xvii
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	7
2.1. IMDB VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR.....	7
2.2. ROTTEN TOMATOES VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR.....	8
2.3. TRIPADVISOR VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR ..	9
2.4. TWITTER COVID-19 VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR.....	9
3. YAPAY ZEKA	11
3.1. ZEKA	11
3.2. AKIL	11
3.3. YAPAY ZEKA	11
3.3.1. Yapay Zeka Nasıl Çalışır?	12
3.4. DOĞAL DİL İŞLEME	12
3.4.1. NLP'nin Çalışma Alanları ve Uygulamaları	13
3.4.2. NLP'nin Kullanıldığı Endüstriler.....	14
3.4.3. NLP Teknolojileri.....	15
3.4.3.1. Makine Öğrenmesi Modelleri	15
3.4.3.2. Derin Öğrenme, Ön Eğitimli Modeller ve Transfer Öğrenme.....	16
3.4.3.3. Ön İşleme Teknikleri	17
3.4.4. NLP'de Kullanılan Programlama Dilleri.....	18
3.4.4.1. Python	18
3.4.5. NLP Geliştirme Ortamları	18
3.4.5.1. TensorFlow ve PyTorch.....	18
3.4.5.2. HuggingFace	19
3.4.5.3. Spark NLP.....	19
3.4.5.4. SpaCy NLP.....	19
3.4.6. Duygu Analizi	19
4. MATERYAL VE METOT	22
4.1. METİN MADENCİLİĞİ VE METİN ÖNİŞLEME	22
4.2. METİN TEMSİL YÖNTEMLERİ.....	23

4.2.1. Frekans Tabanlı Yöntemler	23
4.2.1.1. <i>Terim Frekansı-Ters Belge Frekansı</i>	23
4.2.1.2. <i>Kelime Torbası</i>	24
4.2.2. Yapay Sinir Ağı Tabanlı Metin Temsil Yöntemleri.....	25
4.2.2.1. <i>Word2Vec</i>	25
4.2.2.2. <i>Doc2Vec</i>	28
4.2.2.3. <i>GloVe</i>	30
4.2.2.4. <i>Tekrarlayan Sinir Ağı</i>	30
4.2.2.5. <i>RNN ve Dikkat</i>	31
4.2.2.6. <i>Dönüştürücü Metin Temsil Yöntemi</i>	32
4.3. MAKİNE ÖĞRENMESİ.....	37
4.3.1. Yapay Sinir Ağları	38
4.3.1.1. <i>ANN'nin Yapısı</i>	40
4.3.2. Naïve Bayes	41
4.3.3. Karar Ağacı	41
4.3.4. Rastgele Orman.....	41
4.3.5. Lojistik Regresyon	42
4.3.6. K en Yakın Komşu	43
4.3.7. Destek Vektör Makineleri	43
4.4. DERİN ÖĞRENME.....	43
4.4.1. Derin Öğrenme Mimarileri	45
4.4.1.1. <i>Tekrarlayan Sinir Ağları</i>	45
4.4.1.2. <i>Uzun Kısa Süreli Hafıza</i>	47
4.4.1.3. <i>Kapı Tekrarlayan Geçitler</i>	49
4.4.1.4. <i>Evrşimli Sinir Ağları</i>	50
4.4.1.5. <i>Çift Yönlü Kapı Tekrarlayan Hücreler</i>	51
4.4.1.6. <i>Çift Yönlü Uzun Kısa Süreli Bellek</i>	52
4.5. TOPLULUK ÖĞRENME	53
4.5.1. Oylama	54
4.5.1.1. <i>Çoğunluk Oylaması</i>	54
4.5.1.2. <i>Olasılıklı Oylama</i>	55
4.5.2. Yığınlama	56
4.6. DENEY ORTAM ÖZELLİKLERİ	57
4.6.1. Deney Ortamlarının Yazılım ve Donanım Özellikleri	57
4.6.2. Derin Öğrenme Mimari Kurulumları.....	57
4.7. PERFORMANS METRİKLERİ.....	58
4.7.1. Doğruluk	59
4.7.2. Kesinlik.....	59
4.7.3. Duyarlılık	59
4.7.4. F-ölçü	60
5. DENEYSEL ÇALIŞMALAR VE SONUÇLARI	62
5.1. IMDB VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR.....	62
5.1.1. Deneyde Kullanılan Veri Kümesi	62
5.1.2. Makine Öğrenmesi Deneyi	64
5.1.2.1. <i>Ön işleme ve Metin Temsili</i>	64
5.1.2.2. <i>Deney Kurulumu</i>	65
5.1.2.3. <i>Deney Sonuçları</i>	65
5.1.3. Derin Öğrenme Deneyi	66
5.1.3.1. <i>Ön işleme ve Metin Temsili</i>	69

5.1.3.2. Deney Kurulumu	69
5.1.3.3. Deney Sonuçları.....	69
5.1.4. Çok Katmanlı BiGRU ve Çok Katmanlı CNN Hibrit (MBiGRUMCONV) Deneyi	78
5.1.4.1. Ön işleme ve Metin Temsili.....	78
5.1.4.2. Deney Kurulumu	79
5.1.4.3. Sınama testi ayırım sonuçları	87
5.1.4.4. K-Kat Çapraz Doğrulama Ayırım sonuçları.....	94
5.1.5. Ön Eğitimli Yöntemler Deneyi.....	96
5.1.5.1. Ön işleme ve Metin Temsili.....	96
5.1.5.2. Deney Kurulumu	96
5.1.5.3. Ön Eğitimli GloVe Sonrası Alınan Sonuçlar	97
5.1.5.4. BERT Sonrası MBiGRUCONV ile alınan Sonuçlar	97
5.1.5.5. BERT-CNN Deneyi	98
5.1.5.6. Deney Sonuçlarının Genel Değerlendirilmesi.....	99
5.2. FARKLI VERİ KÜMELERİ ÜZERİNDE YAPILAN DENEYLER.....	100
5.2.1. TripAdvisor ve Rotten Tomatoes Üzerinde Yapılan Deney.....	101
5.2.1.1. Deneyde Kullanılan Veri Kümeleri.....	101
5.2.1.2. Ön işleme ve Metin Temsilleri	101
5.2.1.3. Deney Kurulumu	102
5.2.1.4. Deney Sonuçları.....	102
5.2.2. COVID-19 Twitter Veri Kümesi Üzerinde Yapılan Deney.....	104
5.2.2.1. Deneyde Kullanılan Veri Kümesi	104
5.2.2.2. Deney Kurulumu	106
5.2.2.3. Deney Sonuçları.....	107
6. SONUÇ VE TARTIŞMA.....	115
7. KAYNAKLAR.....	118
ÖZGEÇMİŞ	129

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1. 2023 yılına ait sosyal medya uygulamaları ve veri paylaşım istatistiği [3].....	2
Şekil 1.2. Tezin akış diyagramı.	5
Şekil 3.1. Yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki [55].	12
Şekil 3.2. Duygu analizi yaklaşımları [74].	21
Şekil 4.1. CBOW yöntemi [92].	26
Şekil 4.2. Skip-Gram yöntemi [92].	27
Şekil 4.3. PV-DM yöntemi [92].	29
Şekil 4.4. PV-DBOW yöntemi [92].	29
Şekil 4.5. PV-DM ve PV-DBOW [96].	29
Şekil 4.6. Kodlayıcı-Kod Çözücü mimarisi [65].	31
Şekil 4.7. Gizli katmanlardaki bilginin Kodlayıcı-Kod Çözücü mimarisinde işlenmesi [65].	32
Şekil 4.8. Dönüştürücü mimarisi [93].	33
Şekil 4.9. BERT ön eğitilmiş dönüştürücü [65].	35
Şekil 4.10. Makine öğrenmesi [107].	37
Şekil 4.11. Makine öğrenmesi süreci [107].	38
Şekil 4.12. Biyolojik sinir ağı [107].	38
Şekil 4.13. Yapay sinir ağının matematiksel modeli [107].	39
Şekil 4.14. Aktivasyon fonksiyon grafikleri [110].	40
Şekil 4.15. Yapay sinir ağı [111].	40
Şekil 4.16. Rastgele orman [115].	42
Şekil 4.17. SVM yöntemi [117].	43
Şekil 4.18. Derin öğrenme uygulama süreci [112].	44
Şekil 4.19. RNN mimarisi [114].	45
Şekil 4.20. Yann'ın RNN mimarisi [114].	46
Şekil 4.21. LSTM mimarisi [127].	47
Şekil 4.22. GRU mimarisi [128].	49
Şekil 4.23. Doğal dil işleme de CNN kullanımı [132].	51
Şekil 4.24. Çift Yönlü Tekrarlayan sinir ağları a) GRU, b) BiGRU [135].	52
Şekil 4.25. Çift Yönlü Tekrarlayan sinir ağları a) LSTM, b) BiLSTM [136].	53
Şekil 4.26. Topluluk öğrenme süreci [138].	53
Şekil 4.27. Çoğunluk oylaması süreci [140].	55
Şekil 4.28. Olasılıklı oylama süreci [141].	56
Şekil 4.29. Yığınlama süreci [140].	56
Şekil 4.30. Çapraz doğrulama yöntemi ve test ve eğitim kümesi ayrımı [148].	60
Şekil 4.31. Sınama testi yöntemi ve test ve eğitim kümesi ayrımı [148].	61
Şekil 4.32. Sınama testi yöntemi ve test, eğitim ve doğrulama kümeleri ayrımı [148].	61
Şekil 5.1. IMDB veri kümesi üzerinde yapılan çalışmaların akış diyagramı.	62
Şekil 5.2. IMDB veri kümesi duygu sınıfı dağılımı.	63
Şekil 5.3. Yorumların kelime bulut dağılımı a) pozitif yorumlar b) negatif yorumlar... ..	64
Şekil 5.4. Derin öğrenme deneyi için oluşturulan 7 model diyagramı a) LSTM b) BiLSTM c) GRU d) BiGRU.	68
Şekil 5.5. Hiperparametre kombinasyonlarının çalışma kayıtları.	81

Şekil 5.6. TensorFlow hiperparametre menüsü.	82
Şekil 5.7. TensorFlow parametre menüsündeki paralel koordinat sekmesi.	84
Şekil 5.8. Önerilen model (MBiGRUMCONV).	85
Şekil 5.9. IMDB veri kümesi %70-%30 ve %10 doğrulama ayırımında eğitim ve doğrulama kayıp grafiği.	93
Şekil 5.10. IMDB veri kümesinde yapılan çalışmanın akış diyagramı.	96
Şekil 5.11. IMDB veri kümesinde BERT MBiGRUMCONV eğitim doğrulama doğruluk ve kayıp grafiği.	98
Şekil 5.12. IMDB veri kümesinde BERT-CNN %80-%20 eğitim ve doğrulama kayıp grafiği.	99
Şekil 5.13. TripAdvisor ve Rotten Tomatoes veri kümeleriyle yapılan çalışmanın akış diyagramı.	101
Şekil 5.14. Covid-19 Twitter veri kümesinde yapılan çalışmanın akış diyagramı.	104
Şekil 5.15. Covid-19 Twitter veri kümesi duygu sınıfı dağılım görselleştirilmesi.	105
Şekil 5.16. Duygu sınıfı kelime bulutu.	106



ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 4.1. TF-IDF parametreleri.	24
Çizelge 4.2. BOW parametreleri.....	25
Çizelge 4.3. Word2Vec parametreleri.	27
Çizelge 4.4. Çalışmada kullanılan Word2Vec parametreleri.	28
Çizelge 4.5. LSTM mimarisine ait değişkenler.	48
Çizelge 4.6. Karmaşıklık matrisi.	58
Çizelge 5.1. IMDB veri kümesi.	63
Çizelge 5.2. IMDB veri kümesi detaylı bilgisi.	63
Çizelge 5.3. Word2Vec yöntemi parametreleri.	65
Çizelge 5.4. TF-IDF ve BOW yöntemi parametreleri.	65
Çizelge 5.5. Doğruluk performans sonuçları (%).	65
Çizelge 5.6. Derin öğrenme deneyi LSTM doğruluk sonucu (%80 eğitim, %20 test)...	70
Çizelge 5.7. Derin öğrenme deneyi LSTM doğruluk sonucu (%70 eğitim, %30 test)...	71
Çizelge 5.8. Derin öğrenme deneyi GRU doğruluk sonucu (%80 eğitim, %20 test).	72
Çizelge 5.9. Derin öğrenme deneyi GRU doğruluk sonucu (%70 eğitim, %30 test).	73
Çizelge 5.10. Derin öğrenme deneyi BiLSTM doğruluk sonucu (%80 eğitim, %20 test).	74
Çizelge 5.11. Derin öğrenme deneyi BiLSTM doğruluk sonucu (%70 eğitim, %30 test).	75
Çizelge 5.12. Derin öğrenme deneyi BiGRU doğruluk sonucu (%80 eğitim, %20 test).	76
Çizelge 5.13. Derin öğrenme deneyi BiGRU doğruluk sonucu (%70 eğitim, %30 test).	77
Çizelge 5.14. Derin öğrenme deneyini önceki çalışmalarla karşılaştırma.	78
Çizelge 5.15. Modelde kullanılan parametreler.	79
Çizelge 5.16. Model parametre özet bilgisi.	85
Çizelge 5.17. Önerilen modelin hiperparametre detayı.	86
Çizelge 5.18. Önerilen modelin %80-%20 eğitim-test ve %10 doğrulama ayırımında doğruluk sonuçları.	87
Çizelge 5.19. Önerilen modelin %70-%30 eğitim-test ve %10 doğrulama ayırımında doğruluk sonuçları.	87
Çizelge 5.20. IMDB veri kümesi %80-%20 eğitim-test ayırımında eğitim ve doğrulama kayıp grafikleri.	89
Çizelge 5.21. IMDB veri kümesi %70-%30 eğitim-test ve %10 doğrulama ayırımında eğitim ve doğrulama kayıp grafikleri.	91
Çizelge 5.22. IMDB veri kümesi %80-%20 eğitim-test k-kat çapraz doğrulama doğruluk sonuçları.	94
Çizelge 5.23. IMDB veri kümesi %70-%30 eğitim-test k-kat çapraz doğrulama doğruluk sonuçları.	94
Çizelge 5.24. Önerilen modelin test doğruluk sonuçlarının önceki çalışmalarla karşılaştırılması.	94
Çizelge 5.25. Önerilen modelin doğrulama doğruluk sonuçlarının benzer çalışmalarla karşılaştırılması.	95
Çizelge 5.26. %70-%30 eğitim-test ve %10 doğrulama doğruluk sonuçları.	97
Çizelge 5.27. %80-%20 eğitim-test ve %10 doğrulama doğruluk sonuçları.	97
Çizelge 5.28. IMDB veri kümesinde %70-%30 ve %80-%20 eğitim-test ve %10 doğrulama doğruluk sonuçları.	97

Çizelge 5.29. %70-%30 ve %80-%20 eğitim-test ve %10 sına testi doğruluk sonuçları.	98
Çizelge 5.30. MBiGRUMCONV modelinin, BERT-CNN, BERT-MBiGRUCONV modelleriyle test ve doğrulama doğruluk sonuçlarıyla karşılaştırılması.....	100
Çizelge 5.31. Kullanılan Word2Vec parametreleri.	102
Çizelge 5.32. TF-IDF yöntemi parametreleri.	102
Çizelge 5.33. TripAdvisor veri kümesinde doğruluk sonucu (%80 eğitim, %20 test).	103
Çizelge 5.34. TripAdvisor veri kümesinde doğruluk sonucu (%70 eğitim, %30 test).	103
Çizelge 5.35. Rotten Tomatoes veri kümesinde doğruluk sonucu (%80 eğitim, %20 test).	103
Çizelge 5.36. Rotten Tomatoes veri kümesinde doğruluk sonucu (%70 eğitim, %30 test).	104
Çizelge 5.37. Covid-19 Twitter veri kümesi bilgisi.....	105
Çizelge 5.38. Covid-19 Twitter veri kümesi duygu sınıf dağılımı.	105
Çizelge 5.39. Kullanılan Word2Vec parametreleri.	106
Çizelge 5.40. BOW ve TF-IDF yöntemi parametresi.	107
Çizelge 5.41. Covid-19 veri kümesi üzerinde yapılan deneyin doğruluk sonuçları (%80 eğitim, %20 test).	107
Çizelge 5.42. Covid-19 veri kümesi üzerinde yapılan deneyin doğruluk sonuçları (%80 eğitim, %20 test).	108
Çizelge 5.43. Covid-19 veri kümesi üzerinde yapılan deneyin f-ölçü sonuçları (%80 eğitim, %20 test).	109
Çizelge 5.44. Covid-19 veri kümesi üzerinde yapılan deneyin f-ölçü sonuçları (%70 eğitim, %30 test).	110
Çizelge 5.45. Covid-19 veri kümesi üzerinde yapılan deneyin kesinlik sonuçları (%80 eğitim, %20 test).	110
Çizelge 5.46. Covid-19 veri kümesi üzerinde yapılan deneyin kesinlik sonuçları (%70 eğitim, %30 test).	111
Çizelge 5.47. Covid-19 veri kümesi üzerinde yapılan deneyin duyarlılık sonuçları (%80 eğitim, %20 test).	112
Çizelge 5.48. Covid-19 veri kümesi üzerinde yapılan deneyin duyarlılık sonuçları (80:20 eğitim:test).	113
Çizelge 5.49. Covid-19 veri kümesi verilerine yönelik diğer çalışmalar.	114

KISALTMALAR

ACC	Accuracy (Doğruluk)
ANN	Artificial Neural Network (Yapay Sinir Ağları)
BOW	Bag of Words (Kelime Torbası)
BERT	Bidirectional Encoder Representations from Transformers (Dönüştürücülerden Çift Yönlü Kodlayıcı Temsilleri)
BiGRU	Bidirectional Gated Recurrent Unit (Çift Yönlü Kapı Tekrarlayan Geçitler)
BiLSTM	Bidirectional Long Short Term Memory (Çift Yönlü Uzun Kısa Süreli Bellek)
ChatGPT	Chat Generative Pre-trained transformers (Sohbet Üretken Ön Eğitimli Dönüştürücüler)
CBOW	Continuous Bag of Words (Sürekli Kelime Torbası)
CNN	Convolutional Neural Network (Evrışimli Sinir Ağları)
CONV1D	Convolutional (1 Boyutlu Konvülyasyon)
Doc2Vec	Document Vector (Belge Vektörü)
DT	Decision Tree (Karar Ağacı)
ELU	Exponential Linear Unit (Üstel lineer birim)
FFNN	Feed Forward Neural Network (İleri Beslemeli Sinir Ağı)
GPT	Generative Pre-trained transformers (Ön eğitimli Üretken Dönüştürücüler)
GloVe	Global Vectors (Global Vektörler)
GRU	Gated Recurrent Unit (Kapı Tekrarlayan Geçitler)
HTML	HyperText Markup Language (Hiper Metin İşaretleme Dili)
IMDB	Internet Movie Database
KNN	K Nearest Neighbor (K En Yakın Komşu)
LR	Logistic Regression (Lojistik Regresyon)
LSTM	Long Short Term Memory (Uzun Kısa Süreli Bellek)
ME	Maximum Entropy (Maksimum Entropi)
MLM	Masked Language Modeling (Maskeli Dil Modellemesi)
MV	Majority Voting (Çoğunluk Oylama)
NB	Naïve Bayes
NLP	Natural Language Processing (Doğal Dil İşleme)
NLTK	Natural Language Toolkit (Doğal Dil Araç Takımı)
NSP	Next Sentence Prediction (Sonraki Cümle Tahmini)
PV-DBOW	Distributed Bag of Words Paragraph Vector (Paragraf Vektörünün Dağıtılmış Kelime Çantası Modeli)
PV-DM	Distributed Memory Model of Paragraph Vectors (Paragraf Vektörlerinin Dağıtılmış Bellek Modeli)
PV	Probabilistic Voting (Olasılıksal Oylama)
RF	Random Forest (Rastgele Orman)
ReLU	Rectified Linear Unit (Doğrultulmuş Lineer Birim)
RNN	Recurrent Neural Network (Tekrarlayan Sinir Ağı)
SG	Skip-Gram (Gram Atlama)
SVM	Support Vector Machine (Destek Vektör Makinesi)
SGD	Stochastic Gradient Descent (Stokastik Gradyan İnişi)
Word2Vec	Word Vector (Kelime Vektörü)
TF	Term Frequency (Terim Frekansı)

IDF	Inverse Document Frequency (Ters Belge Frekansı)
TF-IDF	Term Frequency-Inverse Document Frequency (Terim Frekansı-Ters Belge Frekansı)
TPU	Tensor Processing Unit (Tensör İşleme Birimi)
TB	Terabyte



SİMGELER

Σ	Toplam Sembolü
σ	Sigma
b	Bias Değeri
C_t	Mevcut Durum
\hat{C}_t	Hücre Giriş Aktivasyon
$F(x)$	Ağırlık Fonksiyonu
f	Aktivasyon Fonksiyonu
f_t	Unutma Kapısı Aktivasyon Fonksiyonu
h	Gizli Katman
h_t	Akım Çıkışı
h_0	0. Gizli Katman
i_t	Çıkış Kapısı Aktivasyon Fonksiyonu
F_n	False Negative (Yanlış Negatif)
F_p	False Positive (Yanlış Pozitif)
Poz	F_n ve T_p Toplamı
Neg	F_p ve T_n Toplamı
o_t	Çıkış Kapısı
r_t	Reset Kapısı Aktivasyon Fonksiyonu
W	Kelime
w_0	0. Ağırlık
x_0	0. Giriş
$X_{i,j}$	X dokümanında Kelimelerin (i, j) Beraber Bulunma Sayısı
x	Çarpım Sembolü
x_t	Mevcut Giriş
W	Ağırlık Matrisi
t	Zaman
\tanh	Aktivasyon Fonksiyonu
T_n	True Negative (Doğru Negatif)
T_p	True Positive (Doğru Pozitif)
T_{neg}	F_n ve F_p toplam değeri
T_{poz}	T_p ve F_p toplam değeri
z_t	Güncelleme Kapısı Aktivasyon Fonksiyonu

ÖZET

SOSYAL AĞLARDA DUYGU ANALİZİ İÇİN HİBRİT BİR YÖNTEM GELİŞTİRİLMESİ

Muhammet Sinan BAŞARSLAN

Düzce Üniversitesi

Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Dr. Öğr. Üyesi Fatih KAYAALP

Haziran 2023, 128 sayfa

Teknolojinin gelişmesi, insanların sosyal hayatlarında birçok alışkanlığının değişimine sebep olmuştur. Pandeminin de etkisiyle eğitim başta olmak üzere birçok alanda yüz yüze iletişim oldukça azalmıştır. İnsanlar, sosyalleşmek için sosyal medya ve web sitelerine hızlıca erişim sağlayabilecekleri mobil cihazlar ile vakit geçirir hale gelmişlerdir. Pandemi sona ermesine rağmen, insanlar sosyalleşmek için web sitelerini ve sosyal medya uygulamalarını hala yaygın olarak kullanmaya devam edilmektedirler. İnsanlar birlikte izledikleri film, gittikleri restoran gibi birçok deneyimleri hakkında fikirlerini anlık şekilde paylaşmaktadırlar. Bu durum sosyal medya ve web siteleri aracılığıyla sürekli bir veri paylaşımına sebep olmaktadır. Bu veri paylaşımı çok büyük boyutta veri toplanmasına sebep olmaktadır. Bu veriler, pazarlamadan, reklama kadar birçok kurumu yakından ilgilendirmektedir. Bu kurumlar kendileri veya sektörlerine yönelik paylaşılan verileri kullanmak amacıyla çalışmalar yapmaktadırlar. Gelişen yapay zeka teknolojileriyle, bu çalışmalar firmalara katma değer kazandırmaktadır. Bu değeri kazandıran çalışma alanlarından biri de yapay zekanın alt dalı olan doğal dil işleme görevlerinden duygu analizidir. Bu çalışmada, IMDB web sayfasından toplanan veri kümesinde duygu analizi görevine yönelik deneysel çalışmalar gerçekleştirilmiştir. IMDB veri kümesi üzerinde Word2Vec kelime gömme yöntemi sonrası altı adet BiGRU ve sonrasına iki Evrişim Sinir ağı ile MBiGRUMCONV olarak isimlendirilen yeni bir yöntem önerilmiştir. Önerilen bu yöntemin en iyi doğruluk performansını verdiği %80-%20 eğitim-test ve %10 doğrulama kümesinde test sonuçları; %90,59, doğrulama kümesi sonuçları %94,67'dir. Ayrıca önerilen modelin 3, 5 ve 10 kat çapraz geçiş sonuçları da alınmıştır. En iyi sonuç, 5-kat çapraz geçiş ayarlarında %90,67 doğruluk değeri ile alınmıştır. Tüm sonuçlar bir arada değerlendirildiğinde önerilen yöntemin literatür çalışmalarına kıyasla rekabetçi bir performans sergilediği görülmüştür. Çalışma kapsamında TripAdvisor, Rotten Tomatoes ve Twitterdan toplanan açık kaynak olarak sunulan veri kümeleri üzerinde farklı metin temsili yöntemleri sonrasında çeşitli derin öğrenme, makine öğrenmesi ve topluluk öğrenme yöntemleriyle modeller oluşturulmuştur. Yüksek model başarımı almak için gelecek çalışmalarda, hibrit model oluştururken; ön eğitilmiş BERT türevleri ile metin temsili çıkarıldıktan sonra çift yönlü sinir ağlarının kullanılması önerilmektedir.

Anahtar Sözcükler: Derin Öğrenme, Doğal Dil İşleme, Duygu Analizi, Makine Öğrenmesi, Topluluk Öğrenme.

ABSTRACT

DEVELOPMENT OF A HYBRID METHOD FOR SENTIMENT ANALYSIS IN SOCIAL NETWORKS

Muhammet Sinan BAŞARSLAN

Duzce University

Graduate School, Department of Computer Engineering

Doctoral Thesis

Supervisor: Assist. Prof. Dr. Fatih KAYAALP

June 2023, 128 pages

The development of technology has led to changes in many habits in people's social lives. As a result of the pandemic, face-to-face communication has decreased significantly in many areas, especially education. People have started to spend time with mobile devices where they can quickly access social media and websites to socialize. Even though the pandemic is over, people are still widely using websites and social media applications to socialize. People instantly share their opinions about their experiences such as the movie they watched together or the restaurant they went to. This leads to constant data sharing through social media and websites. This data sharing leads to massive data collection. This data is of great interest to many organizations from marketing to advertising. These organizations are working to use the data shared for themselves or their sectors. With the developing artificial intelligence technologies, these studies bring added value to companies. One of the areas of study that adds this value is sentiment analysis, one of the natural language processing tasks, which is a sub-branch of artificial intelligence. In this study, experimental studies were carried out for sentiment analysis task on the dataset collected from IMDB web page. A new method called MBiGRUMCONV was proposed on the IMDB dataset with six BiGRUs and two Convolutional Neural Networks after the Word2Vec word embedding method. The best accuracy performance of this proposed method is 90,59% for the 80%-20% training-testing and 10% validation set, and 94,67% for the validation set. The proposed model was also cross-validation 3, 5, and 10-fold. The 5-fold cross-validation resulted in an accuracy of 90,67%. When all the results are evaluated together, it is seen that the proposed method performs competitively compared to the literature studies. Within the scope of the study, models were created with various deep learning, machine learning and ensemble learning methods after different text representation methods on open-source datasets collected from TripAdvisor, Rotten Tomatoes and Twitter. In order to achieve high model performance, in future studies, it is recommended to use bidirectional neural networks after text representation extraction with pre-trained BERT derivatives when building hybrid models.

Keywords: Deep Learning, Ensemble Learning, Machine Learning, Natural Language Processing, Sentiment Analysis.

EXTENDED ABSTRACT

DEVELOPMENT OF A HYBRID METHOD FOR SENTIMENT ANALYSIS IN SOCIAL NETWORKS

Muhammet Sinan BAŞARSLAN

Duzce University

Graduate School, Department of Computer Engineering

Doctoral Thesis

Supervisor: Assist. Dr. Fatih KAYYALP

June 2023, 128 pages

1. INTRODUCTION

The development of information technology has enabled people to instantly share ideas on many topics. The environments that are popularly used for sharing ideas are social media, which is now characterized as a tool. With the development of devices such as smartphones, it is possible to access social media at any time. People use these devices to access social media applications and websites to instantly share their opinions about movies, restaurants, hotels, health, sports, and many other areas. These posts are made from anywhere, at any time. This creates a huge source of data that can be processed into useful information. This data is processed using various artificial intelligence methods, and artificial intelligence is used to develop informative decision support systems for organizations in various sectors.

2. MATERIAL AND METHODS

Various social media texts were processed using natural language processing methods, a branch of artificial intelligence. Then, for sentiment analysis, which is one of the tasks of natural language processing, models were created using various learning methods such as machine learning, deep learning, and ensemble learning. For sentiment analysis, a balanced and widely used IMDB dataset labeled by Stanford researchers was selected. All experiments on the IMDB dataset were performed in Google Colab. The experiments on the TripAdvisor and Rotten Tomatoes datasets, which were provided as part of other studies conducted during the thesis, were performed on a local machine with an AMD Ryzen 7 processor and 64 GB of RAM. The TensorFlow 2.9.0 and Keras 2.9 libraries and Python version 3.9.13 were used to build the deep learning models. The scikit-learn 1.0.2 library was used for the machine learning models.

3. RESULTS AND DISCUSSIONS

The test result of the model named MBiGRUMCONV, which was created with six Bidirectional Gate Recurrent Unit (BiGRU) and then two Convolutional Neural Networks (CNN) according to the Word2Vec text representation method on the IMDB dataset, in the accuracy performance criterion in the 80%-20% training test and 10% validation separation; 90,59%. The result of the validation set is 94,67%. The proposed model was also cross-validation 3, 5 and 10-fold. The 5-fold cross-validation resulted in an accuracy of 90,67%. With these results, it can be seen that the model can compete with the literature.

4. CONCLUSION AND OUTLOOK

In the experiments created for sentiment analysis on the IMDB dataset, models were created with various machine learning and deep learning methods after various text representation methods such as Word2Vec, Term Frequency-Inverse Document Frequency (TF-IDF), Bag of words (BOW). Among these methods, bidirectional deep neural networks gave better accuracy performance. In the hybrid method aimed at sentiment analysis, bidirectional neural network was preferred.

As future work, it is aimed to develop a high-performance hybrid method using bidirectional neural networks and text representation methods such as (Bidirectional Encoder Representations from Transformers) BERT. For other natural language processing tasks, especially sentiment analysis, it is recommended to use bidirectional neural networks together in hybrid models to create high-performance models.

1. GİRİŞ

Son yıllarda insanların alışkanlıkları, çevresel ve teknolojik faktörlerden dolayı değişkenlik göstermektedir. Teknolojik gelişmelerle, özellikle insanların sosyalleşmeleri neredeyse tamamen akıllı telefonlar aracılığıyla sosyal medya uygulamaları üzerinden gerçekleşmektedir. Öyle ki, insanlar artık sadece resim ve video beğenmekle kalmayıp, sosyal medyadaki diğer kişilerle fotoğraf ve videolarını paylaşmaktadırlar. Bu paylaşımları; filmler, restoranlar, oteller ve spor aktiviteleri gibi birbirinden farklı deneyimleri üzerinedir. Ayrıca farklı ticari ve kamu kuruluşlarından aldıkları hizmetlerden memnuniyetlerini de Twitter, Facebook, Instagram ve Foursquare gibi sosyal medya uygulamalarında ve Youtube gibi platformlarda video olarak ta paylaşmaktadırlar. Bu paylaşımlara ek olarak TripAdvisor, AirBNB gibi hem web hem de mobil uygulama olarak hizmet veren platformlarla da paylaşmaktadırlar. Bu paylaşımlar ile çok büyük miktarlarda veriler ortaya çıkmaktadır. Twitter'da her gün yaklaşık 7 Terabyte (TB), Facebook'ta da yaklaşık 10 TB büyüklüğünde veri topluluğu oluşmaktadır [1]. Ortaya çıkan bu büyük miktarlardaki verinin işlenmesi için geleneksel bilgisayar sistemlerinin işlem gücünün ve analizi için kullanılan klasik makine öğrenmesi yöntemlerinin yetersiz kalması, yeni teknolojilerin ortaya çıkmasına sebep olmuştur. İşlem gücü artışı noktasında bulut bilişim teknolojileri yaygınlaşırken veri analizi noktasında da derin öğrenme yöntemleri yaygınlaşmıştır [2].

İnternet sayesinde haberleşme teknolojilerindeki gelişme insanları birbirine daha da yakınlaştırmıştır. Günümüzden 20 yıl öncesine kadar mektup, telgraf gibi uzun süren iletişim süreci, internet ortamının genişlemesiyle birlikte anlık hale gelmiştir. Akıllı telefon teknolojisi ile beraber ortaya çıkan uygulama yazılımlarından olan sosyal medya uygulamaları sayesinde haberleşme iki kişi arasından çıkıp herkesle iletişim kurulan bir ortam haline gelmiştir. Özellikle insanların gittiği bir film, kıyafet aldığı mağaza ve marka, gittiği restoran ve yediği yemeklerle alakalı olumlu veya olumsuz paylaşım yapması sosyal medya etkileşimlerini oluşturmakta ve kişilerin tercihleri bu durumdan etkilenmektedir. Örnek olarak gidilen bir film, mağaza veya restoran gibi yerler hakkında paylaşım yapma veya olumlu, olumsuz görüş belirtme durumu herkesi etkilemektedir.

İnsanlar, sosyal medyayı iletişimin temel ihtiyaç kaynağı olarak görmeye başlamışlardır. Sosyal medyanın ihtiyaçtan dolayı bir araç olması ve bununla beraber sosyal, ekonomik, sağlık, ürün ve marka gibi kavramlar hakkında sürekli fikir belirtmeleri bu çalışmada da yapılan duygu analizi adını almaktadır. Duygu analizi çalışmalarında metinlerde yer alan duygu ifadeleri tahmin edilmeye çalışılmaktadır. Metinlerdeki çıkarım işlemine de yapay zekanın alt dalı olan doğal dil işleme imkân vermektedir. Farklı bir ifadeyle duygu analizi, doğal dil işleme teknikleri ile geliştirilen ve karşımıza en sık çıkan problemlerden bir tanesi şeklinde de tanımlanabilir. Duygu analizi çalışmaları, 2000’li yılların başlarında ortaya çıkmıştır. 2000’li yılların öncesinde bakış açısı etkileri, duygu sıfatları, metafor yorumlamaları gibi bazı fikirler ortaya atılmış, ama alan ile derinlemesine çalışmalar ilerleyen yıllarda gerçekleştirilmiştir. Yapılan çalışmalarda, yaygın olarak duygu analizi (sentiment analysis) kullanılmasına rağmen; fikir madenciliği (opinion mining), fikir çıkarımı (opinion extraction), duygu madenciliği (sentiment mining) gibi isimlerle de çalışmalar yapılmıştır.

Bu alan ile ilgili çalışmaların çok olmasının sebebi insanların, sosyal medya ve görüş belirtilebilecek forum tarzı siteleri kullanarak büyük boyutta veri oluşturulmasıdır. Şekil 1.1’de, 2023 yılında bir dakika içerisinde insanların ürettikleri veri kaynakları ve miktarları görülmektedir.



Şekil 1.1. 2023 yılına ait sosyal medya uygulamaları ve veri paylaşım istatistiği [3].

Günümüzde birçok kurum, toplumu ilgilendiren konular üzerine yaptıkları çalışmalarını sosyal medya aracılığıyla yapmaktadır. Bu çalışmaların büyük bir bölümü özellikle duygu analizi çalışmalarıdır. Bu çalışma kapsamında IMDB film yorumları üzerinde yapılan duygu analizi çalışmaları anlatılacaktır. Ayrıca restoran, otel ve Covid-19 pandemisinde atılan Tweetler üzerinde yapılan duygu analizi çalışmaları da (5.2.2)'de yer alan farklı veri kümeleri üzerine yapılan çalışmalar başlığı altında anlatılacaktır.

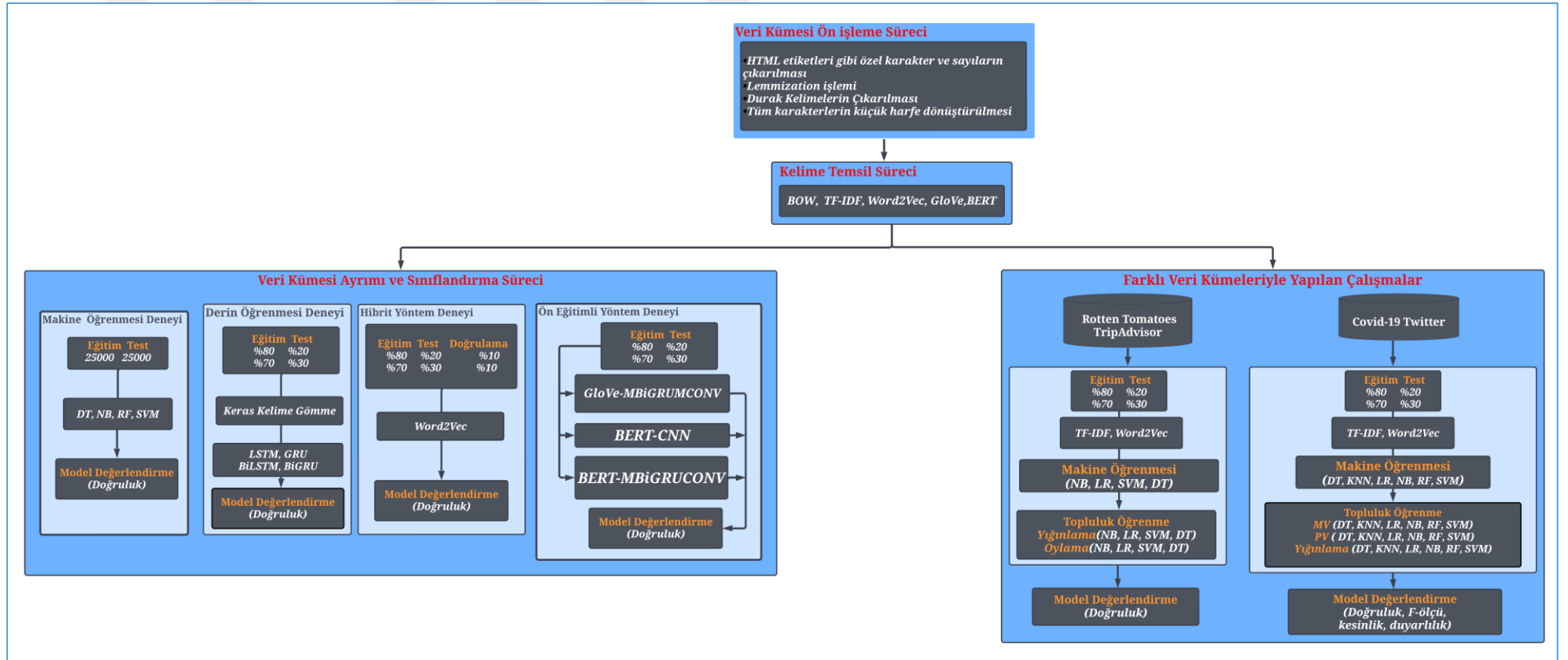
Maas ve arkadaşları [4] tarafından IMDB web sitesinden toplanarak sonrasında etiketlenen filmlere dair görüşleri barındıran veriler üzerinde derin öğrenme ve makine öğrenmesi algoritmalarıyla deneysel çalışmalar yapılmıştır. Bu çalışmalardan birincisi keras metin temsili sonrasında çeşitli optimizasyon fonksiyonları, dönem (epoch), küme boyutu (batch size) gibi parametrelerle çeşitli Sayıda yoğun katman (Dense), seyreltme (Dropout) barındıran derin öğrenme (GRU, LSTM, BiLSTM, BiGRU) modelleri oluşturulmuştur. İkinci çalışmada ise üç farklı vektör boyutunda (100, 200 ve 300) Word2Vec metin temsili sonrası duygu analizi için altı BiGRU ve iki CNN'den oluşan çok katmanlı (Multi-Layer) bir derin öğrenme modeli oluşturulmuştur. Bu çalışmalar (5.1.4)'te yer alan deneysel çalışmalar ve sonuçları bölümünde detaylı olarak anlatılmıştır.

Filmler üzerine görüşlerin yer aldığı Rotten Tomatoes [5] veri kümesi ve seyahat edilen yerlerde kalınan otel, restoran deneyimleri üzerine kullanılan sosyal medya ortamı olan TripAdvisor'dan [6] toplanan yorumlara ait veri kümeleri ile duygu analizi çalışması yapılmıştır. Metin ön işleme sürecinden sonra frekans tabanlı metin temsil yöntemi olan TF-IDF ile tahmine dayalı metin temsil yöntemi olan Word2Vec yöntemiyle kelimeler vektörleştirilmiştir. Sonrasında sınıflandırma işlemi için makine öğrenmesi ve topluluk öğrenmesi yöntemleri kullanılmıştır. Destek Vektör Makineleri (SVM), K en yakın komşu (KNN), Lojistik Regresyon (LR), Naïve Bayes (NB) makine öğrenmesi algoritmalarıyla, bu algoritmaların heterojen şekilde Yığılma (Stacking), Oylama (Voting) gibi topluluk öğrenmesi algoritmalarında bir arada kullanılması ile duygu analizi için deneyler oluşturulmuştur. Bu çalışma (5.2.1)'de yer alan farklı veri kümeleri üzerinde yapılan çalışmalar başlığı altında detaylı olarak anlatılmıştır.

Covid-19 pandemisinde Twitter'dan toplanan veriler [7] üzerinde ön işleme süreçleri sonrasında frekans tabanlı TF-IDF ve BOW metin temsil yöntemleriyle, tahmine dayalı metin temsil yöntemi olan Word2Vec ve Doc2Vec yöntemleriyle kelimeler vektörleştirilmiştir. Sonrasında sınıflandırma işlemi için makine öğrenmesi ve topluluk

öğrenmesi yöntemleri kullanılmıştır. Karar Ağacı (DT), KNN, LR, NB, Rastgele Orman (RF), SVM makine öğrenmesi algoritmalarıyla bu algoritmaların heterojen şekilde Çoğunluk Oylaması (Majority Voting-MV), Olasılıklı Oylama (Probabilistic Voting-PV) ve Yığınlama gibi topluluk öğrenmesi algoritmalarında bir arada kullanarak duygu analizi için deneyler oluşturulmuştur. Bu çalışma (5.2.2)'de yer alan farklı veri kümeleri üzerinde yapılan çalışmalar başlığı altında detaylı olarak anlatılmıştır. Bu veri kümeleri üzerinde yapılmış daha önceki çalışmalar literatür bölümünde anlatılmıştır. Bu çalışmalarını ve tezin akışını özetleyen tez akış diyagramını Şekil 1.2'de verilmiştir.





Şekil 1.2. Tezin akış diyagramı.

2. bölümde, çalışmada kullanılan veri kümeleri üzerinde daha önceden yapılmış duygu analizi çalışmalarıyla ilgili bilgiler derlenmiştir. Materyal ve metot başlıklı 4. bölümde, metin madenciliği ve metin önişleme, metin temsil yöntemleri ve çalışmada kullanılan makine öğrenmesi, topluluk öğrenme ve derin öğrenme yöntemleriyle alakalı geniş teorik bilgiler verilmiştir. Ayrıca 4. Bölümde çalışmada yapılan deneylere ait yazılım ve donanım özellikleri ve oluşturulan modellerin performans değerlendirilmesine yönelik performans metrikleri detaylıca anlatılmaktadır. Çalışmanın deneysel sonuçları 5. bölüm olan deneysel çalışmalar ve sonuçları başlığında anlatılmıştır. 6. Bölüm olan sonuç ve tartışma bölümünde, tez süresince yapılan çalışmaların sonuçlarına yönelik değerlendirmeler ve geleceğe yönelik çalışma hedefleriyle öneriler verilmiştir.



2. LİTERATÜR TARAMASI

Tez çalışmasında kullanılan IMDB [4], Rotten Tomatoes [5], TripAdvisor [6], ve Covid-19 Twitter [7] veri kümelerine ait literatürde yer alan çalışmalar, bu bölümün alt başlıklarında anlatılacaktır.

2.1. IMDB VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR

IMDB veri kümesi, Stanford Araştırmacıları tarafından toplanıp etiketlenerek herkese açık paylaşılan bir veri kümesidir [4]. Bu bölümde IMDB veri kümesi ile yapılmış geçmiş çalışmalar anlatılmıştır.

Vielma ve arkadaşları, tek ve çok katmanlı CNN ve çift yönlü uzun kısa süreli bellek sinir ağı (BiLSTM) ile hibrit model (CNN-BiLSTM) oluşturmuşlardır. Bu modelde, çekirdek (kernel) boyutunun dokuz olduğu durumda %89,54 doğruluk (Accuracy-ACC) performansı almışlardır [8]. Haque, Rakibul ve Mishu, uzun kısa süreli bellek (LSTM) ve CNN sinir ağlarının ayrı ayrı tekil kullanımı ve LSTM-CNN şeklinde hibrit kullanımıyla duygu analizi için çeşitli deneyler yapmışlardır. Bu deneysel çalışmalarda CNN-LSTM modeli %90 doğruluk değeriyle en iyi sonucu vermiştir [9]. Dang ve arkadaşları, dört adet CNN sinir ağını kullanarak yaptıkları duygu analizi çalışmasında %84,5 doğruluk değeri almışlardır [10]. Gao ve arkadaşları ise üç adet CNN sinir ağını bir arada kullanarak %89,4 doğruluk değeri almışlardır [11]. Zabit ve Begonya da tek katmanlı CNN ile %85,32 doğruluk değeri almışlardır [12]. Qaisar ise iki adet LSTM sinir ağını bir arada kullanarak %89,9 doğruluk değeri almıştır [13]. Amit ve arkadaşları, Global Vektör (Global Vector-GloVe), ön eğitilmiş (pre-trained) metin temsili sonrasında BiLSTM ve CNN sinir ağlarıyla oluşturdukları modelde, %90,05 doğruluk değeri almışlardır [14]. Derra, iki adet BiLSTM sinir ağıyla oluşturduğu modelde %87,46 doğruluk değeri almıştır [15]. Prabha ve arkadaşları, Tekrarlayan sinir ağı (Recurrent Neural Networks-RNN) ile oluşturdukları modelde %87 doğruluk değeri almışlardır [16]. Ni ve Cao, LSTM ve GRU sinir ağı ile yaptıkları modelde %87,10 doğruluk değeri almışlardır [17]. Rathee ve arkadaşları, BOW sonrası DT, KNN, LR, RF, SVM gibi makine öğrenmesi algoritması ile oluşturdukları modelde en iyi sonucu %76 doğruluk değeriyle almışlardır [18].

Wenpeng ve arkadaşları, Hiyerarşik CNN adını verdikleri modelde %87,90 doğruluk değeri almışlardır [19]. Amulya ve arkadaşları, RNN sinir ağı ile yaptıkları duygu analizi çalışmalarında %88 doğruluk değeri almışlardır [20]. Sabba ve arkadaşları sıralı CNN modelleriyle oluşturdukları modelde %89,47 doğruluk değeri almışlardır [21]. Hassan ve Mahmood, CNN ve LSTM sinir ağlarını hibrit şekilde bir arada kullanarak oluşturdukları modelde %88,3 doğruluk değeri almışlardır [22]. Camacho-Collados ve Pilehvar, CNN ve LSTM sinir ağlarının hibrit şekilde bir arada kullanarak oluşturdukları modelde %88,90 doğruluk değeri almışlardır [23]. Ghorbani ve arkadaşları, CNN ve LSTM sinir ağlarının farklı sayıda ve bir arada kullanımıyla CNN-LSTM-CNN şeklinde oluşturdukları model ile %89,02 doğruluk sonucu almışlardır [24]. Nehal ve arkadaşları CNN ve LSTM sinir ağlarının CNN-LSTM şeklinde bir arada kullanımıyla oluşturdukları modelde %89,20 doğruluk sonucu almışlardır [25]. Ma ve Fan, BiLSTM sinir ağıyla %89,29 doğruluk değeri alırken [26] Chen ve arkadaşları, BiLSTM sinir ağıyla %90,58 doğruluk değeri almışlardır [27]. Johnson ve Zhang, LSTM sinir ağı ile oluşturdukları modelde %93,49 doğruluk değeri almışlardır [28]. Zhou ve Feng, makine öğrenmesi yöntemlerinden DT ile %89,16 sonuç alırken [29]; Wang ve Manning, SVM ile %89,16 doğruluk sonucu almışlardır [30]. Kumar ve Harish ise SVM ile %85 doğruluk değeri almışlardır [31]. Nguyen ve arkadaşları, SVM ve NB makine öğrenmesi algoritmalarının bir arada kullanımında %87,95 doğruluk değeri almışlardır [32]. Mohaiminul ve Sultana, RF makine öğrenmesiyle %88,66 değeri alırken [33] Narayanan ve arkadaşları, NB ile %88,80 doğruluk değeri almışlardır [34].

2.2. ROTTEN TOMATOES VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR

Çalışmada kullanılan Rotten Tomatoes [5] veri kümesi üzerinde yapılan geçmiş çalışmalara ait literatür bu bölümde anlatılacaktır.

Tiwari ve arkadaşları, Rotten Tomatoes veri kümesi üzerinde gerçekleştirdikleri duygu analizi çalışmalarında metin temsilleri için n-gram yöntemi ve n-gram'ların çeşitli kombinasyonlarını (Bigram, Unigram, Trigram) kullanmışlardır. Çalışmalarında SVM, Maksimum Entropi (ME) ve NB ile modeller oluşturmuşlar ve sırasıyla unigram, bigram ve trigram yöntemlerinin ME ile daha iyi sonuçlar verdiklerini görmüşlerdir. En iyi sonuç olarak unigram ve bigram sonrası ME ile %89,65 doğruluk değeri alınmıştır [35]. Sahu ve Ahuja, Rotten Tomatoes web sitesinden sekiz bin film yorumu toplayarak veri kümesi

oluşturmuşlardır. Bu veri kümesi üzerinde RF, KNN, NB ve Torbalama (Bagging) ile modeller oluşturmuşlardır. Çalışmalarında, en yüksek doğruluk skorunu %95 ile RF vermiştir [36]. Virginia ve arkadaşları, Rotten Tomatoes film yorumları veri kümesi üzerinden yaptıkları duygu analizi çalışmalarında KNN, NB, DT, SVM gibi çeşitli makine öğrenmesi yöntemleri kullanmışlardır. En yüksek doğruluk değerini %95 ile KNN vermiştir [37].

2.3. TRIPADVISOR VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR

Çalışmada kullanılan TripAdvisor [6] veri kümesi üzerinde yapılan geçmiş çalışmalara ait literatür bu bölümde anlatılacaktır.

Mostafa, TripAdvisor web sitesinden, Mısır'ın Aswan kentindeki beş otel hakkında toplam on bir bin dört yüz elli sekiz yorum toplamıştır. Bu yorumların metin temsili için TF-IDF kullanmıştır. Duygu analizi için oluşturduğu SVM, NB ve DT modelleri arasında en yüksek doğruluk değerini %85 ile NB vermiştir [38]. Dehkharghani ve arkadaşları, bin sekiz yüz elli otel hakkında müşteri tarafından sağlanan yaklaşık iki yüz elli bin incelemenden oluşan TripAdvisor veri kümesi kullanmışlardır. Bu çalışmalarında öznellik temelli özellik çıkarımını önermişlerdir. SVM, NB ve DT üzerinden bir model oluşturmuşlar ve SVM, önerdikleri yöntem yardımıyla %87,51 ile en iyi sonucu vermiştir [39]. Raut ve Londhe, TripAdvisor web sitesinden eğitim kümeleri için beş yüz olumlu ve beş yüz olumsuz yorum ile test kümeleri için beş yüz olumlu ve beş yüz olumsuz olmak üzere toplam iki bin yorum toplamışlardır. SVM, NB ve DT ile bu yorumlar üzerinde duygu analizi için model oluşturmuşlardır. SVM ile oluşturdukları modelde %87 doğruluk değeri elde etmişlerdir [40]. Zhou, yüz bin otel yorumunu %70 eğitim kümesi, %10'u doğrulama kümesi ve %20'lik test kümesine ayırmıştır. Sonrasında duygu analizi için GloVe ve FastText ile metin temsilleriyle beraber BiLSTM ve CNN ile modeller oluşturmuştur. En iyi doğruluk performansı, GloVe metin temsili ve BiLSTM modeliyle %73,73 almıştır [41].

2.4. TWITTER COVID-19 VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR

Çalışmada kullanılan açık kaynaklı ve etiketli Covid-19 hakkında atılan tweetlere ait veriler [7] üzerinde yapılan geçmiş çalışmalara ait literatür bu bölümde anlatılacaktır.

Mostafa, Mısırlı öğrencilerin pandemi ile öğrenme sürecine ilişkin tweetlerine yönelik Word2Vec yöntemi sonrası makine öğrenmesi modelleri kullanarak bir duygu analizi çalışması yapmıştır. Word2Vec metin temsilinden sonra NB, SVM ve DT ile oluşturulan modellerde, NB %91 doğruluk değeriyle en iyi sonucu vermiştir [42]. Machuca ve arkadaşları, 2020 yılında Twitter'dan Covid-19 pandemisi sırasında atılan tweetler üzerlerinde, LR ile yaptıkları duygu analizi çalışmalarında %78,5 doğruluk elde etmişlerdir [43]. Siddiqua ve arkadaşları, Twitter verileri üzerinde SVM, NB ve Sıralı Minimum Optimizasyon yöntemlerinin, MV topluluk öğrenme ile bir arada kullanılmasıyla yaptıkları duygu analizine yönelik deneysel çalışmada %87,7 doğruluk değeri ile en iyi sonucu almışlardır [44]. Rahman ve Islam, İngiltere'deki yaklaşık on iki bin tweet verisi üzerinde duygu analizi için, çeşitli topluluk öğrenmesi yöntemleriyle modeller oluşturmuşlardır. Bu modellerden, Yığınlama ile %83,5; Oylama ile %83,3 ve son model olan Torbalama ile %83,2 sonuç almışlardır [45]. Barkur ve arkadaşları, Covid-19 tweetleri üzerinde TF-IDF metin temsili sonrası, LR ile oluşturdukları modelde %74,29 doğruluk değeri elde etmişlerdir [46]. Isnain ve arkadaşları, Covid-19'un ekonomik ve sosyal etkilerini araştırmak için topladıkları tweetler üzerinde, TF-IDF ve n-gram metin temsil yöntemleri sonrası NB ile oluşturdukları modelde %84 doğruluk değeri elde etmişlerdir [47]. Imran ve arkadaşları, Reddit sitesinden toplanan Covid -19 verileri üzerinde duygu analizi için Dönüştürücülerden Çift Yönlü Kodlayıcı Temsilleri (BERT) ve GloVe metin temsil yöntemleri sonrası RNN ve RNN varyasyonlarıyla oluşturdukları modelde %82,4 doğruluk değeri elde etmişlerdir [48]. Buntoro ve arkadaşları, Covid-19 verileri üzerinde duygu analizi için TF-IDF ve Word2Vec metin temsil yöntemleri sonrası SVM ile oluşturdukları modelde %84,38 doğruluk değeri elde etmişlerdir [49]. Vernikou ve arkadaşları, Covid-19 verileri üzerinde duygu analizi için TF-IDF ve BOW metin temsil yöntemleri sonrası LSTM sinir ağıyla oluşturdukları modelde %65 doğruluk değeri elde etmişlerdir [50]. Antonio ve arkadaşları, Covid-19 verileri üzerinde duygu analizi için TF-IDF metin temsil yöntemi sonrası Stokastik Gradyan İnişi (SGD) ile oluşturdukları modelde %65 doğruluk değeri elde etmişlerdir [51].

3. YAPAY ZEKA

3.1. ZEKA

Yapay zekadan bahsedilebilmesi için öncelikle zekanın ne olduğunu bilinmesi gereklidir. Zeka, öğrenme ve öğrenilenden yararlanabilme, yeni durumlara uyarak yeni çözüm bulabilme yeteneği olarak tanımlanır [52]. Zeka kavramı, zihnin birbirinden farklı birçok işlevini temsil eden bir kavramdır. Düşünme, akıl yürütme, kavrama, nesnel gerçekleri anlayarak sonuç çıkarma bu işlevler arasındadır. Zekanın varlığı için işlevlerin hepsi birbirleriyle uyumlu bir şekilde çalışmalıdır. İnsanlar dışındaki diğer canlılar da zeka sahibidir.

Zekanın beyinde yer aldığı kabul edilmektedir. Yapılan çalışmalara göre bir insan beyni 10 milyardan fazla sinir hücresine sahiptir. Her bir sinir hücresi ortalama on bin hücre ile bağlantılı olarak çalışmaktadır. Nöron adı verilen sinir hücreleri birbirleriyle sinyaller göndererek iletişim kurarlar. Bu sinyaller ile bilginin nasıl depolandığı, nasıl kullanıldığı gibi konular açıklanamamıştır ve beynin tam olarak nasıl çalıştığı hâlâ bilinmemektedir [53].

3.2. AKIL

Zeka, birçok özelliği temsil etmektedir. Bu özellikler, genellikle bir eylem ve uygulamalardan oluşmaktadır. Kuramlar ve kurallar akıl sayesinde ortaya çıkarırken bunların pratikte uygulanmasını zeka sağlar. Düşünebilme yeteneği zeka sayesinde olur. Doğru düşünceye ulaşmak ise akılla olur [54].

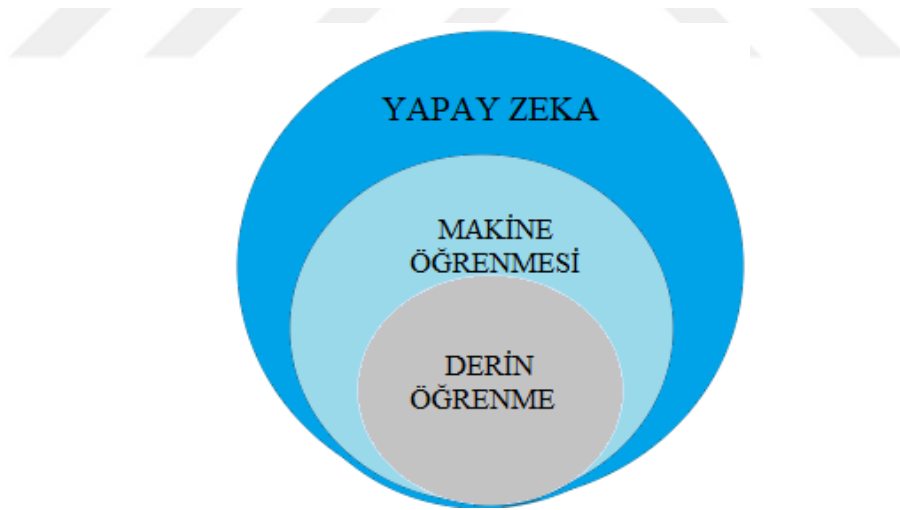
3.3. YAPAY ZEKA

Yapay zeka (Artificial Intelligence) literatürde AI veya YZ olarak kullanılmaktadır. AI, bir bilgisayarın çeşitli eylemleri zeka sahibi bir canlıya benzeyen şekilde veya onu taklit ederek yapması olarak tanımlanabilir. Temel olarak bir yapay zekadan beklenen, kendi kendine öğrenmesi, karar vermesi ve bu kararın sonucuna göre bundan ders çıkararak kendisini geliştirmesidir. Bu işlemleri otonom bir şekilde yapması ve herhangi bir

müdahaleye ihtiyaç duymaması önemlidir. Yapay zeka, dünyayı ele geçiren çok fonksiyonel robotlar akıllara getirirse de, insanların yerine geçmek üzere tasarlanmamıştır. İnsanlara katkı sağlamak üzere çeşitli yazılımlarda kullanılmaktadır. Geliştirilen yapay zeka uygulamalarında, temel hedef bir alanda uzman olmasıdır. Çünkü insan gibi birçok alanda yetkin sistemler tam anlamıyla geliştirilememiş, başarısız olmuşlardır. Alanında uzman yapay zeka uygulamaları birçok farklı sektörde kullanılmakta ve büyük ölçüde kolaylık sağlamaktadır. Yapay zeka alanında çalışmalar devam etmektedir ve bu çalışmalar insan zekasını anlamamıza da katkı sağlamaktadır [54].

3.3.1. Yapay Zeka Nasıl Çalışır?

Yapay zeka programının amacı insan zekasını taklit etmektir ve bu yönüyle diğer yazılımlardan ayrılır. Bunu yapabilmesi için bağımsız bir şekilde öğrenebilmesi ve öğrendiklerini bir problem karşısında çözüme ulaşmak için kullanarak algoritmalar üretmesi gerekir. Bir yazılımın, verileri işleyerek bağımsız bir şekilde öğrenmesi, makine öğrenmesi olarak isimlendirilmektedir. Makine öğrenmesinin kapsamında olan derin öğrenmede ise algoritmalar üretebilmek için kapsamlı yapay sinir ağları kullanılmaktadır. Bu kavramların arasındaki bağ Şekil 3.1.'de gösterildiği gibidir [55].



Şekil 3.1. Yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki [55].

3.4. DOĞAL DİL İŞLEME

Yapay zekanın bir alt dalı olan ve bilgisayarların insan dilini kavraması, üretilmesi ve idare etmesini sağlayan çalışma alanına Doğal Dil İşleme (NLP) denir. NLP insan dilini anlayan, üreten veya idare eden bilgisayarlı oluşumlar inşa etmekle ilgilenen bir mühendislik disiplini [56].

Günlük hayatta farketmeden NLP ile etkileşim kurulmaktadır. NLP teknik ve teknolojileri sayesinde çalışan Siri, Cortana veya Alexa gibi sanal asistanlar ile sorular yöneltilmektedir. Bu asistanların hem kullanıcının talebini anlayıp hem de doğal bir dille yanıt vermesi NLP ile olmaktadır. Web sitesinden arama, istenmeyen e-posta filtreleri, otomatik metin veya konuşma çevirisi, duygu analizi ve dil yazım denetimi NLP uygulama örneklerindedir [56].

1950'li yıllarda şimdiki bilgisayarlara benzer dijital bilgisayarların icadıyla NLP çalışmaları başlamıştır. NLP çalışmalarında dilbilim ve yapay zeka tekniklerinden yararlanır. Ancak son yıllarda verilerden öğrenerek genelleştirme yapan ve yapay zekanın bir dalı olan makine öğrenmesi ve özellikle derin öğrenme kullanılmaktadır. Derin öğrenme, makine öğrenmesinin büyük verilerden örüntüleri öğrenebilen bir türüdür. Web sitelerinden alınan metin verilerinin doğal dil karmaşıklıklarını öğrenmek için en uygun ortamdır [57].

Makine öğrenmesi problemlerinde girdilerin bilgisayarın anlayabileceği dilde olması gerekmektedir. Sayısal olmayan web kaynaklı metin verilerinin, bilgisayarın anlayabileceği hale dönüştürülmesi gerekmektedir. Birçok programlama dilinde NLP yöntemleri ve kütüphaneleri kullanılarak metin verileri bilgisayarın anlayacağı bir biçime dönüştürülür [57].

3.4.1. NLP'nin Çalışma Alanları ve Uygulamaları

NLP, oldukça geniş bir çalışma alanına sahiptir. Metin işleme, bilgi çıkarımı, konuşma tanıma, soru cevaplama, otomatik çeviri, sesten metne çeviri, metinden sese çeviri, başlık sınıflandırma ve duygu analizi gibi birçok alt görev barındırır. NLP'nin, sohbet robotları (chatbot), arama motoru optimizasyonu, sosyal medya analizi, içerik yönetme ve müşteri kayıp analizi gibi birçok uygulama alanı vardır. Bu alanları listeleyecek olursak [57];

- Arama motoru optimizasyonu: NLP, web sayfalarının içeriğini optimize etmek üzere arama motorlarında analiz sonucunda sayfanın daha üstte görünmesine yardımcı olur. NLP teknikleriyle web sayfaları çok yüksek görünürlük elde ederler.
- Sohbet Robotları: gelişmiş NLP algoritmalarıyla donatılmıştır ve kullanıcıların sorularını anlamak, yanıtlamak ve uygun şekilde tepki vermek için doğal dil işleme tekniklerini kullanır. NLP, sohbet robotlarının metinleri anlaması ve kullanıcılarla etkileşim kurması için gereken temel teknolojileri sağlar. Büyük

firmaların müşterileriyle ilişkilerini güçlendirmek adına özellikle son zamanlarda yatırım yapmaktadırlar. Ön Eğitimli Üretici Dönüştürücü Sohbet Robotu (Chat Generative Pre-Trained Transformer-ChatGPT) sohbet robotlarının en son geliştirilen popüler örneğidir. OpenAI firması tarafından geliştirilmiş ve Önce Eğitimli Üretken Dönüştürücüler (Generative Pre-Trained Transformer-GPT) modelinin bir versiyonudur [57].

- Sosyal medya analizleri: NLP müşteri incelemelerini ve sosyal medya yorumlarını analiz ederek yüksek hacimli veriler konusunda bilgi edinilmesini sağlayabilir. Bu alanda yapılan çalışmaların çoğunluğu duygu analizi çalışmalarıdır.
- Pazar rekabet analizi: İşletmeler müşterileriyle nasıl iletişim sağlayabileceklerini analiz etmek üzere NLP'den faydalanmaktadırlar. Müşteri ilgisine yönelik ürünlerin sosyal medya veya web sayfalarındaki yorumları üzerinde duygu analizi gibi NLP görevleriyle çıkarım yaparlar. Bu çıkarımlarıyla ürün tasarımı ve pazarlama için doğrudan uygulanabilir bilgiler edinirler [57]

Tez çalışmasında bu görevlerden duygu analizine ait çalışma gerçekleştirilmiştir. Bu çalışma kapsamında yapılan deneysel çalışmalar alt başlıklarda anlatılacaktır.

3.4.2. NLP'nin Kullanıldığı Endüstriler

NLP çok çeşitli iş süreçlerinde, özellikle e-postalar, anketler, sosyal medya yorumları gibi yüksek hacimli yapılandırılmamış metinlerde işlem yapmayı sağlar. Çeşitli endüstrilerde, NLP sayesinde metin verilerini analiz edilerek doğru karar alınabilir. Örneğin; sağlık alanında, hasta kayıtlarını analiz etmek ve bunlara ilişkin yeni eğilimler elde etmek için NLP'den faydalanabilir. Hukukta, bir davaya hazırlanan avukat çok karmaşık ve büyük dosyalar incelemek ve belirli bir vaka ile ilgili materyalleri aramak zorunda kalmaktadır. NLP teknolojisi yasal keşif sürecini otomatikleştirilir ve büyük hacimlerde belgeleri tarayarak hem harcanan zamanı hem de insan hatalarını azaltabilir. Finans alanında, NLP'den faydalanarak portföylere yönelik alım satım kararları ile ilgili bilgileri çıkarmak için kurumsal belgelerden ve haber bültenlerinden otomatik olarak bilgi taraması yapılabilir. Müşteri hizmetleri alanında ise küçük, büyük kurum fark etmeden müşteri ilişkileri yönetimini ilgilendiren konular başta olmak üzere, sohbet robotları yardımıyla müşteri sorularını ve taleplerini yanıtlayabilirler. Buna ek olarak çeşitli sosyal medya platformları üzerinde kurumları hakkında yapılan yorumlara yönelik NLP teknikleriyle duygu analizi

gibi çalışmalar yapılabilir [58].

Tez çalışmasında, IMDB ve Rotten Tomatoes web sitelerinden; Twitter, TripAdvisor gibi sosyal medya platformlarından toplanmış veri kümeleri üzerinde duygu analizi çalışmaları yapılmıştır. Burada geçen tüm endüstrilere ait firma veya ürünler için de sosyal medya verileri toplanarak NLP teknolojileriyle ön işleme adımları sonrası duygu analizi yapılabilir.

3.4.3. NLP Teknolojileri

Bu bölümde NLP'nin sınıflandırma modellerine yönelik kullanılan makine öğrenmesi, derin öğrenme ve ön eğitilmiş modeller ve transfer öğrenme ile metin ön işleme teknikleri anlatılmıştır.

3.4.3.1. Makine Öğrenmesi Modelleri

Makine öğrenmesi modelleri, bir veri kümesindeki örnekleri genelleştirerek tahminler yapar. Bu veri kümesine eğitim verileri adı verilir ve makine öğrenmesi algoritmaları, bir hedef görevi gerçekleştiren makine öğrenmesi modelini oluşturmak üzere bu eğitim verilerinden eğitilir. Örneğin, duygu analizi cümlelerden olumlu, olumsuz ve nötr başta olmak üzere çeşitli duyguların çıkarımına dayanır. Bir makine öğrenmesi algoritması bu veri kümesindeki cümleleri girdi olarak alıp duygu sınıflarını çıktı olarak veren bir model üretir. Cümleleri girdi olarak alıp, bu girdi için bir etiket çıktısı getiren bu tür modellere belge sınıflandırma modeli adı verilir. Belge sınıflandırıcıları, spor, finans gibi bahsettikleri konulara göre sınıflandırılabilir [59].

Makine öğrenmesi algoritmalarıyla oluşturulan bazı modeller ise belgelerdeki varlıkları tanımak ve sınıflandırmak için kullanılabilir. Bu modeller, belgedeki her bir sözcük için ilgili sözcüğün bir varlık ifadesi olup olmadığını ve varlık ifadesiyse ne tür bir varlıktan bahsedildiğini tahmin eder. Örneğin, "ABC Şirketinin hisseleri bugün 100 ABD Doları karşılığında işlem görmektedir" cümlesinde "ABC Şirketi" bir şirket varlığı, "100 ABD Doları" bir para birimi miktarı ve "bugün" ise bir tarihtir. Varlık tanıma için eğitim verileri her bir sözcüğün, hangi tür varlıklarla ilgili ise o şekilde etiketlendiği bir metin külliyatıdır. Girdideki her bir sözcük için bir etiket üreten bu tür modellere dizi etiketleme modeli adı verilir [59].

Diziden diziye (seq2seq) modelleri, NLP'de güncel kullanılan bir modeldir. Diziden diziye tüm belgeyi, belge sınıflandırıcıda olduğu gibi girdi olarak alır. Ancak çıktı olarak

bir cümle veya başka bir dizi üretir. (Belge sınıflandırıcı ise çıktı olarak yalnızca tek bir sembol üretir.) Seq2seq modellerinin örnek uygulamaları; İngilizce bir cümleyi girdi olarak alıp çıktı olarak Fransızca cümlesini getiren makine çevirisi, belge özetleme (çıkıtı, girdinin bir özetidir) ve semantik ayrıştırma (girdi, İngilizce bir sorgu veya taleptir ve çıktı ise talebi uygulayan bir bilgisayar programıdır) olarak verilebilir [58].

Bu bölümde anlatılan makine öğrenmesi modelleri tez çalışması kapsamında duygu analizi için modeller oluştururken kullanılmıştır.

3.4.3.2. Derin Öğrenme, Ön Eğitimli Modeller ve Transfer Öğrenme

Derin öğrenme, NLP' de en yaygın kullanılan makine öğrenmesi türüdür. 1980'li yıllarda araştırmacılar, yapay sinir ağlarını geliştirmişlerdir. Yapay sinir ağlarında, çok sayıda temel makine öğrenmesi modeli tek bir ağda birleştirilmiştir. Beyin ile benzer bir şekilde basit makine öğrenmesi modelleri "nöronlar" olarak isimlendirilmektedir. Bu nöronlar katmanlar halinde düzenlenir. Bir derin sinir ağı birçok katman içerir. Derin öğrenme, derin sinir ağı modellerini kullanan makine öğrenmesi yöntemidir [58].

Karmaşıklıkları nedeniyle bir derin sinir ağının eğitimi için büyük veri ve bu verilerin işlenmesi için yüksek bilgi işlem gücü ile işlem zamanı gerekir. Derin sinir ağıyla oluşturulan NLP modelleri, Vikipedi gibi web kaynaklı metin verileriyle eğitilir. Eğitim verileri, gigabyte'lar seviyesinde veya daha büyük boyutta olabilir. Derin sinir ağının yüksek performanslı bir kümede eğitimi için uzun süre gerekebilir [59]. Araştırmacılar, daha büyük veri kümelerinden ve derin sinir ağlarıyla oluşturulan modellerin eğitilmesinin daha yüksek performans gösterdiğini ortaya koymuşlardır. Dolayısıyla daha büyük veri kümelerinden daha büyük modellerin eğitilmesi konusunda bir yarış söz konusudur [59].

Derin sinir ağları kapsamlı veri ve bilgi işlem gücü gerektirir. Bu yüzden derin sinir ağı kullanımı ile çalışmak için bunlar göz önüne alınmalıdır. Ancak transfer öğrenme, eğitilmiş bir derin sinir ağının, yeni bir görevi gerçekleştirmek üzere çok daha az eğitim verisi ve bilgi işleme yüküyle daha fazla eğitilebilmesini sağlar. En basit transfer öğrenme türü, ince ayar (fine-tuning) olarak adlandırılır. Temelde önce modeli Vikipedi gibi büyük genel bir veri kümesinde eğitip sonrasında modelle ilgili NLP görevi için etiketli çok daha küçük bir göreve özgü veri kümesinde eğitme işlemidir [59]. İnce ayar veri kümeleri, yüzlerce ve hatta onlarca eğitim örneğinden oluşacak şekilde son derece küçük olabilir. Bu sayede ince ayar eğitimi, tek bir CPU'da dakikalar içinde çalıştırılabilir. Transfer

öğrenme, derin öğrenme modellerinin farklı platform ve çalışmalara dağıtımını kolaylaştırmıştır [57].

Son zamanlarda, farklı diller ve veri kümeleri için ön eğitim kombinasyonlarıyla eğitilmiş derin öğrenme modelleri sunan kapsamlı sağlayıcılar vardır. Bu ön eğitilmiş modeller indirilerek, çok çeşitli hedef görevler için ince ayar uygulanabilir [57].

Tez çalışma kapsamında IMDB veri kümesi üzerinde duygu analizi için önerilen hibrit derin öğrenme modelinde ön eğitilmiş GloVe ve BERT metin temsili kullanılmıştır. Bu ön eğitilmiş metin temsilleriyle oluşturulan hibrit modele ait deney sonuçları ön eğitilmiş yöntemler başlığında anlatılmıştır.

3.4.3.3. Ön İşleme Teknikleri

Bu bölümde, tez çalışmasında kullanılan ön işleme teknikleri anlatılacaktır. Aşağıda NLP aracılığıyla yapılan ön işleme teknikleri listelenmiştir [60]:

- Belirteçlere ayırma: Belirteçlere ayırma (Tokenization) ham metni (bir cümleyi veya belgeyi) sözcükler ya da alt sözcük parçaları gibi belirteç dizisine ayırır. Belirteçlere ayırma tekniği, genellikle NLP veri ön işleme işlemlerindeki ilk adımdır. Belirteçler, daha sonraki işleme adımında atomik birimler olarak kabul edilen, yaygın olarak yinelenen metin dizileridir. Bunlar sözcükler, biçimbirim adı verilen alt sözcük birimleri ("na-" gibi önekler ya da "-yor" gibi sonekler), hatta bağımsız karakterler olabilir.
- Atlanacak sözcük çıkarma: "Atlanacak sözcük" veya "durak sözcük" (Stop Word) daha sonraki süreçlerde yok sayılan belirteçlerdir. Tipik olarak "bir", "ya" gibi tekrarlanan kısa sözcüklerdir. Sözcük torbası modelleri ve arama motorları, genellikle işleme süresini ve veri tabanında depolamayı azaltmak için atlanacak sözcükleri yok sayar. Derin sinir ağları, tipik olarak sözcük sırasını göz önünde bulundurur ve atlanacak sözcükler anlamda ufak ayrımlar yaratabileceği için atlanacak sözcükleri çıkarmaz. Örneğin, "paket kayboldu" ve "bir paket kayboldu" cümleleri atlanacak sözcük çıkarıldıktan sonra aynı görünseler de aynı anlamı taşımaz.
- Kök alma ve kök çözümü: Biçimbirimler dilin en küçük anlamlı öğeleridir. Tipik olarak biçimbirimler sözcüklerden küçüktür. Örneğin, "namümkündü" sözcüğü ön ek olarak "na-", kök olarak "mümkün" ve son ek olarak da geçmiş

zaman ekini "-idi" içerir. Kök alma ve kök çözümlemede (stemming ve lemmization) sözcükler kök biçimlerine eşlenir ("namümkün" + geçmiş). Kök alma ve kök çözümleme, derin öğrenme modellerinde kritik adımlardır ancak derin öğrenme modelleri genellikle bu düzenleri eğitim verilerinden öğrenir ve bu nedenle açık kök alma veya kök çözümleme adımları gerekmez.

Bu çalışmada kullanılan tüm veri kümeleri, İngilizce metinlerden oluşmaktadır. Bu veri kümelerinin ön işleme adımları, Python'da bulunan Doğal Dil Araç Takımı (NLTK) [59] ile gerçekleştirilmiştir. NLTK ile metinlerdeki her kelimenin ekleri kaldırılıp köklere indirgenmiş, noktalama işaretleri, Hiper Metin İşaretleme Dili (HTML) etiketleri gibi özel karakterler kaldırılmıştır. Ayrıca sayılar, durak kelimeleri ve fazladan bırakılan boşluklar (White spaces) da kaldırılmıştır.

3.4.4. NLP'de Kullanılan Programlama Dilleri

Bu bölümde NLP'de kullanılan programlama dilleri hakkında bilgi verilmiştir.

3.4.4.1. Python

Python programlama dilinde, NLP ile ilgili çok sayıda hazır araç ve kütüphane yer almaktadır. Bu yüzden NLP projelerinde genellikle Python kullanılmaktadır. Nesneye dayalı programlamaya yatkın ve üst seviye dil olması sebebiyle Java, Go, JavaScript, Php, C++ gibi diller de tercih edilmektedir. Ayrıca Python dili gibi hazır araçlar ve kütüphanelere sahip Julia ve R gibi diller de sıklıkla tercih edilmektedir [60]. Bu çalışmada Python programlama dilinin 3.10 sürümü kullanılmıştır.

3.4.5. NLP Geliştirme Ortamları

Bu bölümde NLP görevlerine yönelik projelerde kullanılan yazılım altyapılarından bahsedilecektir.

3.4.5.1. TensorFlow ve PyTorch

Tensorflow [61] ve Pytorch [62] sıklıkla kullanılan ücretsiz iki derin öğrenme çerçevesidir. Python dilinin yanında farklı diller aracılığıyla da kullanılabilir. Önceden yazılan hazır bileşenlerden oluşan büyük kütüphaneler, bu altyapının içerisinde yer alır. Grafik işlemci birimi (GPU), hızlandırıcılar içeren yüksek performanslı bilgi işlem altyapılarını destekler [63]. Tez çalışmasında Tensorflow 2.7.0 versiyonu kullanılmıştır.

3.4.5.2. *HuggingFace*

Çok sayıda ve birbirinden farklı ön eğitilmiş, çoğunluğu derin öğrenmeye dayalı NLP modelleri barındıran bir girişimci ekip tarafından sunulan bir ortamdır. Bu ortam, NLP görevleri üzerine çalışacak araştırmacılara hızlı bir şekilde yol gösteren ve kullanıma hazır bir yazılım sağlar [64].

Önceden eğitilmiş BERT modeli için BERT-based ve BERT-large adında iki mimari mevcuttur. Bu mimariler, iki milyon beş yüz bin kelimenin yer aldığı İngilizce Vikipedi dahil olmak üzere geniş etiketsiz metin topluluğu üzerinde önceden eğitilmiştir [65]. IMDB veri kümesinde üzerinde ön eğitilmiş yöntem deneylerinden olan BERT-CNN ve BERT sonrası MBiGRUMCONV çalışmalarında, on iki katman dönüştürücü bloğu, on iki dikkat başlığı (attention head), yedi yüz altmış sekiz gizli katman ve yüz on milyon parametre bulunan BERT-base kullanılmıştır. Bert dışında kullanılan ön eğitilmiş yöntemlerden ikincisi GloVe'dur. GloVe, Stanford araştırmacıları tarafından önceden eğitilmiş 6 milyar kelime bulunmaktadır [66].

IMDB veri kümesi üzerinde yapılan ön eğitilmiş yöntem deneyinde, GloVe' un 100, 200, 300 boyutu ile Google tarafından geliştirilen BERT yöntemi kullanılmıştır. IMDB veri kümesi üzerinde yapılan bu çalışmaların sonuçları, ön eğitilmiş yöntem deney başlığı olan (5.1.5)'te anlatılmıştır.

3.4.5.3. *Spark NLP*

Spark NLP; Python, Java ve Scala programlama dilleri için gelişmiş açık kaynaklı metin işleme kütüphanesidir. Doğal dil işleme için uygulama program arabirimi sağlar. Ön eğitilmiş sinir ağı modelleriyle bazı özel hazırlanmış modellere erişerek, veri kümelerini eğitme desteği sunmaktadır [67].

3.4.5.4. *SpaCy NLP*

NLP için Python ile geliştirilmiş ücretsiz ve açık kaynaklı bir kütüphanedir. Büyük hacimli metinleri işleyebilen ve anlayabilen uygulama yazılımlarının oluşturulmasına yardımcı olmak için tasarlanmıştır. SpaCy, NLP projelerinde yaygın olarak kullanılır [67].

3.4.6. **Duygu Analizi**

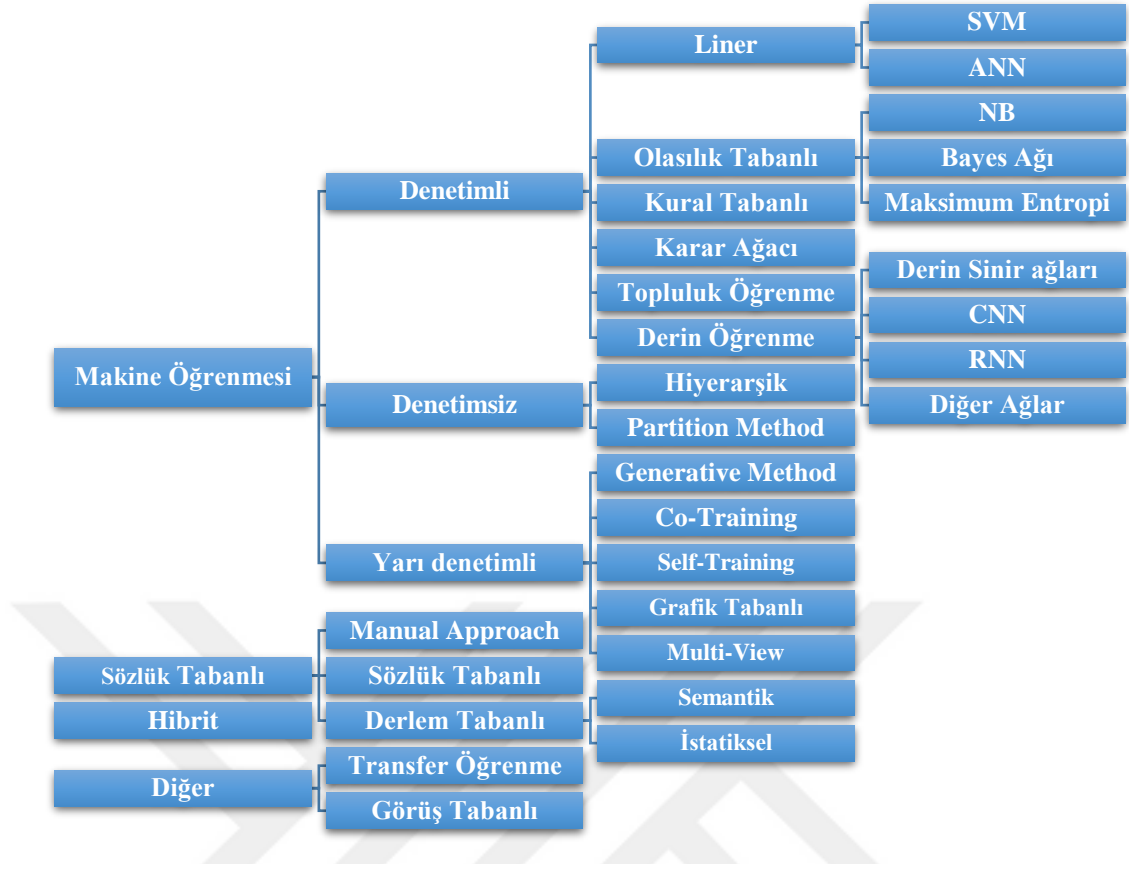
Duygu analizi, uygulama alanlarına bağlı olarak, duyguya dayalı, duygu çıkarma, duygu madenciliği, öznel analiz ve etki analizi gibi birkaç adla anılmaktadır. Ayrıca fikir

madenciliği ve duygu analizi birbirinin yerine kullanılabilir. Duygu, görüş ve fikir terimleri de birbirinin yerine kullanılabilir [68].

Duygu analizi, makine öğrenmesi ve sözlük tabanlı yaklaşımlar olarak iki kategoriye ayrılabilir. Metinlerden duygu çıkarma çalışmalarında çeşitli düzeyler bulunmaktadır. Bu düzeyler; Belge, Cümle, Görüş ve Kavramdır. Kavram [69], Belge [70] ve Görüş [68] düzeyindeki yaklaşımlar için sözlük tabanlı yöntemler kullanılmaktadır. Cümle düzeyinde duygu analizi çalışmalarında, duygu sınıflandırmasındaki varlıkların özelliklerini etkilemez. Bunun için görüş tabanlı (aspect-based) [65]-[67] yöntemler kullanılır. Görüş tabanlı yöntemler, metindeki duygu bütünlüğünü dikkate alarak verileri küçük parçalara bölerek daha detaylı bilgi sağlar [71]. Önce duygu sınıflandırması yapılır, bununla birlikte sadece duygu tanımlama çalışmaları da yapılır. Mutluluk, öfke, sevinç gibi duygular; olumlu, olumsuz ve nötr sınıflardan ziyade daha ayrıntılı olarak belirlenebilir. Duygu algılama özelliğinin kullanıldığı uygulamalarda bu duygular sayesinde kullanıcının tutumu olumlu ya da olumsuz olarak incelenebilmekte ve içinde bulunduğu duygusu analiz edilerek çeşitli aksiyonlar alınabilmektedir [72].

Şekil 3.2'de görüldüğü gibi duygu analizi yaklaşımları, makine öğrenmesi yaklaşımı, sözlük tabanlı yaklaşım ve bunların hibrit kullanımının yanında diğer çalışmalar olmak üzere dört ana başlık altında genelleştirilebilir [73].

Çalışmamızda çeşitli veri kümeleri üzerinde makine öğrenmesi, topluluk öğrenme, derin öğrenme ve derin öğrenme algoritmalarının hibritlenmesi olmak üzere dört farklı yöntem de kullanılmıştır.



Şekil 3.2. Duygu analizi yaklaşımları [74].

Duygu analizi ile yapılan ilk çalışmalarda TF-IDF ve BOW gibi frekans tabanlı metin temsil yöntemlerinden sonra makine öğrenmesi yöntemleri kullanılarak duygu sınıflandırması için DT, LR RF, SGD, SVM, Bernoulli Naïve Bayes (BNB) ve Multinomial Naïve Bayes (MNB) kullanılmıştır. Sonrasında Word2Vec, GloVe gibi tahmine dayalı yöntemler kullanılmaya başlanmıştır. BERT ve gelişmiş dönüştürücülerin ön eğitilmiş modellerindeki duygu analizi görevi dahil olmak üzere NLP'nin diğer görevlerinde kullanılmıştır.

Duygu analizinde denetimli makine öğrenmesi algoritmalarından LR, MNB, RF, SVM; derin öğrenme algoritmalarından RNN, LSTM ve CNN [75], [76] gibi yöntemler de popüler olarak çalışılmaktadır. Duygu analizi çalışmalarında yeni trendler olarak bahsedilebilecek nöro-sembolik AI [77], hızlı girdi (prompt) tabanlı analiz [77] – [79] ve grafik tabanlı [79], [80] yöntemler de son zamanlarda çalışılmaktadır.

Çalışma kapsamında IMDB [4], Rotten Tomatoes [5], TripAdvisor [6] ve Twitter Covid-19 [7] veri kümeleri üzerinde çeşitli makine öğrenmesi, topluluk öğrenme ve derin öğrenme yöntemleriyle duygu analizi deneyleri yapılmıştır.

4. MATERYAL VE METOT

Bu bölümde, Metin Madenciliği ve metin önışleme, metin temsil yöntemleri, makine öğrenmesi, derin öğrenme ve topluluk öğrenmesi algoritmalarına ait bilgiler anlatılmıştır.

4.1. METİN MADENCİLİĞİ VE METİN ÖNİŐLEME

Metin Madenciliği (Text Mining), doğal dil işleme teknikleriyle metinlerden duygu başta olmak üzere çeşitli çıkarımlar yapılma sürecidir. Diğer bir ifadeyle, düzensiz elektronik metin verilerinden düzenli ve yapısal bilgi elde etme sürecidir. Bu süreçte, metinler üzerinden yapılandırılmış veri elde etmek amaçlanmaktadır. Örneğin; metinlerin kümelenmesi, sınıflandırılması metinlerden ilgili olduğu alan tespiti, duygu analizi, metin özetleme ve varlık ilişki modellenmesi gibi çalışmalar hedeflenmektedir [79].

Metin ön işleme, metinler üzerinde metin madenciliğinin daha kolay yapılabilmesi için yapılan hazırlık sürecidir. İstenmeyen kelimelerin ve özel karakterlerin kaldırılması gibi süreçleri içerir. Verilerin, NLP görevleri öncesinde hazırlaması için kullanılır [81].

Bu tez çalışmasında NLP görevlerinden duygu analizi için farklı veri kümeleriyle çeşitli modeller oluşturulmuştur. Bu modellerde daha iyi performans almak için çeşitli metin madenciliği işlemleri kullanılmıştır. Bu işlemler [82]:

- HTML etiketlerinin kaldırılması,
- URL ve email gibi özel ifadelerin kaldırılması,
- Sayıların kaldırılması,
- Kelimeler arası fazla boşlukların kaldırılması,
- Noktalama işaretlerin kaldırılması,
- Belirteçlere ayrılması,
- Kelimelerin tek bir formatta olması için tüm metin verilerinin küçük harfe çevrilmesi,
- Durak kelimelerinin kaldırılması,

- Metinlerin vektörleştirilmesidir.

Metin vektörleştirme işlemleri için, tahmine dayalı, frekans tabanlı ve ön eğitilmiş metin temsil yöntemleri bu çalışmada kullanılmıştır. Bu yöntemlere ait bilgiler bu bölümde anlatılmıştır. Etiket (hashtag), emoji ve emoji işaretleri (smiley) gibi gösterimlerin kaldırılması çalışmanın bilinen kısıtıdır ve çalışma kapsamında devre dışı tutulmuştur.

4.2. METİN TEMSİL YÖNTEMLERİ

Metin temsil yöntemleri diğer bir isimlendirmeye dil modelleri, metinlerin bir bağlamda anlamlandırmaya çalışan kelime grupları üzerinde bir olasılık dağılımıdır. Belirli bir kelime dizisinin bir cümlede meydana gelme olasılığını belirlemek için çeşitli istatistiksel ve olasılıksal teknikler kullanılmaktadır. Dil modelleri, metin sınıflandırma, makine çevirisi, gibi çeşitli NLP uygulamalarında kullanılır [83].

Metin verilerini işleyebilmek için öncelikle verilerin makineler tarafından anlaşılır hale getirilmesi gerekir. Bunun için metin temsil yöntemleri kullanılır. Metin temsil yöntemleri bir dil modelleme tekniğidir. Bu teknik ile sözcükler sayısallaştırılarak birer vektör haline getirilir ve vektör uzayında temsil edilir. Bu temsil yöntemine verilen diğer isimde metin vektörleştirme [84].

Metin temsili yöntemleri aşağıdaki gibi sınıflandırılır;

- Frekans tabanlı yöntemler,
- Tahmin (yapay sinir ağları) tabanlı yöntemler,
- Dönüştürücü modelleridir.

4.2.1. Frekans Tabanlı Yöntemler

Çalışmada kullanılan, kelimelerin cümle içinde geçmesine dayalı yöntemler; TF-IDF ve BOW'dur. Bu bölümde TF-IDF ve BOW anlatılacaktır.

4.2.1.1. Terim Frekansı-Ters Belge Frekansı

TF-IDF, belirli bir belgedeki kelimelerin göreceli sıklığını ve veri kümesindeki bu kelimelerin tersine oranını belirleyerek çalışır. Bir terimin belgedeki önemini gösteren istatistiksel yöntemle hesaplanan ağırlık faktörüdür [85].

TF, Luhn'un [86], IDF ise Jones'un [87] çalışmalarıyla ortaya çıkardıkları iki ayrı

yöntemdir.

TF, bir kelimenin belgede kaç defa geçtiğini belirtir. Bir kelimenin belgede geçme sayısı, belgedeki toplam kelime sayısına bölünerek bulunur. Diğer bir tanımlamayla bir belgedeki terim ağırlıklarını hesaplamak için kullanılan yöntemdir. Denklem (4.1)'de TF yöntemine ait denklem görülmektedir.

$$TF(i, j) = \frac{j \text{ belgesinde } i \text{ terim sayısı}}{j \text{ belgesindeki toplam kelime}} \quad (4.1)$$

IDF ise toplam belge sayısının, ilgili kelimenin geçtiği belge sayısına bölünmesiyle elde edilir. Bu işlem sonucunda çıkan değer sıfıra yaklaştıkça kelimenin birçok yerde geçtiği anlaşılır. Bir değerine yakınsa da daha az geçtiği anlaşılır. Bu da ilgili kelimenin belgede önem derecesi olan IDF değerini gösterir. IDF, Denklem (4.2)'de görülmektedir [88]. TF-IDF skoru, Denklem (4.3)'te verildiği gibi hesaplanır.

$$IDF(i) = \log \left(\frac{\text{Toplam belge}}{\text{belgede geçen } i \text{ terimi}} \right) \quad (4.2)$$

$$j = TF(i, j) * IDF(i) \quad (4.3)$$

Tez çalışmasında, TF-IDF yöntemi için Python sci-kit learn [89] kütüphanesinin 1.0.2 sürümü kullanılmıştır. TF-IDF yöntemine ait parametreler Çizelge 4.1'de verilmiştir [90]:

Çizelge 4.1. TF-IDF parametreleri.

Parametre	Alınacak Değer	Açıklama
max_count	Tam sayı değer girilmelidir	Veri kümesinde en çok geçen kaç kelimenin alınacağına yönelik parametre

4.2.1.2. Kelime Torbası

BOW, metin işlemede yaygın olarak kullanılan frekans tabanlı yöntemdir. Bu yöntemde belgelerin kelime sırası korunmaz, sadece kelime sayıları dikkate alınır [91]. Belgelerdeki kelimelerin sıklığını gösteren BOW, sınıflandırıcı tarafından bir dizi özelliğe sahip bir öğrenme modeli oluşturmak için kullanılır. Bu yöntem ile veri kümesindeki tüm kelimelerin ve bu kelimelerin kaç kez geçtiği öğrenilir. BOW ile metin temsili için kullanılan parametreler sırasıyla aşağıda yer alan Çizelge 4.2'de verilmiştir [90]:

Çizelge 4.2. BOW parametreleri.

Parametre	Alınacak Değer	Açıklama
max_count	Tam sayı değer girilmelidir	Veri kümesinde en çok geçen kaç kelimenin alınacağına yönelik parametre

4.2.2. Yapay Sinir Ağı Tabanlı Metin Temsil Yöntemleri

Yapay sinir ağı tabanlı yöntemlerde, metinler için bir kelime gömme (word embedding) uygulanır. Kelime gömme, bir cümledeki her kelimenin d boyutlu bir vektörle eşleştirildiği, böylece yakın anlamsal benzerliğe sahip kelimelerin gizli uzayda birbirine daha yakın temsil edildiği bir yöntemdir. Bu, düşük boyutlu bir uzayda cümlenin daha fazla bağlamını taşıyan kelime çantasından daha gelişmiş bir tekniktir [92].

Sinir ağları kullanarak en popüler metin temsil tekniklerinden biri Word2Vec'tir. Tomas Mikolov ve ekibi tarafından geliştirilmiştir. Word2Vec'in Gram Atlama (Skip-Gram) ve Sürekli Kelime Torbası (Continuous Bag of Words-CBOW) olmak üzere iki yöntemi vardır. Mikolov [92] ve ekibinin geliştirdiği kelime gömme yöntemi olan Word2Vec'in, üzerinde yapılan çalışmalarla GloVe, fastText gibi yeni yöntemler geliştirilmiştir.

4.2.2.1. Word2Vec

Word2Vec, 2013 yılında Thomas Mikolov ve ekibi tarafından, temelinde yapay sinir ağı ile iki farklı model kullanarak geliştirilmiştir. Word2Vec, CBOW ve Skip-Gram olmak üzere iki yöntemden oluşmaktadır [92].

4.2.2.1.1. CBOW

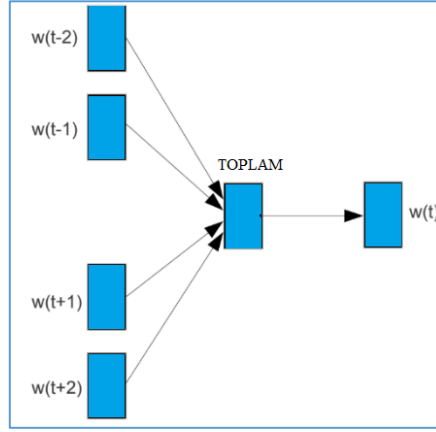
CBOW yöntemi, merkez kelimenin etrafındaki kelimelerden merkezi kelimeyi tahmin etmeye çalışmaya dayanır. CBOW yöntemi küçük veri kümelerinde daha iyi tahmin yapmaktadır. Öte yandan birden fazla anlamı olan kelimeleri anlamakta zorlanır [85].

CBOW yöntemi, geçmiş kelimelerin yanı sıra gelecekteki kelimelerde de kullanılan bir mimaridir. CBOW için amaç fonksiyonu Denklem (4.4)'te verilmiştir [93]:

$$J_{\theta} = \frac{1}{T} = \sum_{t=1}^T \log p(w_t | w_{t-n}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+n}) \quad (4.4)$$

CBOW yönteminde, pencerenin ortasındaki kelimeyi tahmin etmek için bağlamın dağıtılmış temsilleri kullanılır [94]. CBOW yönteminin yapısına ilişkin görsel, Şekil

4.1’de gösterilmektedir [92].



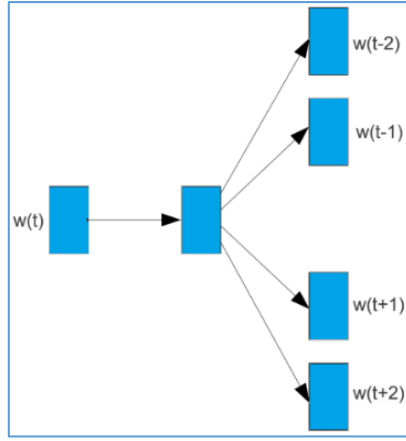
Şekil 4.1. COW yöntemi [92].

4.2.2.1.2. Skip-Gram

Skip-Gram yönteminde, pencere boyutu merkezinde olan kelimeler girdi olarak alınıp pencere boyutu merkezinde olmayan kelimeler çıktı olarak tahmin edilmeye çalışılmaktadır. COW’da olduğu gibi merkez kelimeyi tahmin etmek için çevreleyen kelimeleri kullanmak yerine, bu kelimeleri tahmin etmek için merkezi kelime kullanılır. Skip-Gram amaç fonksiyonu, Denklem (4.5)’te verilen Skip-Gram için hedefi üretmek üzere, hedef kelimenin solundaki ve sağındaki n kelimeyi çevreleyen n'nin logaritmik olasılıklarının toplamıdır [92]:

$$J_{\theta} = \frac{1}{T} = \sum_{t=1}^T \sum_{t=1}^T \log p(w_{j+1} | w_t) \quad (4.5)$$

Skip-Gram yöntemi, bir kelimenin etrafında bulunan kelimeleri tahmin etmeye çalışır. Bu yöntem, daha büyük verilerle daha iyi çalışmaktadır. Aynı zamanda Skip-Gram yöntemi, COW’a göre iki veya daha fazla anlamı olan kelimeleri anlamakta daha başarılıdır. Skip-Gram yönteminin yapısına ilişkin görsel Şekil 4.2’de gösterilmektedir [94].



Şekil 4.2. Skip-Gram yöntemi [92].

Word2Vec'in eğitim aşamasında belirtilmesi gereken bazı parametreleri vardır. Bu parametreler sırasıyla Çizelge 4.3'te verilmiştir [90].

Çizelge 4.3. Word2Vec parametreleri.

Parametre	Alınacak Değer	Açıklama
sentences	ön işlemeden geçirilmiş kelime listesi	Modelin eğitilmesi için kullanılan veri kümesidir.
size	integer	Kelime vektörlerinin boyutu
window	integer	Bir cümle içindeki mevcut ve tahmin edilen kelime arasındaki maksimum mesafedir.
min_count	integer	Parametre değerinden daha düşük frekanstaki kelimeler yok sayılır.
sg	integer	CBOW yöntemiyle eğitim için 0, Skip-Gram yöntemiyle eğitim için 1 kullanılır.

Çalışmalarda, Çizelge 4.4'te görülen parametrelerden Word2Vec için vektör boyutu 100, 200, 300, pencere boyutu (window) beş, veri kümesinde geçen minimum kelime sayısı (Min-count) beş, olarak alınmıştır. IMDB, Covid-19, Rotten Tomatoes ve TripAdvisor veri kümelerine yönelik olan çalışmaların hepsinde Word2Vec kullanılmıştır. Her veri kümesine ait Word2Vec yöntemine ilişkin kullanılan parametre detayları deneysel çalışmalar ve sonuçları başlığı altında anlatılmıştır. Tüm veri kümelerine yönelik yapılan deneylerde kullanılan Word2Vec parametrelerinin tamamı Çizelge 4.4'te verilmiştir.

Çizelge 4.4. Çalışmada kullanılan Word2Vec parametreleri.

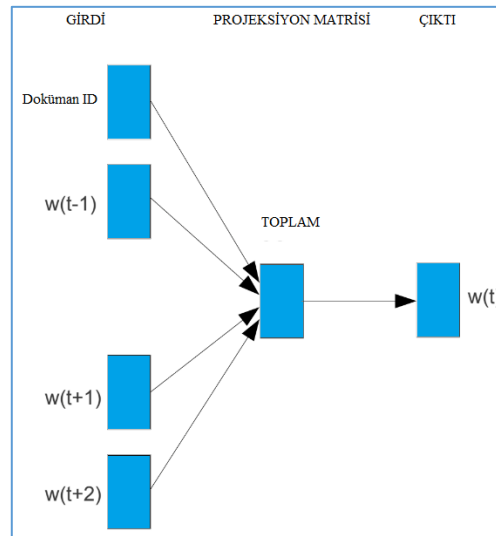
Parametre	Değerler
Size	100, 200, 300
Window	5
Workers	8
min_count	5
sg	0 veya 1

4.2.2.2. Doc2Vec

Doc2Vec, Word2Vec'e dayanır ve Doc2Vec'in ana motivasyonu, belgelerin sayısal bir temsilini oluşturmaktır. Artık sözcüklerin, belgelerdeki mantıksal yapıyı bozmasını çözmek için geliştirilmiştir. Bunun için, Word2Vec modeline başka bir vektörün belge numarası (Document ID) eklenmesi gerekir. Word2Vec ve Doc2Vec arasındaki tek fark budur [95]. Paragraf Kimliği, benzersiz belge kimliği olarak ifade edilebilir. Word2Vec gibi Doc2Vec'in de iki yöntemi vardır. Bunlar; Paragraf Vektörlerinin Dağıtılmış Bellek Modeli (PV-DM) ve Paragraf Vektörünün Dağıtılmış Kelime Çantası yöntemleri (PV-DBOW)'dir [96].

4.2.2.2.1. Paragraf Vektörlerinin Dağıtılmış Bellek Modeli

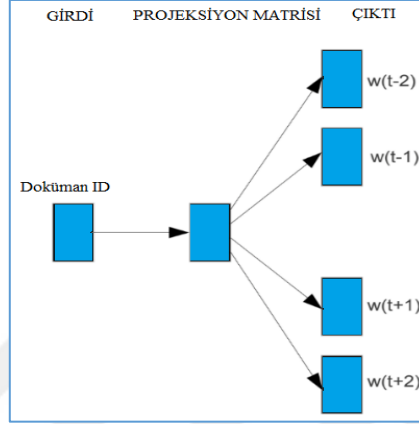
PV-DM, Word2Vec'teki CBOW modeline benzer. Bu model, bir paragraf kimliği [96] ekleyerek komşu sözcüklerden (bağlam sözcükleri) çıktıyı (hedef sözcük) tahmin etmeye çalışır. DM modeli Şekil 4.3'te gösterilmiştir.



Şekil 4.3. PV-DM yöntemi [92].

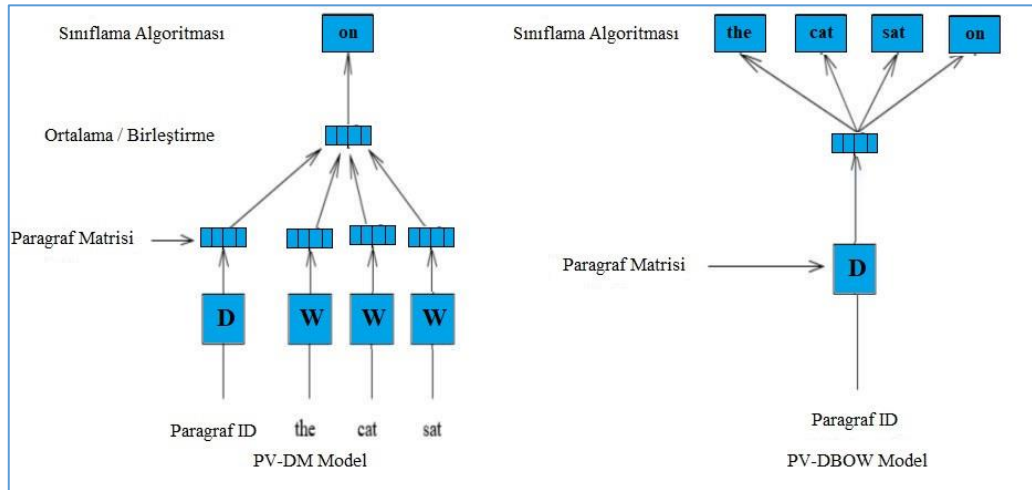
4.2.2.2.2. Paragraf Vektörünün Dağıtılmış Kelime Çantası Modeli

PV-DBOW, Word2vec'in Skip-Gram yöntemine benzer çalışır. PV-DM ile arasında tek bir fark vardır. Hedef kelimeyi girdi olarak kullanmak yerine, PV-DBOW girdi olarak belge numarası alır ve tahmin etmeye çalışır [95]. PV-DBOW modeli Şekil 4.4'te gösterilmiştir.



Şekil 4.4. PV-DBOW yöntemi [92].

Doc2Vec modeli, Word2Vec mimarisini kullanarak her bir doküman için ayrı bir doküman vektörünün eklenmesi ile yaratılmıştır. Bu doküman vektörleri ile derlemdeki dokümanlar uzunluklarından bağımsız bir şekilde vektör uzayında sayısal değerlerle temsil edilmektedirler [97]. PV-DM ve PV-DBOW Şekil 4.5'te beraber gösterilmiştir [98].



Şekil 4.5. PV-DM ve PV-DBOW [96].

Çalışmalarda Doc2Vec, sadece Covid-19 veri kümesinde kullanılmıştır. Bu veri

kümesinde 100, 200 vektör boyutunda hem PV-DM hem de PV-DBOW ile ayrı ayrı deneysel sonuçlar alınmıştır. Bu veri kümesine ait deneysel çalışmalar (5.2.2)'de Covid-19 Twitter veri kümesi üzerinde yapılan çalışmalar başlığı altında verilmiştir.

4.2.2.3. GloVe

Glove'de, amaç fonksiyonu olarak ağırlıklı en küçük kareler toplamı benimsenmiştir. Amaç fonksiyonundaki bu fark Word2Vec metin temsilinden farklıdır. Bu yaklaşımda, kelimelerin beraber görülme oranları başlangıç noktası olarak alınmıştır. Bu şekilde oluşturulan modeller daha iyi başarımlar göstermiştir [66].

Skip-Gram ve CBOW'da, kelimelerin anlamsal birlikteliğine bakılır. Fakat kelimelerin birlikte kullanılma istatistiklerini bakılmaz. Pennington ve arkadaşları tarafından kelimeler arası anlamsal ilişkileri yakalamak için "GloVe" modeli geliştirilmiştir. Bu modelde, olasılık istatistiklerinden yararlanarak Denklem (4.6)'da yer alan yeni bir amaç fonksiyonu oluşturularak bu problemi çözmüşlerdir [66].

$$J = \sum_{i,j=1}^v f(X_{i,j})(w_i^T w_j + b_i + b_j - \log X_{i,j})^2 \quad (4.6)$$

Denklem (4.6)'da; $X_{i,j}$ sözlükteki kelime çiftinin birlikte geçme sayısıdır. $f(X)$ ağırlık fonksiyonunun sahip olacağı gereksinimler listelenecek olursa [66]:

- $f(0) = 0$ durumunda hiçbir terim sonsuz değerine sahip olmamalıdır.
- Aynı konumda beraber geçen kelime ikilileri için ağırlık fonksiyonu azaltıcı olmamalıdır.
- $X_{i,j}$ 'nin değerleri daha küçük olmalıdır.

Çalışmada, Stanford araştırmacıları tarafından önceden eğitilmiş 6 milyar kelime bulunan GloVe [66]'un 100, 200, 300 vektör boyutunda metin temsil işlemi gerçekleştirilmiştir. IMDB veri kümesi üzerinde GloVe sonrası yapılan deneysel çalışmaların sonuçları (5.1.5.3)'te verilmiştir.

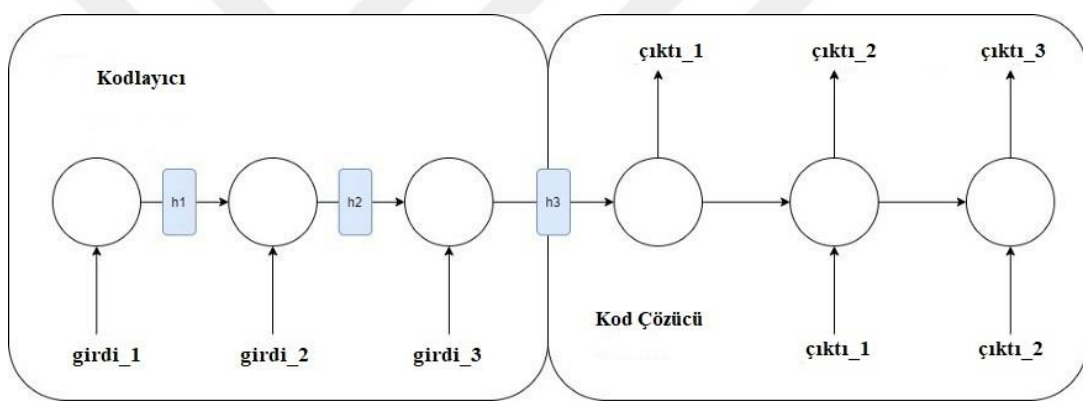
4.2.2.4. Tekrarlayan Sinir Ağı

RNN sıralı verileri işlerken kullanılır. Her bir kelime sırayla RNN'e verilir ve çıktı olarak bir vektör elde edilir. Elde edilen bu vektör ile sonraki kelimenin vektörü tekrar RNN'e girdi olarak verilir. Bu işlem cümle sonuna kadar devam eder. Sadece önceki kelimelere

baktıkları için, RNN'ler bağlam oluşturmada iyi sonuçlar vermezler. Ayrıca RNN sinir ağı eğitilirken donanımı çok optimize kullanamazlar. Bu problemlere çözüm bulan Dönüştürücü dil modelleri geliştirilmiştir [91].

4.2.2.5. RNN ve Dikkat

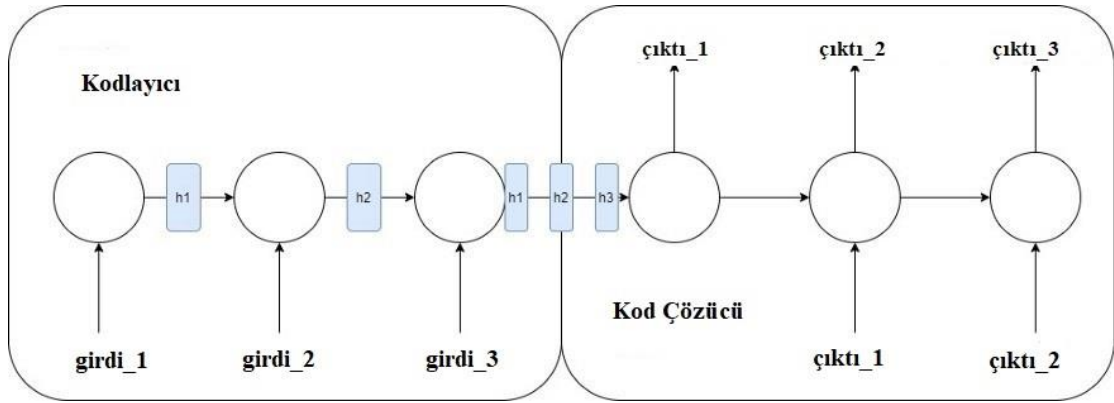
RNN'lerde kullanılan Kodlayıcı-Kod Çözücü (Encoder-Decoder) yapılarının gelişmesiyle, makine dili çevirisi, metinlerin sınıflandırılması gibi NLP görevlerinde alınan sonuçlar yeni bir boyut kazanmıştır [65]. Kodlayıcı ve Kod Çözücü, birer RNN yapısıdır. Bu yapıda öncelikle kelime gömme adı verilen metin temsilleri oluşturulmaktadır. Kodlayıcı bölümünde RNN eğitilir ve bu eğitimin çıktısı olan gizli katman, Decoder'a aktarılmaktadır. Decoder'dan üretilen çıktı, probleme göre değişkenlik gösterebilmektedir. Şekil 4.6'da Kodlayıcı-Kod Çözücü mimarisi gösterilmektedir [65].



Şekil 4.6. Kodlayıcı-Kod Çözücü mimarisi [65].

Şekil 4.6'da verilmiş olan Kodlayıcı-Kod Çözücü mimarisi, NLP görevlerinde çok başarılı performanslar alınmasını sağlamıştır. Fakat bu mimarinin de bazı sorunları vardır. Çok uzun girdi verilince mimaride sorunlar oluşmaktadır. Örneğin, NLP görevlerinden makine çevirisi görevinde yüz kelimenin yer aldığı bir cümle farklı bir dile çevrildiği bir senaryoda cümledeki ilk kelimeyi hatırlama problemine bağlı olarak önemi azalmaktadır. Bu senaryodaki problem LSTM'nin sinir ağını "unutma kapıları" ile çözülebilmektedir. İkinci çözüm yoluysa, Kodlayıcı'da yer alan bilginin bir vektörle ifadesini hatırlama probleminin tamamen olmasa da kalkmasına yardımcı olan Dikkat mekanizmasıdır. Dikkat mekanizması, RNN'nin yaptığı gibi çıktıyı (gizli katmanı) Decoder'a yollamak yerine, tüm gizli katmanları beraber gönderir [99]. Bu durumda ilk ve son kelime olmadan cümlede yer alan kelimelerin önemi korunmuş olur. Bu sayede bilgi bütünlüğü sağlanır. Decoder'da, yer alan bilgi için her seferinde çıktı matrisi oluşturulur. Bu matris Şekil

4.7’de görüldüğü gibi gizli katmanlarla beraber ilgili çıktıyı oluşturmaktadır [100].



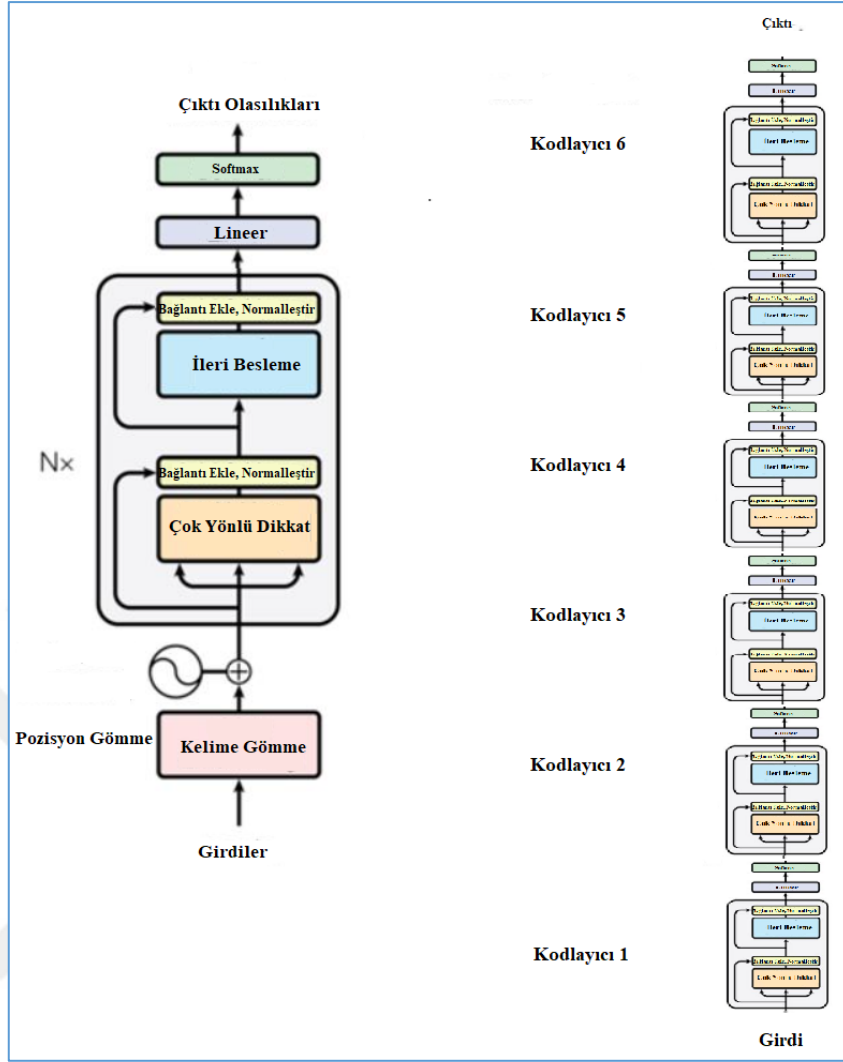
Şekil 4.7. Gizli katmanlardaki bilginin Kodlayıcı-Kod Çözücü mimarisinde işlenmesi [65].

4.2.2.6. Dönüştürücü Metin Temsil Yöntemi

Waswasi ve arkadaşları [100] Dönüştürücü (Transformer) adı verilen ve Dikkat ağına dayanan yeni bir metin temsil yöntemi geliştirmişlerdir. Dönüştürücü’lerde, metin temsili işlemleri paralel şekilde gerçekleştiği için eğitim zamanı da azalır.

Şekil 4.8’de verilen, dönüştürücü mimarisi, altışar tane Kodlayıcı ve Kod Çözücü’den oluşan kapalı bir sistemdir.

Bu mimaride kullanılan altı Kodlayıcı ve altı Kod Çözücü birebir aynı yapıdadır. Fakat en altta yer alan ve Kodlayıcı’ya gelen cümleyi kelime vektörlerine dönüştüren kelime gömme aralarındaki tek farktır. Ayrıca cümlelerdeki kelimelerin konumlarını sayısal olarak koda çeviren pozisyon gömme (position-embedding) olarak isimlendirirken bir parametre yer alır. Bu vektör de kelime gömme vektörü ile aynı uzunluktadır. Bu sayede kelime gömme vektörleriyle toplanan kelimelerin konum bilgisi bilinmiş olur [100].



Şekil 4.8. Dönüştürücü mimarisi [93].

Şekil 4.8’de yer alan görselde, kodlayıcıya gelen cümlelerdeki kelimeler vektörleştirilerek öz-dikkat (self-attention) ismi verilen dikkat mekanizmasında, cümle içinde yer alan kelimelerin aralarındaki anlamsal ilişki çıkarılır. Bu mekanizma ile, anlamsal ilişkisi çıkarılacak kelime dışındaki bütün kelimeler için ilişki skoru çıkarılarak anlamsal ilişki varsa bulunur. Bu skorlar birbiri yerine kullanılabilecek kelimeleri de ortaya çıkarmaktadır.

Dikkat mekanizması ile ilişki hesaplamasının yanında farklı hesaplamalarda yapılmaktadır. Bu hesaplama çok yönlü dikkat (Multi-Headed Attention) şeklinde isimlendirilen kelimelerin farklı yönlerinden de bakılması sağlanmaktadır [101]. Bu işlemler paralel şekilde gerçekleştirilerek oluşturulan çıktılar birleştirilip, bütün dikkat mekanizmalarında oluşturulan bilgi ileri beslemeli sinir ağına (Feed Forward Neural Network-FFNN) gönderilir.

Kodlayıcı-Kod Çözücü mimarisinde, bileşenlerin çıktıları bir sonraki bileşenin girdisi olmak dışında, çıktılarıyla birleşerek normalize edilen başka bir bileşene de girdi olarak ilerler. Kod Çözücü bileşeninde, öz-dikkat ve FFNN bileşenleriyle birlikte Kodlayıcı-Kod Çözücü dikkat (attention) adında bir bileşen bulunur. Bu bileşen, Kodlayıcı'dan gelen değerler arasından önemli olduğunu düşündüğü değerlerle ilgilenir. Kod Çözücü bileşenin çıktısı, bir sonraki Kod Çözücü'nün öz-dikkat bileşenine girdi olarak verilir ve bir sonraki adımda, Kodlayıcı'dan gelen değerlerle birleşerek işlemine devam eder. En üstteki Kod Çözücü'nü çıktısı, Liner ve Softmax katmanlarına iletilir. Liner kısmı, tam bağlı bir sinir ağı mimarisi olan tam bağlı sinir ağı ile birbirine bağlı olan bileşenlerin çıktılarını kullanarak Logit Vektörü'nü oluşturur. Bu vektör, eğitim veri kümesindeki özgün kelime sayısı ile aynı uzunluğa sahiptir ve her bir kelime için bir skor taşır. Softmax kısmında ise bu skorlar kullanılarak her bir kelime için çıktıyı belirleyen olasılık değerleri hesaplanır. En yüksek olasılığa sahip olan kelime, çıktının olduğu zaman birimi olarak belirlenir [65].

Günümüzde en popüler ve en başarılı dil modelleri dönüştürücüler üzerine inşa edilmiştir. Google'ın BERT modeli ve OpenAI'nın GPT-3'ü dönüştürücü üzerine inşa edilen en popüler örneklerdendir [100]. Ayrıca bu modeller, hangi girdilerin birbiriyle ilişkili olduğunu, hangi girdilerin önemli veya önemsiz olduğunu Dikkat adı verilen bir mekanizma kullanarak anlamlandırmaya çalışır. Aynı zamanda bütün girdileri aynı anda alıp RNN gibi bir önceki vektöre ihtiyaç duymadığı için paralelleşebilir ve donanımı daha optimize kullanabilir [65].

Transformer'ların karmaşık mimarisi nedeniyle eğitilmesi uzun zaman almaktadır. Fakat genellikle diğer sinir ağı yöntemlerinden çok daha iyi performans göstermektedir. BERT, ROBERTA, gibi çeşitli Transformer'lar geliştirilmiştir.

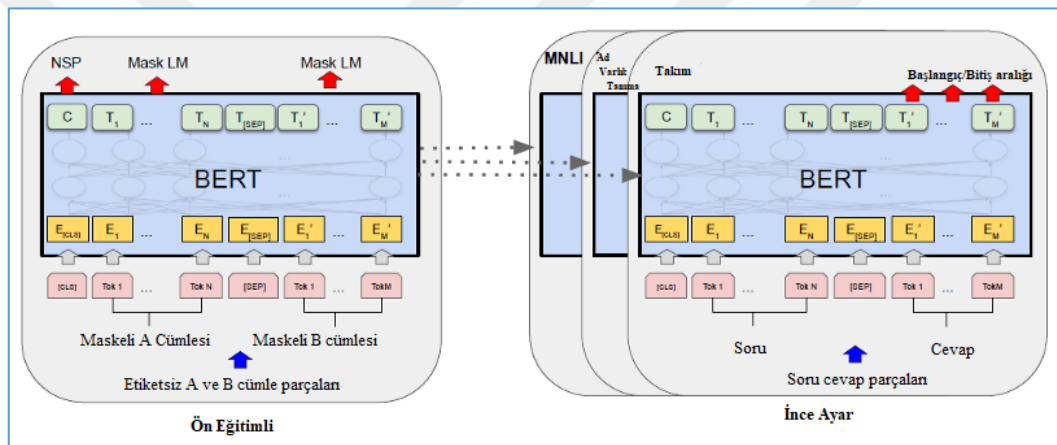
4.2.2.6.1. Dönüştürücülerden Çift Yönlü Kodlayıcı Temsilleri

Google, 2018 yılında Dönüştürücülerden Çift Yönlü Kodlayıcı Temsilleri (Bidirectional Encoder Representations from Transformers -BERT) adı verilen bir modeli duyurmuştur. BERT, diğer modellerden farklı olarak cümleleri hem soldan sağa hem de sağdan sola doğru değerlendirerek kelime ilişkilerini ve anlamlarını daha iyi vektörleştirebilen bir yapıya sahiptir. Bu özelliği sayesinde daha yüksek performanslı sonuçlar elde etmek mümkün olmaktadır.

BERT, BookCorpus adında sekiz milyon kelime ve Vikipedi veri kümesi gibi iki buçuk

milyon kelimeye sahip veri kümesi kullanılarak eğitilmiştir. BERT_large ve BERT_base olarak adlandırılan iki temel model sunulmuştur. BERT_large modeli on altı adet TPU ile, BERT_base modeli ise dört adet TPU ile dört gün boyunca eğitilmiştir. Google, BERT_base'i OpenAI modeli mimarisine benzer bir şekilde oluşturarak performans karşılaştırması yapmak amacıyla geliştirdiklerini belirtmiştir.

BERT, kendi başına bir model olup, Birden Çok Probleme Kullanılabilen (GLM) bir yapıya sahiptir. Ancak chatbot veya metin sınıflandırma gibi problemleri çözmek için BERT üzerine ekstra katmanlar eklenmesi gerekmektedir. BERT'in eğitimi, Maskeli Dil Modellemesi (MLM) ve Sonraki Cümle Tahmini (NSP) adı verilen iki teknikle gerçekleştirilir. Şekil 4.9, BERT'in çift yönlü olmasının yanı sıra, MLM ve NSP teknikleriyle nasıl eğitildiğini göstermektedir.



Şekil 4.9. BERT ön eğitimli dönüştürücü [65].

Şekil 4.9'da gösterilen yöntemlerde, bir cümle modele girdiğinde, cümledeki kelimelerin %15'i MLM tekniği kullanılarak maskelenir. Bu maskelenen kelimelerin %80'i "[MASK]" belirteciyle değiştirilirken, %10'u rastgele başka bir kelimeyle değiştirilir ve geri kalan %10'u ise değiştirilmeden bırakılır. Bu seçim, orijinal makalede ayrıntılı bir şekilde açıklanmıştır [90]. MLM tekniğinde, maskelenen kelimeler açık şekilde modele verilir ve model bu kelimeleri tahmin etmeye çalışır. İkinci teknik NSP ise cümleler arasındaki ilişkiye odaklanır. Eğitim sırasında ikili olarak gelen cümle çiftlerinde, ikinci cümlelerin ilk cümlelerin devamı olup olmadığı tahmin edilir. Bu önceden ikinci cümlelerin %50'sinin rastgele değiştirildiği ve %50'sinin aynı şekilde bırakıldığı bir yaklaşımın yerine kullanılmıştır. Eğitim sürecinde, bu iki teknik kullanılarak elde edilen kaybın minimize edilmesi için optimizasyon işlemi gerçekleştirilir. Önceden eğitilmiş BERT modelleri açık kaynak olarak paylaşılmaktadır. Büyük veri kümesiyle bir dönüştürücü

modelinin eğitilmesi maliyetli olduğundan, bu ön eğitilmiş modeller ince ayar adı verilen bir teknikle yeni problemlerin çözümü için kullanılmaktadır [101].

Ön eğitilmiş modeller, dil veya görüntü işleme gibi diğer problemler için dil özelliklerini sıkıştırılmış bir şekilde sağlayarak daha karmaşık problemlerin çözümünde kullanılmaktadır. Örneğin, bir bilgisayarın çaldığı gitar seslerini üreten bir model oluşturulabilir ve bu model ince ayar yöntemiyle bir insanın çaldığı gitarın notalarını üretebilen bir modele dönüştürülebilir.

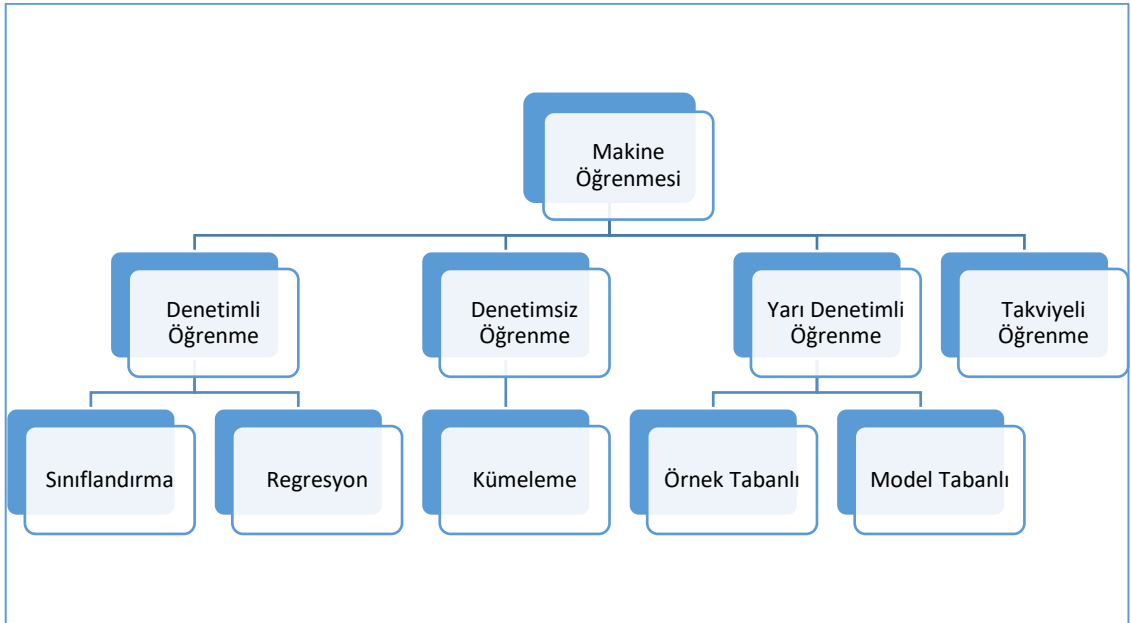
Bu teknik uygulanırken, modele probleme bağlı olarak ekstra bir katman eklenir. Ancak modelin önceden eğitilmiş olduğu bilgilerin korunabilmesi için, bazı bağlantıların değerleri değiştirilmez. Çünkü bir modelde girdi kısmından çıktı kısmına doğru ilerledikçe, öğrenilen özelliklerin karmaşıklığı artar. Örneğin, bir görüntü işleme modeli nesne kenarlarını algılayamıyorsa, oval şekilleri veya dörtgenleri algılaması da mümkün olmayacaktır. Bu nedenle, ön eğitilmiş modellerin ince ayar yöntemiyle yeni problemlere uyarlanması, daha yüksek seviyede özellikleri anlamalarını sağlar. Bu sayede, belirli bir alanda daha karmaşık ve görev özgü problemlerin çözümünde daha iyi performans elde edilebilir [65].

RNN’de kelime bütünlüğü korunmaktadır. Fakat model girdisi uzun olması durumunda uzak kelimeler anlamız görünerek bu kelimeler önemsiz şekilde modele işlenir[102]. Dikkat mekanizmasıyla, bu problem tamamen olmasa da doğru öğrenmeye katkı vermiştir [91]. Dönüştürücü yöntemindeyse paralel şekilde kelimelerin anlamsal ilişkilerinin hesaplamalarının yanı sıra çok yönlü dikkat mekanizması sayesinde cümlenin soldan-sağa ve sağdan-sola şeklinde eğitilmesi sayesinde cümle bağlamına hakim bir model oluşturulur [101]. BERT’te çift taraflı olduğu için kelimelerin öncesi ve sonrasına hakim bir yöntemdir. BERT ile MLM ve NSP ile cümle içeriği hakim bir öğrenme gerçekleştirilmektedir [103], [104].

Tez çalışmasında, BERT-CNN ve BERT sonrası MBiGRUMCONV çalışmalarında, BERT gibi ön eğitilmiş dönüştürücü modeli kullanılmıştır. BERT ile IMDB veri kümesi üzerinde deneysel sonuçlar alınmıştır. BERT sonrası CNN ile sınıflandırma sonucu Bölüm 5.1.5.5’te ve BERT sonrası MBiGRUMCONV sonucu ise Bölüm 5.1.5.4’te verilmiştir.

4.3. MAKİNE ÖĞRENMESİ

Makine öğrenmesi ilk olarak 1943'te Walter Pitts ve Warren McCulloch tarafından tanıtılmıştır [105]. McCulloch, insan beyninde ve sinir ağlarında karar verme sürecini haritalandıran matematiksel bir makale yayınlayan sinirbilimcidir. Makalede, beyindeki her nöron basit bir dijital işlemciye, beyin ise bir bilgi işlem makinesine benzetilmiştir. 1950'de ise matematikçi ve bilgisayar bilimcisi Alan Turing, Turing testini tanıtmıştır. Makine öğrenmesinin ortaya çıkışı ise 1980'lere dayanmaktadır [106]. Makine öğrenmesi modellerinde öğrenme yöntemleri gözetimli öğrenme ve gözetimsiz öğrenme olarak ikiye ayrılmaktadır. Gözetimsiz öğrenmede veriler etiketlenmemiştir. Model verilerini keşfederek bir sonuca varır. Gözetimli öğrenmede ise veriler etiketlenmiştir. Hem girdi hem de çıktı verileri belirtilmiştir. Makine öğrenmesi modellerinin kullanım alanları çok geniştir. Regresyon, sınıflandırma gibi problemlerde sıklıkla kullanılmaktadır. Şekil 4.10'da makine öğrenmesinde kullanılan öğrenme yöntemleri verilmiştir.



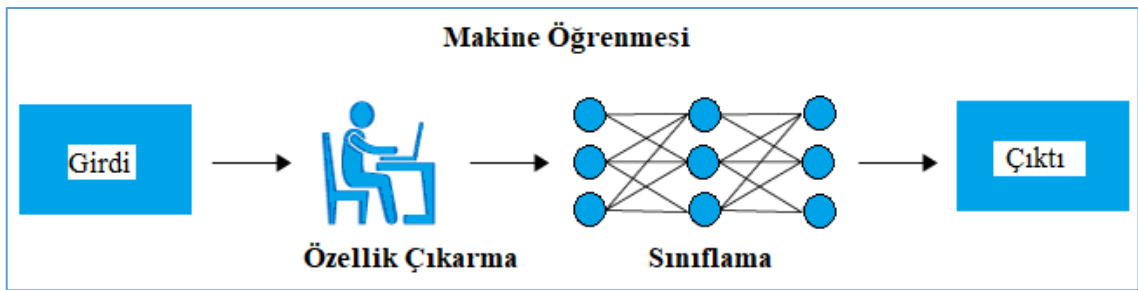
Şekil 4.10. Makine öğrenmesi [107].

Regresyon, denetimli öğrenme yaklaşımlarındandır. Sürekli değişkenleri modelleme ve tahminleme için kullanılabilir. Doğrusal regresyon algoritmasının uygulama örnekleri, gayrimenkul fiyatının tahmini, satış tahmini, borsadaki hisse senedi fiyat hareketlerinin tahmini olabilir. En basit regresyon şekli, veri kümesine düz bir çizgi sığdırılmaya çalışılan ve veri kümesinin değişkenleri arasındaki ilişkinin doğrusal olduğu zaman kullanılabilen bir yöntemdir [108]. Yarı denetimli öğrenme, çok fazla etiketlenmemiş veri

ile etiketlenmiş verinin birlikte kullanılmasıyla oluşturulan öğrenme yöntemidir.

Takviyeli öğrenme, bir ödül-ceza sistemi bulunmasıyla diğer öğrenme yöntemlerinden ayrılmaktadır. Bu öğrenmeden istenilen, tahmini yaparken hatalarından çıkarım yaparak doğruyu bulmasıdır. Bu yöntemler, bunun için ödül ceza sistemi kullanırlar.

Sınıflandırma teknikleri, iki ya da daha fazla kategoriye sahip verilerin çıktısını tahmin eden bir yöntemdir. Örneğin kişinin sağlıklı olup olmadığı, kullandığı Telekom firmasını bırakma durumu gibi tahminlerde kullanılabilir. Tez çalışmasındaysa insanların sosyal medya ve web sitelerinde yorum olarak yazdıkları yorumların olumlu, olumsuz duygu belirtmesi üzerine sınıflandırma işlemi yapılmıştır.

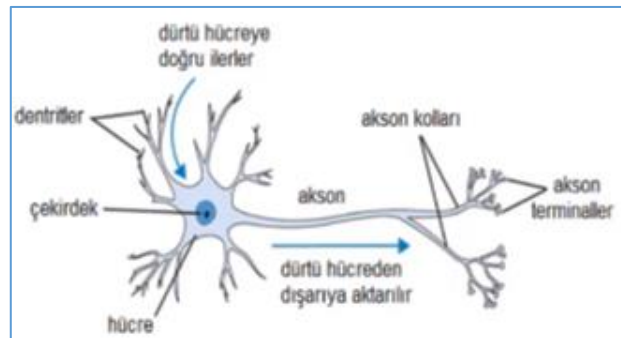


Şekil 4.11. Makine öğrenmesi süreci [107].

Şekil 4.11’de görüldüğü üzere sınıflandırma veya kümeleme işlemi öncesinde öznelilik çıkarma işlemi ek uğraş gerektirmektedir. Metin verilerinde öznelilik çıkarma işleminde çeşitli metin temsil yöntemleri kullanılmaktadır. Yapay sinir ağları (Artificial Neural Networks-ANN), NB, RF, KNN, SVM gibi çalışmada kullanılan makine öğrenmesi algoritmaları bu bölümde anlatılacaktır.

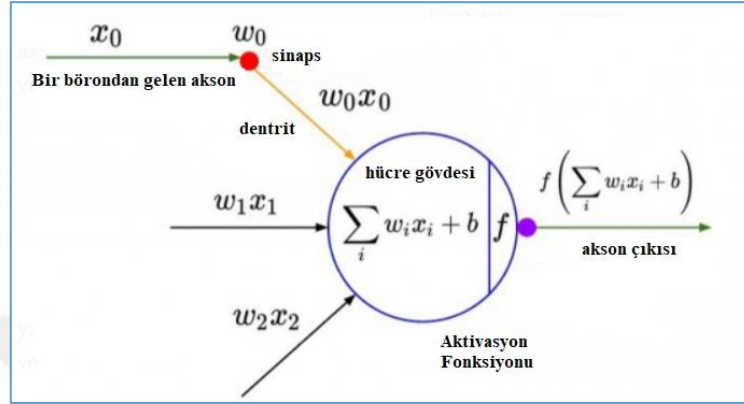
4.3.1. Yapay Sinir Ağları

ANN, insan beyninin çalışmasından esinlenilerek geliştirilmiştir. ANN, insan beyninde bulunan sinir ağlarını taklit eder [109]. Şekil 4.12’de biyolojik sinir ağı görülmektedir.



Şekil 4.12. Biyolojik sinir ağı [107].

Şekil 4.12’de görüldüğü üzere biyolojik sinir ağında, çekirdekten bir akson boyunca iletim yapılmaktadır. Burada çıkış terminallerinde dentrit uçlarından elde edilen algılayıcı verileri çekirdekte ağırlıklandırılarak akson boyunca iletilmektedir. Başka sinir ağlarına bağlanarak sinirler arası iletişim sağlanmaktadır [74]. İnsandaki bir sinir hücresinin matematiksel modeli Şekil 4.13’te gösterilmiştir.



Şekil 4.13. Yapay sinir ağının matematiksel modeli [107].

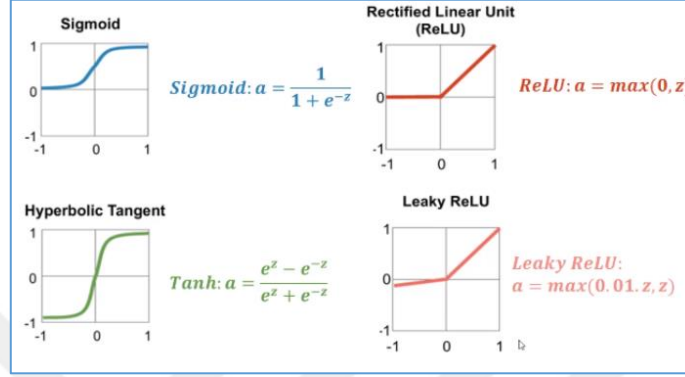
Dentrit denilen yollar boyunca ağırlıkları mevcuttur ve bu dentritlere giren bir başka nöronun da gelmiş olabileceği bir giriş değeri mevcuttur (x_0). Giriş değeri ve dentritteki ağırlık (w_0) çarpıldıktan sonra ($w_0 x_0$) sinir hücresine iletilir ve tüm dentritlerden gelen ağırlık ile giriş çarpımları toplanır. Yani ağırlıklı toplama işlemi yapılır. Ardından bir bias (b) ile toplandıktan sonra aktivasyon fonksiyonundan geçirilerek çıkışa aktarılır. Bu çıkış nihai çıkış olabileceği gibi bir başka hücrenin girişi olabilir. Matematiksel olarak ağırlıklar ile girişler çarpılır artı bir b eklenir. Böylelikle basit bir matematiksel model elde edilir [91], [109].

ANN’de yapılan temel işlem; modelin en iyi skoru vereceği ağırlık parametresi (w) ve b parametrelerinin hesabını yapmaktır.

Her bir sinir hücresi aynı şekilde hesaplanır ve bunlar birbirine seri ya da paralel şekilde bağlanır. Bir yapay sinir hücresi beş bölümden oluşmaktadır [110]. Bu beş bölüm listelenecek olursa:

- Girdiler: Yapay sinir hücresinde yer alan nöronlara verilen ilk verilerdir. Bu veriler toplanmak üzere çekirdeğe gönderilir.
- Ağırlıklar: Yapay sinir hücresine gelen veri girdiler üzerinden sinir hücresi çekirdeğine gelmeden önce bağlantı ağırlıklarıyla çarpılır. Üretilen çıktılar bu sasyede ayarlanabilir.

- Toplama Fonksiyonu: Sinir ağı hücresinin net girdisini hesaplayan bir fonksiyondur.
- Aktivasyon fonksiyonu: Bir önceki katmandan girdilerin ağırlıklı toplamını alarak bir çıkış değeriyle sonraki katmana geçmesini sağlayan fonksiyondur. Şekil 4.14'te popüler olarak kullanılan aktivasyon fonksiyonları verilmiştir.

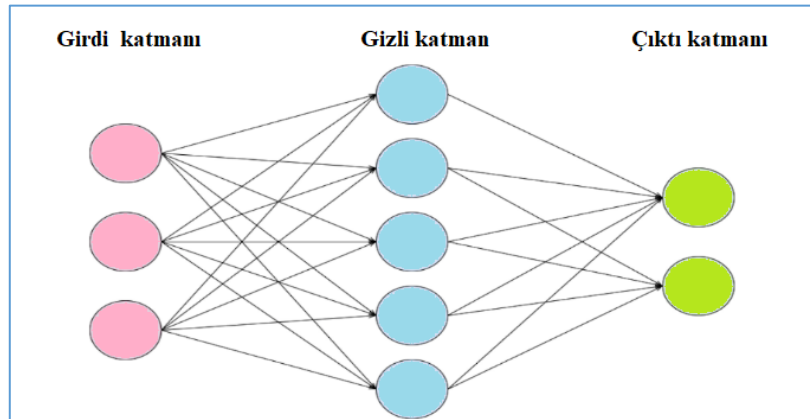


Şekil 4.14. Aktivasyon fonksiyon grafikleri [110].

- **Çıktılar:** Aktivasyon fonksiyonundan çıkan değer hücresinin çıktı değeri olmaktadır. Her hücresinin birden fazla girdisi olmasına rağmen bir tek çıktısı olmaktadır. Bu çıktı istenilen sayıda hücreye bağlanabilir [110].

4.3.1.1. ANN'nin Yapısı

ANN'nin yapısı üç bileşenden oluşur: nöron (yapay sinir hücresi), bağlantılar ve öğrenme algoritması. Nöron, bir yapay sinir ağının temel işlem elemanıdır. Ağdaki nöronlar, probleme göre aldığı girdi değerlerine göre istenilen sayıda sonuç verir. Bağlantılar aracılığıyla nöronların birbirine bağlanması Şekil 4.15'teki gibi ANN oluşturur. Genel bir ANN sisteminde, nöronlar aynı yönde bir araya gelerek katmanlar oluştururlar [110].



Şekil 4.15. Yapay sinir ağı [111].

4.3.2. Naïve Bayes

Algoritma, İngiliz matematikçi Thomas Bayes'in adını almıştır. Bayes algoritmaları, istatistiksel sınıflandırma yöntemleri arasındadır ve istatistiksel Bayes teoremine dayanmaktadır [112]. NB, sürekli veriler yerine ayrık verilerle çalışır. Bu nedenle, sürekli bağımlı veya bağımsız değişkenler kategorik verilere dönüştürülür. Örneğin yaş sayısal bağımsız değişkeni "16-25", "25-44", "45-65", "65+" gibi ayrık hale getirilmelidir.

$X = \{ x_1, x_2, x_3, \dots, x_n \}$ örnek küme, ve $C_1, C_2, C_3, \dots, C_m$ sınıf kümesi olsun. Sınıflandırma işlemine tabi tutulacak kümenin olasılık hesabı Denklem (4.7)'de gösterildiği gibi hesaplanır.

$$P(X|C_i) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (4.7)$$

Her sınıf için hesaplanan olasılığı en yüksek veri örneği o sınıfa aittir [112].

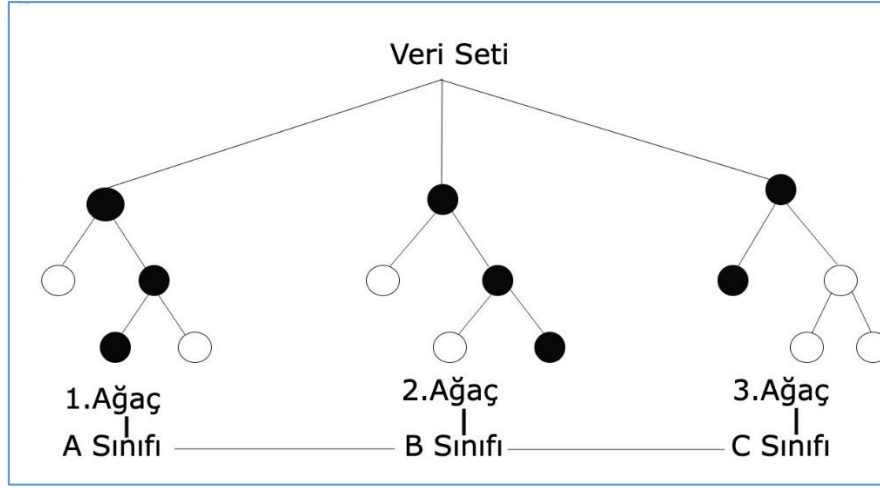
4.3.3. Karar Ağacı

Özellik ve hedefe göre karar düğümleri ve yaprak düğümlerden oluşan ağaç yapısı şeklinde model oluşturan bir sınıflandırma yöntemidir. DT algoritması, veri kümesini daha küçük hatta daha küçük parçalara bölerek geliştirilir. Bir karar düğümü bir veya daha fazla dal içerebilir. İlk düğüme kök düğüm denir. Bir DT hem kategorik hem de sayısal verilerden oluşabilir [113].

DT, insanların karar verme biçimine benzer şekilde, karar vermek için bir dizi kural kullanan denetimli bir algoritmadır [114].

4.3.4. Rastgele Orman

2001 yılında Breiman tarafından tanıtılan [115] RF modeli birçok sınıflandırma ve regresyon problemleri için kullanılmaya başlanmış ve başarılı sonuçlar verdiği gözlemlenmiştir. Yapısal olarak Karar Ağaçları modelini kullanan Rastgele Orman, RF'nin aşırı öğrenme (overfitting) problemine bir çözüm getirmiştir. Veri kümesinden rastgele gruplar seçerek yüzlerce karar ağacı oluşturuluyor ve her bir karar ağacı kendi tahminini üretiyor. Eğer problem regresyonsa, Karar Ağaçları tahminlerin ortalamasını alır, problem sınıflandırmaysa tahminler arasında en çok oy alan seçilir. Şekil 4.16'da RF modeli gösterilmiştir [34].



Şekil 4.16. Rastgele orman [115].

4.3.5. Lojistik Regresyon

LR, bir sonucu belirleyen bir veya birden çok bağımsız değişken içeren bir veri kümesini analiz etmede kullanılan istatistik temelli yöntemdir. Sonuç bir ikili değişken tarafından ölçülür.

LR'nin amacı, bağımlı değişken olan iki yönlü özellik ile bir dizi ilgili bağımsız değişken arasındaki ilişkiyi açıklamak için ilgili modeli bulmaktır. LR, ilgilenilen özelliklerin var olma olasılığını Denklem (4.8)'de yer alan logit dönüşümüyle tahmin etmek için bir formülün katsayılarını (ve bunların standart hatalarını ve önem düzeylerini) üretir [116].

$$\text{logit}(p) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 \dots b_kX_k \quad (4.8)$$

Denklem (4.9)'da p, özelliğin bulunma olasılığıdır.

$$\text{oran} = \frac{p}{1-p} = \frac{\text{özelliğin bulunma olasılığı}}{\text{özelliğin bulunmama olasılığı}} \quad (4.9)$$

Denklem (4.9)'da bulunan p değeri, Denklem (4.10)'da logit dönüşümü yapılır.

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) \quad (4.10)$$

LR'de, karekök hatalarının toplamını en aza indiren parametreleri seçmek yerine, örnek değerlerin gözlemlenme olasılığını en üst düzeye çıkaran parametreler seçilir [107].

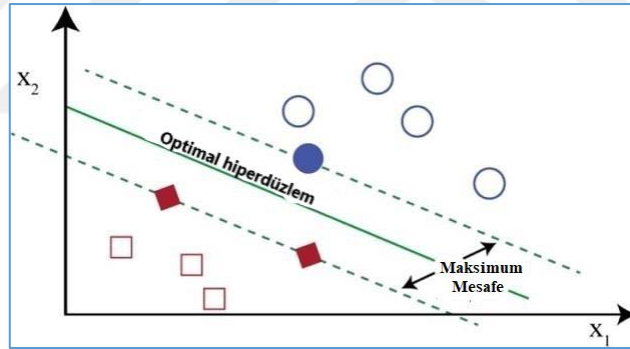
4.3.6. K en Yakın Komşu

KNN, uygulaması kolay denetimli öğrenme algoritmalarından biridir. Hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılmasına rağmen endüstride daha çok sınıflandırma problemlerinin çözümünde kullanılmaktadır.

KNN algoritmaları Cover ve Hart [117] tarafından önerilmiştir. Algoritma, sınıfları bilinen bir örnek kümeden alınan verilerden yararlanılarak kullanılır. Örnek veri kümesine dahil edilecek yeni verinin uzaklığı mevcut veriye göre hesaplanır ve k yakın komşu sayısı kontrol edilir. Genel olarak, mesafe hesaplamaları için Öklid, Manhattan ve Minkowski mesafesi olmak üzere üç tip mesafe fonksiyon kullanılır.

4.3.7. Destek Vektör Makineleri

SVM en düşük yapısal risk ilkesine dayanan denetimli makine öğrenme algoritmasıdır. Eğitim esnasında pozitif ve negatif örnekleri ayırmak için bir hiper düzlem oluşturulur. Bu hiper düzlemden her örneğin yerleştirilmesi gerektiğini belirterek yeni örnekleri sınıflandırır [118]. Şekil 4.17’de SVM algoritmasının mimari işleyişi gösterilmiştir.



Şekil 4.17. SVM yöntemi [117].

SVM, hem doğrusal sınıflandırma hem de doğrusal olmayan sınıflandırma gerçekleştirir. Doğrusal olmayan sınıflandırma, çekirdek işlevi kullanılarak uygulanır. Doğrusal olmayan sınıflandırmada, çekirdekler homojen polinom, karmaşık polinom, Gauss radyal temel fonksiyonu ve hiperbolik tanjant fonksiyonu olarak tanımlanır [119],[120].

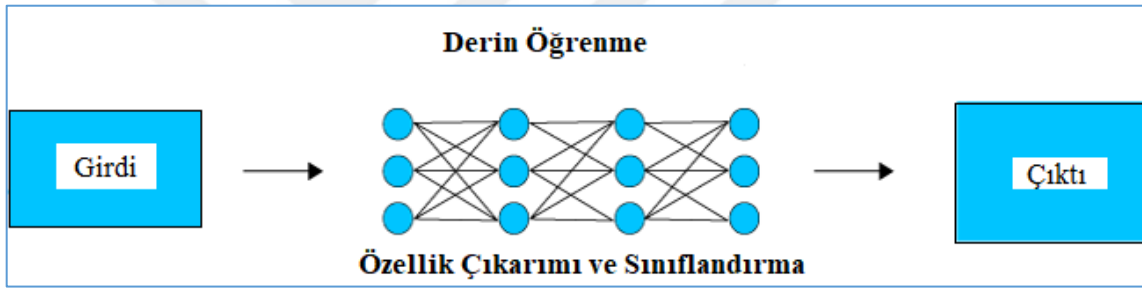
4.4. DERİN ÖĞRENME

Derin öğrenme, Walter Pitts ve Warren McCulloch'un 1943 yılında makinelerinde insanlar gibi düşünebilmesini araştırması sonucu olarak yapay sinir ağları için bir hesaplama modeli geliştirmesiyle ortaya çıkan bir alandır. Derin öğrenme de Makine

öğrenmesinin bir alt dalı olarak ortaya çıkmıştır. Şekil 3.1’de aralarındaki ilişki gösterilmiştir [112].

Makine öğrenmesi yöntemleri tek katmanda gerçekleştirilirken, derin öğrenme ise çok katmanlı sinir ağlarından oluşmaktadır. Veri miktarı artmasıyla beraber derin öğrenme algoritmalarının başarımları artmış ve daha yaygın kullanılmaya başlanmıştır. Ayrıca derin öğrenme algoritmalarının makine öğrenmesi algoritmalarından ayrıldığı diğer önemli nokta, öznelik çıkarma işlemi de sınıflandırma süreciyle beraber yapılmaktadır. Şekil 4.20’de bu işlem görselleştirilmiştir.

Derin öğrenme, geri yayılım algoritmasını kullanarak bir makinenin önceki katmanındaki çıktıdan her katmandaki çıktıyı hesaplamak için sistem parametrelerinin nasıl değiştirilmesi gerektiğini bulur. Bu şekilde büyük veri kümelerindeki karmaşık yapıyı keşfeder [121]. Makine öğrenmesinden farkı Şekil 4.18’de görüldüğü gibi öznelik seçimi öğrenme ile beraber yapılır.



Şekil 4.18. Derin öğrenme uygulama süreci [112].

Derin öğrenme, özellik çıkarmak için doğrusal olmayan çeşitli ardışık katmanlar kullanır. Bu katmanlardan, öncekinin çıktısı sonrakine girdi olur [122]. Derin öğrenme katmanlarında kullanılan algoritmalar denetimsiz veya denetimli öğrenmeden herhangi biri olabilir [121].

Yapay sinir ağlarındaki katman sayılarının artırılmasıyla kurulan çok farklı türde derin öğrenme mimarileri bulunmaktadır [123]. Bu mimariler [85];

- Evrişimli Sinir Ağları (Convolutional Neural Networks)
- Tekrarlayan Sinir Ağları (Recurrent Neural Networks)
- Uzun Kısa Dönemli Bellek (Long Short Term Memory)
- Boltzman Makinesi (Boltzmann Machines)
- Derin İnanç Ağları (Deep Belief Network)

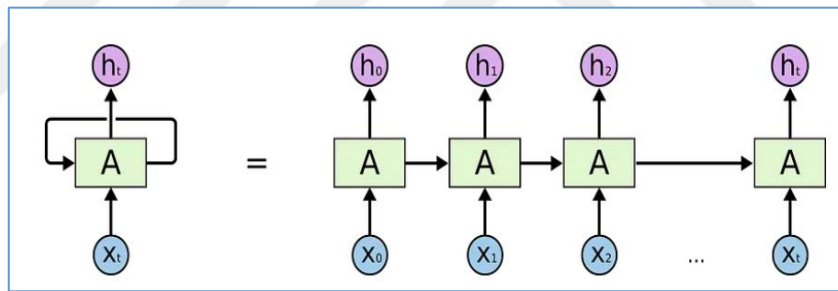
- Dikkat Ağı (Attention Network)
- Kapsül Ağı (Capsule Network) şeklindedir.

4.4.1. Derin Öğrenme Mimarileri

Bu ağlardan çalışmada kullanılan LSTM, RNN, GRU, BiGRU, BiLSTM sinir ağları alt bölümlerde anlatılacaktır.

4.4.1.1. Tekrarlayan Sinir Ağları

Elman tarafından 1990'da ortaya çıkarılan tekrarlayan sinir ağı, nöronlar arasındaki bağlantıları yönlendirilmiş bir döngüye dönüştüren sinir ağları modelidir. Şekil 4.19'da gösterildiği gibi önceki adımdan çıkan sonuç bir sonraki adımı beslemektedir [124]. İleri beslemeli sinir ağlarından farklı olarak RNN, bir dizi girişi işlemek için dahili belleği kullanabilir, bu da onu sıralı bilgilerin işlenmesi gereken problemler için tercih edilmesini sağlar. RNN'nin, her ögesi için aynı görevi gerçekleştirdiği ve her çıktının önceki tüm hesaplamalara bağlı olduğu anlamına gelen bir hafızası vardır. RNN bu hafıza sayesinde o ana kadar işlenenlerle ilgili bilgileri hatırlamış olur [125].



Şekil 4.19. RNN mimarisi [114].

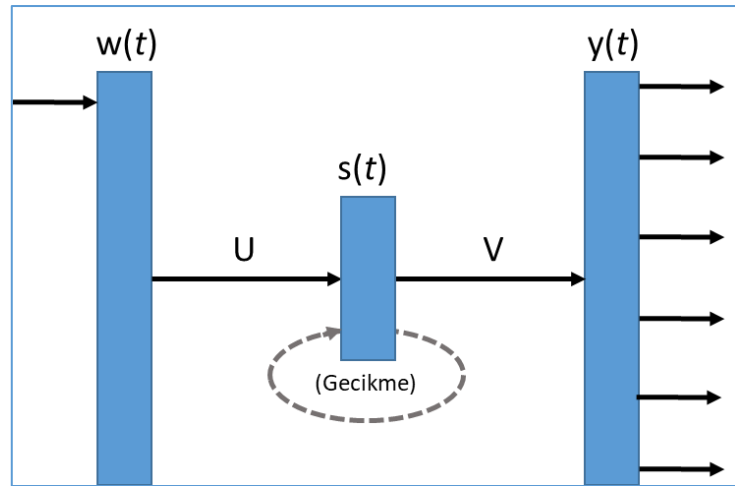
RNN mimarisi açıklanacak olursa; t zamanında görüldüğü gibi, mimari, önceki gizli katmandan h_t 'yi ve t zamanındaki x girişini dahil ederek $t-1$ 'de kullanır. Bu, ağın sırayla geçerli girdinin arkasında bulunan önceki girdilerden gelen bilgileri kullanarak bir sonraki katmana aktarılır. Sıfıncı h vektörünün her zaman 0'ların bir vektörü olarak başlayacağına dikkat etmek önemlidir. Çünkü algoritma dizideki ilk öğeden önce gelen hiçbir bilgiye sahip değildir [126].

RNN'nin normal bir sinir ağından farkı; basit bir sinir ağı bir fotoğraf içerisindeki nesnelere tanımlamak için sadece o anda verilen giriş bilgilerini kullanır. Sisteme daha önce verilen bilgileri kullanmaz. O bilgileri kullanmadan bir tahmin oluşturur. Böylece hedefe gitmek amaçlanır. RNN ise yalnızca o an verilen bilgileri değil, sisteme daha önce

tanıtılan bilgileri ve ayrıca sisteme daha sonra yüklenecek bilgileri de kullanır. Bu sinir ağları bir belleğe sahip olan ağlardır. Eski veriler ile yeni verileri karşılaştırarak sonuca ulaşırlar.

RNN, zaman içerisinde çok farklı tekrar yapabilir. Bu tekrar denemeleri yüzünden veriyle ilişkili olan bazı parametreler veriyi az etkilediği gerekçesiyle sistemden atılabilir. Eğer bu parametreler sinir ağına çok önce eklenmişse artık kullanıcı bu bilgiye ulaşamayabilir. Atılan ve unutulmuş bu bilgileri bir bellekte toplayabilmek için geliştirilmiş mimariler bulunmaktadır. GRU ve LSTM bu mimarilere örnektir.

İnsanlar düşüncelerini, önceden öğrendiği bilgilere dayanarak oluşturur. Örneğin bir cümleyi okuduğunda her kelime bir önceki kelime baz alınarak anlaşılır. Yani önceki baz alarak anlarız yani önceki kelimeyi hatırlayarak sonraki kelime veya kelimeler anlamlandırır. Geleneksel sinir ağları bu işlemi yapamaz. RNN bu işlemleri yapmak için Jeff Elman tarafından 1990'larda geliştirilmiştir [125]. Şekil 4.20'de RNN mimarisi gösterilmiştir.



Şekil 4.20. Yann'ın RNN mimarisi [114].

RNN ağı geri yayılım kullanılarak eğitilir. Giriş, gizli ve çıkış katmanlarını içerir. Bu katmanlardaki değerlerin hesaplamaları Denklem (4.11), Denklem (4.12) ve Denklem (4.13)'te gösterilmektedir.

$$x(t) = [w(t)^T s(t-1)^T]^T \quad (4.11)$$

$$s_j(t) = f\left(\sum_i x_i(t) u_{ji}\right) \quad (4.12)$$

$$y_k(t) = g\left(\sum_j s_j(t) v_{kj}\right) \quad (4.13)$$

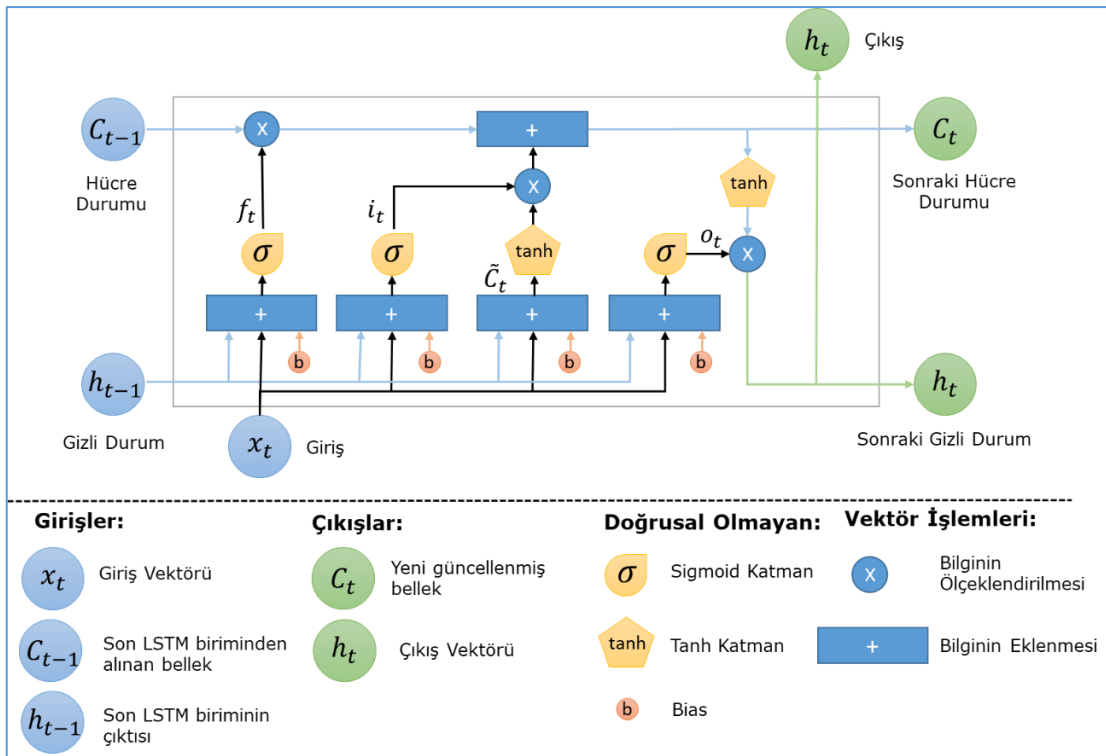
$x(t)$ vektörü, N kodlamanın biri kullanılırken geçerli kelimeyi temsil eden $w(t)$ vektörünün ve önceki zaman adımından gizli katmandaki çıktı değerlerini temsil eden $s(t-1)$ vektörünün birleştirilmesiyle oluşturulur. $s(t)$ gizli katman ve $y(t)$ çıkış vektörünü temsil eder.

4.4.1.2. Uzun Kısa Süreli Hafıza

Hochreiter ve Schmidhuber tarafından tanıtılan RNN'nin kaybolan gradyan problemini çözmek için geliştirmiştir [127].

LSTM hücresi giriş, çıkış ve unut kapısı olarak şu üç kapı içerir. İlk kapı, giriş aktivasyonlarının bellek hücresine akışını kontrol edilen kapı giriş kapısıdır. İkinci kapı, giriş ve önceki çıktındaki bilgileri filtreler ve hangisinin hatırlanacağına veya unutulup çıkarılacağına karar veren kapıdır ve son kapıda, hücre aktivasyonunun çıkış akışını kontrol eden çıkış kapısıdır [127].

Üç kapının yanı sıra, LSTM hücresi, genellikle hücre durumunun bir parçası olan tanh katmanı olan hücre güncellemesi içerir. Her bir LSTM hücresinde; Mevcut giriş (X_t), Önceki çıktı (h_{t-1}), Önceki hücre durumu (C_{t-1}) giriş değerleri olarak yer alır. Çıkış değerleri olarak Akım çıkışı (h_t), Mevcut hücre durumu (C_t) çıkış değerleri olarak yer alır. Şekil 4.21'de LSTM'nin mimari yapısı verilmiştir.



Şekil 4.21. LSTM mimarisi [127].

Şekil 4.21’de verilen LSTM mimarisine ait değişkenler ve bu değişkenlerin açıklaması Çizelge 4.5’ te verilmiştir

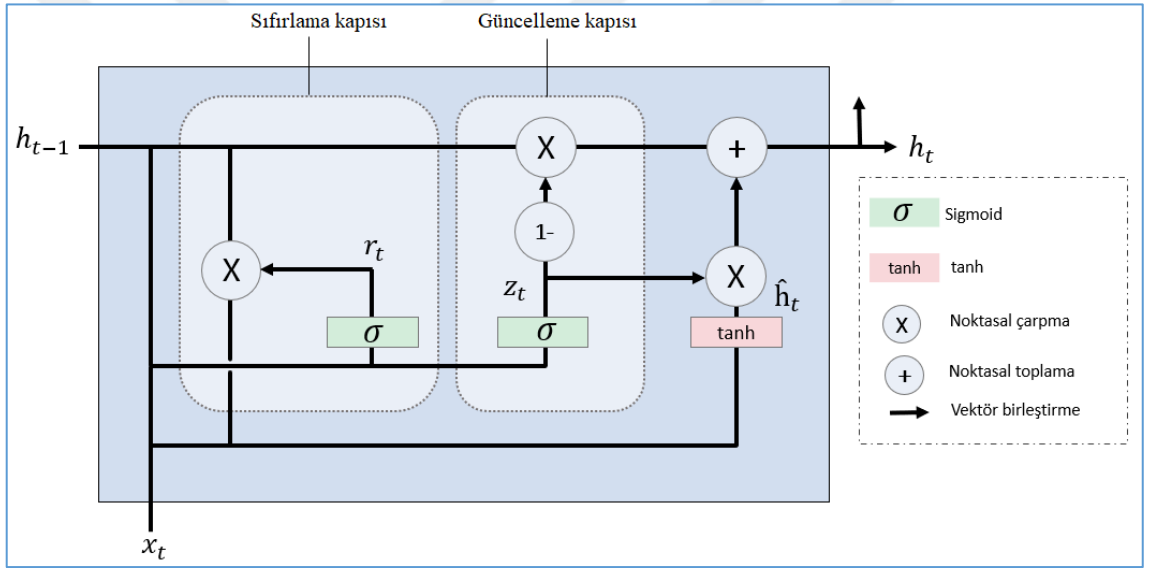
Çizelge 4.5. LSTM mimarisine ait değişkenler.

Değişken	Açıklaması
$X_t \in \mathbb{R}^d$	LSTM birimine giriş vektörü
$f_t \in \mathbb{R}^d$	Unut kapısının aktivasyon vektörü
$i_t \in \mathbb{R}^h$	Çıkış kapısının aktivasyon vektörü
$h_t \in \mathbb{R}^h$	LSTM biriminin çıkış vektörü olarak da bilinen gizli durum vektörü
$\tilde{C}_t \in \mathbb{R}^h$	Hücre girişi aktivasyon vektörü
$C_t \in \mathbb{R}^h$	Hücre durumu vektörü
$W \in \mathbb{R}^{h \times d}$	h ve d üst simgelerinin sırasıyla girdi özelliklerinin sayısını ve gizli birimlerin sayısını ifade ettiği eğitim sırasında öğrenilmesi gereken ağırlık matrisleri ve yanlılık vektörü parametreleri

Hangi bilginin atılacağına karar verme LSTM’nin ilk adımındır. f_t olarak adlandırılan bir sigmoid aktivasyon fonksiyonuna sahip katman aracılığıyla gerçekleştirilir. x_t ve h_{t-1} ye göre, C_{t-1} hücre durumunu her sayı için sıfır, bir arasında bir çıkış üretilir. Eğer çıkış bir ise; Çıkış değeri 1 ile, gelen değer tutulur, sıfır ise değer atılır. LSTM hücresinin diğer adımı, yeni bilginin kalıp kalmayacağına karar vermektir. Bu adım iki bölümdür. Birincisi giriş katmanıdır ve bu katmanda sigmoid ile verilerin güncellenmesine karar verilir. Sonra ki, katmanda tanh uygulanır ve yeni aday vektör oluşturur, \tilde{C}_t , yeni duruma eklenebilir. Bir sonraki adımda, durumu güncellemek i_t ve \tilde{C}_t birleştirilir. Bu durum, her bir değer güncellenme miktarına karar vermek için ölçeklendirilerek bir aday değeri oluşmasını sağlar. C_{t-1} ve f_t çarpılarak bir önceki kararın unutulması sağlanır. Son olarak çıkışa ne gönderileceğine karar verilir. Bu çıkış hücre durumuna bağlıdır. Bu katmanda yer alan Sigmoid’in çıktısı o_t tanh katmanı tarafından hücre durumundan C_t oluşturulan (+1) ile (-1) arasında oluşturan yeni değerle çarpılır. LSTM’ler, RNN algoritmasının hafızalı şeklidir. Metin işleme, el yazısı tanıma, makine çevrisi gibi çeşitli alanlarda kullanılır [127].

4.4.1.3. Kapı Tekrarlayan Geçitler

GRU yinelenen sinir ağlarını kullanan yeni nesil algoritmalarından biridir. GRU, daha az tensor işlemi barındırdığı için LSTM'den daha hızlı çalışır. LSTM modeline çok benzer yapıdadır. LSTM'den farkı ise hücre durumunun (cell state) modelden çıkarılmasıdır. Hücre durumu LSTM modelinde bilgi aktarımını sağlarken, burada bilgi aktarımı gizli durum tarafından yapılmaktadır. Modeldeki diğer bir fark da yalnızca iki kapının bulunmasıdır. Bu kapılar Sıfırlama Kapısı (Reset Gate) ve Güncelleme Kapısıdır (Update Gate). Şekil 4.22'de GRU model yapısı gösterilmiştir [128]. Güncelleme (Update Gate) kapısı, yeni bilgi ekleme ve atmaya karar veren kapıdır. Sıfırlama (Reset Gate) kapısı, geçmiş verilerin ne kadarının unutulacağına karar veren kapıdır. GRU'lar daha az vektör işlemine sahip oldukları için diğer RNN türlerinden biraz daha hızlıdır [128].



Şekil 4.22. GRU mimarisi [128].

GRU'nun model yapısı Şekil 4.22'de gösterilmektedir. Belirli bir t zaman aralığı için, küme boyutu girişi (n: örnek sayısı, d: giriş sayısı) $x_t \in R^{n*d}$ ve son adımın gizli durumu (h: gizli durum sayısı) $h_{t-1} \in R^{n*h}$ olarak ifade edilir.

GRU, ünite içindeki bilgi akışını kontrol eden güncelleme ve sıfırlama kapılarına sahiptir. Güncelleme Kapısı, hangi bilgilerin unutulacağına karar verir [128].

$$U_t = \sigma(W_U h_{t-1} + W_U x_t + b_U) \quad (4.14)$$

Güncelleme Kapısının hesaplanması Denklem. (4.14)'te verilmiştir. Sıfırlama kapısı, geçmişin ne kadarının unutulacağına karar vermek için kullanılan kapıdır [128].

$$R_t = \sigma(W_R h_{t-1} + W_R x_t + b_R) \quad (4.15)$$

Sıfırlama Kapısının hesaplanması Denklem (4.15)'te verilmiştir. Değerin 0'a yakın olması, mevcut hafıza içeriğinde bir önceki ana ait bilgilerin unutulduğunu gösterir. 1'e yakın olduğunda mevcut hafıza içeriğinde kalacağını gösterir [129].

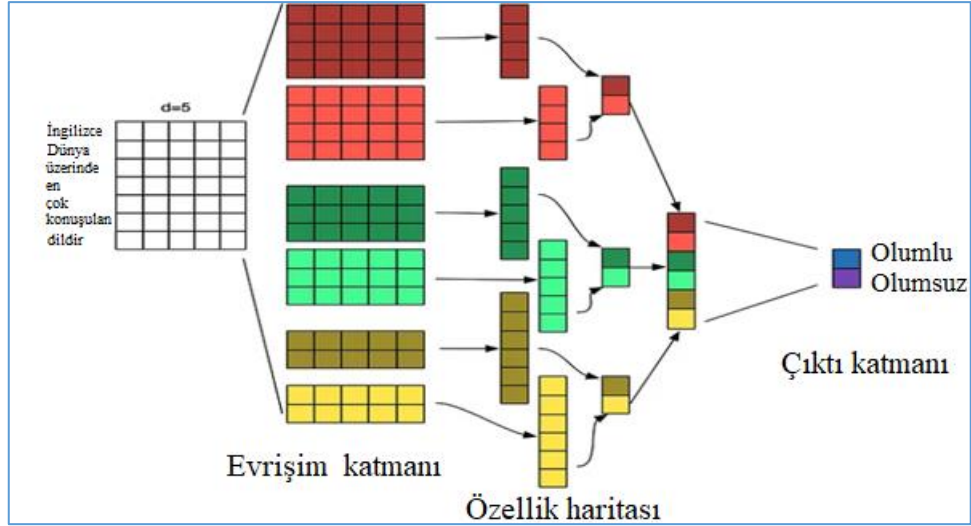
$$\hat{h}_t = \tanh(W \cdot [R_t * h_{t-1}] + W_{x_t}) \quad (4.16)$$

t anındaki belleğin içeriğinin belirlenmesi Denklem (4.16)'da verilmiştir. Sıfırlama ve güncelleme kapıları üzerinden hafızanın ne kadarının unutulacağı ve saklanacağı belirlendikten sonra “tanh” aktivasyon fonksiyonu ile ölçeklendirilir [129]. Denklem (4.17)'de, gizli katman tarafından t zamanında depolanan bilgiler belirlenir.

$$h_t = (1 - U_t) * h_{t-1} + U_t * \hat{h}_t \quad (4.17)$$

4.4.1.4. Evrişimli Sinir Ağları

CNN bir görüntünün analizinin yapılmasını sağlar. Çoğunlukla görüntü işleme ve bilgisayarla görme için kullanılan derin, ileri beslemeli yapay sinir ağlarının bir sınıfıdır. Görüntü işleme ve sınıflandırma için geliştirilmiştir, ancak son zamanlarda metin sınıflandırma görevlerinde de kullanılmaktadır. CNN, model için özellik çıkarıcı olarak evrişim ve havuzlamayı kullanılmaktadır. Evrişim ve havuzlamadan elde edilen çıktı, tamamen bağlı çok katmanlı bir algılayıcıya bağlanır [130]. Evrişimli bir sinir ağında, gizli katmanlar, evrişimleri gerçekleştiren katmanları içermektedir. Bu katmanda, çarpma veya başka bir nokta çarpımı yapan bir bir etkinleştirme işlevi yer alır. Bu katmandaki işlemde sonra gelen katmanlar sırasıyla, havuzlama katmanları, tam bağlantılı katmanlar ve normelleştirme katmanları gibi diğer katmanlar gelmektedir [131]. Kelime gömme sonrası elde edilen metin verilerinde evrişim işlemi uygulamak için, önceden tanımlanmış boyutta bir filtre ve çekirdek seçilir. Sonrasında bu filtre metin değerleri boyunca hareket ettirilir. Bu filtre değerlerine karşılık gelen değerler çarpılarak özellik haritası çıkarılır [131]. Daha sonra havuzlama (Pooling), kullanılarak kelime matrislerinin boyutu küçültülür. Evrişim işleminden sonra dizinin boyutu ile girişin boyutunun aynı tutulması için “same padding” kullanılır. Şekil 4.23'te CNN algoritmasının doğal dil işlemeye uygulanmasına yönelik verilmiştir.



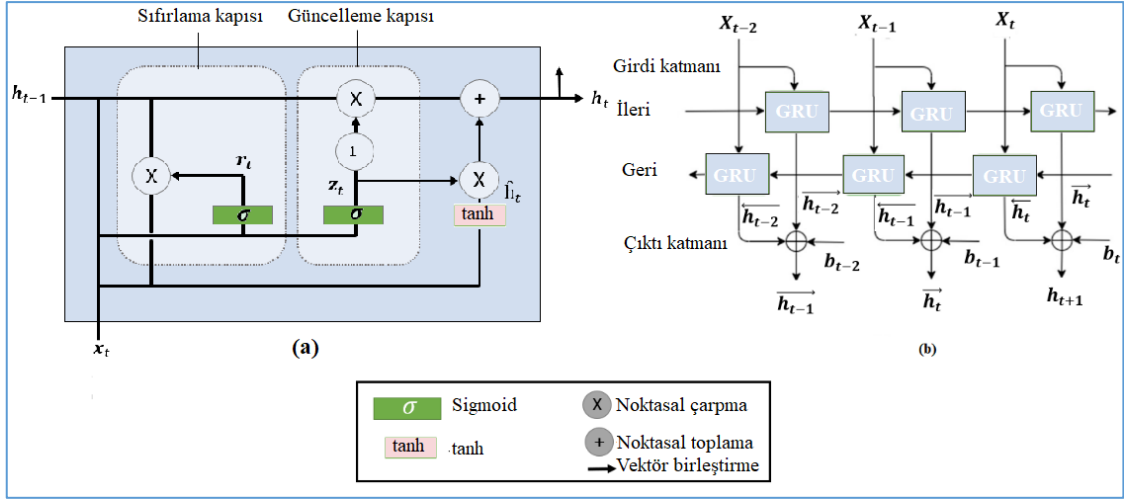
Şekil 4.23. Doğal dil işleme de CNN kullanımı [132].

Görüntü sınıflandırma problemlerinde evrişimli sinir ağlarını temel alan birçok model geliştirilmiştir. Bunlar; LeNet, AlexNet, GoogLeNet, VGGNet gibi modellerdir. Bu modeller transfer öğrenme şeklinde görüntü işleme problemlerinde kullanılmaktadır [133]. CNN, başta görüntü işleme olmak üzere, sinyal işleme, ses işleme ve doğal dil işleme gibi yapay zekanın birçok alt dalında kullanılmaktadır [134].

4.4.1.5. Çift Yönlü Kapı Tekrarlayan Hücreler

Çift yönlü tekrarlayan sinir ağları sadece iki bağımsız GRU'yu bir araya getirir. Bu yapı, ağların her zaman adımında dizi hakkında hem geriye hem de ileriye dönük bilgilere sahip olmasını sağlar.

Çift yönlü GRU kullanarak, girdilerden birinin geçmişten geleceğe diğerininse gelecekte geçmişe olmak üzere iki şekilde çalıştırması sağlanmaktadır. Çift yönlü GRU ile geriye doğru çalışan GRU'da gelecekte gelen bilgiler korunarak iki gizli durum bir araya getirilmiş olur. Zaman içinde herhangi bir noktada hem geçmişten hem de gelecekte gelen bilgiler korunmaktadır. Şekil 4.24'te GRU ve BiGRU'nun yan yana şematik yapısı görülmektedir [135].



Şekil 4.24. Çift Yönlü Tekrarlayan sinir ağları a) GRU, b) BiGRU [135].

4.4.1.6. Çift Yönlü Uzun Kısa Süreli Bellek

BiLSTM, mimarisinin bir türüdür ve dış yinelemeleri ve kapılarla sağladığı iç yinelemeleri olan bir modeldir.

LSTM kapıları arasında unutmaya kapısı (f_t), hangi bilgilerin unutulacağına, giriş kapısı (i_t) hangi bilgilerin ekleneceğine, çıkış kapısı (o_t) ise hangi bilgilerin diğer duruma aktarılacağına karar verir. Ağırlık matrisleri, her bir kapının ne kadar etkileneceğine karar verir. Aşağıda, bu sistemin işlem adımlarını açıklayan LSTM denklemleri Denklem (4.18) ile Denklem (4.22) arasında verilmiştir [131].

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4.18)$$

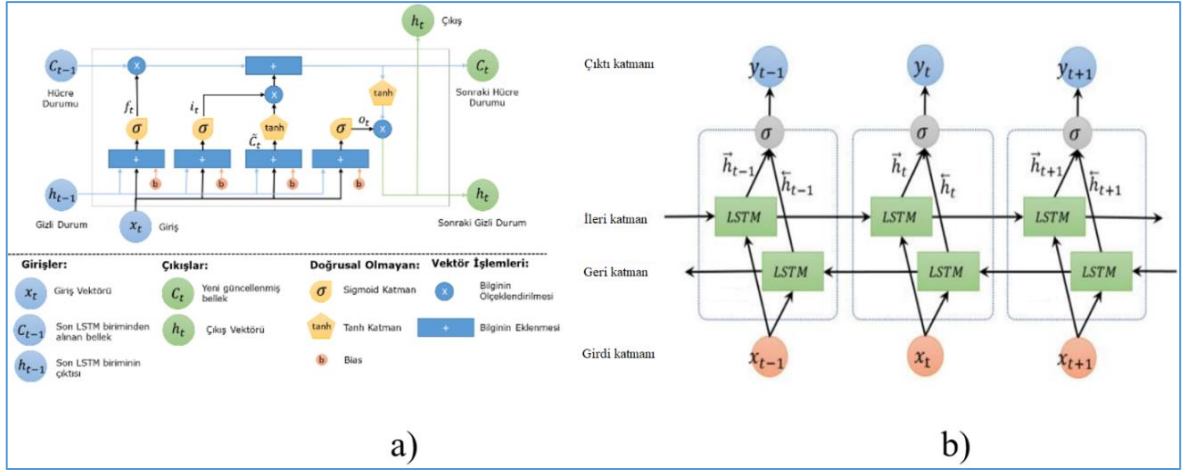
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4.19)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4.20)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4.21)$$

$$h_t = o_t \tanh(c_t) \quad (4.22)$$

BiLSTM modelinin eğitimi sırasında, LSTM'nin çift yönlü olarak çalışmasına dayandığı için Denklem (4.18) ile Denklem (4.22) arasındaki denklemler ileri ve geri yönlü olarak çalıştırılır. Şekil 4.25'te LSTM ve BiLSTM'nin yan yana şematik yapısı görülmektedir [136].

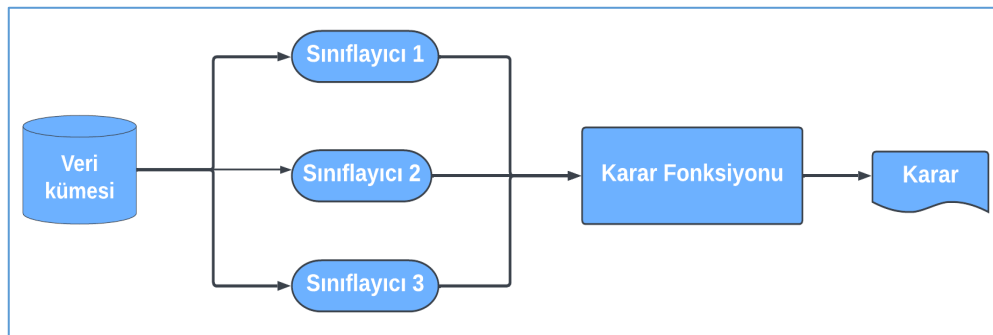


Şekil 4.25. Çift Yönlü Tekrarlayan sinir ağları a) LSTM, b) BiLSTM [136].

4.5. TOPLULUK ÖĞRENME

Topluluk Öğrenimi, temel olarak sınıflandırma ve regresyon problemlerini çözmek için makine öğrenmesi algoritmalarının birlikte kullanılması anlamına gelir. Bu algoritmalar aynı tip (homojen) veya farklı tiplerde (heterojen) olabilir. Topluluk öğrenmesi, tek bir zayıf model kullanılarak elde edilen etkinliği artırmak için çeşitli uzmanların veya makine öğrenmesi modellerinin stratejik bir kombinasyonunu gerçekleştirir [137].

Topluluk öğrenmesi, temel olarak kullanılan modellerin türüne (homojen veya heterojen modeller), veri örneklemesine (değiştirmeli veya değiştirmeden, k-katlama vb.) ve karar işlevine (oylama, ortalama, meta-model) dayanır. Bu nedenle Topluluk öğrenme yöntemleri Torbalama, Güçlendirme (Boosting), Yığınlama olarak sınıflandırılabilir. Bu üç ana kategoriye ek olarak, Oylama ve Harmanlama (Blending) olmak üzere iki önemli türü daha vardır. Oylama ve Karıştırma, Sırasıyla Torbalama ve Yığınlanmanın bir tamamlayıcısı ve bir alt türü olsa da bu yöntemler genellikle Topluluk Öğreniminin doğrudan türleri olarak bulunur ve Şekil 4.26'da öğrenme süreci gösterilmiştir [138].



Şekil 4.26. Topluluk öğrenme süreci [138].

Tez çalışmasında, Oylama ve Yığınlama topluluk öğrenmesi kullanıldığı için bu bölümde bu yöntemler anlatılmıştır.

4.5.1. Oylama

Oylama belki de en basit topluluk öğrenme algoritmasıdır ve genellikle çok etkilidir. Sınıflandırma veya regresyon problemlerinde kullanılabilir. Oylama, iki veya daha fazla alt model oluşturarak çalışır. Her bir alt model, tahminlerin ortalamasını veya modunu alarak, her bir alt modelin sonucunun ne olması gerektiğine ilişkin, tüm sınıflayıcılardan alınan sonuçların oylama mantığıyla en yüksek sınıfı veren sınıf olarak tahminlenir [139].

4.5.1.1. Çoğunluk Oylaması

Çoğunluk Oylaması, her bir sınıflandırıcı bir sınıf için oy kullanır ve çoğunluk kazanır. İstatistiksel terimlerle, topluluğun tahmin edilen hedef etiketi, bireysel olarak tahmin edilen etiketlerin dağılım modudur.

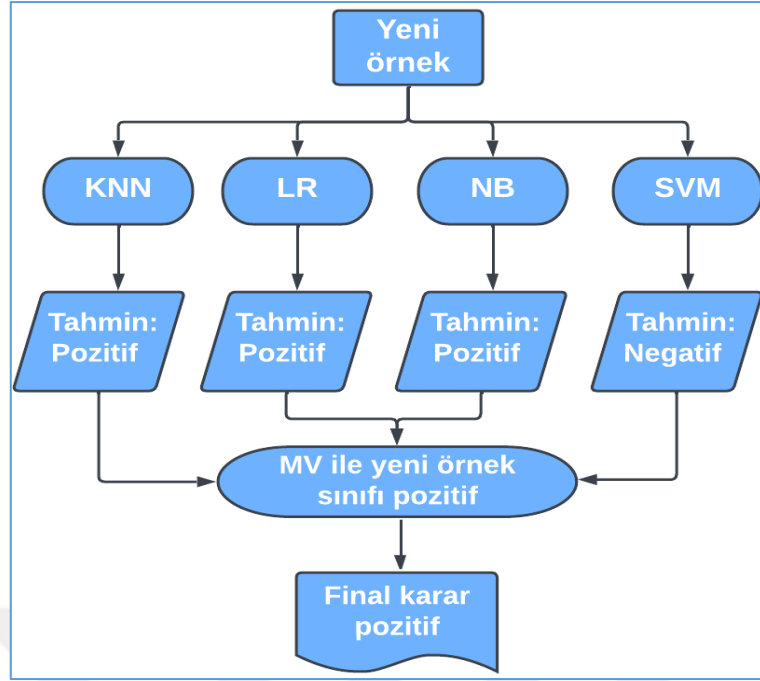
Burada, her sınıflandırıcı C'nin çoğunluk oylaması yoluyla \hat{y} sınıf etiketi tahmin edilmektedir. Bu, Denklem (4.23)'te verilmiştir [91].

$$\hat{y} = mode\{C_1(x), C_2(x), \dots \dots C_m(x)\} \quad (4.23)$$

Denklem (4.24)'te olduğu gibi bir eğitim örneğini sınıflandıran Üç sınıflandırıcının birleştirildiği varsayılacak olursa[91]:

$$\hat{y} = mode\{0,0,1\} = 0 \quad (4.24)$$

Çoğunluk oyu ile, örnek "sınıf 0" olarak sınıflandırılır [140]. Çoğunluk oylama süreci Şekil 4.27'de gösterilmektedir.



Şekil 4.27. Çoğunluk oylaması süreci [140].

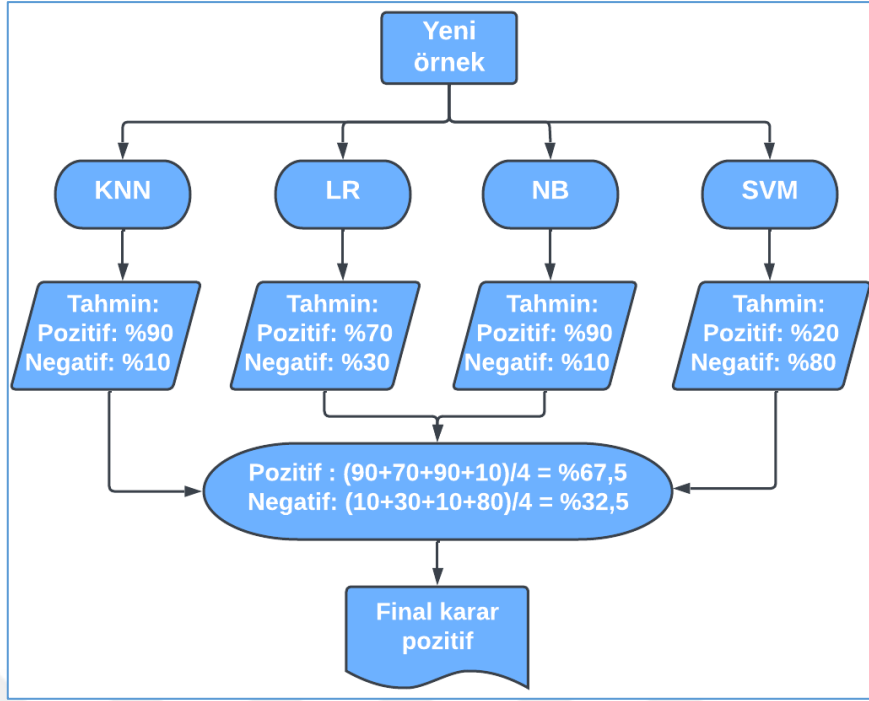
4.5.1.2. Olasılıklı Oylama

Olasılıklı Oylama, her bir sınıflandırıcı, belirli bir veri noktasının belirli bir hedef sınıfa ait olduğuna dair bir olasılık değeri sağlar. Tahminler, sınıflandırıcının önemine göre ağırlıklandırılır. Ardından, ağırlıklı olasılıkların toplamı en yüksek olan hedef etiketi oylamayı kazanır.

Olasılıklı oylama, sınıf etiketlerini sınıflandırıcı için tahmin edilen olasılıklara (p) dayalı olarak tahmin eder. Bu yaklaşım, yalnızca sınıflandırıcılar iyi ayarlanmışsa önerilir.

$$\hat{y} = \arg \max_i \sum_{j=1}^m w_j p_{ij} \quad (4.25)$$

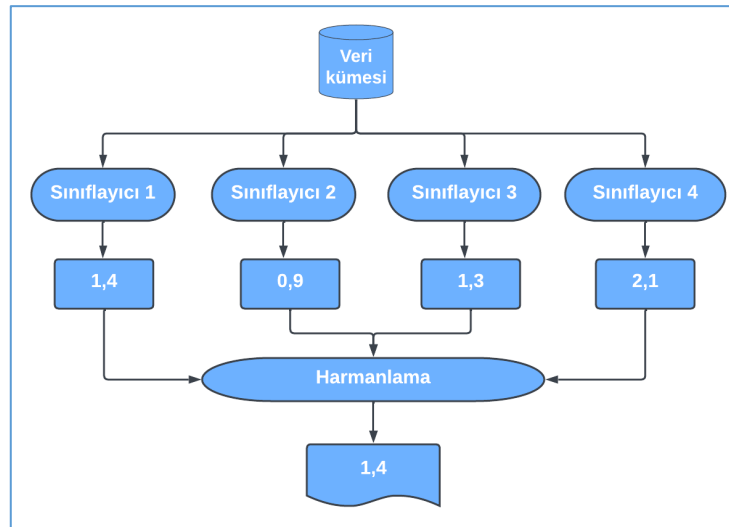
w_j , denklem (4.25) 'teki j. sınıflandırıcıya atanabilen ağırlıktır [141]. Olasılıklı oylama süreci Şekil 4.28'de gösterilmiştir.



Şekil 4.28. Olasılıklı oylama süreci [141].

4.5.2. Yığınlama

Yığınlama, sınıflandırma ve regresyon problemleri için kullanılabilen oylama gruplarının basit bir uzantısıdır. İstifleme, birden çok alt model seçmenin yanı sıra, alt modellerden gelen tahminleri en iyi şekilde nasıl birleştirileceğinin öğrenilmesi için başka bir model belirtilmesine olanak tanır. Alt modellerin (alt modeller) tahminlerini en iyi şekilde birleştirmek için bir meta model kullanıldığından, bu yöntem bazen tahminlerin harmanlanmasında olduğu gibi karıştırma (mixing) olarak adlandırılarak sınıflandırılır [140]. İstifleme işlemi Şekil 4.29'da gösterilmiştir.



Şekil 4.29. Yığınlama süreci [140].

4.6. DENEY ORTAM ÖZELLİKLERİ

Bu bölümde çalışma kapsamında deneylerin yapıldığı ortamlar, teknolojiler ve deneylerdeki oluşturulan modellerin performans değerlendirme metrikleri anlatılmıştır.

4.6.1. Deney Ortamlarının Yazılım ve Donanım Özellikleri

Tez çalışmasında yapılan tüm deneyler Python programlama dili ile yerel bilgisayar veya bulut ortamı olan Google Colab [142] ortamında yapılmıştır. IMDB veri kümesi üzerinde yapılan tüm deneyler Google Colab'te yapılmıştır. Tez çalışmasında TripAdvisor, Rotten Tomatoes ve Covid-19 Twitter gibi farklı veri kümeleri de kullanılmıştır. Bu veri kümeleri üzerinde yapılan deneyler, AMD Ryzen 7 işlemcili, 64 GB RAM'li yerel bilgisayarda gerçekleştirilmiştir. Deneysel çalışmalara ait detaylar ve alınan sonuçlar, 5. Bölümde “Deneysel Çalışmalar ve Sonuçları” başlığı altında anlatılmıştır.

Google Colaboratory (Colab Pro) [142] ve yerel bilgisayarda oluşturulan modellerde TensorFlow 2.9.0 [61] ve Keras 2.9 [143] kütüphaneleri ve Python 3.9.13 [144] sürümü kullanılmıştır. Makine öğrenmesi ve topluluk öğrenmesi modelleri için scikit-learn kütüphanesinin 1.0.2 [89] sürümü kullanılmıştır.

4.6.2. Derin Öğrenme Mimari Kurulumları

Probleme yönelik, hızlı şekilde derin öğrenme mimarileri oluşturmak için Tensorflow ve Pytorch gibi kütüphaneler kullanılır. Mimariler oluştururken sinir düğümü sayısı, gizli katman sayısı, aktivasyon fonksiyonu, optimizasyon fonksiyonu, küme boyutu gibi parametrelerin ayarlanması gerekmektedir. Sinir ağı modelini eğitmek için kullanılan farklı parametrelere hiperparametreler denir [145]. Bu hiperparametreler, optimize edilmiş bir modelle sonuçlanan bir sinir ağının performansını iyileştirmek için düğmeler gibi ayarlanmıştır. Sinir ağında, Gizli katman sayısı (Hidden layer), Gizli katmandaki birim veya düğüm sayısı (Num units), evrişim katmanındaki filtre sayısı, evrişim katmanındaki çekirdek ve adım (stride) değerleri, dolgu (padding) değeri, öğrenme oranı (Learning rate), Seyreltme oranı (Dropout rate), dönem, SGD, Adam gibi optimizasyon fonksiyonları, ReLU, sigmoid, gibi aktivasyon fonksiyonları, küme boyutu gibi hiperparametreler kullanılmıştır.

Manuel search, grid search, random search, bayes optimizasyonu gibi yöntemlerle hiperparametre optimizasyonu yapılabilir [146,147].

Grid Search yöntemi, kartezyen bir çarpımla sonuçlanan ve belirtilen hiperparametrelerin olası tüm kombinasyonlarının kapsamlı bir aramasını sağlamaktadır. Random Search yöntemi, hiperparametreler rastgele seçilir. Sonrasında her hiperparametre kombinasyonu işleme alınmaz. Bu yöntemde hiperparametrelerin sayısı arttıkça, hiperparametrelerin iyi bir kombinasyonuna daha hızlı ulaştığı için rastgele arama daha iyi bir seçenektir. Bayes optimizasyon yönteminde ilgili modelin doğruluğu veya kaybı da dahil olmak üzere, hiperparametreler hakkında önceki verileri birleştirilir. Bu yöntemde, ön bilgiler model için hiperparametre seçimine daha iyi yaklaşmanın belirlenmesine yardımcı olur.

Bu yöntemler için Sckit-Optimizer, Keras Tuner gibi araçlar kullanılır. Bu yöntemlere alternatif olarak Tensorflow için geliştirilmiş TensorBoard kütüphanesi de vardır. TensorBoard, bir derin öğrenme modeli için en iyi Hiperparametreleri bulmada kullanılan ve Grid Search temelli çalışan bir araçtır. Ayrıca bu işlemleri yaparken bir görselleştirme arayüzü sunar [61]. Bu çalışmada, hiperparametre ayarlamaları için TensorBoard kullanılmıştır. Tensorboard aracılığıyla yapılan ayarlamalar ve tüm detaylar 5.1.4 başlığı altında anlatılmıştır.

4.7. PERFORMANS METRİKLERİ

Çalışma kapsamında deneylerin performanslarının değerlendirilmesi için Doğruluk (Accuracy), F-ölçü, Duyarlılık (Recall), Kesinlik (Precision) performans kriterleri kullanılmıştır. Bu performans kriterlerinin hesaplanması Çizelge 4.6'da verilmiştir [85].

Çizelge 4.6. Karmaşıklık matrisi.

		Gerçek değer		
		Pozitif	Negatif	Toplam
Tahmin edilen değer	Pozitif	Tp	Fp	$Tpoz$
	Negatif	$$	$$	$$
	Toplam	$$	$$	

Çizelge 4.6'da, metinlerde yer alan olumlu ve olumsuz duygular üzerinden değerlendirme yapılacak olursa;

- True Pozitif (Tp) ve True Negatif (Tn) modelin pozitif olarak tahminlendiği,
- False Pozitif (Fp) ve False Negatif (Fn) ise modelin negatif olarak tahminlendiği alanlardır.

Tp değeri; duygu sınıfı pozitif olarak tahmin edilen ve pozitif sınıf etiketli metindir. Tn

değeri; duygu sınıfı negatif olarak tahmin edilen ve negatif etiketli metindir. Fp değeri, duygu sınıfı pozitif olarak tahmin edilen ama duygu sınıfı negatif etiketli metindir. Fn değeri, duygu sınıfı negatif olarak tahmin edilen ama duygu sınıfı pozitif etiketli metindir. Bu dört değer birbirine oranlanmasıyla doğruluk, kesinlik, duyarlılık ve F-skor değerleri elde edilir. Bu değerler bu bölümde anlatılmıştır [148].

4.7.1. Doğruluk

Sınıflandırma modellerin performanslarını ölçmek için kullanılan performans metriğidir. Doğruluk değeri, modelin doğru tahmin ettiği değerlerin bütün veri kümesi toplamına oranıdır [128]. Bu metrik denklem (4.26)' da ki gibi hesaplanır.

$$Doğruluk = \frac{Tp + Tn}{Tp + Fp + Fn + Tn} \quad (4.26)$$

Özellikle dengeli sınıf dağılımı olmayan veri kümelerinde, model performansını ölçmede tek başına yeterli değildir. Veri kümesi dengeli dağılmamışsa diğer performans metriklerine de bakılması gerekir [128].

4.7.2. Kesinlik

Kesinlik, pozitif olarak tahminlenen değerlerin gerçekten kaç adedinin Pozitif olduğunu göstermektedir [128]. Denklem (4.27)'de bu değer görülmektedir.

$$Kesinlik = \frac{Tp}{Tpoz} = \frac{Tp}{Tp + Fp} \quad (4.27)$$

Kesinlik değeri, duygu sınıfı pozitif etiketlenip pozitif tahmin edilen Tp değerinin, gerçekte pozitif veya negatif etiketlenmesine rağmen tahmin olarak hep pozitif tahmin edilen $Tpoz$ değerine oranıyla hesaplanmaktadır [128].

4.7.3. Duyarlılık

Duyarlılık, Pozitif olarak tahmin edilen değerlerin ne kadarının Pozitif olarak tahmin edildiğini gösteren bir metriktir [128]. Denklem (4.28)'de Duyarlılık değeri görülmektedir.

$$Duyarluluk = \frac{Tp}{Poz} = \frac{Tp}{Tp + Fn} \quad (4.28)$$

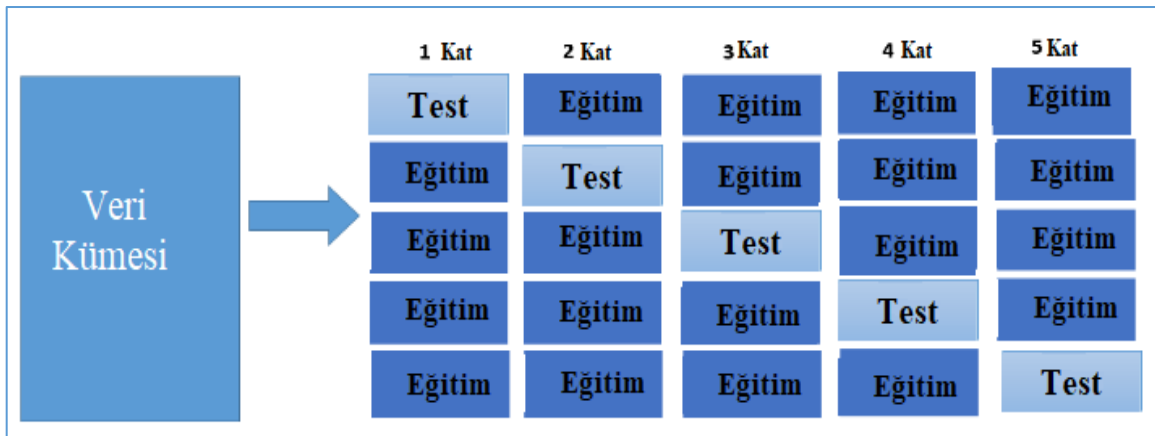
4.7.4. F-ölçü

F-ölçü, F1-skor veya F-skor gibi farklı şekillerde isimlendirilmektedir. Diğer sınıflandırma metrikleri olan duyarlılık ve kesinlik değerlerinin harmonik ortalamasıdır. Duyarlılık ve kesinlik kriterleriyle hesaplandığı için dengeli bir performans kriteridir. F-ölçü değerinin hesaplanması Denklem (4.29)'da verilmiştir [109].

$$F - ölçü = 2 * \frac{Kesinlik * Duyarluluk}{Kesinlik + Duyarluluk} \quad (4.29)$$

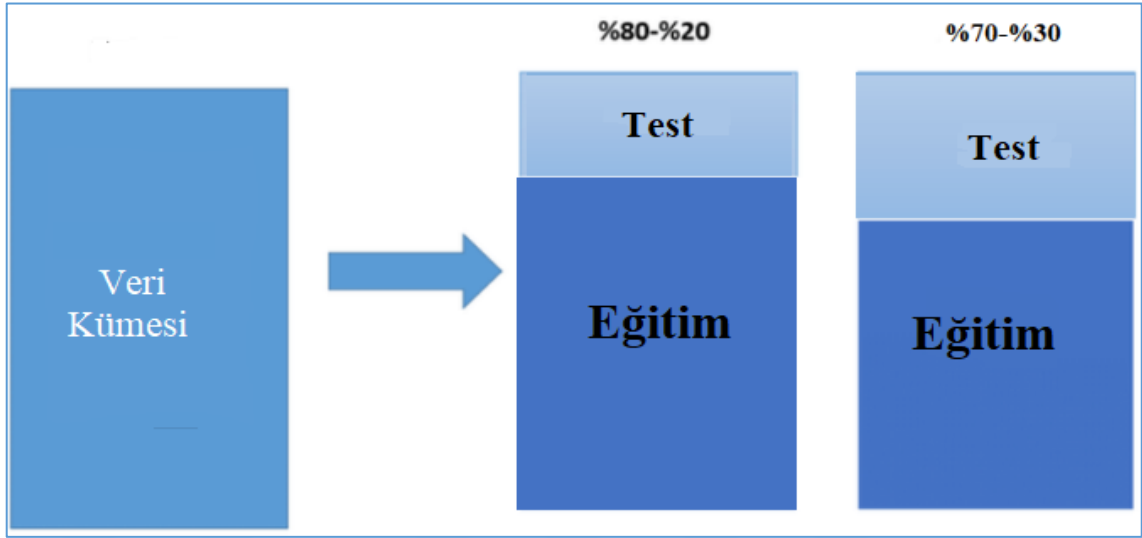
Bu çalışmada, eşit dağılımlı IMDB ve Rotten Tomatoes veri kümesi yanı sıra eşit dağılıma sahip olmayan Twitter veri kümesi kullanılmıştır. Deneysel çalışmalar ve sonuçları ile tez süresince yapılan diğer çalışmalar ismindeki alt bölümlerde Doğruluk performans metriği yanında diğer performans metriklerine de yer verilmiştir.

Sınıflandırıcı algoritmalar ile model kurmak ve daha sonra bu modellerin performansını değerlendirmek için veri kümeleri test ve eğitim olarak ikiye ayrılmıştır. Çalışmalarda sına testi (hold-out) ve k-kat çapraz doğrulama (cross-validation) eğitim-test ayırma yöntemleri tercih edilmiştir. Çalışmada kullanılan k-kat çapraz doğrulama yöntemi ile eğitim-test ayırma yöntemi, Şekil 4.31'de gösterilmektedir. Şekil 4.32'de test ve eğitim ayırımının sına testi olarak isimlendirilen rastgele ayırma gösterilirken; Şekil 4.33'te ise sına testi ayırımının model eğitimi sırasında yapılmasına ait yöntem gösterilmiştir [148].



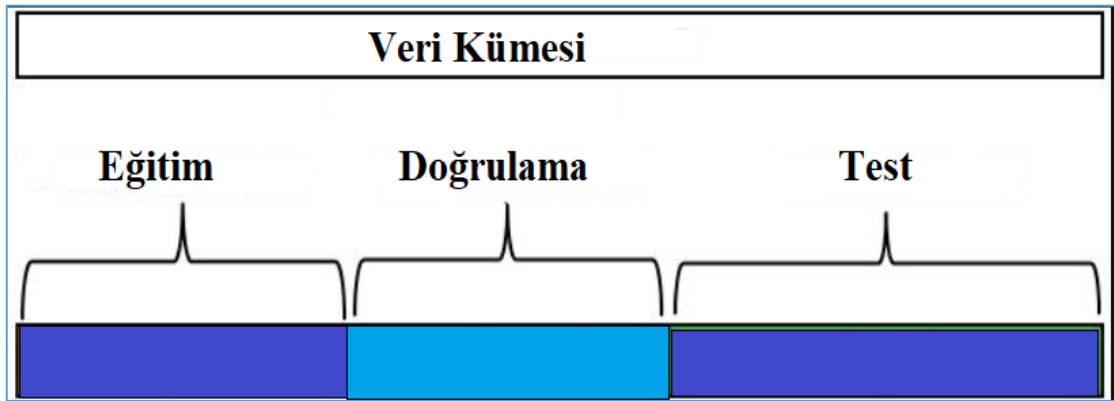
Şekil 4.30. Çapraz doğrulama yöntemi ve test ve eğitim kümesi ayrımı [148].

Şekil 4.30’da görüldüğü gibi eğitim ve test verisi 5 kata bölünerek her katta test veri kümesi değiştirilerek model performansı araştırılmaktadır [148].



Şekil 4.31. Sınama testi yöntemi ve test ve eğitim kümesi ayrımı [148].

Şekil 4.31’de görüldüğü gibi eğitim ve test verisi yüzdesel olarak bölünmüştür. %80 eğitim ve %20 test ayrımı ile %70 eğitim ve %30 test veri kümesi olarak ayırım yapılmıştır [148]. Şekil 4.31’e alternatif olarak Şekil 4.32’de gösterildiği gibi eğitim veri kümesi başlangıçta veya model eğitimi sırasında, belirli bir bölümü de doğrulama için ayrılabilir.



Şekil 4.32. Sınama testi yöntemi ve test, eğitim ve doğrulama kümeleri ayrımı [148].

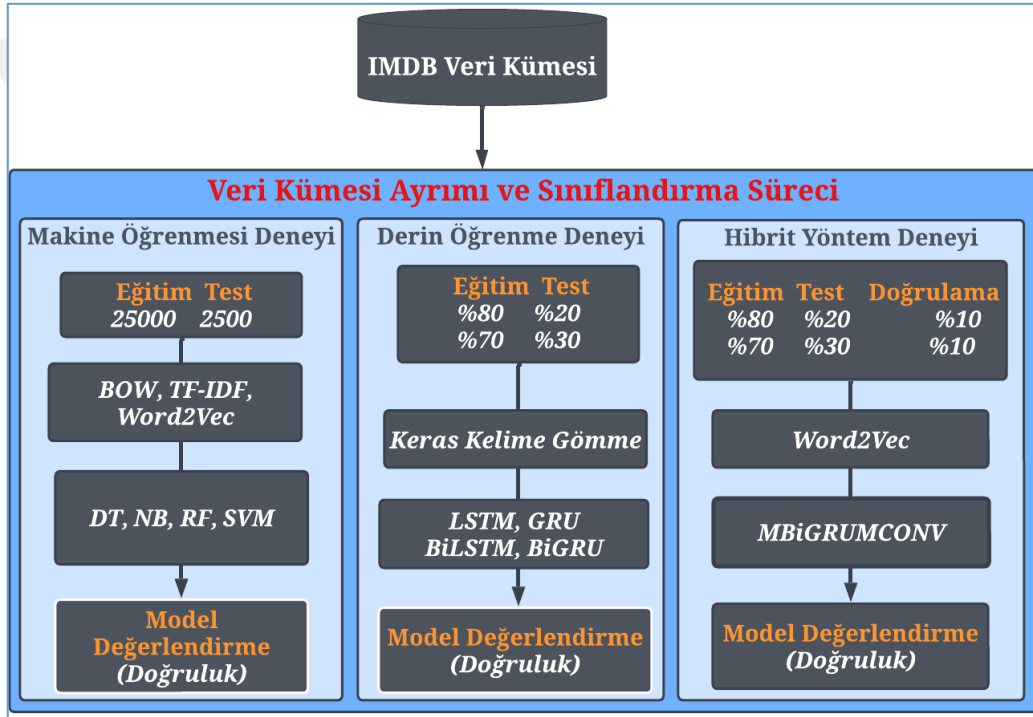
Şekil 4.32’deki eğitim kümesinin belirli bir bölüme ayrılmasına, model eğitimine başlamadan doğrulama küme ayırımı (validation_split) metoduyla da doğrulama ayırımı yapılabilir [149].

5. DENEYSEL ÇALIŞMALAR VE SONUÇLARI

Çalışma kapsamında kullanılan IMDB [4], Rotten Tomatoes [5], TripAdvisor [6] ve Twitterdan toplanan Covid-19 [7] veri kümeleri üzerinde yapılan deneysel çalışmalara ait detay bilgiler bu bölümde anlatılmıştır.

5.1. IMDB VERİ KÜMESİ ÜZERİNDE YAPILAN ÇALIŞMALAR

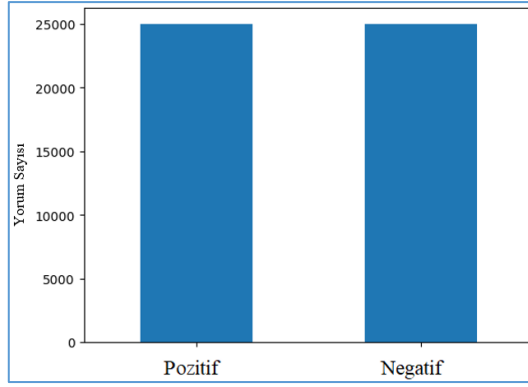
Bu bölümde IMDB [4] veri kümesi üzerinde yapılan çalışmalar anlatılmıştır. Şekil 5.1’de çalışmanın akış diyagramı görülmektedir.



Şekil 5.1. IMDB veri kümesi üzerinde yapılan çalışmaların akış diyagramı.

5.1.1. Deneyde Kullanılan Veri Kümesi

IMDB veri kümesi, 25 000 olumlu ve 25 000 olarak etiketlenmiş film yorumlarının bulunduğu ve duygu analizi çalışmalarında popüler olarak kullanılan Stanford araştırmacıları tarafından toplanmış, güvenilir ve dengeli bir veri kümesidir [4]. Veri kümesinin duygu sınıfı dağılımı Şekil 5.2’de görülmektedir.



Şekil 5.2. IMDB veri kümesi duygu sınıfı dağılımı.

Şekil 5.2’de görüldüğü üzere IMDB veri kümesinin [4] sınıf dağılımı dengeli olduğu için bu bölümde yapılan tüm deneylerde, performans değerlendirilmesinde sadece doğruluk kullanılmıştır. Stanford araştırmacıları tarafından etiketlenmiş ve yaygın kullanımından dolayı önerilen mimari (MBiGRUMCONV) başta olmak üzere, tüm deneyleri literatürdeki çalışmalarla karşılaştırabilmek için bu veri kümesi tercih edilmiştir. Çizelge 5.1’de IMDB veri kümesi içerisindeki özniteliklerle ilgili bilgiler verilmiştir.

Çizelge 5.1. IMDB veri kümesi.

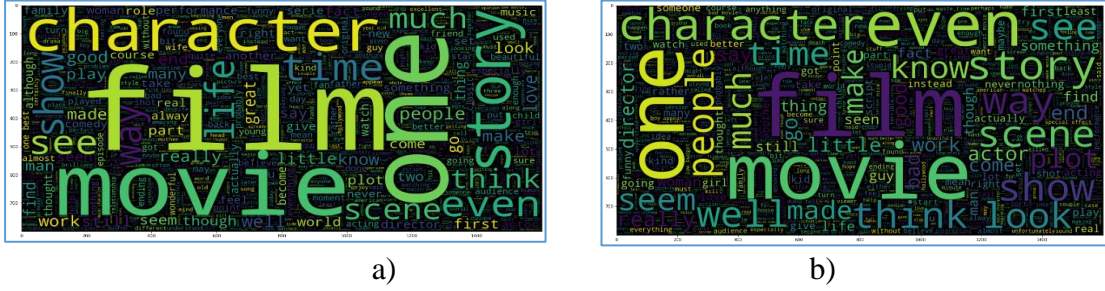
Öznitelik Adı	Açıklama	Öznitelik Türü
Sentiment	Metinlerin etiketlenmiş duygu sınıfları	Tam sayı
Review	Yorumlar	Metin

Çizelge 5.1’de duygu sınıfı özniteliğinde; bir ise pozitif, sıfır ise negatif duygu sınıfına sahip olduğu şeklinde işaretleme yapılmıştır. Çizelge 5.2’de veri kümesine ait yorum sayısı ve kelime uzunluğu gibi ilgili daha detaylı bilgiler yer almaktadır.

Çizelge 5.2. IMDB veri kümesi detaylı bilgisi.

Parametre	Değerler
Sınıf Sayısı	2
Yorum sayısı	50 000
Pozitif yorum	25 000
Negatif yorum	25 000
Maksimum yorum uzunluğu	13 704
Minimum yorum uzunluğu	6
Benzersiz kelime sayısı	49 576

Veri kümesinde yer alan yorumların, negatif ve pozitif kelime dağılımlarına ait kelime bulutu dağılımı Şekil 5.3'te verilmiştir.



Şekil 5.3. Yorumların kelime bulutu dağılımı a) pozitif yorumlar b) negatif yorumlar.

Şekil 5.3'te görüldüğü üzere pozitif ve negatif etiketli verilerin dağılımında “good”, ”great” gibi kelimeler pozitif etiketli; “even”, “though” gibi kelimeler negatif etiketlenmiştir. “film”, ”scene”, “story” gibi kelimelerin ise hem pozitif hem negatif etiketlendiği kelime bulutunda görülmektedir.

5.1.2. Makine Öğrenmesi Deneyi

Stanford araştırmacıları tarafından toplanan IMDB veri kümesi [4] üzerinde ön işleme sürecinden sonra, çeşitli metin temsil yöntemleri ve çeşitli makine öğrenmesi yöntemleriyle duygu analizi çalışması yapılmıştır. Bu bölümde yapılan çalışma anlatılacaktır.

5.1.2.1. Ön işleme ve Metin Temsili

Bu deneyde, materyal metot bölümünde anlatılan metin ön işleme süreçleri uygulanmıştır. Bu işlemler için Python'da yer alan NLTK kütüphanesi kullanılmıştır.

IMDB veri kümesinin Stanford araştırmacıları tarafından oluşturulduğu şekliyle; yirmi beş bin eğitim, yirmi beş bin test kümesine ayrılarak duygu analizi çalışması yapılmıştır.

Model oluşturmadan önce veri kümesi; BOW, TF-IDF, Word2Vec gibi metin temsil yöntemleriyle vektörleştirilmiştir. TF-IDF ve BOW metin temsilleri oluşturmak için scikit-learn [132] kütüphanesinin 1.0.2 versiyonu kullanılmıştır. Word2Vec metin temsili için Gensim [133] kütüphanesi kullanılmıştır. Bu deneyde, Çizelge 5.3'te verilen Word2Vec Parametreleri kullanılmıştır.

Çizelge 5.3. Word2Vec yöntemi parametreleri.

Parametre	Değerler
Size	100
Window	5
Workers	8
min_count	2
sg	0

Çizelge 5.3'te verilmiş olan Word2Vec parametrelerinden vektör boyutu olarak 100, pencere boyutu için beş, veri kümesinde geçen minimum kelime sayısı iki ve CBOW yöntemi kullanılmıştır. Çizelge 5.4'te bu deneyde kullanılan diğer temsil yöntemlerinden olan TF-IDF ve BOW yöntemlerine ait parametreler yer almaktadır.

Çizelge 5.4. TF-IDF ve BOW yöntemi parametreleri.

Parametre	Değer
max_count	10 000

5.1.2.2. Deney Kurulumu

Veri kümesinin eğitim-test ayrımından sonra, ön işleme ve metin temsil yöntemleriyle vektörleştirdikten sonra SVM, NB, DT ve RF makine öğrenmesi algoritmalarıyla duygu analizi için modeller oluşturulmuştur.

5.1.2.3. Deney Sonuçları

Bu modellerden elde edilen doğruluk performans karşılaştırılması Çizelge 5.5'te verilmiştir. Modellerin sonuçları yerel bilgisayarda alınmıştır.

Çizelge 5.5. Doğruluk performans sonuçları (%).

	BOW	TF-IDF	Word2Vec
SVM	80	84	88
NB	82	83	85
DT	71	73	74
RF	81	83	84

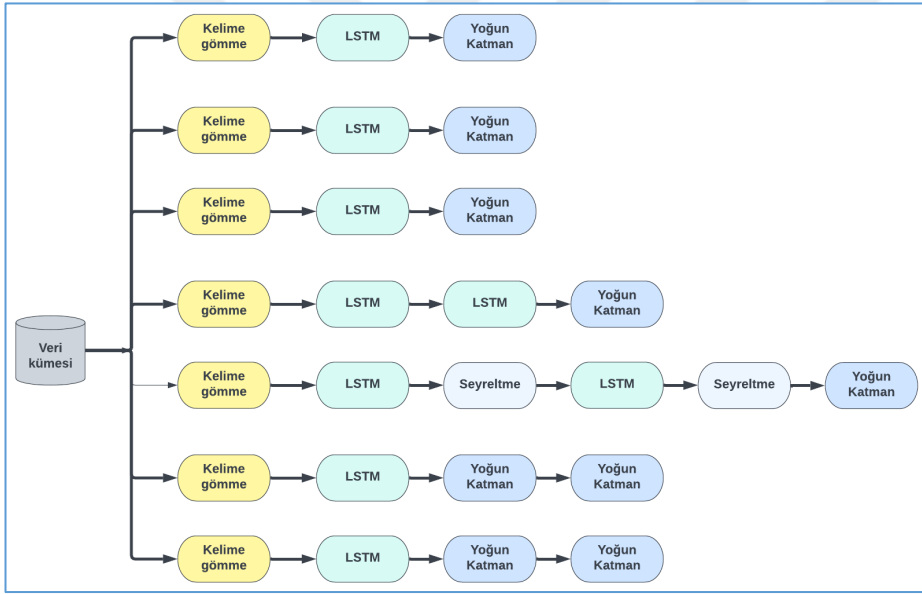
Çizelge 5.5'te görüldüğü üzere metin temsil yöntemlerinden sonra makine öğrenmesiyle oluşturulan modellerden SVM ve Word2Vec ile olanlar, diğer modellerden daha iyi sonuç vermiştir. Çizelge 5.5'te BOW temsil sonrası oluşturulan makine öğrenmesi modelinde

NB, RF, SVM, DT sıralamasıyla doğruluk performansı vermiştir. TF-IDF sonrasında SVM ilk sırada gelirken, NB ve RF eşit ve ikinci, DT ise son sırada doğruluk performansı vermiştir. Word2Vec metin temsili sonrası makine öğrenme modellerinin performansları sırasıyla, SVM, NB, RF ve DT şeklindedir.

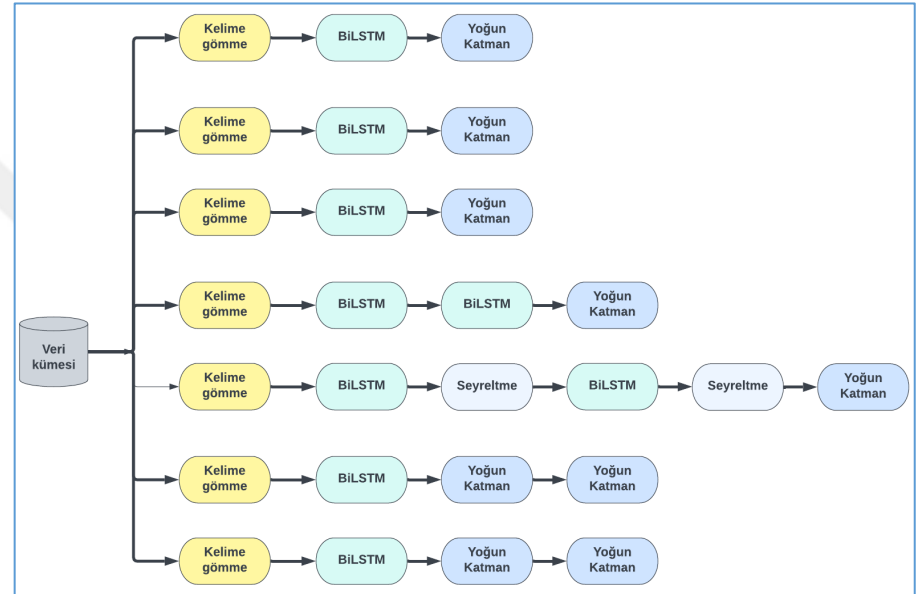
5.1.3. Derin Öğrenme Deneyi

IMDB [4] veri kümesi üzerinde çeşitli optimizasyon fonksiyonları, dönem, küme boyutu gibi parametrelerde çeşitli sayıda Yoğun Katman, Seyreltme ve çeşitli derin öğrenme sinir ağlarıyla (GRU, LSTM, BiLSTM, BiGRU) oluşturulan birden çok katmandan oluşan farklı yedi model oluşturulmuştur. Şekil 5.4'te oluşturulan bu yedi modelin diyagram olarak gösterimleri yer almaktadır. Bu deneyde aynı küme boyutu ve dönemde farklı optimizasyon ile farklı aktivasyon fonksiyonlarının model performansına etkileri araştırılmıştır. Bu yüzden, Model 2 ve Model 3'te optimizasyon fonksiyonu Adadelta kullanılırken; diğer modellerde Adam kullanılmıştır. Aktivasyon fonksiyonu olarak Model 5'te ReLU, Model 6'da Softmax alınırken; diğer modellerde Sigmoid kullanılmıştır.

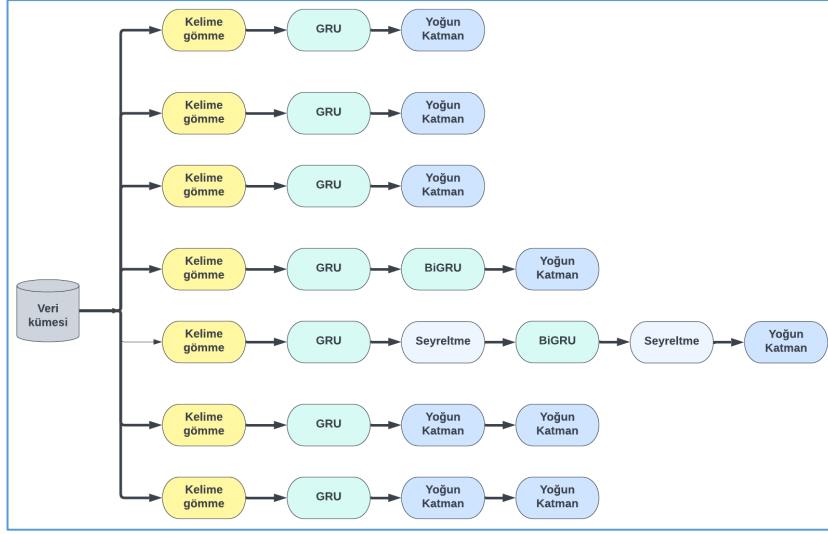




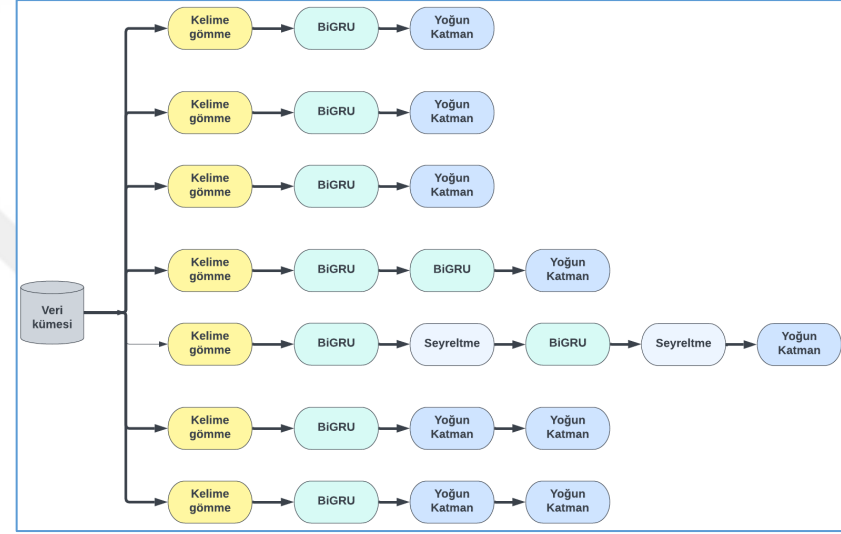
a) LSTM 7 model



b) BiLSTM 7 model



c) GRU 7 model



d) BiGRU 7 model

Şekil 5.4. Derin öğrenme deneyi için oluşturulan 7 model diyagramı a) LSTM b) BiLSTM c) GRU d) BiGRU.

5.1.3.1. Ön işleme ve Metin Temsili

IMDB veri kümesi, bu çalışmada Keras kelime gömme ile kullanılmıştır. Metin ön işleme ve metin madenciliği bölümünde anlatıldığı gibi işlemler yapılmıştır.

5.1.3.2. Deney Kurulumu

Deneyisel analizde, ön işleme ve öznitelik çıkarımı yapılan IMDB veri kümesi, Sınama testi (%80-%20 ve %70-%30) eğitim-test ayırımından sonra topluluk ve makine öğrenmesi sınıflandırıcıları ile sınıflandırılmış ve ardından performanslar, doğruluk kriteri ile değerlendirilmiştir. Bütün modellerde, küme boyutu değeri 128, dönem değeri beş olarak alınmıştır. Deneyler Google Colab [128] üzerinde yapılmıştır.

5.1.3.3. Deney Sonuçları

Çalışmada elde edilen sonuçlar Çizelge 5.6 ile Çizelge 5.13 arasında verilmiştir. Bu Çizelgeler, kullanılan optimizasyon fonksiyonları, küme boyutu, katmanlar ve alınan doğruluk değerlerini içerir.



Çizelge 5.6'da görüldüğü gibi LSTM ile oluşturulan yedi model arasından, LSTM ve iki Yoğun katmanlarından oluşan model 6'da %87,26 doğruluk değeriyle en iyi sonuç alınmıştır. Yoğun katman sayısındaki artış, LSTM sinir ağı katman sayısındaki artış kadar model performansını olumlu etkilememiştir

Çizelge 5.6. Derin öğrenme deneyi LSTM doğruluk sonucu (%80 eğitim, %20 test).

LSTM	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	LSTM (128) Yoğun (1)	LSTM (128) Yoğun (1)	LSTM (128) Yoğun (1)	LSTM (128) LSTM (128) Yoğun (1)	LSTM (128) Seyreltme (0,3) LSTM (128) Seyreltme (0,3) Yoğun (1)	LSTM (128) Yoğun (16) Yoğun (1)	LSTM (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	87,04	79,05	87,11	87,69	86,66	87,26	82,17

Çizelge 5.7'de görüldüğü gibi LSTM ile oluşturulan yedi model arasında, iki LSTM ve Yoğun katmanlarıyla oluşturulan model 4, %87,85 doğruluk değeriyle en iyi sonucu verdiği görülmüştür. LSTM katman sayısındaki artış doğruluk değerini artırırken, Yoğun katman sayısındaki artış model performansını olumlu yönde etkilememiştir. Ayrıca Seyreltme katmanının kullanılması, model performanslarını olumlu etkilemektedir.

Çizelge 5.7. Derin öğrenme deneyi LSTM doğruluk sonucu (%70 eğitim, %30 test).

LSTM	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	LSTM (128) Yoğun (1)	LSTM (128) Yoğun (1)	LSTM (128) Yoğun (1)	LSTM (128) LSTM (128) Yoğun (1)	LSTM (128) Seyreltme (0,3) LSTM (128) Seyreltme (0,3) Yoğun (1)	LSTM (128) Yoğun (16) Yoğun (1)	LSTM (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	87,59	87,06	87,66	87,85	87,70	87,81	87,16

Çizelge 5.8'de görüldüğü gibi GRU ile oluşturulan yedi modelden, iki GRU ve Yoğun katmanlarından oluşan model 4, %88,10 doğruluk ile en iyi sonucu vermiştir. GRU sinir ağı katman sayısındaki artış doğruluk değerini artırırken, Yoğun katman sayısındaki artış, GRU sinir ağı katman sayısındaki artış kadar model performansını olumlu etkilememiştir.

Çizelge 5.8. Derin öğrenme deneyi GRU doğruluk sonucu (%80 eğitim, %20 test).

GRU	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	GRU (128) Yoğun (1)	GRU (128) Yoğun (1)	GRU (128) Yoğun (1)	GRU (128) GRU (128) Yoğun (1)	GRU (128) Seyreltme (0,3) GRU (128) Seyreltme (0,3) Yoğun (1)	GRU (128) Yoğun (16) Yoğun (1)	GRU (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	86,76	82,71	86,84	88,10	87,62	87,21	88,16

Çizelge 5.9'da görüldüğü gibi GRU ile oluşturulan yedi modelden, iki GRU ve Yoğun katmanlarından oluşan model 4, %88,13 doğruluk değeri ile en iyi sonucu vermiştir. GRU sinir ağı katman sayısındaki artış doğruluk değerini artırırken, Yoğun katman sayısındaki artış GRU sinir ağı katman sayısındaki artış kadar model performansını olumlu etkilememiştir.

Çizelge 5.9. Derin öğrenme deneyi GRU doğruluk sonucu (%70 eğitim, %30 test).

GRU	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	GRU (128) Yoğun (1)	GRU (128) Yoğun (1)	GRU (128) Yoğun (1)	GRU (128) GRU (128) Yoğun (1)	GRU (128) Seyreltme (0,3) GRU (128) Seyreltme (0,3) Yoğun (1)	GRU (128) Yoğun (16) Yoğun (1)	GRU (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	87,69	81,27	87,85	88,13	87,69	88,04	88,26

Çizelge 5.10'da görüldüğü gibi BiLSTM ile oluşturulan yedi modelden, iki BiLSTM iki Seyreltme ve Yoğun katmanlarından oluşan model 5, %88,21 doğruluk değeriyle en iyi sonucu vermiştir. BiLSTM sinir ağı katman sayısındaki artış doğruluk değerini artırırken, Yoğun katman sayısındaki artış kadar model performansını olumlu etkilememiştir. Adam optimizasyon fonksiyonu diğer optimizasyon fonksiyonlarına göre model performansları üzerinde olumlu etki göstermiştir.

Çizelge 5.10. Derin öğrenme deneyi BiLSTM doğruluk sonucu (%80 eğitim, %20 test).

BiLSTM	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	BiLSTM (128) Yoğun (1)	BiLSTM (128) Yoğun (1)	BiLSTM (128) Yoğun (1)	BiLSTM (128) BiLSTM (128) Yoğun (1)	BiLSTM (128) Seyreltme (0,3) BiLSTM (128) Seyreltme (0,3) Yoğun (1)	BiLSTM (128) Yoğun (16) Yoğun (1)	BiLSTM (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	88,08	80,03	87,64	86,72	88,21	87,60	87,70

Çizelge 5.11'de görüldüğü gibi BiLSTM ile oluşturulan yedi modelden, iki BiLSTM, iki Seyreltme ve Yoğun katmanlarından oluşan model 5, %87,72 doğruluk değeri ile en iyi sonucu vermiştir. BiLSTM sinir ağı katman sayısı ve Seyreltme katman sayısındaki artış, model performansını olumlu yönde etkilemiştir. Ayrıca Adam optimizasyon fonksiyonu ve iki Seyreltme kullanılan bir önceki deneyde olduğu gibi (Çizelge 5.10) performans artışına etki ettiği gözlemlenmiştir.

Çizelge 5.11. Derin öğrenme deneyi BiLSTM doğruluk sonucu (%70 eğitim, %30 test).

BiLSTM	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	BiLSTM (128) Yoğun (1)	BiLSTM (128) Yoğun (1)	BiLSTM (128) Yoğun (1)	BiLSTM (128) BiLSTM (128) Yoğun (1)	BiLSTM (128) Seyreltme (0,3) BiLSTM (128) Seyreltme (0,3) Yoğun (1)	BiLSTM (128) Yoğun (16) Yoğun (1)	BiLSTM (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	87,54	79,45	87,01	87,32	87,72	87,30	86,65

Çizelge 5.12'de olduğu gibi BiGRU ile oluşturulan yedi modelden, iki BiGRU ve Yoğun katmanlarından oluşan model 4, %88,20 doğruluk değeri ile en iyi sonucu vermiştir. BiGRU sinir ağı katman sayısındaki artış, Seyreltme veya Yoğun katman sayısındaki artışa göre daha iyi bir model performansı alınmasını sağlamıştır. Ayrıca, adam optimizasyon fonksiyonu diğer optimizasyon fonksiyonlarına göre model performansları üzerinde olumlu etki göstermiştir.

Çizelge 5.12. Derin öğrenme deneyi BiGRU doğruluk sonucu (%80 eğitim, %20 test).

BiGRU	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Foksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	BiGRU (128) Yoğun (1)	BiGRU (128) Yoğun (1)	BiGRU (128) Yoğun (1)	BiGRU (128) BiGRU (128) Yoğun (1)	BiGRU (128) Seyreltme (0,3) BiGRU (128) Seyreltme (0,3) Yoğun (1)	BiGRU (128) Yoğun (16) Yoğun (1)	BiGRU (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	88,15	86,51	87,19	88,20	86,63	87,43	86,57

Çizelge 5.13'te olduğu gibi BiGRU ile oluşturulan yedi modelden, iki BiGRU ve Yoğun katmanlarından oluşan model 4, %88,18 doğruluk ile en iyi sonucu vermiştir. BiGRU sinir ağı katmanlarının sayısındaki artış, Seyreltme veya Yoğun katmanların sayısındaki artıştan daha iyi bir model performansı alınmasını sağlamıştır. Ayrıca, adam optimizasyon fonksiyonu diğer optimizasyon fonksiyonlarına göre model performansları üzerinde olumlu etki göstermiştir.

Çizelge 5.13. Derin öğrenme deneyi BiGRU doğruluk sonucu (%70 eğitim, %30 test).

BiGRU	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
Optimizasyon Foksiyonu	Adam	Adadelta	Adadelta	Adam	Adam	Adam	Adam
Aktivasyon Fonksiyonu	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid	ReLU	Softmax
Küme Boyutu	128	128	128	128	128	128	128
Dönem	5	5	5	5	5	5	5
Katmanlar	BiGRU (128) Yoğun (1)	BiGRU (128) Yoğun (1)	BiGRU (128) Yoğun (1)	BiGRU (128) BiGRU (128) Yoğun (1)	BiGRU (128) Seyreltme (0,3) BiGRU (128) Seyreltme (0,3) Yoğun (1)	BiGRU (128) Yoğun (16) Yoğun (1)	BiGRU (128) Yoğun (16) Yoğun (1)
Doğruluk (%)	87,70	80,46	86,24	88,18	86,11	87,37	85,91

Derin öğrenme deneyinde, en iyi sonuçları veren %80-%20 eğitim-test ayrımında elde edilen BiLSTM (Çizelge 5.10'daki model 5) ve BiGRU (Çizelge 5.12'deki model 4) modelleri vermişyir. Bu deneyde alınan en iyi model sonuçlarıyla, IMDB ile daha önce yapılan çalışmalarla oluşturulan modellere ait metin temsil yöntemi ve alınan doğruluk performans sonucuna yönelik karşılaştırılma Çizelge 5.15'te verilmiştir.

Çizelge 5.14. Derin öğrenme deneyini önceki çalışmalarla karşılaştırma.

Referans	Metin Temsil Yöntemleri	Model	Doğruluk (%)
[20]	TF-IDF, BOW, Keras kelime gömme	CNN, RNN, LSTM, LR, MNB, SVM, XGBoost	88 (RNN)
[16]	Word2Vec	RNN	87
[150]	BOW	MLP	86,67
[33]	TF-IDF	BNB, LR, MNB, RF, SGD, SVM	83,66 (RF)
[19]	D-boyutlu yoğun vektör, n girdi, $d \times n$ özellik matrisi	CNN	87,90
En iyi Model	Keras kelime gömme	BiLSTM (Çizelge 5.10'daki model 5)	88,21
		BiGRU (Çizelge 5.12'deki Model 4)	88,20

Deney kapsamında yapılan genel çıkarımlar sıralanacak olursa;

- Adam optimizasyon fonksiyonu, aynı küme boyutu ve dönemde, Adadelta fonksiyonuna göre daha iyi performans göstermiştir.
- Yoğun katman sayısındaki artışa rağmen GRU veya LSTM sinir ağı katman sayısındaki artış model performansını artırdığı gözlemlenmiştir.
- Seyreltme kullanımının performansa olumlu etkisi olmasına rağmen; BiGRU sinir ağının tek başına kullanılmasıyla daha iyi sonuçlar alınmıştır.

GRU ve LSTM sinir ağı kullanımını yerine BiGRU veya BiLSTM kullanımının duygu analizi için daha yüksek performanslı modeller oluşturmada kullanılabileceği görülmüştür. Bu deney, duygu analizi üzerine oluşturulacak hibrit model için BiGRU seçilmesine sebep olmuştur.

5.1.4. Çok Katmanlı BiGRU ve Çok Katmanlı CNN Hibrit (MBiGRUMCONV) Deneyi

5.1.4.1. Ön işleme ve Metin Temsili

Metin ön işleme ve metin madenciliği bölümünde bu işlemler anlatılmıştır. Kelime

vektörleştirme işlemleri için Word2vec'in CBOW ve Skip-Gram yöntemleri 100, 200, 300 vektör boyutlarında kullanılmıştır,

5.1.4.2. Deney Kurulumu

Oluşturulan modele ilişkin deneyler, Google Colaboratory (Colab) [142] üzerinde TensorFlow 2.9.0 [61] ve Keras 2.9 [143] kütüphaneleri ve Python'ın 3.9.13 [144] versiyonu kullanılarak yapılmıştır. Yapılan deneylerin daha hızlı ve kesintisiz sonuçlanması için Colab Pro sürümü kullanılmıştır. Ayrıca daha iyi sonuç veren bir model oluşturmak için çeşitli geri çağırım (Callback) kullanılmıştır. Bunlar; ReduceLRonPlateau [152] ve Erken Durdurma (Early Stopping) [143]'tir.

Modelin eğitimi süresince takip edilen doğrulama doğruluk değerinde bir iyileşme görülmezse, eğitim işlemi belirtilen adım sayısından önce durdurulur. Bu işleme Erken Durdurma denir [113]. Keras kütüphanesinde yer alan Erken Durdurma fonksiyonu ile yapılır. Çalışmada beş adım boyunca izlenen doğrulama doğruluk değeri (validation accuracy) değerinde bir iyileşme olmuyorsa eğitim durdurulur [112].

Eğitim esnasında takip edilen doğrulama kayıp değeri (validation loss) boyunca iyileşme yoksa, öğrenme hızı belirlenen bir katsayıyla çarpılarak küçültülür. Çalışmada beş adım boyunca doğrulama kayıp verisi başarımında bir iyileşme olmadığında öğrenme hızı 0,1 ile çarpılarak azaltılmıştır. Literatür taraması başlığı altında anlatılan IMDB veri kümesi üzerinde yapılan önceki çalışmalar göz önüne alınarak parametreler belirlenmiştir. Bu parametreler Çizelge 5.15'te gösterilmektedir.

Çizelge 5.15. Modelde kullanılan parametreler.

Parametreler	Değerler
Küme Boyutu	32, 64 , 128
Dönem Sayısı	10, 20 , 30
Aktivasyon Fonksiyonu	ELU, Leaky ReLU, ReLU , Sigmoid, tanh
Optimizasyon Fonksiyonu	Adam , RMSprop, SGD
Sinir Ağı Düğüm Sayısı ve Filtre Sayısı	64, 128 , 256
Çekirdek Değeri	3, 4, 5
Seyreltme Oranı	0,3 ; 0,4; 0,5; 0,6
Adım	1 , 2
Dolgu	same , valid

Çizelge 5.15'teki parametrelerin en optimumunu bulmak için Grid Search'ü grafikleştirerek sunan Tensorflow'un sunduğu Tensorboard kullanılmıştır [154].

TensorBoard, makine öğrenme çalışmalarında kayıp ve doğruluk, model grafiği

görselleştirme, hızlı ve çok parametrelili model kurulumu gibi deney metriklerinin izlenmesini sağlayan Tensorflow'un bir aracıdır [153]. TensorBoard kütüphanesi kullanılarak Çizelge 5.15'in ilk sütunundaki parametrelere ikinci sütundaki değerlerin iç içe döngüler şeklinde atamaları yapılarak çalıştırılmış ve her bir parametre için en optimum değerin bulunması sağlanmıştır. Bu optimum değerler, Çizelge 5.15'te kalınlaştırılarak gösterilmiştir.

Model oluşturulurken, küme boyutu, dönem sayısı, aktivasyon fonksiyonu, optimizasyon fonksiyonu, düğüm sayısı ve Evrişimli sinir ağı filtre sayısı, çekirdek sayısı ve adım değeri, seyreltme oranı gibi parametrelere, verilen çeşitli değerlerden en optimum değerlerin bulunması hedeflenmiştir. Bu işleme hiperparametre optimizasyon işlemi denir.

Her hiperparametre için farklı değerlerden oluşan rasgele değişkeni model mimarisine yerleştirerek, BiGRU katmanındaki sinir ağı sayısı (num_units), iki Evrişimli sinir ağı katmanında yer alan filtre sayısı, çekirdek değeri, adım ve dolgu değerleri, optimizasyon fonksiyonu (optimizer), küme boyutu, dönem sayısı gibi parametrelerin en optimumunun seçilmesi hedeflenmiştir. Her parametreye ait kombinasyonların çalışma kayıtları (log runs) bir üst dizinde kaydedilir. Bu işleme çalışma kayıtlarının örnek görüntüsü Şekil 5.5'te verilmiştir.

```

--- Starting trial: run-0
{'num_units': 64, 'kernel': 3, 'optimizer': 'adam', 'activation': 'relu', 'epoch': 10, 'batch': 32, 'dropout': 0.3}
Epoch 1/10
1125/1125 [=====] - 6s 5ms/step - loss: 0.4940 - accuracy: 0.7735 - val_loss: 0.3443 - val_accuracy: 0.8508
Epoch 2/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3591 - accuracy: 0.8395 - val_loss: 0.3378 - val_accuracy: 0.8508
Epoch 3/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3340 - accuracy: 0.8546 - val_loss: 0.3185 - val_accuracy: 0.8602
Epoch 4/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3189 - accuracy: 0.8609 - val_loss: 0.3182 - val_accuracy: 0.8633
Epoch 5/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3021 - accuracy: 0.8692 - val_loss: 0.3164 - val_accuracy: 0.8637
Epoch 6/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.2943 - accuracy: 0.8744 - val_loss: 0.3162 - val_accuracy: 0.8605
Epoch 7/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.2818 - accuracy: 0.8815 - val_loss: 0.3130 - val_accuracy: 0.8677
Epoch 8/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.2755 - accuracy: 0.8852 - val_loss: 0.3144 - val_accuracy: 0.8593
Epoch 9/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.2652 - accuracy: 0.8898 - val_loss: 0.3178 - val_accuracy: 0.8595
Epoch 10/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.2611 - accuracy: 0.8902 - val_loss: 0.3209 - val_accuracy: 0.8635
313/313 [=====] - 1s 2ms/step - loss: 0.3212 - accuracy: 0.8604
--- Starting trial: run-1
{'num_units': 64, 'kernel': 3, 'optimizer': 'adam', 'activation': 'relu', 'epoch': 10, 'batch': 32, 'dropout': 0.5}
Epoch 1/10
1125/1125 [=====] - 6s 5ms/step - loss: 0.5557 - accuracy: 0.7514 - val_loss: 0.3659 - val_accuracy: 0.8407
Epoch 2/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3849 - accuracy: 0.8279 - val_loss: 0.3437 - val_accuracy: 0.8482
Epoch 3/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3641 - accuracy: 0.8390 - val_loss: 0.3346 - val_accuracy: 0.8510
Epoch 4/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3521 - accuracy: 0.8452 - val_loss: 0.3236 - val_accuracy: 0.8612
Epoch 5/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3399 - accuracy: 0.8519 - val_loss: 0.3243 - val_accuracy: 0.8595
Epoch 6/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3317 - accuracy: 0.8553 - val_loss: 0.3435 - val_accuracy: 0.8500
Epoch 7/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3245 - accuracy: 0.8609 - val_loss: 0.3174 - val_accuracy: 0.8625
Epoch 8/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3190 - accuracy: 0.8626 - val_loss: 0.3209 - val_accuracy: 0.8618
Epoch 9/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3123 - accuracy: 0.8652 - val_loss: 0.3142 - val_accuracy: 0.8655
Epoch 10/10
1125/1125 [=====] - 5s 5ms/step - loss: 0.3086 - accuracy: 0.8671 - val_loss: 0.3168 - val_accuracy: 0.8630
313/313 [=====] - 1s 2ms/step - loss: 0.3129 - accuracy: 0.8657

```

Şekil 5.5. Hiperparametre kombinasyonlarının çalışma kayıtları.

Şekil 5.5'te farklı hiperparametre kombinasyonlarının nasıl denendiğine dair bir ekran görüntüsü gösterilmiştir. Denenen parametrelere ait

kaydedilen Yapılan işlemlere dair işlem kayıtlarının performansları Tensorflow hiperparametre menüsü (hparams dashboard) ile görselleştirilmesine ait çıktılar Şekil 5.6’da gösterilmiştir.

The screenshot shows the TensorBoard HParams dashboard. The left sidebar contains a tree view of hyperparameters and their values, along with filters for metrics, status, and sorting. The main area displays a table of hyperparameter configurations and their resulting accuracies.

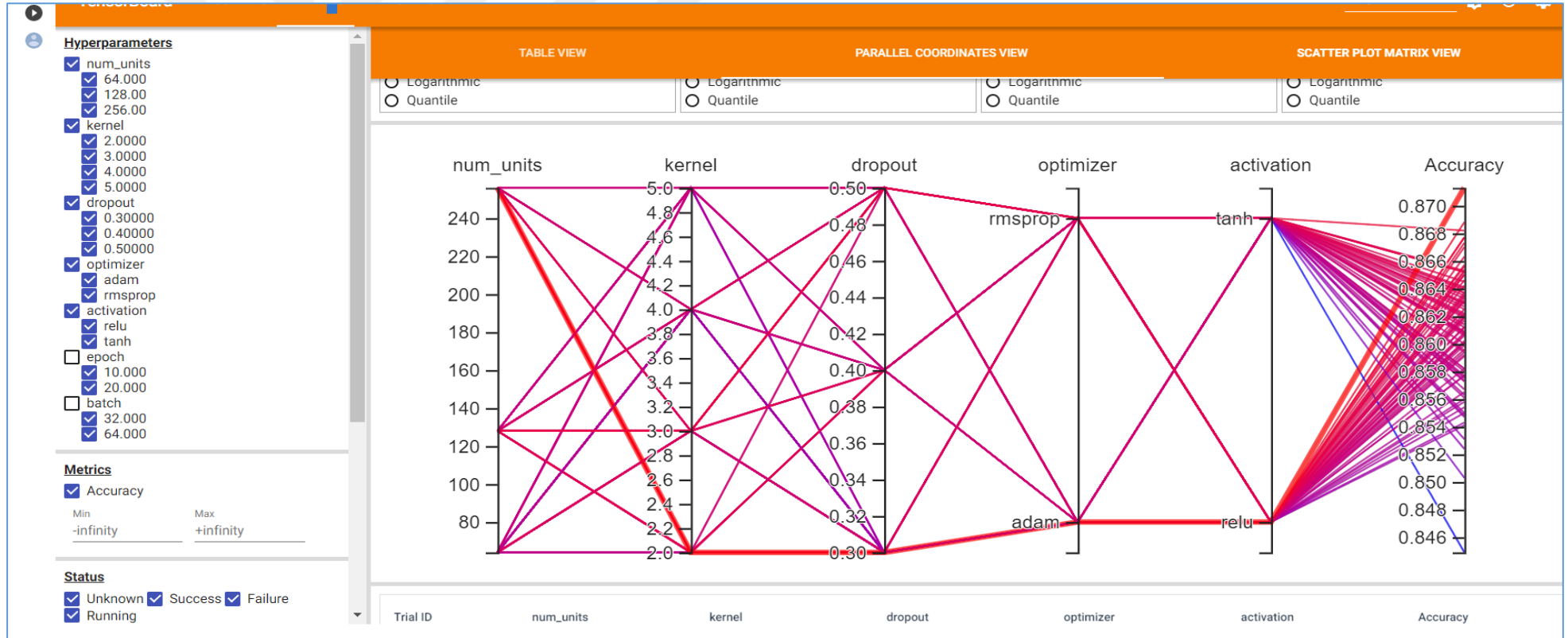
Trial ID	Show Metrics	num_units	kernel	dropout	optimizer	activation	Accuracy
041c86d6c65f5...	<input type="checkbox"/>	128.00	5.0000	0.30000	adam	relu	0.86090
04da2413b8313...	<input type="checkbox"/>	128.00	5.0000	0.50000	adam	relu	0.86180
0a49c1aafc746...	<input type="checkbox"/>	128.00	3.0000	0.30000	adam	relu	0.85750
130436743fe3e...	<input type="checkbox"/>	128.00	5.0000	0.50000	adam	relu	0.86200
14368c2c54373...	<input type="checkbox"/>	64.000	3.0000	0.50000	adam	relu	0.86200
21bec00a54077...	<input type="checkbox"/>	64.000	3.0000	0.50000	adam	relu	0.86710
384bf8c053f88...	<input type="checkbox"/>	64.000	3.0000	0.30000	adam	relu	0.86170
3b73f13d4851e...	<input type="checkbox"/>	128.00	3.0000	0.30000	adam	relu	0.86350
3e0b313cb282f...	<input type="checkbox"/>	64.000	5.0000	0.50000	adam	relu	0.85710
400ecfb3913e...	<input type="checkbox"/>	64.000	5.0000	0.50000	adam	relu	0.85860
412afa5295ffa...	<input type="checkbox"/>	128.00	3.0000	0.30000	adam	relu	0.86810
42147c311e54b...	<input type="checkbox"/>	128.00	3.0000	0.50000	adam	relu	0.86310
425615ae60c3a...	<input type="checkbox"/>	128.00	5.0000	0.30000	adam	relu	0.85350
49cf1f133ec71b...	<input type="checkbox"/>	128.00	5.0000	0.50000	adam	relu	0.86570
50200f26bb60e...	<input type="checkbox"/>	64.000	5.0000	0.30000	adam	relu	0.85420
54571f6cd759a...	<input type="checkbox"/>	128.00	3.0000	0.30000	adam	relu	0.86530
6e020a89c0c84...	<input type="checkbox"/>	64.000	3.0000	0.30000	adam	relu	0.86060
72b224c64c6bf...	<input type="checkbox"/>	128.00	5.0000	0.50000	adam	relu	0.86210
74958afd299b7...	<input type="checkbox"/>	64.000	5.0000	0.50000	adam	relu	0.85940
79cfe94ad7741...	<input type="checkbox"/>	128.00	5.0000	0.30000	adam	relu	0.86130
7dde21f5cfe8c5...	<input type="checkbox"/>	64.000	3.0000	0.30000	adam	relu	0.86040

Şekil 5.6. TensorFlow hiperparametre menüsü.

Şekil 5.6'da görüldüğü gibi hiperparametre performans çıktıları tablo, koordinat ve dağılım grafiği halinde listenebilir. Ayrıca sol menüde yer alan onay kutularıyla (checkbox) performans metriği, sıralama ölçütü ve parametre seçimi başta olmak üzere denenen tüm parametrelere (hparams) yönelik seçim yapılabilir. Bu seçimlere yönelik anlık performans sonuçları Şekil 5.5'teki gibi görülmektedir.

Şekil 5.6'da tablo görünümü (table view) ile hiperparametre sonuçları görülmektedir. Eğer paralel koordinat görünümü (paralel coordinate view) sekmesi seçilirse, parametrelerin optimum değerlerine ait çeşitli hiperparametrelerde alınan performansların anlık grafiksel dağılımları Şekil 5.7'deki gibi görülür.





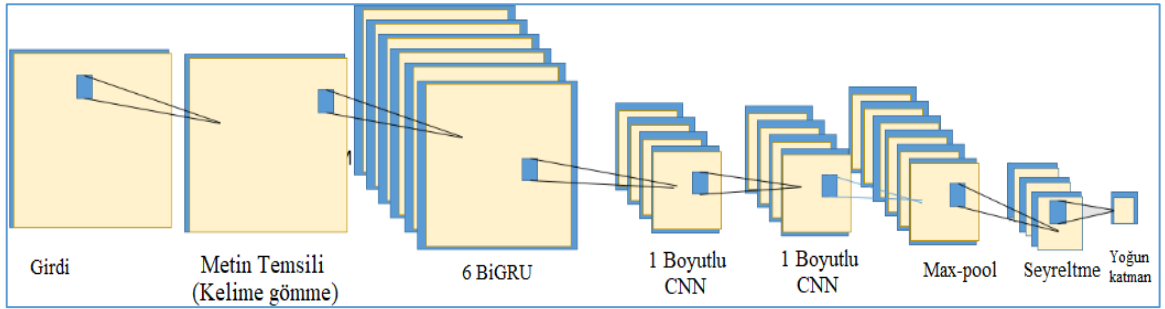
Şekil 5.7. TensorFlow parametre menüsündeki paralel koordinat sekmesi.

Şekil 5.7'ye göre ilgili parametrelerin hangileri ile birlikte optimum değer vereceği görülmektedir. Çizelge 5.16'da en iyi sonuç veren parametrelere ait özet bilgi, şekil 5.5 ve şekil 5.6'daki TensorFlow parametre menüsün (Hparams Dashboard)'den elde edilmiştir. TensorBoard ile belirlenen ve çalışmada kullanılan tüm modelin özet bilgisi Çizelge 5.16'da verilmiştir.

Çizelge 5.16. Model parametre özet bilgisi.

Parametreler	Değerler
Eğitim-Test ayrımı	% 70-% 30, % 80-% 20
Doğrulama ayrımı	% 10
Kelime gömme yöntemi	Word2Vec (Skip-Gram ve CBOW)
Kelime gömme boyutu	100, 200, 300
Çekirdek değeri	5
Sinir ağı düğüm sayısı	128
Evrişim Sinir Ağı Filtre Sayısı	128
Adım	1
Dolgu	same
Optimizasyon Fonksiyonu	Adam
Küme Boyutu	64
Dönem Sayısı	20
Seyreltme Oranı	0,3
Kelime Uzunluğu	200
Çıktı	1
Performans metriği	Doğruluk
Aktivasyon fonksiyonu	ReLU
Kayıp fonksiyonu	Binary Cross-Entropy
Çıkış fonksiyonu	Sigmoid

IMDB veri kümesi üzerinde 3 farklı vektör boyutunda (100, 200 ve 300) Word2Vec (Skip-Gram ve CBOW) kelime gömme metim temsil sonrası duygu sınıflandırma işlemi için çok katmanlı bir derin öğrenme modeli (MBiRUMCONV) önerilmiştir. Önerilen modelin görseli Şekil 5.8’dir. Önerilen model ait detaylar bu bölümde anlatılacaktır.



Şekil 5.8. Önerilen model (MBiGRUMCONV).

Önerilen modelinin girdi katmanında metin verisinin alındığı ilk katmandır. Sonrasında gelen kelime gömme katmanı ikinci katmandır. Metin verilerini belirli bir kelimeye karşılık gelecek şekilde kelime vektörüne dönüştürülür. Bu gerçek değerli vektörleri bir araya toplayarak, bir kelime gömme katmanı oluşturulur. Çalışmada, 100, 200 ve 300 vektör boyutunda Word2Vec kelime gömme yöntemi kullanılmıştır. Sonrasında, sırasıyla çift yönlü BiGRU, Evrişim, Seyreltme ve Yoğun katmanları gelmektedir. BiGRU katmanı sıralı altı adettir. Evrişim katmanı bir boyutlu, özellik haritasının yapısına dayalı

olarak tüm giriş kanalları ve filtreler arasındaki nokta çarpımını hesaplayarak elde edilen hacmi yöneten katmandır. Seyreltme katmanıysa belirli nöronların sonraki katmana bağlılığını geçersiz kılarken diğerlerini değiştirmeden bırakan bir katmandır. TensorBoard [61] ile hiperparametre ayarlamasında Seyreltme oranı 0,3 olarak alınmıştır. Son katman olan çıkış katmanında, önerilen modelle yapılan tahminlerin elde edildiği son aşamadır. Bu katmanda, bir düğümden oluşan ikili çıktıya (olumlu, olumsuz) göre sınıflandırma sürecinin sonlanır. Önerilen modelin hiperparametre detayları Çizelge 5.17’de verilmiştir.

Çizelge 5.17. Önerilen modelin hiperparametre detayı.

#Katman No	Katman	Hiperparametreler	Çıkış Boyutu	Parametre Sayısı
1	Kelime Gömme	Kelime gömme yöntemi= Word2Vec Maksimum kelime uzunluğu=200 Kelime matris uzunluğu=300 workers=8, min_count=5)	(None, 200,300)	15 000 000
2	BiGRU	Aktivasyon Fonksiyonu: ReLU	(None, 200,256)	330 240
3	BiGRU	Aktivasyon Fonksiyonu: ReLU	(None, 200,256)	296 448
4	BiGRU	Aktivasyon Fonksiyonu: ReLU	(None, 200,256)	296 448
5	BiGRU	Aktivasyon Fonksiyonu: ReLU	(None, 200,256)	296 448
6	BiGRU	Aktivasyon Fonksiyonu: ReLU	(None, 200,256)	296 448
7	BiGRU	Aktivasyon Fonksiyonu: ReLU	(None, 200,256)	296 448
8	Conv1D	Filtre sayısı: 32 Aktivasyon Fonksiyonu: ReLU Çekirdek boyutu: 5 Adım: 1	(None, 196,128)	163 968
9	Conv1D	Filtre sayısı: 32 Aktivasyon Fonksiyonu: ReLU Çekirdek boyutu: 5 Adım: 1	(None, 192,128)	82 048
10	Max Pooling	Havuz sayısı: 2	(None, 128)	0
11	Seyreltme	Oran: 0,3	(None, 16)	2 064
12	Yoğun Katman	Düğüm sayısı: 1 Aktivasyon Fonksiyonu: sigmoid	(None,1)	17

Bu çalışmada Word2Vec kelime gömme sonrası duygu analizi için sınıflayıcı model oluşturulmuştur. Bu modele ait sınama testi ayırım sonuçları Çizelge 5.18 ve Çizelge

5.19’da k kat çapraz geçerleme sonuçları Çizelge 5.22 ve Çizelge 5.23’te verilmiştir.

5.1.4.3. Sınama testi ayırım sonuçları

Yapılan deneylere ilişkin sonuçlar Çizelge 5.18 ve 5.19’da verilmiştir. 80%-%20 Eğitim-Test ve %10 doğrulama veri kümesi ayırımları sonuçları bu bölümde anlatılacaktır.

Çizelge 5.18. Önerilen modelin %80-%20 eğitim-test ve %10 doğrulama ayırımında doğruluk sonuçları.

Performans metriği	Veri kümesi ayırımı	CBOW			Skip-Gram		
		100	200	300	100	200	300
Doğruluk (%)	Eğitim	90,99	93,20	94,38	91,42	92,08	95,32
	Doğrulama	88,20	87,63	89,75	89,65	89,15	94,67
	Test	88,08	87,99	89,19	89,83	89,05	90,59

Çizelge 5.18’de görüldüğü üzere CBOW ve Skip-Gram yöntemlerinin 200 vektör boyutları dışında doğrulama ve test ayırımlarında vektör boyutu arttıkça model performansı artmıştır.

CBOW ve Skip-Gram’ın 100 vektör özelinde tüm ayırımlarında Skip-Gram daha iyi doğruluk performansı vermiştir. 200 vektör boyutu özelinde ise, eğitim ayırımında CBOW öndeyken; doğrulama ve test ayırımında Skip-Gram daha iyi sonuçlar vermiştir. 300 vektör boyutundaysa üç ayırında da (test, eğitim ve doğrulama) Skip-Gram, CBOW’dan daha iyi sonuç vermiştir. %80-%20 eğitim-test ve %10 doğrulama ayırımında önerilen modelin en iyi sonucu, Skip-Gram metin temsili yöntemi ile 300 vektör boyutunda tüm veri kümesi ayırımlarında alınmıştır.

Çizelge 5.19. Önerilen modelin %70-%30 eğitim-test ve %10 doğrulama ayırımında doğruluk sonuçları.

Performans metriği	Veri kümesi ayırımı	CBOW			Skip-Gram		
		100	200	300	100	200	300
Doğruluk (%)	Eğitim	92,58	93,28	94,21	90,40	90,93	94,68
	Doğrulama	89,03	89,11	90,19	89,49	89,43	90,37
	Test	89,30	89,29	89,78	89,09	90,53	89,80

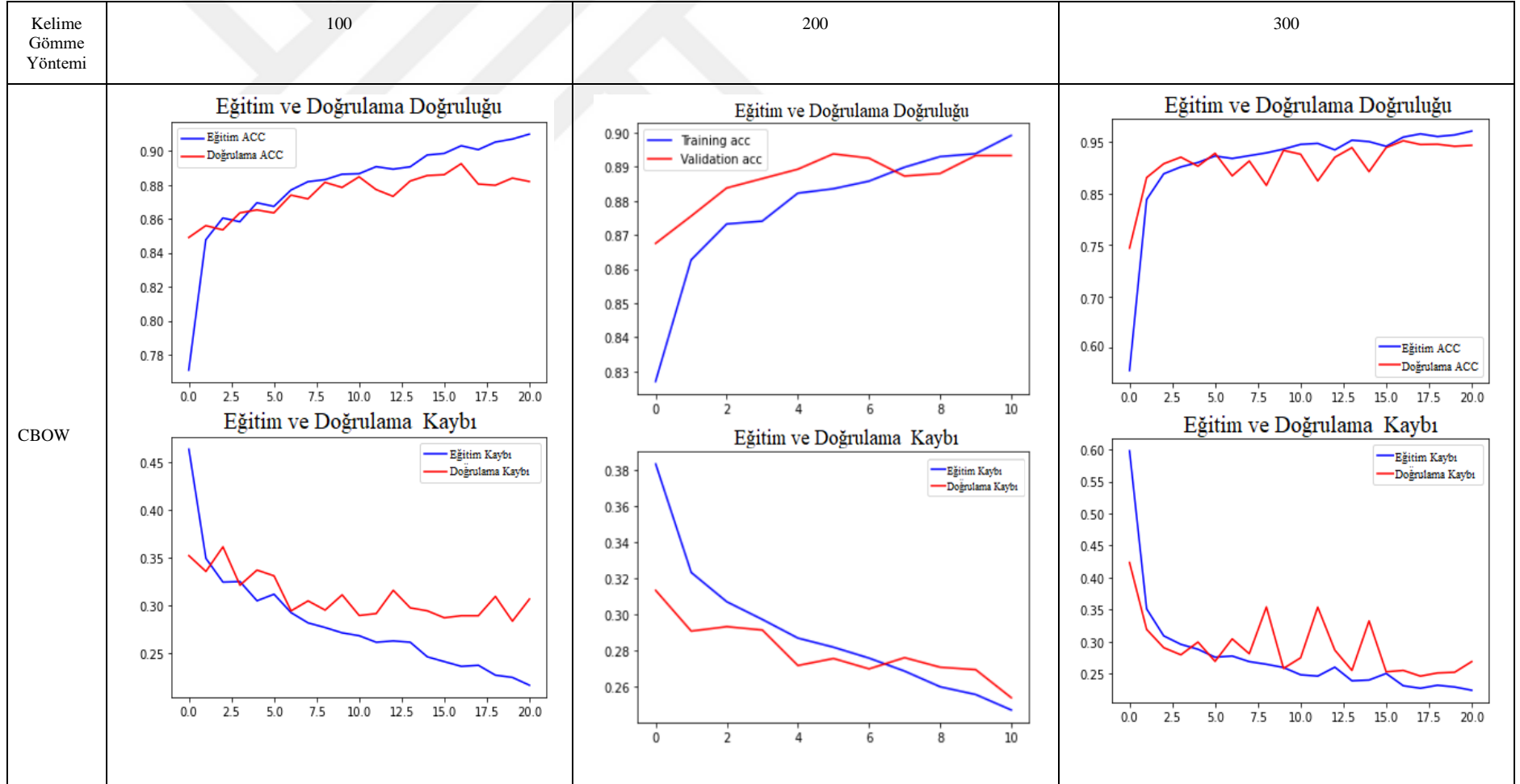
Çizelge 5.19’da görüldüğü üzere tüm veri kümesi ayırımlarında (eğitim, test ve doğrulama) Skip-Gram ve CBOW yöntemlerinin ikisinde de de vektör boyutu arttıkça performans değeri artmıştır.

CBOW ve Skip-Gramın 100 vektör özelinde test ve eğitim ayırımında CBOW daha iyi doğruluk performansı verirken; doğrulama ayırımında Skip-Gram daha iyi sonuç vermiştir. 200 vektör boyutunda ise eğitim ayırımında CBOW öndeyken; Doğrulama ve test ayırımında ise Skip-Gram daha iyi sonuç vermiştir. 300 vektör boyutundaysa eğitim ve doğrulama ayırımında Skip-Gram, test küme ayırımındaysa CBOW da daha iyi sonuç vermiştir. %70-%30 eğitim-test ve %10 doğrulama ayırımında önerilen modelin en iyi sonucu, Skip-Gram metin temsili ile 300 vektör boyutunda tüm veri kümesi ayrımlarında alınmıştır.

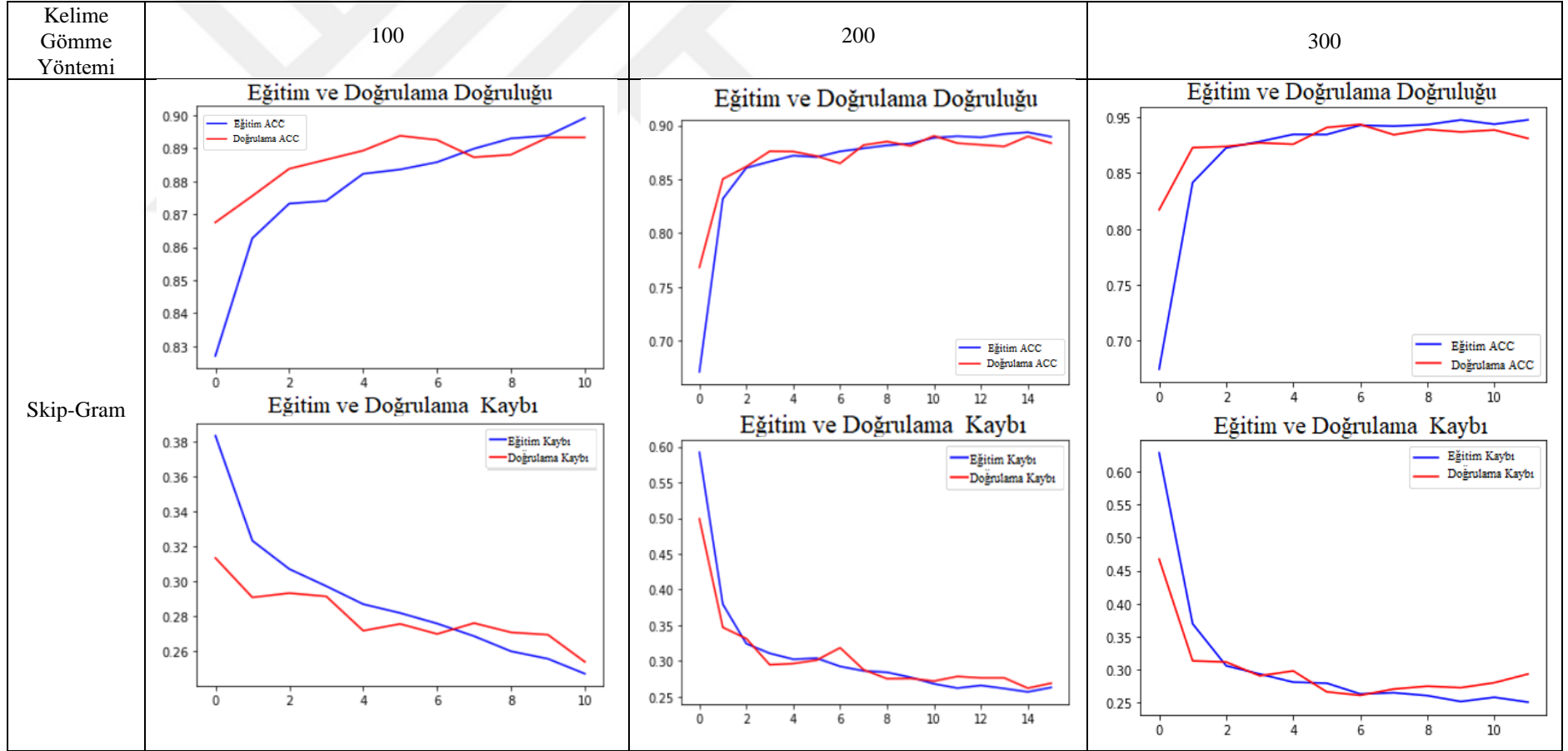
Çizelge 5.18 ve Çizelge 5.19 beraber değerlendirildiğindeyse %80-%20, %70-%30 eğitim-test ayırımında alınan sonuçlara göre daha iyi sonuç vermiştir. İki farklı %80-%20 ve %70-%30 eğitim-test ayırımında Skip-Gram ve CBOW yöntemlerinin ikisinde de de vektör boyutu arttıkça performans değeri arttığı görülmüştür. İki ayırmda da en iyi sonuçları, Skip-Gram yöntemi ve 300 vektör boyutunda alınmıştır.

Önerilen modelin tüm metin temsili ve vektör ayrımlarına yönelik, eğitim ve doğrulama doğruluk ve kayıp grafikleri Çizelge 5.20 ve Çizelge 5.21’de verilmiştir. %70-%30 eğitim-test ve %10 doğrulama ayırımında en iyi sonucu veren Skip-Gram 300 vektör boyutunda alınan eğitim ve doğrulama kayıp grafiği şekil 5.9’da gösterilmiştir.

Çizelge 5.20. IMDB veri kümesi %80-%20 eğitim-test ayrımında eğitim ve doğrulama kayıp grafikleri.

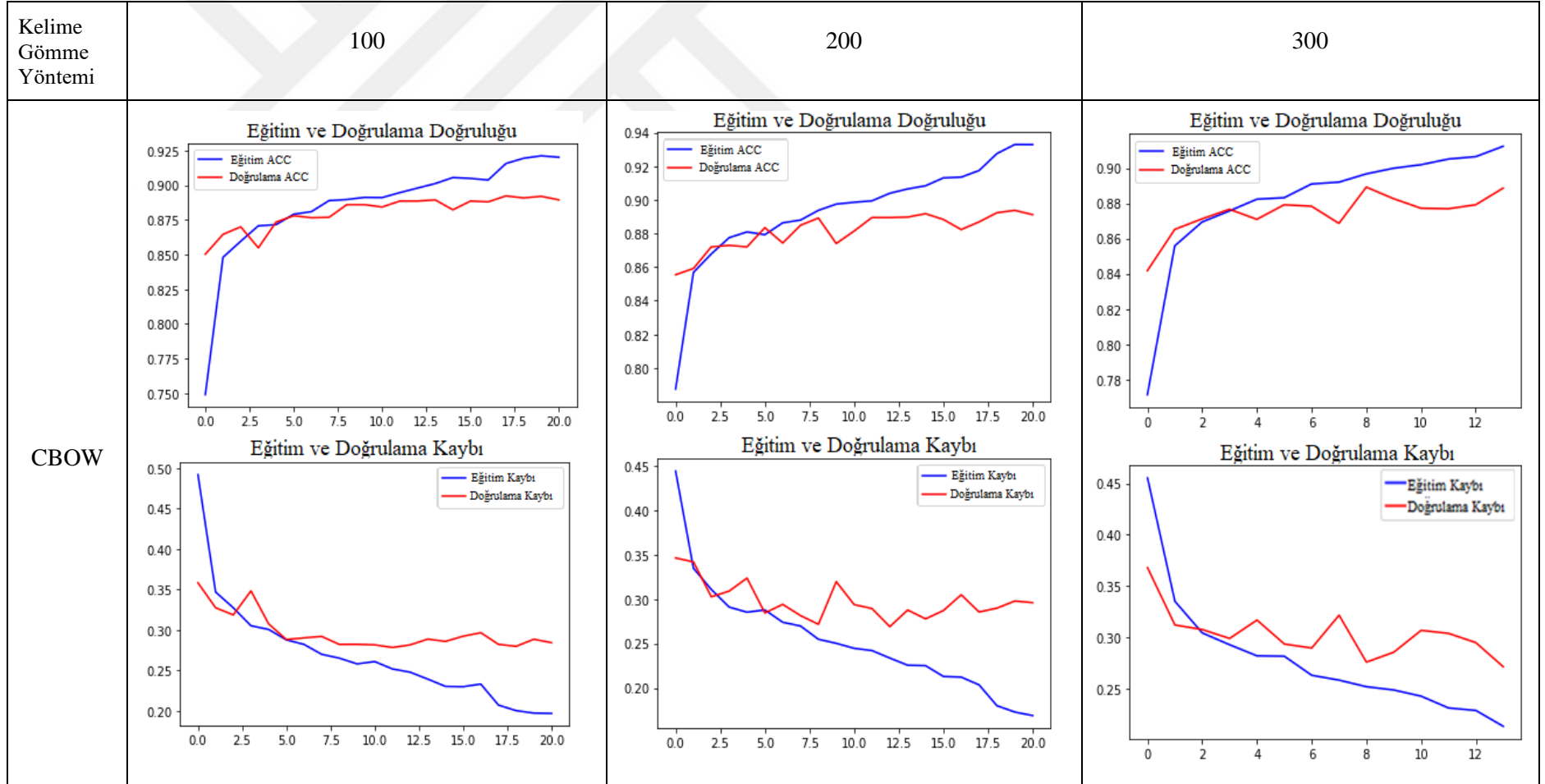


Çizelge 5.20. (devam). IMDB veri kümesi %80-%20 eğitim-test ayrımında eğitim ve doğrulama kayıp grafikleri.

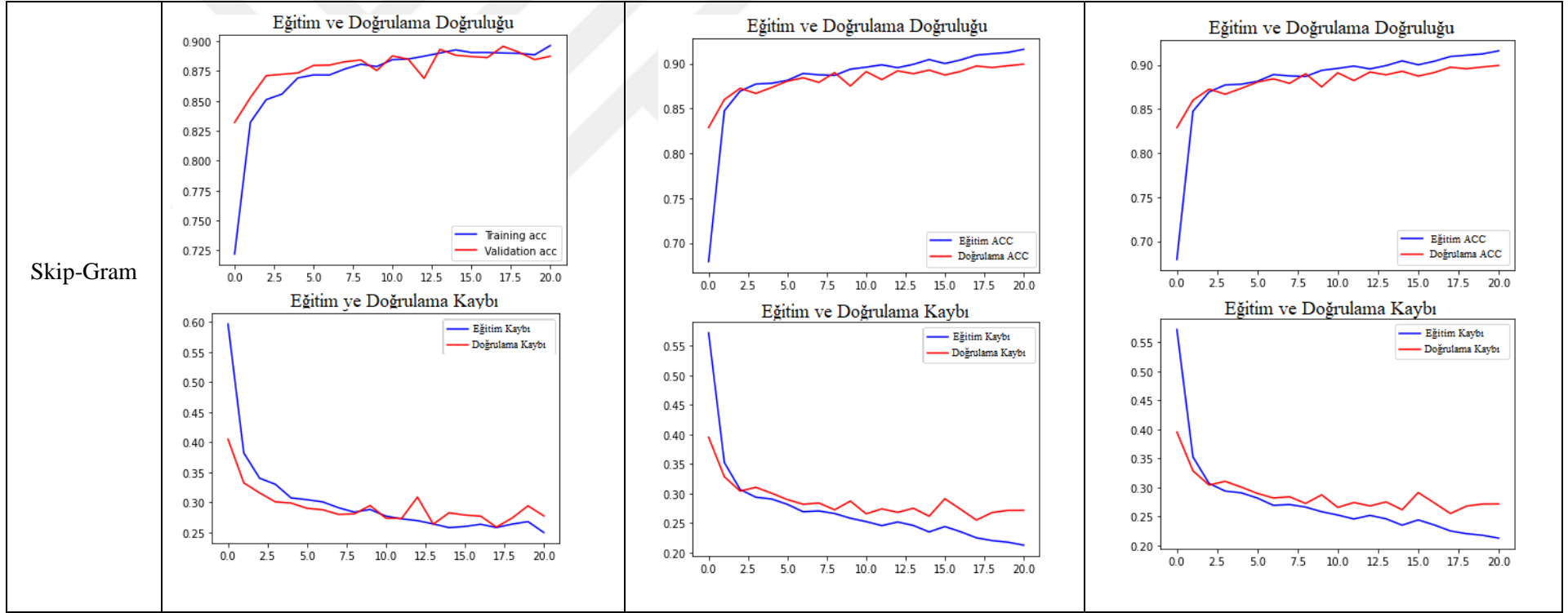


Çizelge 5.20'deki grafiklerden, tüm vektör boyutlarında (100, 200, 300) Skip-Gram, CBOW'a göre tüm vektör boyutlarında daha iyi (robust) model elde edildiği görülmektedir.

Çizelge 5.21. IMDB veri kümesi %70-%30 eğitim-test ve %10 doğrulama ayırımında eğitim ve doğrulama kayıp grafikleri.

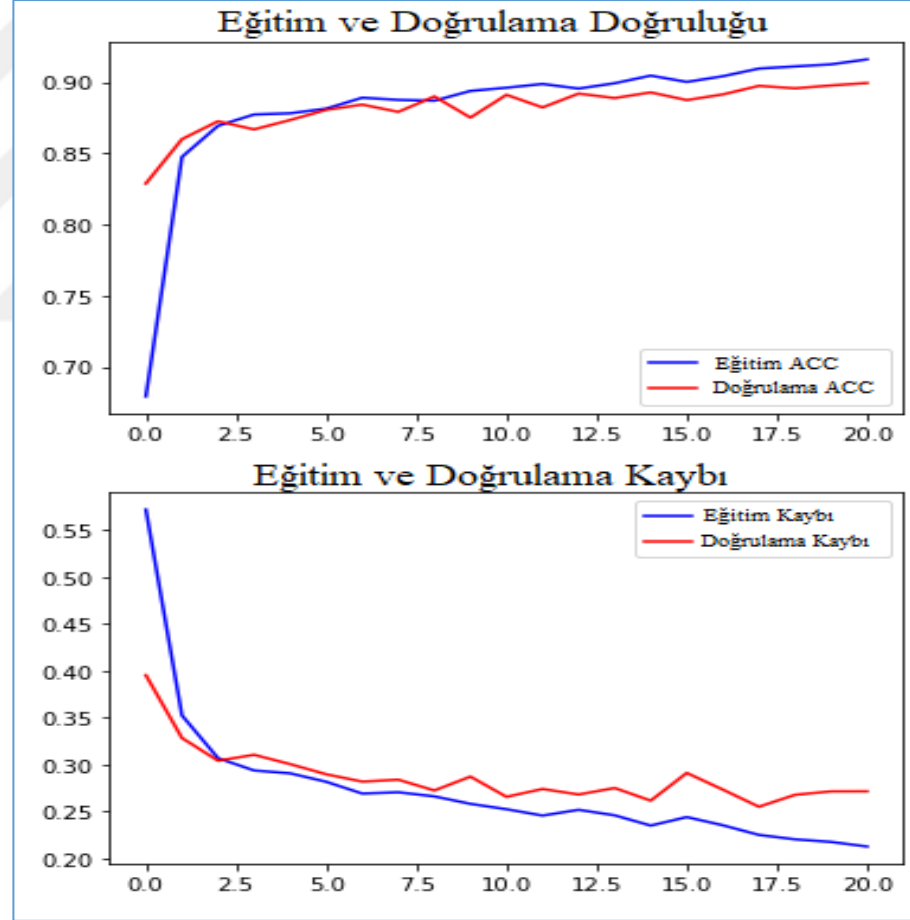


Çizelge 5.21. (devam). IMDB veri kümesi %70-%30 eğitim-test ve %10 doğrulama ayırımında eğitim ve doğrulama kayıp grafikleri.



Çizelge 5.21'deki grafiklerden, tüm vektör boyutlarında (100, 200, 300) Skip-Gram, CBOW'a daha daha iyi model elde edildiği görülmektedir.

Şekil 5.9'da en iyi sonucu veren modelin eğitim-test doğruluk kayıp grafiği verilmiştir.



Şekil 5.9. IMDB veri kümesi %70-%30 ve %10 doğrulama ayırımında eğitim ve doğrulama kayıp grafiği.

Şekil 5.9’da IMDB veri kümesi %70-%30 eğitim-test ve %10 doğrulama ayırımında grafikleri en iyi sonucu Skip-Gram 300 vektör boyutunda alınan eğitim ve doğrulama kayıp grafik sonucu gösterilmiştir.

5.1.4.4. K-Kat Çapraz Doğrulama Ayrım sonuçları

Word2Vec sonrası 3 kat, 5 kat ve 10 kat çapraz geçiş sonuçları Çizelge 5.22, Çizelge 5.23'te verilmiştir.

Çizelge 5.22. IMDB veri kümesi %80-%20 eğitim-test k-kat çapraz doğrulama doğruluk sonuçları.

Word2Vec Türü	CBOW			Skip-Gram		
Vektör boyutu	100	200	300	100	200	300
3 kat	89,49	89,63	89,69	89,32	89,39	90,56
5 kat	89,32	89,86	89,87	89,45	89,58	90,67
10 kat	89,18	89,25	89,48	89,28	89,42	89,51

Çizelge 5.22'de görüldüğü üzere, sınamaya testi ayırma sonuçlarına benzer şekilde 3, 5 ve 10 kat çapraz doğrulama sonuçlarına göre Skip-Gram 300 vektör boyutu CBOW'a göre daha iyi sonuç vermiştir.

Çizelge 5.23. IMDB veri kümesi %70-%30 eğitim-test k-kat çapraz doğrulama doğruluk sonuçları.

Word2Vec Türü	CBOW			Skip-Gram		
Vektör boyutu	100	200	300	100	200	300
3 kat	89,24	89,66	89,85	89,35	89,47	89,87
5 kat	89,32	89,75	89,94	89,56	89,65	89,99
10 kat	89,06	89,78	89,90	89,32	89,52	89,77

Çizelge 5.23'te görüldüğü üzere, sınamaya testi ayırma sonuçlarına benzer şekilde 3, 5 ve 10 kat çapraz doğrulama sonuçlarına göre Skip-Gram 300 vektör boyutu CBOW'a göre daha iyi sonuç vermiştir. Çizelge 5.23'te IMDB veri kümesi üzerinde benzer içerikli çalışmalarla önerilen modelin test kümesi ayırma için doğruluk sonucu verilmiştir.

Çizelge 5.24. Önerilen modelin test doğruluk sonuçlarının önceki çalışmalarla karşılaştırılması.

Yazar	Model	Doğruluk (%)
Rathee ve arkadaşları [18]	Paragraf Vektör	76
Kumar ve Harish [31]	SVM	85
Zabit ve Begonya [12]	CNN	85,32
Vielma ve arkadaşları [8]	CNN-BiLSTM	89,54
Haque, Lima ve Mishu [9]	CNN	90
Dang ve arkadaşları [10]	4 CNN	84,5
Gao ve arkadaşları [11]	3 CNN	89,4
Qaisar [13]	2 LSTM	89,9
Amit ve arkadaşları [14]	BiLSTM-CNN	90,5

Çizelge 5. 24. (devam). Önerilen modelin test doğruluk sonuçlarının önceki çalışmalarla karşılaştırılması.

Yazar	Model	Doğruluk (%)
Derra [15]	2 BiLSTM	87,46
Prabha ve arkadaşları [16]	RNN	87
Ni ve Cao [17]	SLTM-GRU	87,10
Wenpeng ve arkadaşları [19]	CNN	87,90
Nguyen ve arkadaşları [32]	SVM-NB	87,95
Amulya ve arkadaşları [20]	RNN	88
Hassan ve Mahmood [22]	CNN-LSTM	88,3
Mohaiminul ve Sultana [33]	RF	88,66
Narayanan ve arkadaşları [34]	NB	88,80
Camacho-Collados ve Pilehvar [23]	CNN-LSTM	88,90
Ghorbani ve arkadaşları [24]	CNN-LSTM-CNN	89,02
Zhou ve Feng [29]	DT	89,16
Wang ve Manning [30]	SVM	89,16
Nehal ve arkadaşları [25]	CNN-LSTM	89,20
Ma ve Fan [26]	Bi-LSTM	89,29
Sabba ve arkadaşları [21]	CNN	89,47
Chen ve arkadaşları [27]	BiLSTM	90,58
Önerilen Model	Sınama testi ayrımı	90,59
	5 kat doğrulama ayrımı	90,67

Çizelge 5.24'te görüldüğü gibi önerilen modelin doğruluk performansı aynı veri kümesi üzerinde benzer içerikli model sonuçlarına göre öndedir. Çizelge 5.25'te doğrulama veri kümesi ayırımına ait doğruluk sonucu verilmiştir.

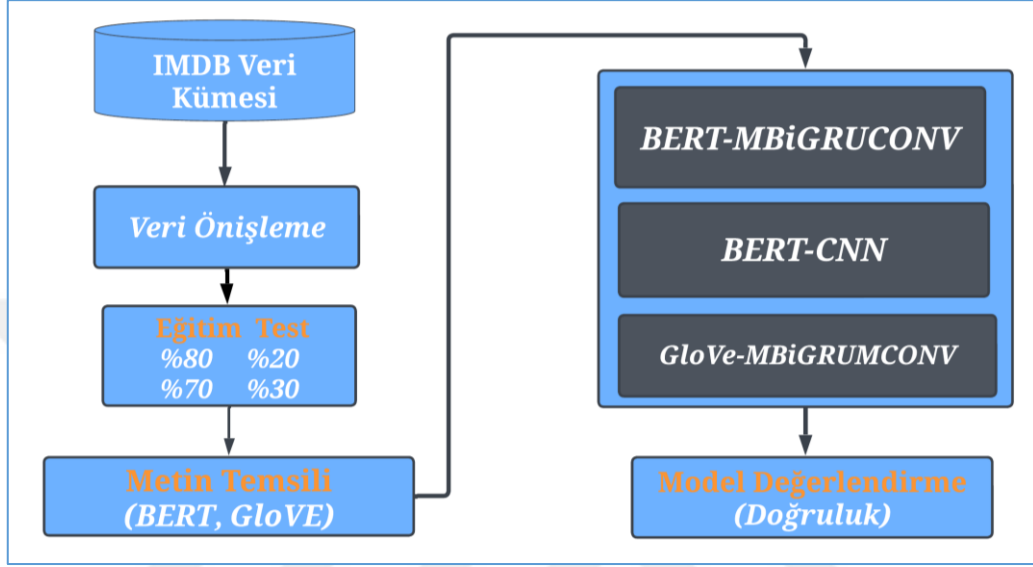
Çizelge 5.25. Önerilen modelin doğrulama doğruluk sonuçlarının benzer çalışmalarla karşılaştırılması.

Yazar	Model	Doğruluk (%)
Chen ve arkadaşları [27]	CNN-BiLSTM	82,30
Shaukat ve arkadaşları [150]	MLP	86,67
Pang ve Lee [155]	NB-SVM	87,20
Derra [15]	2 BiLSTM	88,36
Ghorbani ve arkadaşları [24]	CNN-LSTM-CNN	89,02
Rathor ve Agrawal [156]	1 CNN	89,27
Yenter ve Verma [157]	CNN-LSTM	89,50
Ghosh	Linear SVM	91,10
Johson ve Zhang [28]	LSTM	93,49
Amit [14]	MBiLSTM GloVe	93,55
Önerilen Model	MBiGRUMCONV	94,67

Çizelge 5.25'te önerilen modelin doğrulama ayırımında doğruluk performans sonucu özellikle çok katmanlı önerilen modellerle karşılaştırılmıştır. Bu karşılaştırmalarda benzer çalışmaları geride bıraktığı görülmektedir.

5.1.5. Ön Eğitimli Yöntemler Deneyi

IMDB veri kümesi özelinde Word2Vec sonrası MBiGRUMCONV modeline ek olarak Bu bölümde ön eğitimli metin temsil yöntemleriyle oluşturulan modellerle yapılan deneysel çalışmalar anlatılacaktır. Şekil 5.10'da çalışmanın akış diyagramı görülmektedir.



Şekil 5.10. IMDB veri kümesinde yapılan çalışmanın akış diyagramı.

5.1.5.1. Ön işleme ve Metin Temsili

IMDB veri kümesi ile BERT-CNN ve BERT sonrası MBiGRUMCONV çalışmalarında, on iki katman dönüştürücü bloğu, on iki dikkat başlığı, yedi yüz altmış sekiz gizli katman ve yüz on milyon parametre bulunan BERT-base kullanılmıştır. GLoVe için de altı milyar kelime bulunan GLoVe [95] versiyonunun 100, 200, 300 boyutu ile Google tarafından önceden eğitilmiş BERT kullanılmıştır.

5.1.5.2. Deney Kurulumu

GLoVe sonrası MBiGRUMCONV ve BERT sonrası MBiGRUMCONV ile %70-%30 ve %80-%20 eğitim-test ayırımında modeller oluşturulmuştur. Bunlara ek olarak BERT-CNN ile de model oluşturulmuştur.

Bu bölümde yapılan BERT-CNN, ayrı ayrı ön eğitimli GLoVe ve BERT sonrası MBiGRUMCONV modellerine ait deneyler, daha hızlı ve kesintisiz sonuçlanması için Colab Pro [142] üzerinde yapılmıştır. GLoVe ve BERT sonrası MBiGRUMCONV modeliyle alınan sonuçlarla BERT-CNN sonrası alınan sonuçlar alt başlıklarda anlatılmıştır.

5.1.5.3. Ön Eğitimli GloVe Sonrası Alınan Sonuçlar

IMDB veri kümesinde, ön eğitimli GloVe sonrası %70-%30, %80-%20 eğitim-test ayrımının her ikisinde de %10 geçерleme sonuçları Çizelge 5.26 ve Çizelge 5.27’de verilmiştir.

Çizelge 5.26. %70-%30 eğitim-test ve %10 doğrulama doğruluk sonuçları.

Performans metriği		100	200	300
Doğruluk	Eğitim	96,86	97,40	97,21
	Test	87,30	87,35	87,43
	Doğrulama	87,10	87,35	87,39

Çizelge 5.26’da görüldüğü üzere, %70-%30 eğitim-test sınama testi ayrım sonuçlarına göre Test ve doğrulama kümesinde 300 vektör boyutunda daha iyi sonuç vermiştir. Eğitim kümesinde ise 200 vektör boyutu daha iyi sonuç vermiştir.

Çizelge 5.27. %80-%20 eğitim-test ve %10 doğrulama doğruluk sonuçları.

Performans metriği		100	200	300
Doğruluk	Eğitim	97,23	97,77	98,12
	Test	86,57	87,74	89,10
	Doğrulama	88,10	88,15	88,58

Çizelge 5.27’de görüldüğü üzere, %80-%20 eğitim-test sınama testi ayrım sonuçlarına göre test ve doğrulama kümesinde 300 vektör boyutu daha iyi sonuç vermiştir.

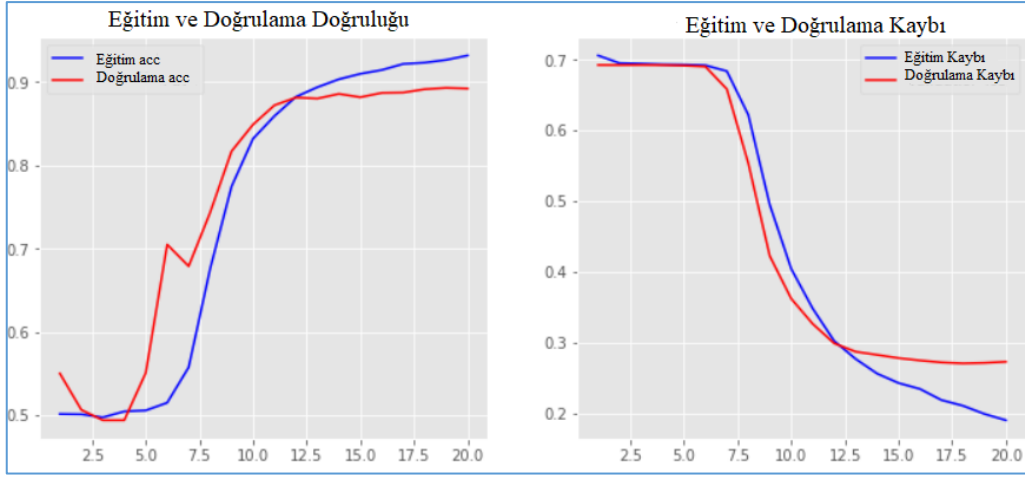
5.1.5.4. BERT Sonrası MBiGRU CONV ile alınan Sonuçlar

BERT sonrası MBiGRU CONV yöntemiyle, %70-%30, %80-%20 eğitim-test ayrımının her ikisinde de %10 geçерleme sonuçları Çizelge 5.28’ de verilmiştir.

Çizelge 5.28. IMDB veri kümesinde %70-%30 ve %80-%20 eğitim-test ve %10 doğrulama doğruluk sonuçları.

Eğitim-Test ayrımı	%70-%30	%80-%20
Eğitim	95,49	95,59
Test	88,81	89,63
Doğrulama	88,63	90,43

Çizelge 5.28’de görüldüğü üzere %80-%20 eğitim-test ayrımı, %70-%30 eğitim-test ayrımına göre daha iyi sonuç vermiştir. BERT sonrası IMDB veri kümesi üzerinde %80-%20 eğitim-test ayrımında en iyi sonucu veren modelin, eğitim doğrulama kayıp grafiği Şekil 5.11’de verilmiştir.



Şekil 5.11. IMDB veri kümesinde BERT MBiGRUMCONV eğitim doğrulama doğruluk ve kayıp grafiği.

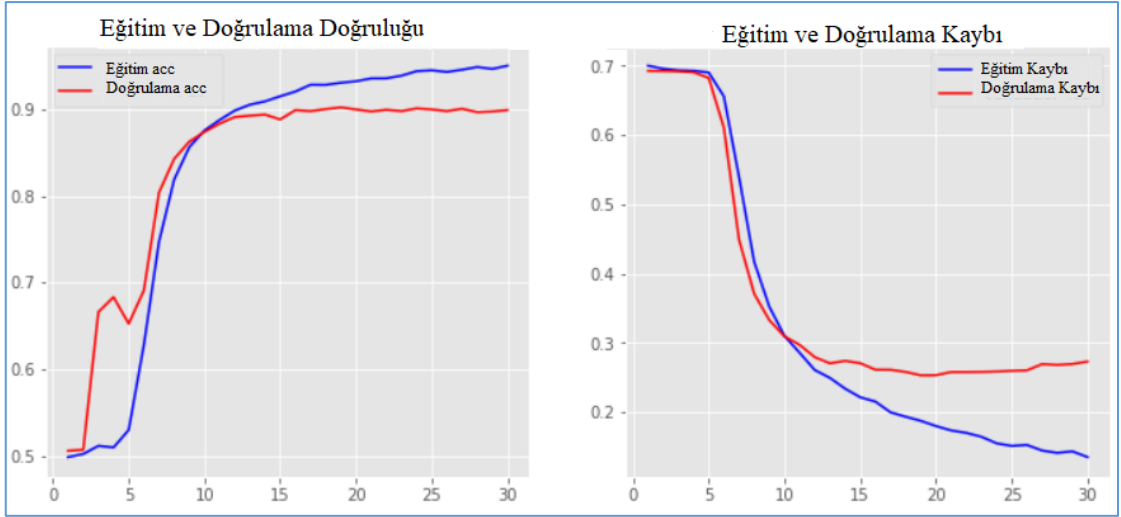
5.1.5.5. BERT-CNN Deneyi

BERT-CNN ile %70-%30 ve %80-%20 eğitim-test ve %10 geçерleme ayırım sonuçları Çizelge 5.29’da verilmiştir.

Çizelge 5.29. %70-%30 ve %80-%20 eğitim-test ve %10 sınaama testi doğruluk sonuçları.

	%70-%30	%80-%20
Eğitim	96,24	96,60
Test	88,72	90,72
Doğrulama	88,42	94,75

Şekil 5.12’de BERT-CNN sonrası IMDB veri kümesi %80-%20 eğitim-test ayırımında en iyi sonucu veren eğitim doğrulama kayıp grafiği verilmiştir.



Şekil 5.12. IMDB veri kümesinde BERT-CNN %80-%20 eğitim ve doğrulama kayıp grafiği.

5.1.5.6. Deney Sonuçlarının Genel Değerlendirilmesi

IMDB veri kümesi üzerinde yapılan Makine öğrenmesi, Derin öğrenme, çok katmanlı BİGRU ve Çok katmanlı CNN ve ön eğitilmiş olmak üzere yapılan tüm deneylerin değerlendirilmesi bu bölümde yapılacaktır.

Yapılan Makine öğrenmesi deneyinde BOW ve Word2Vec metin temsili sonrası NB, RF, SVM, DT makine öğrenmesi yöntemleriyle oluşturulan modeller doğruluk performansı ile değerlendirilmiştir. TF-IDF ve Word2Vec temsillerinden sonra SVM daha iyi sonuç vermiştir.

Çalışmada yapılan ikinci deneyde çeşitli optimizasyon fonksiyonları, dönem, küme boyutu gibi parametrelerde farklı sayıda Yoğun Katman, Seyreltme ve çeşitli derin öğrenme sinir ağlarıyla (GRU, LSTM, BiLSTM, BiGRU) oluşturulan birden çok katmandan oluşan farklı yedi model oluşturulmuştur. Bu modellerin doğruluk performansı bakımından en iyi sonucu BiLSTM ve BiGRU derin öğrenme yöntemleriyle oluşturulan modeller vermiştir. Bu sonuç ile önerilen modelde BiGRU sinir ağı kullanılmasına önemli bir etken olmuştur.

IMDB veri kümesi üzerinde üçüncü deneyde, Word2Vec metin temsil işlemi sonrası derin öğrenme temelli yeni bir model önerilmiştir. Derin öğrenme yöntemi olarak çok katmanlı BiGRU ve CNN sinir ağları kullanılmıştır. Önerilen model; altı adet BiGRU ve iki CNN sinir ağlarından oluşturulmuştur. Deneyler sırasında veri kümesi %80-%20 ve %70-%30 sınıma testi eğitim-test ve her iki ayırma da %10 doğrulama ayırma

bölünerek sonuçları ayrı iki Çizelgede (Çizelge 5.17, Çizelge 5.18) verilmiştir. Ayrıca 3-5-10 kat çapraz geçirme sonuçları da Çizelge 5.21 ve Çizelge 5.22’de verilmiştir. Bir sonraki deneyde önerilen MBiGRUMCONV modelinde Word2Vec yerine GloVe ve BERT ön eğitilmiş yöntemleriyle metin temsilleri çıkarılmıştır. Önerilen modelin ön eğitilmiş GloVe sonucu; Çizelge 5.26 ve Çizelge 5.27’de, BERT kullanılarak oluşturulan BERT-MBiGRUMCONV modelinin sonucu Çizelge 5.28’de verilmiştir. Ayrıca Çizelge 5.29’da NLP görevlerinde güncel yaklaşım olarak sıklıkla kullanılan BERT-CNN sonucu verilmiştir.

BERT-MBiGRUMCONV, BERT-CNN ve önerilen yöntem olan Word2Vec sonrası MBiGRUCONV modelleriyle alınan sonuçların kendi aralarında test ve doğrulama doğruluklarına ait karşılaştırılmaları Çizelge 5.30’da verilmiştir.

Çizelge 5.30. MBiGRUMCONV modelinin, BERT-CNN, BERT-MBiGRUCONV modelleriyle test ve doğrulama doğruluk sonuçlarıyla karşılaştırılması.

Model		Test (%)	Doğrulama (%)
BERT MBiGRUMCONV		89,63	90,43
BERT-CNN		90,72	94,75
MBiGRUMCONV	Sınama testi	90,59	94,67
	5 kat çapraz geçirme	90,67	-

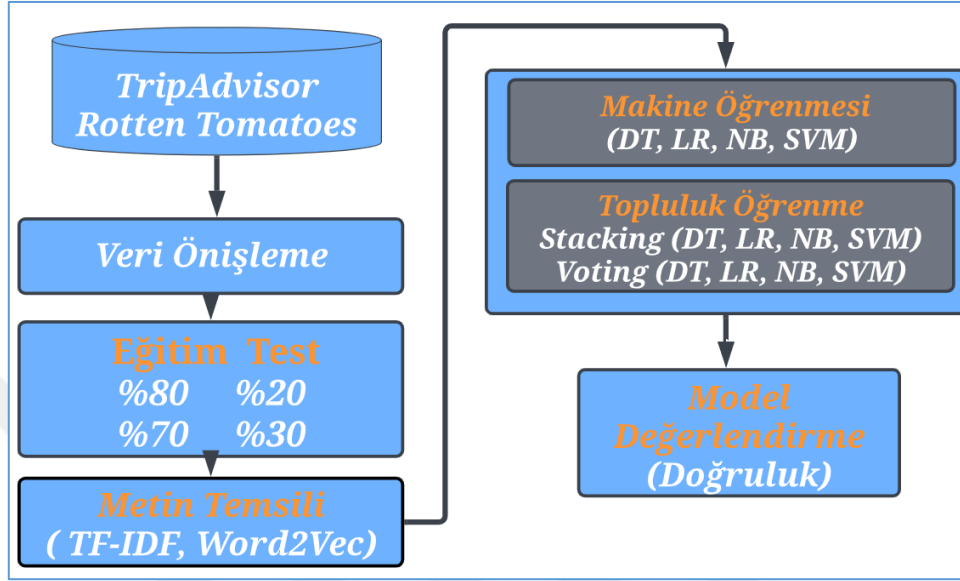
Çizelge 5.30’da görüldüğü gibi, BERT-CNN literatürdeki gibi öndedir. Fakat Word2Vec sonrası MBiGRUCONV, BERT sonrası MBiGRUCONV’un test ayırımında bir adım önündedir. Test ayırımında ki bu sonuç doğrulama ayırımında da benzer şekildedir. Test setinde en iyi sonucu BERT-CNN %90,72 test doğruluk değeri vermiştir. Word2Vec MBiGRUMCONV modeli de %80-%20 sınama test ayırımında %90,59 doğruluk değeri 300 vektör boyutlu Skip-Gram ile %90,59 test doğruluk değerleri verirken; 5 kat çapraz geçirme ayırımında %90,67 doğruluk sonucunu yine 300 vektör Skip-Gram yöntemiyle vermiştir. BERT-MBiGRUMCONV modeli %89,63 ile önerilen modeli takip etmektedir. Bu sonuçlar; Word2Vec sonrası MBiGRUMCONV ile Bert sonrası MBiGRUMCONV birbirine yakın sonuçlar verdiğini göstermiştir.

5.2. FARKLI VERİ KÜMELERİ ÜZERİNDE YAPILAN DENEYLER

Bu bölümde çalışma IMDB dışındaki TripAdvisor, Rotten Tomatoes ve Covid-19 veri kümeleri üzerinde yapılan deneylerden bahsedilecektir.

5.2.1. TripAdvisor ve Rotten Tomatoes Üzerinde Yapılan Deney

Bu bölümde TripAdvisor ve Rotten Tomatoes veri kümesi ve bu veri kümeleri üzerinde yapılan deneysel çalışma anlatılacaktır. Şekil 5.13'te çalışmanın akış diyagramı görülmektedir.



Şekil 5.13. TripAdvisor ve Rotten Tomatoes veri kümeleriyle yapılan çalışmanın akış diyagramı.

5.2.1.1. Deneyde Kullanılan Veri Kümeleri

Çalışmada, film ve otel inceleme sitelerinden toplanan ve bu sitelerden toplayanlar tarafından açık kaynak olarak paylaşılan yorumlar kullanılmıştır. TripAdvisor otel incelemesi veri kümesi [6], yirmi bin dört yüz doksan otel yorumundan elde edilen verilerden oluşmaktadır. Bir ve iki yıldızlı yorumlar negatif; üç yıldızlı yorumlar nötr; dört ve beş yıldızlı yorumlar pozitif olarak işaretlenmiştir. Rotten Tomatoes film incelemeleri veri kümesi, iki yüz kırk bin olumlu (fresh) ve iki yüz kırk bin olumsuz (rotten) olarak etiketlenmiş film incelemelerine sahiptir [5]. Bu veri kümelerine dair bilgi literatür taramasının alt bölümü olan bölüm (2.2.1)'de yer almaktadır.

5.2.1.2. Ön işleme ve Metin Temsilleri

Dokümanlarda metnin temsili için kullanılan yöntemler, sınıflandırmada başarılı sonuçların alınmasına zemin hazırlamaktadır. Bu çalışmada, Word2Vec ve TF-IDF metin temsil yöntemleri kullanılmıştır. Bu deneyde, Çizelge 5.31'de verilen Word2Vec Parametreleri kullanılmıştır.

Çizelge 5.31. Kullanılan Word2Vec parametreleri.

Parametre	Değerler
Size	300
Window	5
Workers	8
min_count	5
sg	0

Çizelge 5.31’de görüldüğü üzere, çalışmada Word2Vec yöntemi için vektör boyutu 300, pencere boyutu beş, veri kümesinde geçen minimum değer beş ve CBOW yöntemi kullanılmıştır. Çalışmada kullanılan diğer temsil yöntemi için Çizelge 5.32’de verilen TF-IDF parametresi kullanılmıştır.

Çizelge 5.32. TF-IDF yöntemi parametreleri.

Parametre	Değer
max_count	10 000

5.2.1.3. Deney Kurulumu

Bu deneyde, TripAdvisor ve Rotten Tomatoes veri kümeleri, %80-%20 ve %70-%30 eğitim-test ayırımına bölünmüştür. Duygu analizi deneyinde, deney ortam özellikleri ve performans kriterleri başlığında anlatıldığı üzere yerel bilgisayarda, Python programlama dili ve Jupyter IDE’si ile gerçekleştirilmiştir. Deneyde duygu analizi modelleri için; makine öğrenmesi algoritmalarından NB, SVM, LR ve KNN sınıflandırıcıları kullanılmıştır. Tekil makine öğrenmesi algoritmalarının bir arada kullanımını sağlayan topluluk öğrenme yöntemlerinden Yığılma ve Oylama kullanılmıştır. Oluşturulan modeller ve tüm NLP işlemleri, AMD RYZEN 7 2700 işlemcili ve 64 GB RAM’e sahip bir bilgisayarda çalıştırılmıştır. Deneyde model performansları doğruluk kriteriyle değerlendirilmiştir.

5.2.1.4. Deney Sonuçları

TripAdvisor ve Rotten Tomatoes veri kümelerininin %80-%20 ve %70-%30 eğitim-test ayırımında, TF-IDF ve Word2Vec metin temsilleri sonrasında duygu analizi modelleri oluşturulmuştur. Bu modeller, tekil makine öğrenmesi (NB, KNN, LR, SVM) ve bu tekil makine öğrenmesi yöntemlerininin heterojen şekilde bir arada kullanımı için topluluk öğrenme algoritmalarından oluşan Yığılma ve Oylama’dır. TripAdvisor için

oluşturulan modellerin doğruluk performansları Çizelge 5.33 ve Çizelge 5.34'te, Rotten Tomatoes üzerinde oluşturulan modellerin doğruluk performansı Çizelge 6.10 ve Çizelge 6.11'de verilmiştir.

Çizelge 5.33. TripAdvisor veri kümesinde doğruluk sonucu (%80 eğitim, %20 test).

Metin temsil yöntemi	NB	KNN	LR	SVM	Yığınlama	Oylama
TF-IDF	83,1	76,6	88,2	88,5	89,7	90,4
Word2Vec	84,3	78,6	89,3	89,5	90,5	91,5

Çizelge 5.33 incelendiğinde, tekil makine öğrenmesi yöntemleri arasında en iyi doğruluk performans sonucu Word2Vec sonrası SVM ile oluşturulan modelin verdiği görülmektedir. Topluluk öğrenme yöntemleriyle tekil makine öğrenmesi yöntemleri karşılaştırıldığında, her iki topluluk öğrenme yönteminin de tekil makine öğrenme yöntemlerine göre daha iyi performans verdiği görülmektedir. Ayrıca Oylama, topluluk öğrenme yöntemleri arasında Yığınlama'ya göre daha iyi sonuçlar vermiştir.

Çizelge 5.34. TripAdvisor veri kümesinde doğruluk sonucu (%70 eğitim, %30 test).

Metin temsil yöntemi	NB	KNN	LR	SVM	Yığınlama	Oylama
TF-IDF	83,6	76,1	86,2	88,4	90,8	91,2
Word2Vec	85,2	77,2	86,4	89,7	90,4	91,4

TripAdvisor veri kümesinin, %70-%30 sına testi eğitim-test ayırımı sonuçları, Çizelge 5.34 incelendiğinde, TF-IDF ile tekil makine öğrenmesi yöntemleri arasında performansı en iyi olan SVM'dir. Tekil makine öğrenmesiyle Word2Vec sonrası oluşturulan modellerde sırasıyla SVM, LR, NB, KNN şeklinde doğruluk performansları alınmıştır. İki temsil yöntemi sonrası topluluk öğrenme yöntemleriyle oluşturulan modellerde; Oylama, Yığınlama'ya göre daha iyi sonuçlar vermiştir.

Çizelge 5.35. Rotten Tomatoes veri kümesinde doğruluk sonucu (%80 eğitim, %20 test).

Metin temsil yöntemi	NB	KNN	LR	SVM	Yığınlama	Oylama
TF-IDF	81,5	76,8	81,7	83,8	83,9	87,2
Word2Vec	81,6	80,6	81,9	87,8	88,5	88,7

TripAdvisor veri kümesinin, %80-%20 sına testi eğitim-test ayırımı sonuçları Çizelge 5.35 incelendiğinde, TF-IDF ile tekil makine öğrenmesi yöntemleri arasında doğruluk sonucu en iyi olan SVM'dir. Ancak SVM, Word2Vec ile en iyi sonuçları elde eder. Word2Vec temsilinde model performans sıralaması SVM, LR, NB, KNN şeklindedir. Topluluk öğrenme yöntemleri ile tekil makine öğrenmesi yöntemleri karşılaştırıldığında,

Yığınlama ve Oylama daha iyi sonuç vermiştir. Topluluk öğrenme yöntemleri kendi arasında karşılaştırıldığında Oylama, Yığınlama'ya göre daha iyi sonuçlar vermiştir.

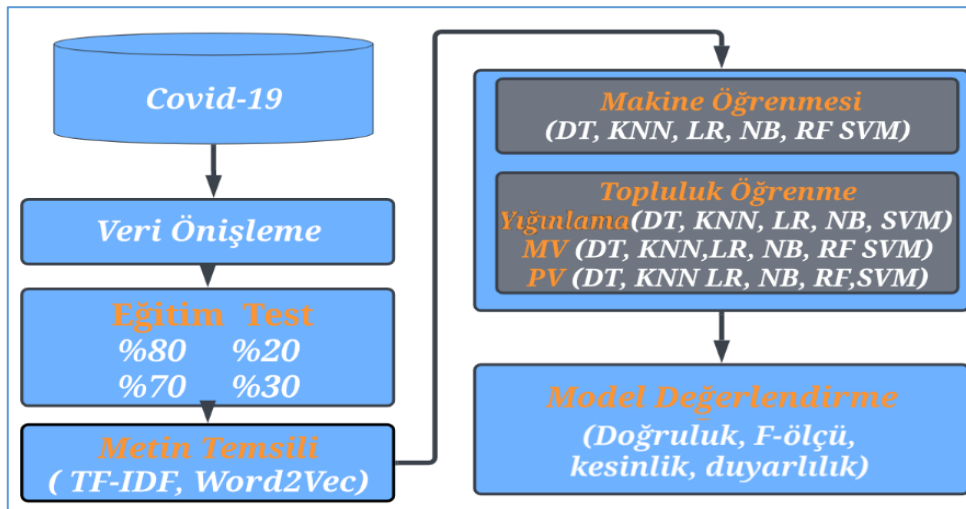
Çizelge 5.36. Rotten Tomatoes veri kümesinde doğruluk sonucu (%70 eğitim, %30 test).

Metin temsil yöntemi	NB	KNN	LR	SVM	Yığınlama	Oylama
TF-IDF	80,5	74,3	80,7	82,2	83,8	85,4
Word2Vec	81,3	80,1	81,6	82,7	85,8	87,5

Çizelge 5.36 incelendiğinde, TF-IDF ve Word2Vec metin temsil yöntemleri sonrasında oluşturulan modellerin doğruluk performanslarına ait sıralama benzer şekilde alınmıştır. Bu sıralama sırasıyla Oylama, Yığınlama, SVM, LR, NB ve KNN şeklindedir. Ayrıca Oylama, topluluk öğrenme yöntemleri arasında Yığınlama'ya göre daha iyi sonuçlar göstermiştir. Bu deney göstermiştir ki; makine öğrenmesi yöntemlerini tekil kullanmak yerine heterojen şekilde topluluk yöntemleriyle bir arada kullanmak model performansı olumlu etkilemektedir. Ayrıca yapılan deneyde; Word2Vec'in, TF-IDF'den daha iyi performans gösterdiği görülmektedir.

5.2.2. COVID-19 Twitter Veri Kümesi Üzerinde Yapılan Deney

Bu deneyde heterojen topluluk sisteminde çeşitlilik sağlamak için SVM, DT, RF, LR, NB, KNN algoritmaları kullanılmaktadır. Topluluk öğrenme yöntemlerinden Çoğunluk MV, PV ve Yığınlama kullanılmıştır. Çalışmanın akış şeması Şekil 5.14'te verilmiştir.



Şekil 5.14. Covid-19 Twitter veri kümesinde yapılan çalışmanın akış diyagramı.

5.2.2.1. Deneyde Kullanılan Veri Kümesi

Açık kaynaklı paylaşılan etiketli veriler üzerinde çalışılmıştır. Veri kümesi, Çizelge

6.13'te görüldüğü gibi 44955 veri ve 6 özellik içermektedir. Çalışmada sadece tweet içeriği ve duygu sınıfını barındıran etiket özelliği kullanılmıştır. Diğer dört özellik kaldırılmıştır. Çizelge 5.37'de, Twitter verileri hakkında bilgi vermektedir.

Çizelge 5.37. Covid-19 Twitter veri kümesi bilgisi.

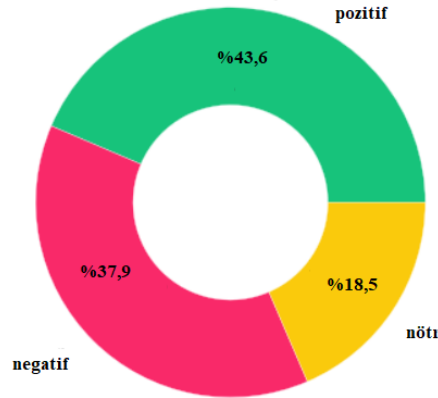
Öznitelik	Öznitelik Açıklaması
Username	Twitter kullanıcı adı
Screen Name	Twitter kullanıcılarına görünen isim
Location	Tweetin atıldığı konum
Tweet At	Tweetin ne zaman atıldığına dair bilgi
Tweet	Tweet içeriği
Label	Duygu sınıfı

Çizelge 5.38'de görüldüğü gibi tweet verilerinin duygu sınıfı dağılımı olumlu, olumsuz ve nötr olmak üzere üç sınıftan oluşmaktadır.

Çizelge 5.38. Covid-19 Twitter veri kümesi duygu sınıf dağılımı.

Duygu Sınıfı	Kelime Sayısı
Pozitif	19 592
Negatif	17 031
Nötr	8 332

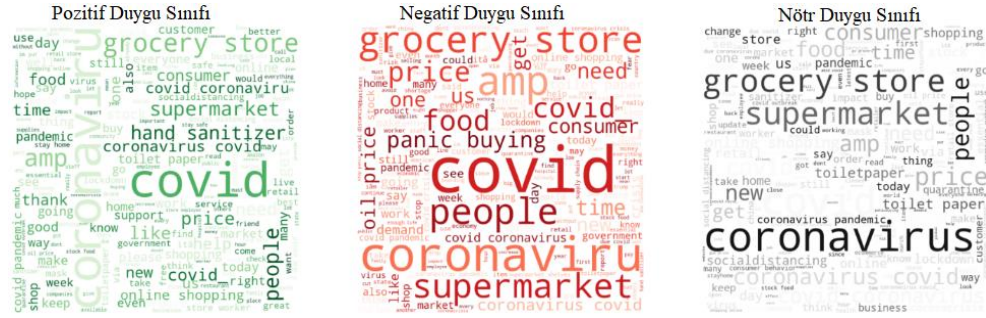
Covid-19 tweet verilerinin negatif, pozitif ve nötr toplamda üç sınıfına ait yüzde dağılımı, Şekil 5.15'te gösterilmektedir.



Şekil 5.15. Covid-19 Twitter veri kümesi duygu sınıfı dağılım görselleştirilmesi.

Şekil 4.15'te görüldüğü üzere veri kümesi etiketleri birbirine yakın gibi olmasına rağmen dengesiz bir dağılımı sahiptir. Bu dağılımı ölçmek için Çarpıklık (Skewnes) değeri kullanılır [85]. Bu veri kümesinin çarpıklık değeri -0,14'tür. Bu değerde -1 ile -0,5 arasında yer aldığı için orta derece çarpıktır. Şekil 5.16'da atılan tweetlerin duygu sınıflarına göre kelime bulutları yer almaktadır. Her üç duygu sınıfında da görüldüğü gibi

“süpermarket”, “pandemi”, “bakkal”, “mağaza”, “alışveriş” gibi temel ihtiyaçlar için kelimeler kullanılmaktadır. Bu durum sağlığı korumanın yanı sıra kendisinin ve ailesinin başta gıda olmak üzere hijyen gibi ihtiyaçlarını karşıladıkları yerlere yönelik duygularını da ortaya çıkarmıştır. Bu durum incelenmesi gereken bir diğer araştırma konusudur.



Şekil 5.16. Duygu sınıfı kelime bulutu.

5.2.2.2. Deney Kurulumu

Bu bölümde metin temsilleri, makine ve topluluk öğrenme algoritmaları ve veri kümeleri ele alınacaktır.

Sınıflandırma öncesinde 44955 Twitter kaydı ön işleme tabi tutulmuştur. Tweetlerdeki semboller, sayılar ve noktalama işaretleri, yasak kelimeler kaldırılmıştır. Karakterler küçük harfe dönüştürülmüş ve her kelimenin kökü bulunmuştur. “@username” gibi ifadeler ile “#coronavirus” gibi hashtag’ler kaldırılmıştır. Bu işlemler için Python NLTK kütüphanesi kullanılmıştır.

Bu çalışmada frekans tabanlı metin temsil yöntemlerinden BOW ve TF-IDF kullanılmıştır. Ayrıca metin temsil yöntemlerinden Word2Vec (CBOW ve Skip-Gram) ve dokümanların doğrudan vektörleştirilmesini sağlayan Doc2Vec (PV-DM, PV-DBOW) yöntemi kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Bu deneyde, Çizelge 5.39’da verilen Word2Vec Parametreleri kullanılmıştır.

Çizelge 5.39. Kullanılan Word2Vec parametreleri.

Parametre	Değerler
Size	100, 200
Window	5
Workers	8
Min_count	5
Sg	0 veya 1

Çizelge 5.40’da görüldüğü üzere, çalışmada Word2Vec yöntemi için vektör boyutu 100

ve 200 ve Skip-Gram ile CBOW yöntemi kullanılmıştır. Ayrıca pencere boyut değeri beş, veri kümesinde en çok geçen kelime değeri beş olarak alınmıştır. Çalışmada kullanılan diğer metin temsil yöntemleri için Çizelge 5.40’da verilen aynı parametre kullanılmıştır.

Çizelge 5.40. BOW ve TF-IDF yöntemi parametresi.

Parametre	Değer
max_count	10 000

Bu deney, AMD RYZEN 7 2700 işlemcili ve 64 GB RAM'e sahip yerel bir bilgisayarda çalıştırılmıştır.

5.2.2.3. Deney Sonuçları

Covid-19 veri kümesinin %70-%30 ve %80-%20 eğitim-test ayrımları yapıldıktan sonra duygu sınıflandırmaya yönelik modeller oluşturulmuştur. Sınıflandırma öncesi frekans tabanlı TF-IDF ve BOW ile tahmin tabanlı Word2Vec ve Doc2Vec yöntemleri kullanılmıştır. Makine öğrenmesi algoritmalarından; NB, KNN, DT, RF tekil sınıflandırmada kullanılmıştır. Bu algoritmaların bir arada kullanımına imkan veren MV, PV, ve Yığınlama topluluk öğrenmesi algoritmalarıyla da modeller oluşturulmuştur. Bu modellerin performanslarını değerlendirmek için doğruluk, kesinlik ve F-skor kullanılmıştır ve sonuçları bu bölümde anlatılmıştır. Doğruluk sonuçları; Çizelge 5.41 ve Çizelge 5.42’de, F-ölçü sonuçları; Çizelge 5.43 ve Çizelge 5.44’te, Kesinlik sonuçları; Çizelge 5.45 ve Çizelge 5.46’da Duyarlılık sonuçları; Çizelge 5.47 ve Çizelge 5.48’de verilmiştir.

Çizelge 5.41. Covid-19 veri kümesi üzerinde yapılan deneyin doğruluk sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	78	75	67	69	67	68	62	64	61	67
SVM	77	77	67	69	66	68	62	64	61	66
RF	79	78	72	74	72	73	67	67	64	71
NB	77	77	70	72	65	71	64	65	63	66
KNN	63	69	67	68	69	69	62	62	59	67
DT	71	69	62	61	60	61	65	65	64	69

Çizelge 5.41. (devam). Covid-19 veri kümesi üzerinde yapılan deneyin doğruluk sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
Yığınlama	82	80	84	89	84	85	79	81	83	83
MV	83	82	83	84	82	84	75	76	73	79
PV	81	81	83	84	82	84	75	76	83	79

Çizelge 5.41’de Covid-19 veri kümesinin frekans tabanlı metin temsili (TF-IDF, BOW) ve 100 ve 200 vektör boyutunda tahmin tabanlı metin temsil yöntemleri (Word2Vec ve Doc2Vec) sonrasında %80-%20 eğitim-test ayrımı ile oluşturulan duygu analizi modellerinin doğruluk sonuçları verilmiştir. MV; BOW ve TF-IDF yöntemlerinden sonra sınıflandırmada daha iyi performans göstermiştir. Yığınlama ise, tüm metin temsil yöntemlerinde benzer şekilde en iyi performansı göstermiştir. Tekil makine öğrenmesi yöntemleri kendi arasında değerlendirdiğinde RF tüm metin temsil yöntemlerinde daha iyi sonuç vermiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.42. Covid-19 veri kümesi üzerinde yapılan deneyin doğruluk sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	78	76	69	70	67	69	62	64	61	67
SVM	79	78	73	74	74	74	67	67	65	68
RF	75	73	72	72	73	73	65	66	63	67
NB	77	77	68	70	67	69	62	63	61	66
KNN	62	69	67	67	68	67	60	69	69	67
DT	71	70	61	61	61	61	64	66	64	69
Yığınlama	83	82	84	85	84	85	81	80	83	83
MV	82	81	83	84	83	84	76	77	73	79
PV	82	80	82	83	80	83	75	77	73	79

Çizelge 5.42'de Covid-19 veri kümesinin, %80-%20 eğitim-test ayrımı sonrasında frekans tabanlı metin temsili (TF-IDF, BOW) ve 100 ve 200 vektör boyutunda tahmin tabanlı metin temsili (Word2Vec, Doc2Vec) yöntemleriyle oluşturulan duygu analizi modellerinin doğruluk sonuçları görülmektedir. Tüm metin temsil yöntemlerinde sonra oluşturulan modellerde Yığınlama daha iyi performans göstermiştir. Tekil makine öğrenmesi yöntemleri kendi arasında değerlendirdiğinde; SVM tüm metin temsil yöntemlerinde daha iyi sonuç göstermiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.43. Covid-19 veri kümesi üzerinde yapılan deneyin f-ölçü sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	80,76	80,76	72,39	73,11	71,12	73,19	72,24	69,32	69,33	71,45
SVM	81,01	81,54	72,65	73,86	72,31	73,52	72,44	70,35	70,44	72,35
RF	80,91	81,03	72,5	73,66	71,25	73,18	72,25	69,64	69,25	71,64
NB	76,91	76,64	72,64	73,64	71,64	73,10	64,17	65,22	62,53	66,1
KNN	62,63	69,24	67,45	67,76	68,64	68,77	61,92	62,12	68,68	66,68
DT	71,38	69,47	61,87	61,07	60,1	61,05	54,96	54,81	64,29	66,88
Yığınlama	82,26	82,07	75,7	76,44	72,49	77,85	72,54	73,63	72,71	74,64
MV	81,93	81,52	73,89	75,74	73,69	75,29	48,92	70,08	72,01	72,61
PV	81,32	81,32	73,7	74,44	72,49	73,85	65,14	66,07	71,55	72,46

Çizelge 5.43'te Covid-19 veri kümesinin, %80-%20 eğitim-test ayrımı sonrasında frekans tabanlı metin temsili (TF-IDF, BOW) ve 100 ve 200 vektör boyutunda tahmin tabanlı metin temsili (Word2Vec, Doc2Vec) yöntemleriyle oluşturulan duygu analizi modellerinin F-ölçü sonuçları verilmiştir. Tüm metin temsili yöntemlerinden sonra oluşturulan modellerde; Yığınlama topluluk öğrenme modeli, diğer modellere göre daha iyi sonuçlar vermiştir. Tekil makine öğrenmesi modelleri arasında en iyi sonuçları SVM vermiştir. Ancak LR ve RF makine öğrenmesi yöntemleri tüm metin temsil yöntemlerinde SVM'e oldukça yakın sonuçlar vermiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.44. Covid-19 veri kümesi üzerinde yapılan deneyin f-ölçü sonuçları (%70 eğitim, %30 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	82,67	81,21	76,06	76,81	75,32	74,19	72,75	73,28	72,33	74,64
SVM	82,01	82,03	75,95	76,74	75,32	74,13	72,75	73,21	72,22	74,51
RF	81,91	81,57	74,11	74,79	75,29	74,43	71,19	71,76	72,25	74,56
NB	74,19	74,17	74,17	74,17	74,17	74,17	65,25	66,22	62,53	66,1
KNN	61,9	69,18	66,82	67,17	67,97	67,37	59,86	59,44	58,68	66,68
DT	70,55	69,78	60,66	60,88	60,83	60,62	54,34	55,8	54,29	58,88
Yığınlama	82,92	82,42	76,79	76,91	75,98	76,3	73,02	74,1	72,73	77,61
MV	81,96	81,96	74,81	74,27	75,22	74,22	65,44	66,15	63,07	68,64
PV	81,86	81,69	74,81	74,27	75,22	74,22	65,44	66,11	71,07	72,64

Çizelge 5.44'te Covid-19 veri kümesinin %70-%30 eğitim-test ayrımı ile frekans tabanlı metin temsilinden (TF-IDF, BOW) ve 100 ve 200 vektör boyutlarında tahmin tabanlı metin temsili (Word2Vec, Doc2Vec) sonrasında oluşturulmuş duygu analizi modellerinin F-ölçü sonuçları verilmiştir. Yığınlama, tüm metin temsil yöntemleriyle oluşturulan modellerin hepsinde daha iyi performans göstermiştir. Ancak Yığınlamaya; MV, PV topluluk öğrenmesi yöntemleriyle LR, RF, SVM makine öğrenmesi yöntemleri BOW ve TF-IDF metin temsillerinde yakın sonuçlar vermiştir. Ayrıca Word2Vec ve Doc2Vec metin temsil yöntemlerinin 100 ve 200 vektör boyutlarında da SVM makine öğrenmesi yöntemi Yığınlama topluluk öğrenmesine oldukça yakın sonuç vermiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.45. Covid-19 veri kümesi üzerinde yapılan deneyin kesinlik sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	80	81	73	74	72	73	66	69	67	71
SVM	78	78	68	74	71	72	66	69	67	71
RF	79	80	71	72	71	71	65	67	62	71
NB	77	77	73	74	66	74	65	66	62	66

Çizelge 5.45. (devam). Covid-19 veri kümesi üzerinde yapılan deneyin kesinlik sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
KNN	63	69	67	67	68	67	60	60	69	67
DT	72	72	60	61	61	60	64	66	64	69
Yığınlama	83	82	74	75	74	75	71	70	70	72
MV	82	82	73	74	73	74	66	68	64	70
PV	81	80	71	74	72	72	65	67	64	70

Çizelge 5.45'te, Covid-19 veri kümesinin %80-%20 eğitim-test ayrımı sonrasında frekans tabanlı metin temsilden (TF-IDF, BOW) ve kelime 100 ve 200 vektör boyutlarında (Word2Vec, Doc2Vec) oluşturulmuş duygu analizi modellerinin kesinlik sonuçları verilmiştir. Yığınlama, tüm metin temsil yöntemleriyle oluşturulan bütün modellerde daha iyi performans göstermiştir. Yığınlamaya, tüm metin temsil yöntemlerinde MV, PV topluluk öğrenmesi yöntemleriyle, LR, SVM ve RF makine öğrenmesi yöntemleri Yığınlamaya yakın sonuçlar vermiştir. Tekil makine öğrenmesi yöntemleri kendi arasında karşılaştırıldığında LR daha iyi sonuç vermiştir. Topluluk öğrenmesi ile alınan sonuçlar tekil makine öğrenmesi yöntemleriyle alınan sonuçların önündedir.

Çizelge 5.46. Covid-19 veri kümesi üzerinde yapılan deneyin kesinlik sonuçları (%70 eğitim, %30 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	80	80	71	70	70	72	64	65	60	65
SVM	80	80	71	72	71	71	65	66	61	66
RF	80	80	71	71	70	70	64	65	60	65
NB	64	79	70	68	68	69	62	62	60	65
KNN	73	71	72	61	60	61	61	62	59	60
DT	78	80	73	74	72	72	60	60	59	62
Yığınlama	81	82	74	75	74	75	67	70	68	73
MV	79	81	71	72	70	72	66	66	62	69
PV	80	80	71	72	70	72	65	66	60	66

Çizelge 5.46’da, Covid-19 veri kümesinin %70-%30 eğitim-test ayrımı sonrasında frekans tabanlı metin temsilinden (TF-IDF, BOW) ve kelime 100 ve 200 vektör boyutlarından (Word2Vec, Doc2Vec) sonra duygu analizi modellerinin kesinlik sonuçları verilmiştir. Yığınlama, tüm metin temsil yöntemleriyle oluşturulan modellerde daha iyi performans göstermiştir. Tüm metin temsil yöntemlerinde; MV, PV topluluk öğrenmesi yöntemleriyle, LR, SVM ve RF makine öğrenmesi yöntemleri yığınlamaya yakın sonuçlar vermiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.47. Covid-19 veri kümesi üzerinde yapılan deneyin duyarlılık sonuçları (%80 eğitim, %20 test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	77	74	66	68	65	67	60	61	60	61
SVM	76	77	66	68	65	67	60	61	60	61
RF	80	80	72	71	65	72	72	66	72	66
NB	77	76	72	74	64	71	64	65	62	66
KNN	63	69	67	67	68	68	62	62	60	67
DT	72	70	62	61	60	61	61	60	62	60
Yığınlama	82	81	74	75	73	75	70	70	70	72
MV	81	80	72	74	72	73	66	67	64	69
PV	81	80	72	74	72	73	66	67	72	69

Çizelge 5.47’ de, frekans tabanlı metin temsili (TF-IDF, BOW) ve 100 ve 200 vektör boyutları (Word2Vec, Doc2Vec) sonrasında %80-%20 eğitim-test ayrımı ile rastgele oluşturulmuş duygu analizi modellerinin duyarlılık sonuçları verilmiştir. Yığınlama, tüm metin temsil yöntemleriyle oluşturulan modellerde daha iyi performans göstermiştir. Ancak MV, PV ve RF; BOW ve TD-IDF metin temsil yöntemlerinde yığınlamaya yakın sonuçlar vermiştir. Tekil makine öğrenmesi yöntemleri kendi arasında değerlendirildiğinde; bütün metin temsil yöntemlerinde RF daha iyi sonuç vermiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.48. Covid-19 veri kümesi üzerinde yapılan deneyin duyarlılık sonuçları
(80:20 eğitim:test).

Sınıflandırma Algoritmaları	Metin Temsil Yöntemleri									
	BOW	TF-IDF	Word2Vec				Doc2Vec			
			CBOW		Skip-Gram		PV-DM		PV-DBOW	
			100	200	100	200	100	200	100	200
LR	77	74	67	68	65	67	60	61	62	64
SVM	76	77	66	68	65	67	60	60	61	64
RF	80	78	71	72	72	72	66	66	61	68
NB	77	77	73	74	66	74	62	66	62	66
KNN	63	69	67	67	68	67	60	60	60	67
DT	72	71	60	61	61	60	61	66	64	60
Yığınlama	83	82	74	75	74	75	64	70	65	72
MV	82	82	73	74	73	74	65	68	64	69
PV	82	82	73	74	73	74	66	67	64	69

Çizelge 5.48’de, frekans tabanlı metin temsili (TF-IDF, BOW) ve 100 ve 200 vektör boyutları (Word2Vec, Doc2Vec) sonrasında %70-%30 eğitim-test ayrımı ile rastgele oluşturulmuş duygu analizi modellerinin duyarlılık sonuçları verilmiştir. Bütün metin temsil yöntemlerinde, topluluk öğrenme modelleri, tekil makine öğrenmesi modellerinden daha iyi sonuç vermiştir. Tekil makine öğrenmesi modelleri kendi içinde değerlendirildiğinde; RF daha iyi sonuç vermiştir. Topluluk öğrenme modelleri kendi içinde değerlendirildiğinde; tüm metin temsil yöntemlerinde Yığınlama öndedir. Fakat MV ve PV topluluk öğrenmesi sonuçları da bütün metin temsil yöntemlerinde yığınlamaya oldukça yakındır.

Çalışmada en iyi sonuçları veren yığınlama %80-%20 eğitim-test ayrımına sahip 200 vektör boyutlu CBOW Word2Vec modelidir. En iyi sonucu veren modelin literatürde karşılaştırılması Çizelge 6.24’tedir. Çizelge 6.24’te yer alan önceki çalışmalar, Twitter platformundan toplanan Covid-19 pandemisinde atılan tweetlerden oluşmaktadır. Çizelge 5.49’da çalışmayı yapan yazar, geliştirdiği model ve doğruluk sonucu verilmiştir. Topluluk öğrenmesi ile alınan sonuçlar, tekil makine öğrenmesi yöntemleriyle alınan sonuçlarına göre daha öndedir.

Çizelge 5.49. Covid-19 veri kümesi verilerine yönelik diğer çalışmalar.

Yazar	Model	Doğruluk (%)
Machuca ve arkadaşları [43]	LR	78,5
Siddiqua ve arkadaşları [44]	MV	87,7
Rahman ve Islam [45]	Torbalama	83,2
	Oylama	83,3
	Yığınlama	83,5
Barkur ve arkadaşları [46]	LR	74,29
Marga ve arkadaşları [47]	NB	84
Imran arkadaşları [48]	RNN	82,4
Buntoro arkadaşları [49]	SVM	84,38
Vernikou ve arkadaşları [50]	LSTM	65
Antonio ve arkadaşları [51]	SGD	65
En iyi sonuç veren model	CBOW ile Yığınlama	89

Çizelge 5.49’da görüldüğü üzere, çalışma da yapılan deneylerde en iyi sonucu veren model daha iyi sonuç vermiştir. Bu sonuç; literatürle rekabet eden model oluşturulduğunu göstermektedir. Tekil makine öğrenmesi algoritmalarının heterojen şekilde topluluk öğrenme algoritmalarında bir arada kullanılmasının, tekil sınıflandırıcılara göre sınıflandırma performansı açısından önemli bir avantaj elde ettiği görülmüştür. Tüm temsil yöntemleri içinde Word2Vec sonrası Yığınlama, bütün modeller arasında en iyi performansı vermiştir.

6. SONUÇ VE TARTIŞMA

IMDB veri kümesi üzerinde MBiGRUMCONV adıyla önerilen yöntemle ait yapılan deneyler sırasında veri kümesi, %80-%20-%10 ve %70-%30-%10 sına testi eğitim-test-doğrulama ayrımlarına bölünerek bölünmesinin yanında 3-5-10 kat çapraz doğrulama sonuçları da alınmıştır. Önerilen yöntemde metin temsil yöntemlerinden Word2Vec kullanılmıştır. Tez çalışmasında GloVe, BERT gibi ön eğitimli metin temsil yöntemleriyle de sonuçlar alınmış ve deneysel çalışmalar ve sonuçları bölümde sonuçları verilmiştir. Önerilen yöntemde Word2Vec yerine BERT kullanımı ve NLP görevlerinde güncel yaklaşım olan BERT-CNN modelleriyle alınan sonuçlar karşılaştırılmıştır. BERT-CNN literatürde olduğu gibi yaygın kullanılan Word2Vec yöntemine göre daha iyi sonuç vermiştir. Fakat word2Vec sonrası MBiGRUONV, BERT sonrası MBiGRUONV'un test ayırımında bir adım önündedir. Test ayırımında ki bu sonuç, doğrulama ayırımında da benzer şekildedir. Test setinde en iyi sonucu BERT-CNN %90,72 doğruluk değeri ile veririrken; Word2Vec sonrası MBiGRUMCONV modelinde sına testi ayırımında doğruluk değeri %90,59 ve 5 kat çapraz geçirme ayırımında testi doğruluk değeri %90,67'dir. BERT-MBiGRUMCONV yöntemi %89,63 ile üçüncüdür. Word2Vec sonrası MBiGRUMCONV ile BERT sonrası MBiGRUMCONV birbirine yakın sonuçlar vermiştir. Önerilen modelde dönüştürücü gibi daha güncel yöntemlere yerine kelime gömme kullanılmasıyla da yüksek başarılı modeller oluşturulabileceği görülmüştür.

IMDB veri kümesinde alınan sonuçlar değerlendirildiğinde, performans artışının sebeplerinden biri olarak Word2Vec'in kendi içinde cümledeki kelimeyi tahmin etmek için bağlamın dağıtılmış temsillerini kullanan CBOW'un yerine hedef sözcüğün solundaki ve sağındaki çevreleyen sözcüklerin olasılıkları toplamından elde edilen Skip-Gram'ın daha verimli çalışmasından dolayı olduğu değerlendirilmektedir. İkinci bir sebep de vektör boyutunun artmasının sınıflandırmaya olumlu etki ettiği değerlendirilmiştir.

Tez kapsamında yapılan diğer çalışmalardan olan BERT'in ön eğitimli olması ve daha çok kelime gömme katmanına karşılık gelen ve içinde dikkat mekanizması barındırması sebebiyle literatüre benzer şekilde daha iyi sonuç vermiştir.

Gerek literatür çalışmalarının gerekse önerilen modelin sonuçları değerlendirildiğinde; çok katmanlı ve hibrit modeller, tekil sinir ağı modellerinin kullanımlarına göre daha iyi sonuçlar verdiği görülmüştür. Bu yüzden, bu alanda çalışma yapacak olan araştırmacılara çok katmanlı ve hibrit yapıda sinir ağı mimarileri kullanmaları önerilmektedir. Ayrıca daha yüksek performanslı modeller oluşturmak için, BERT ve benzeri dönüştürücülerle metinler vektörleştirildikten sonra çift yönlü sinir ağlarıyla topluluk öğrenme modelleri destekli yeni hibrit yöntemler geliştirilebilir.

Tez süresince yapılan diğer çalışmalardan, TripAdvisor ve Rotten Tomatoes veri kümeleri üzerinde yapılan deneylerde alınan sonuçlarda; Word2Vec'in TF-IDF'den daha iyi performans gösterdiği görülmektedir. Word2Vec gibi tahmine dayalı metin temsil yöntemleri saymaya dayalı yöntemlere göre öncelikli olarak tercihe dileyebilir. Ayrıca word2Vec yöntemine alternatif olarak fastText gibi tahmine dayalı yöntemler de gelecek çalışma olarak planlanmıştır. Bu deneyin diğer sonucu, makine öğrenme yöntemlerini tekil kullanmak yerine heterojen şekilde bir arada topluluk öğrenme algoritmalarında kullanmak daha başarılı sonuçlar elde edilmesini sağlamıştır.

Tez süresince yapılan diğer çalışmalardan bir diğeri olan Covid-19 veri kümesi üzerinde alınan sonuçlar ışığında; makine öğrenmesi algoritmalarının heterojen şekilde topluluk öğrenme algoritmalarıyla bir arada kullanımı, bu algoritmaların tekil kullanımına göre sınıflandırma performansı açısından önemli bir avantaj elde ettiği görülmüştür. TripAdvisor ve Rotten Tomatoes veri kümeleri üzerinde yapılan deney çalışmaları da, tekil kullanıma oranla daha iyi sonuç alınması tekrar etmiştir. Bu iki deney sonrası çalışmada kullanılmayan güçlendirme gibi farklı topluluk yöntemleri kullanımıyla yeni modeller oluşturulması gelecek çalışma olarak düşünülmektedir. Bu çalışmada kısıtıdır. Daha sonraki hedef olarak bu tür işaretlemelerde dikkate alınarak NLP'nin diğer görevlerinde yeni model geliştirme yapılacaktır.

Bu çalışma göstermiştir ki; NLP görevlerinden olan duygu analizinde Word2Vec gibi metin temsillerinin farklı vektör boyutu ve farklı yöntemlerde ince ayar yapılarak en iyi sonuç alınırken ön eğitilmiş BERT sonrası hibrit modellerle daha hızlı şekilde ve performansı daha iyi modeller oluşturulabilir.

-Tek katmanlı mimari yerine çok katmanlı hibrit mimariler tercih edilmesi model performansını olumlu yönde etki ettiği görülmüştür.

-Sinir ağlarının çift yönlü versiyonlarının model performansına etkisinin olumlu olduğu

görülmüştür.

-Evrışimli ağların, NLP görevlerinden duygu analizi üzerine oluşturulan modellerin performansına olumlu etki ettiği görülmüştür.

Geleceğe yönelik öneriler;

-Başlık sınıflandırma, soru cevaplama gibi NLP'nin farklı görevlerinde, çalışmada önerilen yöntem gibi çift yönlü sinir ağlarından oluşan çok katmanlı hibrit derin öğrenme mimarilerinde öncelikli olarak tercih edilmesi tavsiye edilmektedir.

-Roberta, Distilbert, Bart, Albert gibi farklı Transformers yapıları kullanılabilir.

- TF-IDF, Doc2Vec veya Word2Vec gibi metin temsil yöntemlerine yerine ön eğitilmiş dönüştürücü temelli yöntemlerin tercih edilmesi başarılı modeller oluşturmaya katkı sağlayabileceği için tavsiye edilmektedir.



7. KAYNAKLAR

- [1] D. Chaffey. (2023, 1 Şubat). *Global social media statistics research summary* [Online]. Erişim: <https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research>.
- [2] M. S. Başarslan & F. Kayaalp, “MBi-GRUMCONV: A novel multi Bi-GRU and multi CNN-Based deep learning model for social media sentiment analysis,” *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1-16, 2023.
- [3] D. Austin. (2023, 20 Nisan). *2023 Internet Minute Infographic, by eDiscovery Today and LTMG: eDiscovery Trends* [Online]. Erişim: <https://ediscoverytoday.com/2023/04/20/2023-internet-minute-infographic-by-ediscovery-today-and-ltmg-ediscovery-trends/>.
- [4] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, & C. Potts, “Learning word vectors for sentiment analysis,” *ACL-HLT Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, USA, 2011, pp. 142–150.
- [5] N. Gervais. (2019, 10 Haziran). Rotten Tomatoes Dataset [Online]. Erişim: <https://github.com/nicolas-gervais/rotten-tomatoes-dataset>.
- [6] M. H. Alam, W.J. Ryu, & S. Lee, “Joint multi-grain topic sentiment: modeling semantic aspects for online reviews,” *Information Sciences*, vol. 339, no. 1, pp. 206–223, 2016.
- [7] A. Miglani, (2021, 5 Ocak). *Coronavirus tweets NLP- Text Classification* [Online]. Erişim: <https://www.kaggle.com/datatattle/covid-19-nlp-textclassification>.
- [8] C. Vielma, A. Verma, & D. Bein, “Single and multibranch CNN-Bidirectional LSTM for IMDb sentiment analysis,” in *17th International Conference on Information Technology--New Generations (ITNG 2020)*, İsviçre: Springer International Publishing, 2020, pp. 401–406.
- [9] M. R. Haque, S. Akter Lima, & S. Z. Mishu, “Performance analysis of different neural networks for sentiment analysis on IMDB movie reviews,” *3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, Rajshahi, Bangladesh: Springer, 2019, pp. 161–164.
- [10] N. C. Dang, M. N. Moreno-García, & F. D. Prieta, “Sentiment analysis based on deep learning: a comparative study,” *Electronics (Basel)*, vol. 9, no. 3, pp. 30-48, 2020.
- [11] Y. Gao, W. Rong, Y. Shen, & Z. Xiong, “Convolutional neural network based sentiment analysis using adaboost combination,” *Proceedings of the International Joint Conference on Neural Networks*, Vancouver, Canada, 2016, pp. 1333–1338.
- [12] Z. Hameed & B. Garcia-Zapirain, “Sentiment classification using a single-layered BiLSTM model,” *IEEE Access*, vol. 8, no. 1, pp. 73992–74001, 2020.

- [13] S. M. Qaisar, "Sentiment analysis of IMDb movie reviews using long short-term memory," *2020 2nd International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2020, pp. 2020–2033.
- [14] A. Pimpalkar & J. R. Raj R, "MBiLSTM GloVe: embedding GloVe knowledge into the corpus using multi-layer BiLSTM deep learning model for social media sentiment analysis," *Expert Systems with Application*, vol. 203, no.4, pp. 117-141, 2022.
- [15] N. D. Derra & D. Baier, "Working in detail: how LSTM hyperparameter selection influences sentiment analysis results," *Archives of Data Science, Series A*, vol. 6, no. 1, pp. 1–22, 2020.
- [16] S. Prabha, K. S & P. N. Karthikayan, "For Movie reviews, a sentiment analysis using long short term memory networks," *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 9, pp. 1758–1766, 2021.
- [17] R. Ni & H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," in *2020 39th Chinese Control Conference (CCC)*, Shenyang, China, 2020, pp. 7492–7497.
- [18] N. Rathee, N. Joshi and J. Kaur, "Sentiment analysis using machine learning techniques on Python," *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2018, pp. 779-785.
- [19] W. Li, W. Shao, S. Ji, & E. Cambria, "BiERU: Bidirectional emotional recurrent unit for conversational sentiment analysis," *Neurocomputing*, vol. 467, no. 2, pp. 73–82, 2022.
- [20] K. Amulya, S. B. Swathi, P. Kamakshi, & Y. Bhavani, "Sentiment analysis on IMDB movie reviews using machine learning and deep learning algorithms," *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2022, pp. 814–819.
- [21] S. Sabba, N. Chekired, H. Katab, N. Chekkai, & M. Chalbi, "Sentiment analysis for IMDb reviews using deep learning classifier," in *2022 7th International Conference on Image and Signal Processing and their Applications (ISPA)*, Mostaganem, Algeria, 2022, pp. 1–6.
- [22] A. Hassan & A. Mahmood, "Deep learning approach for sentiment analysis of short texts," *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, Nagoya, Japan, 2017, pp. 705–710.
- [23] J. Camacho-Collados & M. T. Pilehvar, "On the role of text preprocessing in neural network architectures: an evaluation study on text categorization and sentiment analysis," *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Stroudsburg, USA: Association for Computational Linguistics, 2018, pp. 40–46.
- [24] M. Ghorbani, M. Bahaghighat, Q. Xin, & F. Özen, "ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing," *Journal of Cloud Computing*, vol. 9, no. 1, pp. 1-16, 2020.
- [25] N. Mohamed Ali, M. M. A. El Hamid, & A. Youssif, "Sentiment analysis for movies reviews dataset using deep learning models," *International Journal of Data Mining & Knowledge Management Process*, vol. 9, no. 3, pp. 19–27, 2019.
- [26] Y. Ma, H. Fan, & C. Zhao, "Feature-Based fusion adversarial recurrent neural networks for text sentiment classification," *IEEE Access*, vol. 7, no. 1, pp. 132542–

132551, 2019.

- [27] P. Cen, K. Zhang, & D. Zheng, “Sentiment analysis using deep learning approach,” *Journal on Artificial Intelligence*, vol. 2, no. 1, pp. 17–27, 2020.
- [28] Johnson, R., & Zhang, T. Semi-supervised convolutional neural networks for text categorization via region embedding. *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Kanada, 2015.
- [29] Z. H. Zhou & J. Feng, “Deep forest,” *National Science Review*, vol. 6, no. 1, pp. 74–86, 2019.
- [30] S. Wang & C. D. Manning, “Baselines and bigrams: Simple, good sentiment and topic classification,” *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012- Proceedings of the Conference*, vol. 2, no. 2, pp. 90–94, 2012.
- [31] B. S. Harish & K. Kumar, “Sentiment analysis on IMDb movie reviews using hybrid feature extraction method,” vol. 5 no. 5, pp. 109-114, 2018,
- [32] D. Q. Nguyen, D. Q. Nguyen, T. Vu, & S. B. Pham, “Sentiment Classification on Polarity Reviews: An Empirical Study Using Rating-based Features,” in *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Stroudsburg, Portland, USA: Association for Computational Linguistics, 2014, pp. 128–135.
- [33] M. Mohaiminul & N. Sultana, “Comparative Study on machine learning algorithms for sentiment classification,” *International Journal Computer Applications*, vol. 182, no. 21, pp. 1–7, 2018.
- [34] V. Narayanan, I. Arora, & A. Bhatia, “Fast and accurate sentiment classification using an enhanced Naive Bayes model,” *International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2013 Hefei, Çin*, 2013, pp. 194–201.
- [35] P. Tiwari, B. K. Mishra, S. Kumar, & V. Kumar, “Implementation of n-gram methodology for rotten tomatoes review dataset sentiment analysis,” in *Cognitive Analytics*, USA:IGI Global, 2020, pp. 689–701.
- [36] T. P. Sahu & S. Ahuja, “Sentiment analysis of movie reviews: A study on feature selection & classification algorithms,” in *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*, Durgapur, India, 2016, pp. 1–6.
- [37] O. R. H., G. Virginia, and A. R. C., “Sentiment classification of film reviews using IB1,” *2016 7th International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, Bangkok, Thailand, 2016, pp. 78–82.
- [38] L. Mostafa, “Machine learning-based sentiment analysis for analyzing the travelers reviews on Egyptian hotels,” *Kahire, Mısır*, 2020, pp. 405–413.
- [39] R. Dehkharghani, B. Yanikoglu, D. Tapucu, & Y. Saygin, “Adaptation and Use of Subjectivity Lexicons for Domain Dependent Sentiment Classification,” in *2012 IEEE 12th International Conference on Data Mining Workshops*, Brüksel, Belçika, 2012, pp. 669–673.
- [40] V. B. Raut & D. D. Londhe, “Opinion mining and summarization of hotel reviews,” in *2014 International Conference on Computational Intelligence and*

Communication Networks, Bhopal, India, 2014, pp. 556–559.

- [41] Y. Zhou, “Sentiment classification with deep neural network,” Yüksek lisans tezi, Faculty of Information Technology and Communication Sciences, Department of Computer Science, Tampere University, Tampere, Finland, 2019.
- [42] L. Mostafa, “Egyptian student sentiment analysis using Word2vec during the Coronavirus (Covid-19) Pandemic,” *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, İsviçre: Springer International Publishing, 2020, pp. 195–203.
- [43] C. R. Machuca, C. Gallardo, & R. M. Toasa, “Twitter sentiment analysis on coronavirus: machine learning approach,” *Journal of Physics*, vol. 1828, no. 1, pp. 333-342, 2021.
- [44] U. A. Siddiqua, T. Ahsan, & A. N. Chy, “Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog,” in *2016 19th International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2016, pp. 304–309.
- [45] M. M. Rahman & M. N. Islam, “Exploring the performance of ensemble machine learning classifiers for sentiment analysis of Covid -19 Tweets,” Singapur: Springer, 2022, pp. 383–396.
- [46] G. Barkur, Vibha, & G. B. Kamath, “Sentiment analysis of nationwide lockdown due to Covid-19 outbreak: Evidence from India,” *Asian Journal of Psychiatry*, vol. 51, no. 3, pp. 102089, 2020.
- [47] A. R. Isnain, N. S. Marga, & D. Alita, “Sentiment analysis of government policy on corona case using Naive Bayes algorithm,” *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 1, pp. 55-64, 2021.
- [48] A. S. Imran, S. M. Daudpota, Z. Kastrati, & R. Batra, “Cross-Cultural polarity and emotion detection using sentiment analysis and deep learning on Covid-19 related tweets,” *IEEE Access*, vol. 8, nol: 2, pp. 181074–181090, 2020.
- [49] G. A. B. & M. B. S. R. Mahendrajaya, “Gopay User Sentiment Analysis Using the Lexicon Based Method and Support Vector Machine,” *Komputek*, vol. 3, no. 2, pp. 52-60, 2019.
- [50] S. Vernikou, A. Lyras, & A. Kanavos, “Multiclass sentiment analysis on Covid-19-related tweets using deep learning models,” *Neural Computing Applicaitons*, vol. 34, no. 22, pp. 19615–19627, 2022.
- [51] V. Dwi Antonio, S. Efendi, & H. Mawengkang, “Sentiment analysis for Covid-19 in Indonesia on Twitter with TF-IDF featured extraction and stochastic gradient descent,” *International Journal of Nonlinear Analysis and Applications*, vol. 13, no. 1, pp. 2008–6822, 2022.
- [52] A. Yörükoğlu, *Çocuk ruh sağlığı*. İstanbul: Özgür Yayınları, 2004.
- [53] Ş. Ş. Elmas, “Sosyal medya mesajlarının veri madenciliği yöntemi ile duygu analizi,” Yüksek lisans tezi, Yönetim Bilişim Sistemleri, Sosyal Bilimler Enstitüsü, Cumhuriyet Üniversitesi, Sivas, Türkiye, 2019.
- [54] K. Arslan, “Eğitimde yapay zekâ ve uygulamaları,” *Batı Anadolu Eğitim Bilimleri Dergisi*, vol. 11, no. 1, pp. 71–80, 2020.
- [55] K. Ozturk & M. Şahin, “Yapay sinir ağları ve yapay zekâ’ya genel bir bakış”

- Takvim-i Vekayi*, vol. 6, no. 2, pp. 25–36, 2018.
- [56] D. W. Otter, J. R. Medina, & J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.
- [57] S. E. Seker, “Doğal dil işleme,” *YBS Ansiklopedi*, vol. 2, no. 4, pp. 14–31, 2015.
- [58] Oracle, (2019, 1 Ekim), *Doğal Dil İşleme* [Online]. Erişim: <https://www.oracle.com/tr/artificial-intelligence/what-is-natural-language-processing/>
- [59] S. Bird, “NLTK,” *Proceedings of the COLING/ACL on Interactive presentation sessions-*, Morristown, New Jersey, USA, 2006, pp. 69–72.
- [60] J. Thanaki, *Python natural language processing*. USA: Packt Publishing, 2017.
- [61] M. Abadi, “TensorFlow: learning functions at scale,” *In Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming*, New York, USA, 2016, pp. 1-1.
- [62] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, G., ... & S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, Vancouver, Kanada, 2019, pp. 1-12.
- [63] H. Haddad, A. C. Rouhou, A. Messaoudi, A. Korched, C. Fourati, A. Sellami, ... & F. Ghriss, “TunBERT: Pretraining BERT for tunisian dialect understanding,” *SN Computer Science*, vol. 4, no. 2, pp. 194-206, 2023.
- [64] V. Moscato, M. Postiglione, C. Sansone, & G. Sperli, “TaughtNet: learning multi-task biomedical named entity recognition from single-task teachers,” *Journal of Biomedical and Health Informatics*, vol. 27 no. 5. pp. 2512–2523, 2023.
- [65] J. Devlin, M. W. Chang, K. Lee, & K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” *NAACL HLT 2019- Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Minnesota, USA, 2019, pp. 4171–4186.
- [66] J. Pennington, R. Socher, & C. D. Manning, *GloVe: Global Vectors for Word Representation. Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, 2014, pp. 1532-1543.
- [67] M. Garg, A. K. Gupta, & R. Prasad, *Graph Learning and Network Science for Natural Language Processing*, 1. Baskı, Boca Raton: CRC Press, 2022.
- [68] T. A. Rana, & Y. N. Cheah, “Aspect extraction in sentiment analysis : comparative analysis and survey,” *Artifiial Intelligence Review*, vol. 46, no. 4, pp. 459–483, 2016.
- [69] G. D’Aniello, M. Gaeta, & I. La Rocca, “KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis,” *Artifiial Intelligence Review*, vol. 55, no. 7, pp. 5543–5574.
- [70] S. Behdenna, F. Barigou, & G. Belalem, “Document level sentiment analysis: a survey,” *Endorsed Transactions on Context-aware Systems and Applications*, vol. 4, no. 13, p. 154-339, 2018.

- [71] E. Cambria, D. Das, S. Bandyopadhyay, & A. Feraco, “Affective computing and sentiment analysis,” *A Practical Guide to Sentiment Analysis. Socio-Affective Computing*, İsviçre: Springer, 2017, pp. 1–10.
- [72] B. Liang, H. Su, L. Gui, E. Cambria, & R. Xu, “Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks,” *Knowledge Based Systems*, vol. 235, no:1, pp. 107-143.
- [73] F. Hemmatian & M. K. Sohrabi, “A survey on classification techniques for opinion mining and sentiment analysis,” *Artificial Intelligence Review*, vol. 52, no. 3, pp. 1495–154, 2019.
- [74] M. Birjali, M. Kasri, & A. Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” *Knowledge Based Systems*, vol. 226, no: 3, pp. 107-134.
- [75] Q. Ye, Z. Zhang, & R. Law, “Sentiment classification of online reviews to travel destinations by supervised machine learning approaches,” *Expert Systems with Applications*, vol. 36, no. 3, pp. 6527–6535.
- [76] Q. Liu, J. Lu, G. Zhang, T. Shen, Z. Zhang, & H. Huang, “Domain-specific meta-embedding with latent semantic structures,” *Informations Sciences*, vol. 555, pp. 410–423, 2021.
- [77] M. Gaur, K. Gunaratna, S. Bhatt, & A. Sheth, “Knowledge-Infused Learning: A sweet spot in neuro-symbolic AI,” *IEEE Internet Computing*, vol. 26, no. 4, pp. 5–11, 2022.
- [78] R. Li, W. Deng, Y. Cheng, Z. Yuan, J. Zhang, & F. Yuan, “Exploring the upper limits of text-based collaborative filtering using large language models: Discoveries and Insights,” *arXiv PrePrint arXiv:2305.11700*, 2023.
- [79] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, & G. Neubig, ‘Pre-Train, Prompt, and Predict: A systematic survey of prompting methods in natural language processing’, *ACM Computing Surveys*, vol. 55, no. 9, pp. 1-35, 2023.
- [80] B. Yu & S. Zhang, “A novel weight-oriented graph convolutional network for aspect-based sentiment analysis,” *Journal Supercomputing*, vol. 79, no. 1, pp. 947–972, 2023.
- [81] T. Parlar, “Nitelik seçimi kullanarak twitter kullanıcısının cinsiyet sınıflandırması üzerine bir çalışma,” *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi*, vol. 14, no. 3, pp. 204–210, 2022.
- [82] Y. Samet, “Kelime Vektör Temsilleriyle Cümle Üretme,” Yüksek lisans tezi, Bilgisayar Mühendisliği, Ege Üniversitesi, İzmir, Türkiye, 2022.
- [83] C. R. Aydın, “Türkçe için kapsamlı bir duygu analizi çatısı geliştirme,” Doktora tezi, Bilgisayar Mühendisliği, Boğaziçi Üniversitesi, İstanbul, Türkiye, 2020.
- [84] H. Karayığit, “Sosyal medyada Türkçe nefret söylemlerinin ve Covid-19 yorumlarının makine öğrenmesi, derin öğrenme ve bert teknikleri ile analizi,” Doktora tezi, Elektrik ve Elektronik Mühendisliği, Mersin Üniversitesi, Mersin, Türkiye, 2022.
- [85] M. S. Başarslan & F. Kayaalp, “Sentiment analysis on social media reviews datasets with deep learning approach,” *Sakarya University Journal of Computer and Information Sciences*, vol. 4, no. 1, pp. 35-49, 2021.

- [86] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of Research and Development*, vol. 1, no. 4, pp. 309–317, 1957.
- [87] K. S. Jones, "A Statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.
- [88] R. Sjögren, K. Stridh, T. Skotare, & J. Trygg, "Multivariate patent analysis using chemometrics to analyze collections of chemical and pharmaceutical patents," *Journal of Chemometrics*, vol. 34, no. 1, pp.30-41, 2020.
- [89] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, ... & E. Duchesnay, "Scikit-learn: Machine learning in Python," *The Journal of machine Learning research*, vol. 12, no. 1, pp. 2825-2830, 2011.
- [90] M. Tezgider, B. Yildiz, & G. Aydin, "Improving word representation by tuning Word2Vec parameters with deep learning model," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Türkiye, 2018, pp. 1–7.
- [91] A. Onan, "Mining opinions from instructor evaluation reviews: A deep learning approach," *Computer Applications in Engineering Education*, vol. 28, no. 1, 2020, pp. 117–138.
- [92] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, & J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems 26 (NIPS)*, Nevada, USA, 2013, pp. 1–9.
- [93] G. Mesnil, T. Mikolov, M. A. Ranzato, and Y. Bengio, "Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews," *3rd International Conference on Learning Representations, ICLR 2015- Workshop Track Proceedings*, California, USA, 2015, pp. 1-5.
- [94] M. S. Başarslan & F. Kayaalp, "Sentiment Analysis with Ensemble and Machine Learning Methods in Multi-domain and Dataset," *Turkish Journal of Engineering*, vol. 7, no. 2, pp. 141-148, 2023.
- [95] H. ER, "Musicassette interchangeability. the facts behind the facts," *AES: Journal of the Audio Engineering Society*, vol. 19, no. 5, pp. 417–425, 1971.
- [96] T. H. Hidayat, Y. Ruldeviyani, A. R. Aditama, G. R. Madya, A. W. Nugraha, & M. W. Adisaputra, "Sentiment analysis of twitter data related to Rinca Island development using Doc2Vec and SVM and logistic regression as classifier," *Procedia Computer Science*, vol. 197, no.3, pp. 660–667, 2022.
- [97] A. Güran & D. Kınık, "TF-IDF ve Doc2Vec tabanlı türkçe metin sınıflandırma sisteminin başarımlı değerinin ardışık kelime grubu tespiti ile artırılması," *European Journal of Science and Technology*, vol. 21 no. 1, pp. 323–332, 2021.
- [98] Y. Liu, G. Zhang, X. Jin, & H. Yuan, "Multi-info fusion based video recommendation system," *Journal of Physics*, vol. 1229, no. 1, pp. 1-10, 2019.
- [99] Z. Niu, G. Zhong, & H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, no. 5, pp. 48–62, 2021.
- [100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones. A. N. Gomez, L. Kaiser & I. Polosukhiet, "Attention Is All You Need," *31st Conference on Neural Information Processing Systems, Long Beach,, California, USA, 2017*, pp. 30-41.

- [101] B. Cui, Y. Li, M. Chen, & Z. Zhang, “Fine-tune BERT with Sparse Self-Attention Mechanism,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China, 2019, pp. 3546–3551.
- [102] W. Yin, K. Kann, M. Yu, & H. Schütze, “Comparative Study of CNN and RNN for Natural Language Processing,” *arXiv PrePrint*, arxiv:1702.01923. 2017
- [103] X. Fu, J. Yang, J. Li, M. Fang, & H. Wang, “Lexicon-Enhanced LSTM With Attention for General Sentiment Analysis,” *IEEE Access*, vol. 6, no.4, pp. 71884–71891, 2018.
- [104] Q. Ni & X. Cao, “MBGAN: An improved generative adversarial network with multi-head self-attention and bidirectional RNN for time series imputation,” *Engineering Applications of Artificial Intelligence*, vol. 115, no.3, pp. 105-232, 2022.
- [105] W. S. McCulloch & W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull Math Biophys*, vol. 5, no. 4, pp. 115–133, 1943.
- [106] F. Man, C. Yılmaz, & M. N. Kurutkan, “İnsan kaynakları yönetiminde eğitim ve geliştirme alanının bibliyometrik analizi,” *International Journal of Social Sciences and Education Research*, c. 9, s 1, ss. 15–35, 2023.
- [107] Ü. Ağbulut, “Forecasting of transportation-related energy demand and CO2 emissions in Turkey with different machine learning algorithms,” *Sustain Production Consumption*, vol. 29, pp. 141–157, no. 1. 2022.
- [108] S. Ray, “A quick review of machine learning algorithms,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 35–39.
- [109] Ç. Elmas, *Yapay Sinir Ağları Kuram, Mimari, Uygulama*. İstanbul, Türkiye: Seçkin Yayıncılık 2003.
- [110] E. Öztemel, *Yapay Sinir Ağları*. İzmir, Türkiye: Papatya Yayıncılık, 2012.
- [111] M. S. Basarslan & F. Kayaalp, “Sentiment analysis with machine learning methods on social Media,” *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 9, no. 3, pp. 5–15, 2020.
- [112] Ü. Ağbulut, A. E. Gürel, & Y. Biçen, “Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison,” *Renewable and Sustainable Energy Reviews*, vol. 135, no. 1, pp. 110-114, 2022.
- [113] E. Uzun, H. V. Agun, & T. Yerlikaya, “Web content extraction by using decision tree learning,” in *2012 20th Signal Processing and Communications Applications Conference (SIU)*, Mugla, Türkiye, 2012, pp. 1–4.
- [114] M. C. Castro S. Kim, L. Barberia, A. F. Ribeiro, S. Gurzenda, K. B. Ribeiro, ... & B. H. Singer, “Spatiotemporal pattern of Covid -19 spread in Brazil,” *Science*, vol. 372, no. 6544, pp. 821–826, 2021.
- [115] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [116] C. Zhang, Y. Tian, & N. Deng, “The new interpretation of support vector machines on statistical learning theory,” *Science in China Series A Mathematics*, vol. 53, no. 1, pp. 151–164, 2010.

- [117] T. Cover & P. Hart, "Nearest neighbor pattern classification," *IEEE transactions Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [118] S. M. H. Dadgar, M. S. Araghi, & M. M. Farahani, "A novel text mining approach based on TF-IDF and Support Vector Machine for news classification," in *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, Coimbatore, India, 2018, pp. 112–116.
- [119] M. S. Bakay & Ü. Ağbulut, "Electricity production based forecasting of greenhouse gas emissions in Turkey with deep learning, support vector machine and artificial neural network algorithms," *Journal Cleanear Production*, vol. 285, no. 3, pp. 125-244, 2021.
- [120] H. Park, N. Kim, & J. Lee, "Parametric models and non-parametric machine learning models for predicting option prices: empirical comparison study over KOSPI 200 index options," *Expert System with Application*, vol. 41, no. 11, pp. 5227–5237, 2014.
- [121] Y. LeCun, Y. Bengio, & G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [122] Ü. Ağbulut, A. E. Gürel, & Y. Biçen, "Prediction of daily global solar radiation using different machine learning algorithms: Evaluation and comparison," *Renewable and Sustainable Energy Reviews*, vol. 135, no. 1, pp. 110-114, 2021.
- [123] F. Sönmez, *Yapay Sinir Ağ Yapıları*, 1. baskı. İstanbul, Türkiye: Nobel akademik Yayıncılık, 2018.
- [124] Q. Yao, D. D. C. Lu, & G. Lei, "A Surface Temperature Estimation Method for Lithium-ion Battery Using Enhanced GRU-RNN," *IEEE Transactions on Transportation Electrification*, vol. 9, no. 1, 2022, pp. 1103-1102.
- [125] L. Zhang, S. Wang, & B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 4, pp. 1253-1265, 2018.
- [126] A. B. Dieng, J. Gao, C. Wang, & J. Paisley, "TopicRNN: A recurrent neural network with long-range semantic dependency," *5th International Conference on Learning Representations, ICLR 2017- Conference Track Proceedings*, Toulon, Fransa, 2017, pp. 1–13.
- [127] S. Hochreiter & J. Schmidhuber, "Long Short-Term Memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [128] H. Canlı & S. Toklu, "Deep learning-based mobile application design for smart parking," *IEEE Access*, vol. 9, pp. 61171–61183, 2021.
- [129] H. Canlı & S. Toklu, "Design and implementation of a prediction approach using big data and deep learning techniques for parking occupancy," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 1955–1970, 2022.
- [130] Y. Liu, W. Sun, & L. J. Durlofsky, "A deep-learning-based geological parameterization for history matching complex models," *Math Geosciences*, vol. 51, no. 6, pp. 725–766, 2019.
- [131] T. Chen, R. Xu, Y. He, & X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN," *Expert System with Application*, vol. 72, no.1, pp. 221–230, 2017.
- [132] J. Lin & K. C. Wong, "Off-target predictions in CRISPR-Cas9 gene editing using

- deep learning,” *Bioinformatics*, vol. 34, no. 17, pp. 656–663, 2018.
- [133] D. Cireşan, U. Meier, & J. Schmidhuber, “Multi-column deep neural networks for image classification,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Rhode Island, USA. 2012, pp. 3642–3649.
- [134] I. Wallach, M. Dzamba, & A. Heifets, “AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery”, *arXiv PrePrint* arXiv:1510.02855 2015.
- [135] J. Yan, J. Liu, Y. Yu, & H. Xu, “Water Quality Prediction in the Luan River Based on 1-DRCNN and BiGRU Hybrid Neural Network Model,” *Water (Basel)*, vol. 13, no. 9, pp. 12-33, 2021.
- [136] M. S. Islam & N. A. Ghani, “A Novel BiGRUBiLSTM Model for multilevel sentiment analysis using deep neural network with BiGRU-BiLSTM,” *Lecture Notes in Electrical Engineering*, vol. 730, no. pp. 403–414, 2022.
- [137] M. Kumar. V. Tan, & P. Steinbach, *Introduction to Data Mining*. ABD: Addison Wesley, 2005.
- [138] A. Mohammed & R. Kora, “A comprehensive review on ensemble deep learning: Opportunities and challenges,” *Journal of King Saud University- Computer and Information Sciences*, vol. 35, no. 2, pp. 757–774, 2023.
- [139] Z. H. Zhou, *Ensemble methods: foundation and algortihms*. ABD: CRC Press, 2012.
- [140] R. Polikar, “Ensemble based systems in decision making,” *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.
- [141] G. Dogan, I. Cay, S. S. Ertas, Ş. R. Keskin, N. Alotaibi, & E. Sahin, “Where are you?,” in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, New York, USA, 2020, pp. 301–304.
- [142] Google LLC, (2020, 1 Ekim) *Colab* [Online]. Erişim: <https://colab.research.google.com/>
- [143] F. Chollet, *Deep learning with Python*. USA: Simon and Schuster, 2021.
- [144] G. Rossum. (2021, 1 Ekim). *Python* [Online]. Erişim: <https://www.python.org/downloads/>
- [145] E. Tanyildizi & F. Demirtas, “hiper parametre optimizasyonu hyper parameter optimization,” in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, Ankara, Turkey, 2019, pp. 1–5.
- [146] A. Gülcü & Z. Kuş, “Konvolüsyonel sınır ağlarında hiper-parametre optimizasyonu yöntemlerinin incelenmesi,” *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, vol. 7, no. 2, pp. 503–522, 2019.
- [147] R. Khandelwal, (2022,1 Aralık). *Tensorboard Hyperparameter Optimization* [Online]. Erişim: <https://towardsdatascience.com/tensorboard-hyperparameter-optimization-a51ef7af71f5>.
- [148] M. S. Başarslan & F. Kayaalp, “MBi-GRUMCONV: A novel Multi Bi-GRU and Multi CNN-Based deep learning model for social media sentiment analysis,”

Journal of Cloud Computing, vol. 12, no. 1, pp. 1-16, 2023.

- [149] Y. Özkan, *Uygulamalı Derin Öğrenme*. İstanbul, Türkiye: Papatya Yayıncılık, 2021.
- [150] Z. Shaukat, A. A. Zulfiqar, C. Xiao, M. Azeem, & T. Mahmood, “Sentiment analysis on IMDB using lexicon and neural networks,” *SN Application Science*, vol. 2, no. 2, pp. 1-10, 2020.
- [152] K. Team. (2022, 6 Şubat). *Keras Documentation: ReduceLRonPlateau*. [Online]. Erişim: https://keras.io/api/callbacks/reduce_lr_on_plateau/.
- [153] J. Chi, X. Li, H. Wang, D. Gao, & P. Gerstoft, “Sound source ranging using a feed-forward neural network trained with fitting-based early stopping,” *The journal of the acoustical society of America*, vol. 146, no.3, pp. 258-264, 2019.
- [155] B. Pang & L. Lee, “A sentimental education,” *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics- ACL '04*, Morristown, NJ, USA: Association for Computational Linguistics, Barcelona, İspanya, 2004, pp. 271-278.
- [156] S. Rathor & S. Agrawal, “Sense understanding of text conversation using temporal Convolution Neural Network,” *Multimedia Tools Appliicaitons*, vol. 81, no. 7, pp. 9897–9914, 2022.
- [157] A. Yenter & A. Verma, “Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis,” in *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2017, pp. 540–546.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Muhammet Sinan BAŞARSLAN

Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Bilgisayar Müh.	Düzce Üniversitesi	2023
Y. Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2017
Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2015
Lise	Fen Bilimleri	Yalova Çiftlikköy Atatürk Anadolu Lisesi	2009

YAYINLAR

Uluslararası Hakemli Dergilerde Yayınlanan Makaleler

1. M. S. Basarslan, & F. Kayaalp, "Sentiment analysis with machine learning methods on social media", *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 9, no. 3, pp. 5–15, 2020.
2. M. S. Başarslan & F. Kayaalp, "MBi-GRUMCONV: A novel Multi Bi-GRU and Multi CNN-Based deep learning model for social media sentiment analysis," *Journal of Cloud Computing*, vol. 12, no. 5, pp. 1-16, 2023.
3. M. S. Basarslan & F. Kayaalp, "Sentiment analysis using a deep ensemble learning Model", *Multimedia Tools and Applications*, Değerlendirmede.

Ulusal Hakemli Dergilerde Yayınlanan Makaleler

1. M. S. Basarslan & F. Kayaalp, "Sentiment analysis on social media reviews datasets with deep learning approach. *Sakarya University Journal of Computer and Information Sciences*, vol. 4, no. 1, pp. 35-49, 2021.

2. M. S. Bařarslan ve F. Kayaalp, "Sentiment analysis with ensemble and machine learning methods in multi-domain datasets", *Turkish Journal of Engineering*, c. 7, sayı. 2, ss. 141-148, 2023.
3. M. S. Bařarslan & F. Kayaalp, "Sentiment analysis of coronavirus data with ensemble and machine learning methods", *Eskiřehir Technical University Journal of Science and Technology A- Applied Sciences and Engineering*, Deęerlendirmede.

Uluslararası Bilimsel Toplantılarda Sunulan ve Bildiri Kitaplarında Basılan Bildiriler:

1. M. S. Basarslan & F. Kayaalp, "Sentiment analysis with various deep learning models on movie reviews," *2022 International Conference on Artificial Intelligence of Things (ICAIoT)*, İstanbul, Türkiye, 2022, ss. 1-5.

