

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BİLİM DALI

YAPAY ZEKÂ İLE ÜRETİLMİŞ TÜRKÇE BİR METNİ
TESPİT ETMEK

DOKTORA TEZİ

Rezzan Nisa ER

İstanbul
Ekim - 2025

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BİLİM DALI

**YAPAY ZEKÂ İLE ÜRETİLMİŞ TÜRKÇE BİR METNİ TESPİT
ETMEK**

DOKTORA TEZİ

Rezzan Nisa ER

Tez Danışmanı
Dr. Öğr. Üyesi Kevser Nur ÇOĞALMIŞ

İstanbul
Ekim - 2025

Lisansüstü Eğitim Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman Dr. Öğr. Üyesi Kevser Nur ÇOĞALMIŞ

Üye Dr. Öğr. Üyesi Şengül BAYRAK

Üye Dr. Öğr. Üyesi Mohammed SALEM

Üye Dr. Öğr. Üyesi Sümeyra BEDİR

Üye Dr. Öğr. Üyesi Muhammed DAVUD

Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. Erhan İÇENER

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Doktora tezi olarak hazırladığım “**YAPAY ZEKÂ İLE ÜRETİLMİŞ TÜRKÇE BİR METNİ TESPİT ETMEK**” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Rezzan Nisa ER

ÖN SÖZ

Bu tezin ortaya çıkmasında her zaman yanımda olan, destekleyen, bilgisiyle yolumu aydınlan sevgili danışmanım Dr. Öğr. Üyesi Kevser Nur Çoğalmış' a gönülden teşekkür ederim. Bu süreçte bana ilham veren ve yol gösteren gerçek bir rehber oldu. Katkıları ve içten yönlendirmeleri için kendisine ne kadar teşekkür etsem azdır.

Ayrıca tez sürecinde yapıcı değerlendirmeleri ve desteklerinden dolayı değerli tez izleme jüri üyelerimiz Dr. Öğr. Üyesi Şengül Bayrak ve Dr. Öğr. Üyesi Mohammed Salem' e de teşekkürlerimi borç bilirim.

Bu süreçte birlikte çalıştığım, motivasyonumu yüksek tutmamdaki katkılarını unutmayacağım, her zaman yanımda olan çalışma arkadaşlarıma hem arkadaşlıkları hem de destekleri için çok teşekkür ederim.

En büyük teşekkürü beni bugünlere getiren, her zaman yanımda olan, sevgileri ve fedakârlıklarıyla bana güç veren, beni sabırla, sevgiyle ve inançla yetiştiren canım annem ve babam Leyla Er ve Zafer Cengiz Er'e ediyorum. Onların desteği, bu yolculuğun en kıymetli dayanağı oldu. Ayrıca kardeşlerime de destekleri ve her zaman yanımda oldukları için teşekkür ederim.

Eğitim hayatım boyunca sağladığı imkânlar, akademik ortamı ve değerli hocalarıyla gelişimime katkıda bulunan İstanbul Sabahattin Zaim Üniversitesi'ne de teşekkürlerimi sunarım.

İstanbul, Ekim, 2025

Rezzan Nisa ER

ÖZET

YAPAY ZEKÂ İLE ÜRETİLMİŞ TÜRKÇE BİR METNİ TESPİT ETMEK

Rezzan Nisa ER

Doktora, Bilgisayar Bilimleri ve Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Kevser Nur ÇOĞALMIŞ

Ekim, 2025 - 99 + XV Sayfa

Dil modeli teknolojilerinin hızla gelişmesiyle birlikte, yapay zekâ tarafından üretilen metinleri insan yazımı metinlerden ayırt etmek giderek daha zor bir yetenek haline gelmektedir. Bu tez çalışması, ChatGPT-4 tarafından oluşturulan akademik makale özetleri ile insan makale özetlerini tespit ederek sınıflandırmayı amaçlamaktadır. Veri seti oluşturulurken, DergiPark Akademik platformunda yayımlanmış sağlık, hukuk, sosyoloji ve mühendislik alanlarının her birinden 250'şer adet makale seçilmiştir. Veri seti içerisinde toplam 2000 adet olan özetlerin 1000 tanesi Dergipark'tan seçilen bahsi geçen 1000 adet insan yazımı özet iken, kalan 1000 tanesi ise her makalenin başlığı ve anahtar kelimeleri ChatGPT-4 uygulamasına verilerek oluşturulan yapay özetlerdir. Metin sınıflandırması için LSTM, CNN, GRU, GPT-2 ve BERTurk gibi oldukça popüler olan derin öğrenme modelleri kullanılmıştır. Modellerin başarıları doğruluk, f1-skoru gibi sınıflandırma metrikleriyle; kaynak kullanımı ise eğitim süresi ve donanım tüketimi metrikleri ile değerlendirilmiştir. Hem doğruluk hem de modellerin eğitim süresi birlikte değerlendirildiğinde en başarılı ve etkili olan model LSTM olmuştur. Bu çalışma, yapay zekâ tarafından üretilen akademik metinlerin tespiti için farklı derin öğrenme modellerinin farklı veri setleri üzerindeki performans ölçümleri ve karşılaştırmalarının yanı sıra insan-yapay zekâ sınıflandırması için etkili bir yaklaşım sunmaktadır.

Anahtar Kelimeler: Yapay Zekâ, Makine Öğrenmesi, Doğal Dil İşleme, Derin Öğrenme

ABSTRACT

DETECTING A TURKISH TEXT PRODUCED WITH ARTIFICIAL INTELLIGENCE

Rezzan Nisa ER

PhD, Computer Science and Engineering

Supervisor: Asst. Prof. Dr. Kevser Nur OĐALMIŐ

October, 2025 - 99 + XV Pages

With the rapid development of language modeling technologies, distinguishing AI-generated text from human-written text is becoming increasingly challenging. This thesis aims to identify and classify academic article abstracts generated by ChatGPT-4 and human-written abstracts. To create the dataset, 250 articles from each of the fields of health, law, sociology, and engineering published on the DergiPark Academic platform were selected. Of the 2000 abstracts in the dataset, 1000 were human-written abstracts selected from DergiPark, while the remaining 1000 were artificial abstracts generated by feeding the title and keywords of each article to ChatGPT-4. Popular deep learning models such as LSTM, CNN, GRU, GPT-2, and BERTurk were used for text classification. Model performance was evaluated using classification metrics such as f1-score and accuracy, while resource utilization was evaluated using training time and hardware consumption. When both accuracy and training time of the models are evaluated together, the most successful and effective model is the LSTM. This study provides performance measurements and comparisons of different deep learning models on different datasets for the detection of academic texts generated by AI, as well as an effective approach for human-AI classification.

Keywords: Artificial Intelligence, Machine Learning, Natural Language Processing, Deep Learning,

İÇİNDEKİLER

TEZ ONAYI	i
BİLİMSEL ETİK BİLDİRİMİ	ii
ÖN SÖZ	iii
ÖZET	iii
ABSTRACT	vi
İÇİNDEKİLER	vi
TABLO LİSTESİ	viii
ŞEKİL LİSTESİ	x
KISALTMALAR	xiv
SEMBOLLER	xv
BİRİNCİ BÖLÜM	
GİRİŞ	1
1.1. Kapsam.....	2
1.2. Katkı.....	3
1.3. Doğal Dil İşleme (NLP- Natural Language Processing).....	4
İKİNCİ BÖLÜM	
LİTERATÜR İNCELEMESİ	6
ÜÇÜNCÜ BÖLÜM	
KULLANILAN YÖNTEMLER	10
3.1. Veri Seti	10
3.1.1. Metin Ön İşleme (Pre-processing).....	15
3.2. Metin Temsili (Word Embedding).....	17
3.2.1. Word2Vec	17
3.2.2. FastText	18

3.2.3. Bidirectional Encoder Representations from Transformers (BERT)	19
3.2.4. TF-IDF (Term Frequency-Inverse Document Frequency, Terim Frekansı- Ters Belge Frekansı).....	21
3.3. Sınıflandırma Yöntemleri.....	22
3.3.1. Makine Öğrenmesi Algoritmaları.....	22
3.3.2. Derin Öğrenme Modelleri.....	28
3.3.3. Dönüştürücü (Transformer)Tabanlı Modeller	31

DÖRDÜNCÜ BÖLÜM

DENEYSEL ÇALIŞMALAR..... 34

4.1. Kelime Gömme Yöntemlerinin Belirlenmesi (Embedding)	34
4.2. Modeller ile Sınıflandırma	38
4.2.1. CNN Modeli	38
4.2.2. LSTM Modeli	48
4.2.3. GRU Modeli	55
4.2.4. GPT-2 Modeli	64
4.2.5. BERTurk Modeli	73
4.3. Modellerin Performanslarının Karşılaştırılması.....	80
4.4. Bert + (CNN, LSTM , GRU) karşılaştırması	81
4.5. Caude ai ile modeli test	82

SONUÇ..... 85

KAYNAKÇA

ÖZGEÇMİŞ..... 98

TABLO LİSTESİ

Tablo 4.1: Colab ortamında çalıştırılan CNN modelinin kaynak kullanımı	389
Tablo 4.2: epoch = 5 değeri için CNN modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	41
Tablo 4.3: epoch =10 için CNN modelinin doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	43
Tablo 4.4: epoch = 20 değeri için CNN modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	45
Tablo 4.5: epoch = 16 değeri için CNN modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	47
Tablo 4.6: Colab ortamında çalıştırılan LSTM modelinin kaynak kullanımı	48
Tablo 4.7: epoch = 5 değeri için LSTM modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	50
Tablo 4.8: epoch = 10 değeri için LSTM modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	52
Tablo 4.9: epoch = 20 değeri için LSTM modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	54
Tablo 4.10: Colab ortamında çalıştırılan GRU modelinin kaynak kullanımı	55
Tablo 4.11: epoch= 5 için GRU modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	57
Tablo 4.12: GRU epoch= 10 için precision, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	59
Tablo 4.13: epoch = 12 değeri için GRU modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	61
Tablo 4.14: epoch = 20 değeri için GRU modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	62
Tablo 4.15: Colab ortamında çalıştırılan GPT - 2 modelinin kaynak kullanımı.....	64
Tablo 4.16: epoch = 5 değeri için GPT-2 modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri.....	66
Tablo 4.17: epoch =10 için GPT-2 modelinin modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri	68
Tablo 4.18: epoch =14 için GPT-2 modelinin modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri	70

Tablo 4.19: epoch = 20 için GPT-2 modelinin modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri	72
Tablo 4.20: Colab ortamında çalıştırılan BERTurk modelinin kaynak kullanımı	73
Tablo 4.21: epoch = 5 değeri için BERTurk modelinin doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri	75
Tablo 4.22: BERTurk modelinde epoch =10 değeri için karmaşıklık matrisi	77
Tablo 4.23: BERTurk modelinde epoch =20 değeri için presicion duyarlılık, f1-skoru ve örnek sayısı değerleri.....	79
Tablo 5.1: Modellerin early stop epoch değerleriyle çalıştırıldığındaki doğruluk, f1-skoru ve eğitim süresi.....	86
Tablo 5.2: Literatürdeki benzer çalışmalarla yapılan karşılaştırma tablosu.....	88



ŞEKİL LİSTESİ

Şekil 3.1: v0 veri seti; özetler cümlelere bölünerek tutulmuş yapay özetler Chatgpt 3.5 tarafından oluşturulmuştur.	11
Şekil 3.2: v1 veri setinden örnek bir satır görülmektedir, aynı konulu makalenin insan ve ai (chatgpt-4) tarafından yazılmış içeriği.	12
Şekil 3.3: v2 veri setinden örnek bir satır görülmektedir, aynı konulu makalenin insan, Chatgpt ve claud.ai tarafından yazılmış içeriği. Toplamda 300 satırlık veri setinden 3 satırı örnek olarak gösterilmiştir.	14
Şekil 3.4: Word2Vec mimarisinde kullanılan CBOW ve Skip-gram modelleri. CBOW, bağlam kelimelerden hedef kelimeyi tahmin ederken; Skip-gram, hedef kelimedenden bağlam kelimeleri tahmin etmeye çalışır (Sangole, 2024).	18
Şekil 3.5: BERT modelinin pre-training (öğretim) sürecini özetlemektedir (Nvidia, 2019)	21
Şekil 3.6: $k=5$ iken 3-sınıflı bir problem için Knn gösterimi (Raschka, 2018)	213
Şekil 3.7: Doğrusal ayrılabilme durumu (IBM, t.y.)	24
Şekil 3.8: Doğrusal olarak ayrılamama durumu	25
Şekil 3.9: 2 Sınıf için lojistik regresyonun temel bir diyagramı görülmektedir. (Hasib vd., 2023)	26
Şekil 4.1: Her bir makine öğrenmesi algoritması sırasıyla tüm embedding yöntemleriyle birlikte çalıştırılmıştır	34
Şekil 4.2: Gömme yöntemlerinin makine öğrenmesi algoritmalarıyla birlikte çalıştırıldıklarında elde edilen doğruluk değerleri	35
Şekil 4.3: Gömme yöntemlerinin makine öğrenmesi algoritmalarıyla beraber çalıştırıldığında geçen sürelerin karşılaştırılması.....	36
Şekil 4.4: Gömme yöntemlerinin makine öğrenmesi modelleri ile çalıştırıldıklarında elde edilen f1-skoru değerleri	37
Şekil 4.5.a: epoch = 5 için CNN modelinin accuracy (doğruluk) değeri grafiği	40
Şekil 4.5.b: epoch = 5 için CNN modelinin kayıp (loss) değeri grafiği.....	40
Şekil 4.6: Karmaşıklık matrisi epoch = 5 için CNN modelin in.....	41
Şekil 4.7.a: epoch = 10 için CNN modelinin accuracy (doğruluk) değeri grafiği ...	42

Şekil 4.7.b: epoch = 10 için CNN modelinin kayıp (loss) değeri grafiği.....	42
Şekil 4.8: epoch =10 için CNN modelinin Karmaşıklık matrisi.....	43
Şekil 4.9.a: epoch =16 için CNN modelinin accuracy (doğruluk) grafiği	44
Şekil 4.9.b: - epoch =16 için CNN modelinin kayıp (loss) gafiği.....	44
Şekil 4.10: epoch =16 için CNN modelinin Karmaşıklık matrisi.....	45
Şekil 4.11.a: epoch =20 için CNN modelinin accuracy (doğruluk) grafiği	46
Şekil 4.11.b: epoch =20 için CNN modelinin kayıp (loss) grafiği.....	46
Şekil 4.12: epoch =20 için CNN modelinin Karmaşıklık matrisi.....	47
Şekil 4.13.a: epoch =5 için LSTM modelinin accuracy (doğruluk) değeri grafiği ...	49
Şekil 4.13.b: epoch =5 için LSTM modelinin kayıp (loss) değeri grafiği.....	49
Şekil 4.14 epoch =5 için LSTM modelinin Karmaşıklık matrisi.....	50
Şekil 4.15.a: epoch = 10 için LSTM modelinin accuracy (doğruluk) değeri grafiği	51
Şekil 4.15.b: epoch =10 için LSTM modelinin kayıp (loss) değeri grafiği.....	51
Şekil 4.16: epoch = 10 için LSTM modelinin Karmaşıklık matrisi	52
Şekil 4.17.a: epoch= 20 için LSTM modelinin accuracy (doğruluk) değeri grafiği .	53
Şekil 4.17.b: epoch= 20 için LSTM modelinin kayıp (loss) değeri grafiği.....	53
Şekil 4.18: epoch= 20 için LSTM modelinin Karmaşıklık matrisi	54
Şekil 4.19.a: epoch= 5 için GRU modelinin accuracy (doğruluk) değeri grafiği.....	56
Şekil 4.19.b: epoch= 5 için GRU modelinin kayıp (loss) değeri grafiği	56
Şekil 4.20: epoch= 5 için GRU modelinin Karmaşıklık matrisi.....	57
Şekil 4.21.a: epoch= 10 için GRU modelinin accuracy (doğruluk) değeri grafiği ..	58
Şekil 4.21.b: epoch= 10 için GRU modelinin kayıp (loss) değeri grafiği.....	58
Şekil 4.22: epoch= 10 için GRU modelinin Karmaşıklık matrisi.....	59
Şekil 4.23.a: epoch= 12 için GRU modelinin accuracy (doğruluk) değeri	60
Şekil 4.23.b: epoch= 12 için GRU modelinin kayıp (loss) değeri	60

Şekil 4.24: epoch= 12 için GRU modelinin Karmaşıklık matrisi.....	61
Şekil 4.25.a: epoch= 20 için GRU modelinin accuracy (doğruluk) değeri grafiği ...	62
Şekil 4.25.b: epoch= 20 için GRU modelinin kayıp (loss) değeri	62
Şekil 4.26: epoch= 20 için modelinin Karmaşıklık matrisi	63
Şekil 4.27.a: GPT-2 modelinin epoch =5 için accuracy (doğruluk) değeri grafiği...	65
Şekil 4.27.b: GPT - 2 modelinin epoch =5 için kayıp (loss) değeri grafiği	65
Şekil 4.28: GPT - 2 modelinin epoch =5 için karmaşıklık değeri	66
Şekil 4.29.a: GPT-2 modelinin epoch =10 için accuracy (doğruluk) değeri grafiği .	67
Şekil 4.29.b: GPT-2 modelinin epoch =10 için kayıp (loss) değeri grafiği	67
Şekil 4.30: GPT-2 modelinin epoch =10 için Karmaşıklık matrisi	68
Şekil 4.31.a: GPT-2 modelinin epoch =14 için accuracy (doğruluk)değerleri	69
Şekil 4.31.b: GPT-2 modelinin epoch =14 için kayıp (loss) değerleri.....	69
Şekil 4.32: GPT-2 modelinin epoch =14 için karmaşıklık matrisi	70
Şekil 4.33.a: GPT-2 modelinin epoch =20 için doğruluk değeri grafiği.....	71
Şekil 4.33.b: GPT-2 modelinin epoch =20 için loss değeri grafiği.....	71
Şekil 4.34: GPT-2 modelinin epoch =20 için Karmaşıklık matrisi	72
Şekil 4.35.a: epoch =5 için BERTurk modelinin doğruluk grafiği	74
Şekil 4.35.b: epoch =5 için BERTurk modelinin kayıp grafiği.....	74
Şekil 4. 36: epoch =5 için BERTurk modelinin Karmaşıklık matrisi	75
Şekil 4.37.a: epoch =10 için BERTurk modelinin doğruluk değeri grafiği	76
Şekil 4.37.b: epoch =10 için BERTurk modelinin kayıp değeri grafiği.....	76
Şekil 4.38: BERTurk modelinde epoch =10 değeri için Karmaşıklık matrisi.....	77
Şekil 4.39.a: BERTurk modelinde epoch =20 değeri için doğruluk değeri grafiği...	78
Şekil 4.39.b: BERTurk modelinde epoch =20 değeri için kayıp değeri grafiği.....	78
Şekil 4.40: BERTurk modelinde epoch =20 değeri için karmaşıklık matrisi.....	79

Şekil 4.41: Modellerin doğruluk değerlerinin karşılaştırılması.....	80
Şekil 4.42: Colab ortamında çalışan modellerin kaynak tüketimi	80
Şekil 4.43: BERT + CNN birlikte çalışan modelin doğruluk ve kayıp değerleri	81
Şekil 4.44: BERT + LSTM birlikte çalışan modelin doğruluk ve kayıp değerleri...	81
Şekil 4.45: BERT + GRU birlikte çalışan modelin doğruluk ve kayıp değerleri	82
Şekil 4.46: 100 adet claude.ai ve 100 adet insan tarafından oluşturulan v2 veri seti ile sınıflandırma sonucunda doğruluk ve kayıp değerleri	83
Şekil 4.47: 100 adet chatgpt-4 ve 100 adet insan tarafından oluşturulan v2 veri seti ile sınıflandırma sonucunda doğruluk ve kayıp değerleri	83
Şekil 4.48: 100 adet claude.ai ve aynı başlık ve anahtar kelimeli 100 adet insan etiketli verilerin LSTM modeli ile sınıflandırma sonucu ortaya çıkan karmaşıklık matrisi ..	84
Şekil 4.49: 100 adet chatgpt-4 ve aynı başlık ve anahtar kelimeli 100 adet insan etiketli verilerin LSTM modeli ile sınıflandırma sonucu ortaya çıkan karmaşıklık matrisi ..	84

KISALTMALAR

AI	: Artificial Intelligence (Yapay Zekâ)
BERT	: Bidirectional Encoder Representations from Transformers (Transformerlardan Çift Yönlü Kodlayıcı Gösterimleri)
CBOW	: Continuous Bag of Words (Sürekli Kelime Torbası)
CNN	: Convolutional Neural Networks (Evrışimli Sinir Ağları)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
GELU	: Gaussian Error Linear Unit (Gauss Hatalı Doğrusal Birim)
GLTR	: Giant Language model Test Room (Büyük Dil Modeli Test Odası)
GPU	: Graphics Processing Unit (Grafik İşlem Birimi)
GRU	: Gated Recurrent Unit (Kapılı Tekrarlayan Birim)
GPT	: Generative Pre-trained Transformer (Önceden Eğitilmiş Üretici Transformer)
KNN	: K-Nearest Neighbors (K-En Yakın Komşu)
LLM	: Large Language Model (Büyük Dil Modeli)
LR	: Logistic Regression (Lojistik Regresyon)
LSTM	: Long Short-Term Memory (Uzun Kısa Süreli Bellek)
ML	: Machine Learning (Makine Öğrenimi)
NLG	: Natural Language Generation (Doğal Dil Üretimi)
NLU	: Natural Language Understanding (Doğal Dil Anlama)
NLP	: Natural Language Processing (Doğal Dil İşleme)
RELU	: Rectified Linear Unit (Düzeltilmiş Doğrusal Birim)
RAM	: Random Access Memory (Rastgele Erişimli Bellek)
RF	: Random Forest (Rastgele Orman)
RNN	: Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
SHAP	: SHapley Additive exPlanations (Shapley Katkısal Açıklamaları)
SVM	: Support Vector Machine (Destek Vektör Makineleri)
TF-IDF	: Term Frequency–Inverse Document Frequency (Terim Frekansı– Ters Belge Frekansı)

SEMBOLLER

b	: Bias (sabit) terimi
d	: Belge (document)
$df(t)$: t terimini içeren belge sayısı
e	: Euler sayısı (≈ 2.718), üstel fonksiyon tabanı
$f(t, d)$: d belgesinde t teriminin geçiş (frekans) sayısı
$IDF(t, d)$: Ters belge frekansı – bir terimin tüm belgelerdeki nadirlik ölçüsü
N	: Koleksiyondaki toplam belge sayısı
$P(A)$: A olayının olasılığı
$P(A B)$: B olayı gerçekleştiğinde A olayının koşullu olasılığı
$P(B)$: B olayının olasılığı
$P(B A)$: A olayı gerçekleştiğinde B olayının koşullu olasılığı
$\sigma(z)$: Sigmoid aktivasyon fonksiyonu
$\sum f(w, d)$: d belgesinde bulunan tüm kelimelerin toplam frekans sayısı
t	: İncelenen terim (kelime)
$TF(t, d)$: Terim frekansı – bir terimin belirli bir belgede göreceli sıklığı
$TF - IDF(t, d)$: Bir terimin belirli belgede önemini ölçen birleşik ağırlık değeri
w	: d belgesinde bulunan herhangi bir terim (kelime)
w_i	: i . özelliğe (değişkene) ait ağırlık katsayısı
x_i	: i . özellik (girdi değişkeni)
z	: Doğrusal kombinasyon sonucu elde edilen giriş değeri

BİRİNCİ BÖLÜM

GİRİŞ

Büyük Dil Modelleri (LLM- Large Language Models), intihal, e-ticaret platformlarına sahte incelemeler yerleştirmek veya seçim sonuçlarını etkileyebilecek sahte sosyal medya gönderileri oluşturmak gibi çeşitli suistimal riski taşıyan metinler, gerçekliği olmayan ve tereddütsüz okuyucular tarafından doğru kabul edilen metinler üretebilme kabiliyetindedir. Bir metnin makine tarafından üretilip üretilmediğini tespit etmek bu nedenle giderek daha önemli hale gelmiştir (Yu vd., 2023). Dil modeli teknolojilerinin metin oluşturma yetenekleri sürekli olarak gelişirken, makine tarafından oluşturulan metinler ile insanlar tarafından yazılan metinler arasındaki ayrımı fark etmek, giderek daha da zor hale gelmektedir. Bu zorluk beraberinde alanındaki tespit yöntemlerinin yeni tekniklerle genişleyip değişimini getirmektedir. Ancak, her konuşma dilinin farklı morfolojik yapısı ve anlamsal kuralları, bir dil modeli tarafından üretilen metinlerin hızla algılanması ve doğrulanması konusunda önemli zorluklar ortaya çıkarmaktadır. Bu alandaki ilerlemeler, yenilikçi çözümlerin geliştirilmesini gerektirirken, büyük dil modelleri özellikle eğitim, gazetecilik ve sanat gibi çeşitli alanlarda insan yazarları giderek daha fazla geri planda bırakmaya başlayacak araçlar haline gelmektedir. Fakat, belirli içeriklerin özellikle de gerçekliğin hayati önem içerdiği durumlarda, oluşturulan içeriklerin doğrulanmış ve insan kaynaklı olması gerektiği göz önünde bulundurulmalıdır. Bu nedenle, dil modeli teknolojilerinin metin üretimlerinde meşru kullanımlarını doğrulamak için araçların ve mekanizmaların geliştirilmesi ihtiyacı ortaya çıkmıştır.

Yapay Zekâ (AI) teknolojisinin yükselişi ve bunun eğitim üzerindeki etkisi, son yıllarda artan bir endişe konusu olmuştur. Chatbot gibi yeni nesil yapay zekâ sistemleri internet üzerinden daha erişilebilir ve yetenek açısından daha güçlü hale gelmiştir. Okullarda ödev, proje veya akademik makaleler oluşturmak için chatbotların, özellikle de ChatGPT'nin kullanılması, üretilen ve yayınlanan bilginin doğruluğu, yazar tarafından ortaya konulup konulmadığı konusunda soru işaretlerine ve kaygılara sebep olmuştur (Khalil ve Er, 2023). ChatGPT güçlü metin üreten bir diyalog sistemidir. Doğal dil işleme (NLP) alanında geliştirilen bu model, kullanıcıların verdiği metinlere insan benzeri yanıtlar üretme yeteneğine sahiptir. Modelin temelini, Üretken Önceden Eğitilmiş Dönüştürücüler (Generative Pre-Trained Transformers) olarak bilinen bir

yapay zeka mimarisi oluşturur. Geniş çapta internetten toplanan konuşma verileriyle eğitilen sistem, metnin devamını tamamlama, çeviri ve soru cevaplama gibi çok çeşitli işleri başarıyla gerçekleştirebilir. Bu model, Transformer adı verilen sinir ağı mimarisi üzerinde çalışır ve çeşitli kaynaklardan (örneğin kitaplar, makaleler, web siteleri) elde edilen metinlerle dilin yapısını öğrenir. Model, önceki kelimelere bakarak bir sonraki kelimeyi tahmin eder; böylece verilen bağlama uygun yeni metinler oluşturur. Bu üretim süreci, anlamlı bir cümle ya da istenen uzunlukta bir metin ortaya çıkana kadar devam eder (An, Ding ve Lin, 2023).

1.1. Kapsam

Doğal dilin anlaşılmasına yönelik önceden eğitilmiş dil modellerinin başarısı, benzeri görülmemiş derecede büyük dil modellerinin eğitilmesine yol açmıştır. Bu büyük dil modelleri, uzman olmayan bir okuyucunun insan tarafından yazılan metinden ayırt edilemeyecek metinsel çıktılar üretme potansiyeline sahiptir. Yapay zekâ dil modellerinin kökleri yapay zekânın ilk günlerine kadar uzanmaktadır. Eliza dil modeli ve yapay zekâ dil modelinin en eski örneklerinden biri olup, 1966'da Massachusetts Institute of Technology tarafından tanıtılmıştır. Tüm dil modellerinde ilk olarak veri üzerinde eğitimi yapılır. Daha sonra ilişkileri anlamak için çeşitli tekniklerden yararlanır ve sonrasında eğitilen verilere dayalı olarak yeni içerik üretilir. Dil modelleri, kullanıcının bir sonuç oluşturmak için doğal dilde bir sorgu girdiği doğal dil işleme (NLP) uygulamalarında yaygın olarak kullanılır. Büyük Dil Modelleri yani yaygın bilinen adıyla LLM'ler, eğitim ve çıkarım için kullanılan verileri önemli ölçüde genişletmekte, yapay zekâdaki dil modeli konseptinde evrim yaratmış olmasının yanı sıra yapay zekâ modelinin yeteneklerinde büyük bir artış sağlamaktadır. Her ne kadar eğitimde kullanılacak veri setinin büyüklüğü için net ve evrensel bir standart bulunmasa da, büyük dil modelleri (LLM'ler) genellikle en az bir milyar parametreyi veya daha fazlasını içerecek şekilde yapılandırılır. Parametreler, üzerinde eğitim yapılan modelde mevcut olan ve yeni içerik çıkarmak için başvuru değişkenler için kullanılan bir makine öğrenimi terimidir ("Large Language Models", t.y.). LLM'lerdeki son gelişmeler (ChatGPT gibi) teknolojinin bireyler ve toplum üzerindeki potansiyel etkilerini artırmıştır. Örneğin; Wharton Business School'daki MBA sınavları gibi zorlu testlerdeki başarılı performansıyla kanıtlanan ChatGPT'nin yetenekleri, çeşitli disiplinlerde profesyonel yardım sağlama potansiyelini ortaya koymaktadır. Özellikle sağlık alanında, ChatGPT uygulamaları verimlilikteki basit

iyileştirmelerin çok ötesine uzanmaktadır. ChatGPT, yalnızca tıbbi belgelerin hazırlanmasını kolaylaştırarak (örneğin; hasta kayıtları, klinik ilerleme notları ve taburcu raporları gibi) sağlık alanındaki belge süreçlerini daha verimli hale getirmekle kalmaz; aynı zamanda hasta bilgilerini toplama ve bu verileri analiz etme sürecinde de sağlık uzmanlarına destek olur. Bu sayede, hasta bakımıyla ilgili karar verme süreci daha bilinçli şekilde yürütülebilir (Tang, Chuang ve Hu, 2023). OpenAI tarafından geliştirilen bu yapay zekâ tabanlı sohbet platformu, insan-teknoloji etkileşiminde dönüşüm yaratma potansiyeline sahiptir. ChatGPT, derin öğrenme ve NLP yöntemlerine başvurusuyla makinelerle insanların doğal bir diyalog kurmasına olanak tanır. ChatGPT, 2022 yılında geniş kitlelerce tanıtılmış ve özellikle 2023 yılında popülerlik kazanmıştır. ChatGPT, bağlamı, amacı ve duyguyu anlayabilme yeteneği sayesinde, herhangi bir programlama bilgisine ya da bilgisayar bilimi geçmişine sahip olmayan kullanıcıların bile farklı dillerde doğal bir biçimde iletişim kurmasına olanak tanır. Eğitim, müşteri destek hizmetleri, sağlık, finans, eğlence gibi birçok sektörde kullanılacak esnekliğe sahiptir. Halihazırda piyasada yer alan geleneksel sohbet botları ve sesli asistanlara kıyasla önemli iyileştirmeler sunar. Özellikle, sohbetlerin bağlamını kavrama yetisi sayesinde, diğer çözümlere göre daha isabetli yanıtlar verebilir. Bu yönüyle, ChatGPT ile çalışan sistemlerde hata payı azalırken, kullanıcı deneyimi de kayda değer şekilde artar. Bu teknoloji farklı sektörlerde faydalı bir şekilde kullanılabilir. Eğitim alanında, her öğrencinin ihtiyacına uygun şekilde kişiselleştirilmiş öğretim desteği sunan yapay zekâ destekli ders hizmetleri kullanılabilir. Eğlence ve sağlık sektörlerinde ise kullanıcı tercihlerini anlayarak deneyimleri daha etkileşimli hale getiren sanal asistanlar ile tıbbi verilerdeki kalıpları insanlardan daha hızlı tespit eden tahmine dayalı analitik araçları, daha akıllı hasta bakımı çözümleri sağlayabilir. Finans alanında ise müşterilere, bireysel risk profillerine dayalı olarak yatırım kararları konusunda otomatik tavsiyeler sunulabilme ChatGPT' nin farklı endüstriler ve sektörlerde faydalı bir şekilde kullanılabilceğinin örnekleridir (George ve George, 2023).

1.2. Katkı

Bu çalışma, yapay zekâ tarafından üretilen akademik metinlerin tespitine odaklanan ve Türkçe dilinde gerçekleştirilen az sayıdaki araştırmadan biri olma özelliğini taşımaktadır. Mevcut literatürde, İngilizce metinler üzerinde insan - yapay zekâ ayrımını belirlemeye yönelik çeşitli yöntemler geliştirilmiş olmakla birlikte, Türkçe

metinler için hazırlanmış kapsamlı ve dengeli bir veri setine rastlanmamıştır. Bu açıdan, çalışmanın en temel katkısı; Türkçe akademik makale özetlerinden oluşan, hem insan eliyle yazılmış hem de ChatGPT-4 tarafından üretilmiş metinleri dengeli biçimde içeren, 2000 örnekten oluşan özgün bir veri seti sunmasıdır.

İkinci olarak, çalışma; LSTM, CNN, GRU, GPT-2 ve BERTurk gibi farklı derin öğrenme mimarilerinin Türkçe metin sınıflandırma görevlerindeki performanslarını karşılaştırmalı olarak analiz etmektedir. Bu kapsamda, modellerin doğruluk oranı, f1-skoru, eğitim süresi ve donanım gereksinimleri gibi çok boyutlu ölçütler dikkate alınarak ayrıntılı bir değerlendirme yapılmıştır.

Üçüncü önemli bulgu ise, LSTM mimarisinin Türkçe akademik metinlerde yapay zekâ üretimini tespit etmede hem yüksek doğruluk hem de verimlilik açısından öne çıkmasıdır. Bu sonuç, özellikle sınırlı kaynaklara sahip uygulama ortamlarında, etkili ve hafif bir çözüm önerisi olarak değerlendirilebilir.

Son olarak, çalışma yalnızca Türkçe doğal dil işleme (NLP) alanında yapay zekâ-insan ayrımı konusuna katkı sağlamakla kalmamakta, aynı zamanda gelecekte benzer araştırmalarda kullanılacak ölçeklenebilir bir metodoloji de ortaya koymaktadır. Bu metodoloji, farklı dil ve disiplinlere uyarlanabilir nitelikte olup, akademik bütünlük ve içerik güvenilirliği konularında önemli bir başvuru kaynağı olma potansiyeline sahiptir.

1.3. Doğal Dil İşleme (NLP- Natural Language Processing)

Doğal dil, insanların çevrelerinden öğrendikleri ve birbirleriyle iletişim kurmak için kullandıkları herhangi bir dildir. Doğal Dil İşleme (NLP) ise bilgisayarlar ve doğal dil arasındaki etkileşimlerle ilgilenen bilgisayar bilimi, yapay zekâ ve dil biliminin bir dalıdır. İletişimin şekli ne olursa olsun, bilgimizi ve duygularımızı ifade etmek, tepkilerimizi diğer insanlara ve çevremize iletmek için doğal dilleri kullanırız (Reshamwala, Mishra ve Pawar, 2013).

Dijitalleşme, araştırmacılara çağdaş olguları araştırmak için zengin metinsel veriler toplama konusunda benzeri görülmemiş fırsatlar sağlamaktadır. İnsan dilini anlamayı ve analiz etmeyi hedefleyen bilgisayar destekli bir analitik teknik olan NLP, bilim adamlarının külfetli hesaplamalı çalışmalardan kaçınırken metinsel veri kümelerinde bulunan faydalı bilgileri kolayca çıkarmasına olanak tanımaktadır (Kang, Cai, Tan, Huang ve Liu, 2020).

Doğal Dil İşleme (NLP), dilin hem anlaşılmasını hem de üretilmesini hedefleyen iki temel bileşenden oluşur: Doğal Dil Anlama (NLU), metinlerin ne ifade ettiğini çözümlmeye odaklanırken; Doğal Dil Üretimi (NLG), bu anlamdan yola çıkarak yeni metinler oluşturmayı sağlar. Doğal Dil Anlama (NLU), makinelerin insan dilini yorumlamasına olanak tanır; bu sayede metinlerdeki kavramları, duyguları, varlıkları ve anahtar kelimeleri belirleyerek içerikleri anlamlı bir şekilde analiz eder. Dilbilim dilin anlamını, dilin bağlamını ve dilin çeşitli biçimlerini inceleyen bilimdir. Bu nedenle, NLP'nin önemli terminolojilerini ve NLP'nin farklı seviyelerini anlamak önem derecesi yüksek bir görevdir (Khurana, Koli, Khatter ve Singh, 2023).

Doğal dil üretme süreci, bilgisayarlaştırılmış veri tabanlarındaki verilerin insanlar tarafından kolayca anlaşılabilir bir dile dönüştürülmesini içerir. Doğal dili anlamak için metin parçalarının bilgisayar programcılarının yönetmesi, çok daha basit olan birinci dereceden mantıksal yapılara dönüştürülmesi gerekir. Ayrıca, NLP doğal dil ifadeleri kullanılarak elde edilen çeşitli olası anlamları tanıma ile ilgilenir ve bu tanımlamalar genellikle doğal dillerin kavramlarında bulunan düzenli notasyonlar biçimindedir. Dillerde ontolojiler ve meta-modeller oluşturmak, ampirik karakterde mükemmel çözümler sunar. Doğal dilin biçimsel hale getirilmesi için, kapalı ve açık terimler ile nesnel ve öznel arasındaki farklılıklar gibi özellikler ile ilgili varsayımlar yapılarak, anlamın biçimsel hale getirilmesi gereklidir (Bharadiya, 2023).

İKİNCİ BÖLÜM

LİTERATÜR İNCELEMESİ

Son yıllarda, yapay zekâ tarafından üretilen metni tespit etmeye yönelik araştırmalar büyük bir ilgi görmüştür. Bu çalışma, mevcut literatürdeki boşlukları doldurmayı ve bilgi birikimini genişletmeyi amaçlamaktadır. İlgili konu hakkında literatürde yapılan çalışmalar incelendiğinde, Ma ve arkadaşlarının 2023 yılında NLG modellerinin kötüye kullanılmasını ve bilim camiasında NLG modellerinin ürettiği yanlış bilgileri önlemek için, yapay zekâ tarafından oluşturulan bilimsel metin ile insan tarafından yazılan metin arasındaki boşluk konusunu araştırdıkları görülmektedir. Yapay zekâ tarafından üretilen bilimsel içeriğin gerçek konularda hatalar içerme olasılığının daha yüksek olduğunu tespit etmiş, yapay zekâ tarafından oluşturulan bilimsel metin ile insan tarafından yazılan bilimsel metin arasında bir bir “yazma stili” farkı olduğunu keşfetmişlerdir.

2019 yılında Gehrman, Strobelt ve Rush, bir metnin bir model tarafından oluşturulup oluşturulmadığını tespit etmek için insanlara destek olacak bir araç olan Dev Dil modeli Test Odası (GLTR - Giant Language model Test Room) geliştirmişlerdir. Açık kaynaklı, genel kullanıma açık ve oluşturulan çıktıları tespit etmek için halihazırda kullanılan GLTR, üretim yapaylıklarını tespit edebilen bir dizi temel istatistiksel yöntem kullanmaktadır. İnsanlar ile birlikte yaptıkları bir çalışmada, GLTR tarafından sağlanan açıklama şemasının herhangi bir ön eğitim gerektirmeksizin sahte metnin insanlar tarafından tespit edilme oranını %54’ten %72’ye çıkardığını göstermişlerdir.

Olsson ve Engelbrektsson 2022 yılında yaptıkları çalışmada esas olarak insanların yapay zekâ tarafından oluşturulan makaleleri insanlar tarafından yazılan makalelerden ayırt edip edemeyeceğine ve bunların kullanımını engellemek için hangi önlemlerin kullanılabileceği konusuna odaklanmışlardır. Halmstad Üniversitesi’ndeki öğretmenlere insan ve yapay zekâ tarafından üretilen metinler verilmiş, daha sonra bu öğretmenlerden sunulan metnin kaynağını tahmin etmeleri istenmiş ve deneyde öğretmenlerin yapay zekâ tarafından oluşturulan metni insan tarafından yazılan metinden ayırt etme kabiliyetinin yetersiz olduğu sonucuna varmışlardır. Çalışma GPT-2 tarafından oluşturulan metinler için mevcut dedektörler olan bilgisayar tabanlı

algılama yöntemlerinin, GPT-3 metnini algılamak için bir dereceye kadar umut verici olduğunu göstermiş olsa da mevcut hata oranları akademik sahtekarlıktan şüphelenilen bir durumda kesin kanıt olarak yöntemlerin son derece yetersiz kaldığını göstermiştir. Aynı zamanda, dedektörlerin doğruluk ve daha da önemlisi genelleştirilebilirlik açısından kat etmesi gereken uzun bir yolu olduğunu iddia etmişlerdir. Bu iddialarını GPT-3'ün şu anda NLP için önde gelen bir model olmasına rağmen, diğer modellerin muhtemelen önümüzdeki yıllarda hızlı ve büyük ölçüde gelişecek olması ile ilişkilendirmişlerdir.

2023'te Chaka, GPTZero, OpenAI Metin Sınıflandırıcı, Writer.com'un Yapay Zekâ İçerik Dedektörü, Copyleaks AI Tespit Aracı ve Giant Language Model Test Room olmak üzere beş farklı yapay zekâ içerik algılayıcısının doğruluk düzeylerini değerlendirmiştir. Bu süreçte, ChatGPT, YouChat ve Chatsonic kullanılarak İngilizce dil eğitimiyle ilgili istemler aracılığıyla oluşturulan yanıtlar analiz edilmiştir. Ardından, ChatGPT tarafından üretilen bu yanıtlar Google Translate aracılığıyla Almanca, Fransızca, İspanyolca, Güney Sotho ve isiZulu dillerine çevrilmiş ve içeriklerin yapay zekâ tarafından oluşturulup oluşturulmadığını belirlemek amacıyla GPTZero'ya yüklenmiştir. Ek olarak, Almanca, Fransızca ve İspanyolcaya çevrilen içerikler, AI üretimini tespit etmek üzere Copyleaks'in içerik dedektörüne de gönderilmiştir. ChatGPT, YouChat ve Chatsonic tarafından oluşturulan yanıtlar için Copyleaks AI İçerik Dedektörü, beş AI içerik dedektörü arasında en iyi performansı göstermiş olup, bir sonraki başarılı yöntem ise OpenAI'nin AI Metin Sınıflandırıcısı olmuştur. Google tarafından beş dile çevrilen ve ChatGPT tarafından oluşturulan yanıtlarla ilgili olarak GPTZero bu yanıtların tamamının insan yapımı olduğu yönünde yanlış sınıflandırma yapmıştır. Google'ın Almanca, Fransızca ve İspanyolca'ya çevirdiği ve ChatGPT tarafından oluşturulan yanıtlar için Copyleaks AI İçerik Dedektörü Almanca'ya çevrilmiş metinlerden üçünü, Fransızca'ya çevrilmiş metinlerden beşini ve tüm İspanyolca çevrilmiş metinleri yapay zekâ tarafından oluşturulmuş olarak doğru bir şekilde tanımlamıştır. Bu çalışma, AI kaynaklı içeriklerin doğru ve ikna edici bir biçimde tespit edilmesinin hâlâ tam anlamıyla mümkün olmadığını açıkça göstermektedir. Bu durum, özellikle akademik çalışmalarda ve makale yazımında yapay zekânın ürettiği yanlış olabilecek sonuçlar ve intihal açısından oldukça ciddi sonuçları olabileceğini göstermektedir.

Elkhatat ve arkadaşları 2023'te OpenAI, Writer, Copyleaks, GPTZero ve CrossPlag dahil olmak üzere AI metin içeriği algılayıcılarının performansını değerlendirmişler ve sonuç olarak araçların metni yapay zekâ tarafından oluşturulmuş veya insan tarafından yazılmış olarak doğru bir şekilde tanımlama ve kategorilere ayırma becerisinde önemli farklılıklar olduğunu göstermişlerdir. Ayrıca genel eğilimin, GPT 3.5 tarafından oluşturulan içeriği sınıflandırmanın GPT 4 tarafından oluşturulan içeriğe kıyasla daha iyi bir performans gösterdiğini vurgulamışlardır.

2023'te Bhattacharjee ve Liu çalışmalarında, ChatGPT'nin yapay zekâ tarafından oluşturulan bir metin üzerinde dedektör olarak performansını araştırmışlardır. İnsan tarafından yazılan ve yapay zekâ tarafından oluşturulan metin algılama görevinde ChatGPT'nin sıfır atış performansı değerlendirilmiş ve halka açık veri kümeleri üzerinde deneyler gerçekleştirilmiştir. Deneylerinde, ChatGPT'nin yapay zekâ tarafından oluşturulan metni tanımlamakta zorlansa da insan tarafından yazılan metinlerde iyi performans gösterdiğine dair ilginç bir bulguyu ortaya koymuşlardır. ChatGPT'nin bu asimetrik performansı, insan tarafından yazılan metni tanımlamaya odaklanan dedektörler oluşturmak ve böylece dolaylı bir şekilde de olsa yapay zekâ tarafından oluşturulan metin algılama sorununu etkili bir şekilde çözmek için kullanılabilir olduğu sonucuna varılmıştır.

Yang ve arkadaşları 2023'te yaptıkları çalışmalarında, ChatGPT ile düzeltilmiş metinler gibi insan-makine iş birliği yoluyla oluşturulan metinlerin de ayırt edilmesinin güç olduğunu iddia etmiş ve bu boşluğu ele alarak, daha sağlam bir ChatGPT dedektörü oluşturmaya odaklanmışlardır. Human-ChatGPT Polished Paired abstracTs adı verilen bir veri kümesi geliştirmişlerdir. Daha sonra, insan tarafından yazılan metinleri ChatGPT ile düzeltilmiş metinlerden ayırt edebilen RoBERTa tabanlı bir model geliştirmişlerdir. Deneylerinin sonucunda önerdikleri dedektör yardımıyla kullanıcıların şüpheli metinler hakkında doğru ve ikna edici bir yargıya varabileceğini iddia etmişlerdir.

2023 yılında, bir dedektörü ve bir yorumlayıcıyı ortaklaşa eğitmek için çekişmeli öğrenmeyi benimseyen Radar adı verilen bir yapay zekâ metin dedektörü eğitim sistemi sunmuş olan Hu ve arkadaşlarının çalışmasında, yorumlayıcının amacı yapay zekâ metni tespitinden kaçınmak için gerçekçi bir içerik oluşturmaktır. RADAR, dedektörden gelen geri bildirimini kullanarak yorumlayıcıyı günceller ve bunun tersini de yapar. 8 farklı dil modeli ile 4 veri kümesinde değerlendirilen deneysel sonuçlar,

özellikle yorumlayıcı devredeyken Radar'ın mevcut yapay zekâ metin tespit yöntemlerini (GPT-J, Pythia, Vicuna, LLaMA, Dolly 2.0, Camel ve Dolly 1.0) önemli ölçüde geride bıraktığı gösterilmiş olsa da tespit sonucunun her zaman doğru tahminler yapamayacağı için kanıt olarak kullanılması gibi ciddi durumlarda daha ileri doğrulama adımlarının gerekli olduğunu da belirtmişlerdir.

Liang ve arkadaşları 2023 yılında yaptıkları çalışmalarında, anadili İngilizce olan ve olmayan yazı örnekleri üzerinde yaygın olarak kullanılan çeşitli GPT algılayıcılarının performansını değerlendirmiş olup dedektörlerin anadili İngilizce olmayan yazı örneklerini sürekli olarak yapay zekâ tarafından oluşturulmuş olarak yanlış sınıflandırıldığını ancak ana dili İngilizce olan yazı örneklerini doğru bir şekilde tanımladığını ortaya koymuşlardır.

2023 yılında yaptıkları çalışma ile Mitrović ve arkadaşları, bir makine öğrenme modelinin, özellikle metnin kısa olduğu durumlarda insan ve ChatGPT tarafından üretilen metni doğru bir şekilde ayırt etmek için etkili bir şekilde eğitilip eğitilemeyeceğini araştırmışlar ve ML modelinin kullanılmasının gerekli olduğunu göstermişlerdir. ML modelinin yeni metni, orijinal insan metinlerini yeniden ifade ederek oluşturmak yerine, müşteri sorgularına dayalı olarak oluşturulduğunda daha iyi ayırım yapabildiği belirtilmiştir. Tahminlerin SHAP (Shapley Additive Explanations - Shap ile Açıklanabilir Yapay Zekâ) açıklamalarına bakmak, ChatGPT' nin yazma stili hakkında bazı bilgiler vermektedir. Model, bir özgünlük puanı tabanlı yaklaşımla karşılaştırmış ve tekrar ifade edilmiş metin kullanıldığında insan ve ChatGPT tarafından üretilen metinler arasındaki ayrımın yapılmasının ML modeli için daha zor olduğunu göstermiştir. Önerilen yaklaşım %79 doğruluk elde etmiştir. Mevcut yapay zekâ modellerinin, insan dilinin zenginliği ve inceliklerini tam anlamıyla ifade etme konusunda sınırlamalarının bulunduğunu ayrıca ironi, metafor gibi durumları ayırt edebilecek kadar insan dilinin derinliğine sahip olmadığı iddia edilmiştir.

Otomatik olarak üretilen bilimsel makalelerin tespiti üzerine yapılan Kashnitsky ve arkadaşlarının 2022'deki çalışmasında, genel bir üretilmiş içerik tespit aracı geliştirmekten uzak olduğumuz iddia edilmiş olup, eğitim verilerine sahip olmadığımız her yeni modelin (mesela, GPT-4) yeni bir zorluk oluşturduğu ve herhangi bir tespit aracının bu tür bir modelle üretilmiş olan içeriği tanıma konusunda, veri kaymasından dolayı başarısız olma olasılığının yüksek olduğu yargısına varılmıştır.

ÜÇÜNCÜ BÖLÜM

KULLANILAN YÖNTEMLER

3.1. Veri Seti

Bu tez kapsamında hazırlamış olduğumuz veri setimizin 3 farklı versiyonu bulunmaktadır. Bunların her birinin adlandırılması v0, v1, v2 olmak üzere veri setlerinin oluşturulma sırasına göre yapılmıştır. Tüm veri setleri temelde gerçek özet ve yapay özet olarak iki temel gruptan oluşmaktadır. Gerçek özet kısmı Dergipark Akademik platformundan belirlenen farklı bilim dallarında yayınlanmış Türkçe makalelerin özet bölümlerinden oluşmaktadır.

v0 olarak adlandırılan veri setinin ilk versiyonu için Dergipark'tan Bilgisayar Mühendisliği Sosyoloji ve Tıp alanlarında yayınlanmış makalelerin başlıkları ve anahtar kelimeleri gibi ayırt edici bilgiler seçilerek yapay zekâ metin oluşturucuya (Chatgpt 3.5) verilmiş ve bu bilgiler doğrultusunda her bir başlık için bir makale özeti oluşturması istenmiştir. Oluşturulan bu özetler yapay zekâ özetleri (yapay zekâ tarafından oluşturulan metin) olarak veri setine eklenirken, Dergipark'taki orijinal doküman özeti de insan özeti olarak aynı veri setine eklenmiştir. Dergipark platformundan alınan makale özetleri seçilirken yayın tarihleri de dikkate alınmıştır. Veri setine eklenen özetlerin yazarı yapay zekâ mı yoksa insan mı olduğu bilgisinin yazarın beyanı dışında doğruluğundan emin olunması için, makale özetinin oluşturulduğu tarihte yapay zekâ araçlarının kullanımda olmaması önem teşkil etmektedir. Bu nedenle Chatgpt'in lansmanından (Kasım 2022) öncesine ait yayınlar seçilmiştir. Veri seti için Bilgisayar Mühendisliği, Sosyoloji ve Tıp bilim dallarında yayınlanmış ve insan olarak etiketlenen toplam 300 adet makale seçilmiştir. Bu makalelerin başlıklarının ve anahtar kelimelerinin ChatGPT'ye verildiği örnek bir sorgu aşağıdaki şekildedir:

“Makale başlığı” başlığı ve “anahtar kelimeler” anahtar kelimeleri ile bir makale özeti yaz. Ya da “Makale başlığı” başlığa sahip ve anahtar kelimeler : “anahtar kelimeler olan bir makale özeti yazar mısın?”.

Bu sorgulardan birinin sonucu olarak “ai” etiketli yapay özetleri ortaya çıkmıştır. Veri seti v0 5946 satırlık bir metin tablosundan oluşmaktadır. Bu tablodaki verilerin 3029 satırı insanlar tarafından elde edildiği için “human” olarak etiketlenirilmiş, geri kalan

2918 satırlık veri ChatGPT 3.5 tarafından oluşturulduğu için “ai” olarak etiketlenmiştir.

metin	etiket
İçinde bulunduğumuz teknoloji çağında veri güvenliği artık hükümetlerin, şirketlerin ve bireylerin en hayati ve en önemli konusu haline gelmiştir.	human
Veri türleri arasında görseller içerdikleri önemli bilgiler nedeniyle özel bir öneme sahiptir.	human
Görüntülerin aktarılması veya saklanması ekstra güvenlik önlemleri gerektirmektedir.	human
Bu çalışmada, bazı görüntü şifreleme yöntemleri görüntü işleme alanında en popüler ve en çok kullanılan görüntülere uygulanarak performansları karşılaştırılmıştır.	human
Dört farklı deney gerçekleştirilmiştir.	human
Yedi farklı şifreleme yönteminin performansları birbirleri ile karşılaştırılarak yapılan ölçümler ve elde edilen gözlemler sunulmuştur.	human
Bu makalede, kriptoloji ve veri güvenliği alanlarında kullanılan bazı görüntü şifreleme tekniklerinin performans analizi ve uygulamaları incelenmektedir.	ai
Günümüzde, veri güvenliği büyük önem taşımaktadır ve özellikle görüntü verilerinin korunması giderek artan bir endişe kaynağı haline gelmiştir.	ai
Makalede, farklı görüntü şifreleme tekniklerinin nasıl çalıştığı ve bu tekniklerin güvenlik düzeyi üzerindeki etkisi detaylı bir şekilde ele alınmaktadır.	ai
Kriptoloji prensipleri temel alınarak geliştirilen bu teknikler, görüntü verilerini şifreleyerek yetkisiz erişimden korumayı amaçlamaktadır.	ai
Ayrıca, makalede bu şifreleme tekniklerinin performansları çeşitli ölçütlere göre karşılaştırılmaktadır.	ai
Görüntü işleme algoritmalarıyla birlikte kullanılan bu tekniklerin işlem hızı, şifreleme/çözme kalitesi ve kaynak tüketimi gibi faktörler incelenerek performansları değerlendirilmektedir.	ai
Sonuç olarak, makale, farklı görüntü şifreleme tekniklerinin uygulamalı performans analizini sunarak, bu tekniklerin güvenlik açısından etkinliğini ve pratik uygulanabilirliklerini değerlendirmeyi amaçlamaktadır.	ai
Bu çalışma, kriptografi ve veri güvenliği alanında araştırma ve uygulama yapacak olanlar için önemli bir kaynak olabilir.	ai

Şekil 3.1: v0 veri seti; özetler cümlelere bölünerek tutulmuş yapay özetler Chatgpt 3.5 tarafından oluşturulmuştur.

Şekil 3.1’de v0 veri setinden örnek bir kesit yer almaktadır. Bir makalenin hem insan (human) hem de Chatgpt 3.5 (ai) tarafından oluşturulmuş özetlerin cümle cümle ayrılmış olduğu görülmektedir. Veri setinde her bir özeti cümleler halinde bölünüp, yapay zekâ aracının bağımsız cümle oluşturma performansı incelenmiş, konu

bütünlüğü önemsenmemiştir. Veri seti v0 hazırlanırken henüz Chatgpt-4 piyasaya çıkmadığı için yapay özetler Chatgpt 3.5 tarafından ürettirilmiştir.

Chatgpt-4'ün piyasaya çıkması ile bu çalışmanın ilerleyen zamanlarında v1 veri seti geliştirilmeye başlanmıştır. Veri seti v1 Sağlık, Hukuk, Mühendislik ve Sosyoloji olmak üzere 4 bilim dalından seçilen makalelerle oluşturulmuştur. Her bir bilim dalından 250'şer adet makale özeti alınarak toplamda 1000 adet insan yazımı makale özeti toplanmıştır. Yine veri seti v0'da olduğu gibi, seçilen bu makalelerin anahtar kelimeleri ve başlık bilgisi kullanılarak ChatGPT-4 modelinden makale özeti oluşturması istenmiştir. Aşağıda çalıştırılmış olan örnek bir sorgu verilmiştir.

“Ses veya Arayüz Yardımı ile Kontrol Edilebilen Mobil Robot Kol Tasarımı başlıklı ve anahtar kelimeleri şunlar olan makalenin özetini yazar mısın? Anahtar kelimeler: Ses ile kontrol, Mobil araç, Robot kol, Arayüz.”

metin	etiket
Robot teknolojisinin hızlı gelişimine paralel olarak mobil araç, robot kol ve ses işleme teknolojisi de hızlı bir gelişim göstermiştir. Bu teknolojik gelişimde robotlardan beklenen en önemli parametrelerin başında güvenlik, çözüm üretme ve hız gelmektedir. Bu çalışmada, mobil araç üzerine bir robot kol yerleştirilmesi ve bu sistemlerin tasarlanan arayüz ve ses sistemi ile kontrol edilerek daha verimli ve hızlı çalışması amaçlanmıştır. Amaçlar doğrultusunda önce üzerinde bir robot kol bulunan bir mobil araç tasarımı gerçekleştirilmiş, sonra da bu mobil aracın ve robot kolun hem arayüz hem de ses komutları ile kontrol edilmesi sağlanmıştır. Yapılan test sonuçları incelendiğinde ses komutları ile kontrol sisteminin, arayüz ile kontrol sistemine göre daha verimli olduğu gözlemlenmiştir.	insan
Bu çalışma, ses komutları veya kullanıcı arayüzü yardımıyla kontrol edilebilen mobil robot kol tasarımını ele almaktadır. Mobil araç üzerine entegre edilen robot kol, kullanıcı tarafından sesli komutlar veya grafiksel bir arayüz üzerinden kontrol edilebilecek şekilde geliştirilmiştir. Bu sistem, özellikle endüstriyel otomasyon, uzaktan manipülasyon ve engelli bireyler için destekleyici teknolojiler gibi çeşitli alanlarda kullanılacak esnek bir yapıya sahiptir. Çalışmada, ses tanıma algoritmaları kullanılarak robot kola verilen komutların algılanması ve hareketlerin hassas bir şekilde gerçekleştirilmesi sağlanmıştır. Ayrıca, alternatif bir grafik kullanıcı arayüzü (GUI) geliştirilerek manuel kontrol imkanı sunulmuştur. Sonuçlar, ses ile kontrol edilen sistemlerin kullanım kolaylığı sağladığını ve robot kolun mobil platformlarla entegrasyonunun çeşitli uygulamalar için verimli bir çözüm sunduğunu göstermektedir.	ai

Şekil 3.2: v1 veri setinden örnek bir satır görülmektedir, aynı konulu makalenin insan ve ai (chatgpt-4) tarafından yazılmış içeriği.

Şekil 3.2' de veri setindeki bir yayınlanmış makale özeti ve aynı makale için oluşturulmuş yapay zekâ özetini içeren bir örnek verilmiştir. Veri seti v1'de metin içerisinde anlam bütünlüğünün korunup korunmadığı konusunun, o metnin insan ya

da ai tarafından mı yazıldığına tespiti için belirleyici bir unsur olduğunu düşündüğümüz için, özetler paragraf halinde tutulmuş olup cümlelerine ayrılmamış, v1 veri setinde her bir satır tam bir özet olarak saklanmıştır. Her bir makale için özet cümlelerinin oluşturduğu paragraf konu bütünlüğünü korumak adına tek seferde model tarafından işlenmek üzere veri setine yerleştirilmiştir. Seçilmiş olan 1000 adet insan yazımı makale özetinin her biri için üretilen yapay zekâ özetleriyle beraber veri seti 2000 adet satırdan oluşmaktadır. Bu makalelerde yayın yılları yine 2014 yılı ve öncesine ait olanlar seçilmiştir.

Son olarak model başarısının objektif olarak ölçülebilmesi ve modelin performansının farklı veri setleri üzerindeki karşılaştırmasının yapılabilmesi için farklı bir yapay zeka metin oluşturma platformu kullanarak v2 veri seti oluşturulmuştur. Claude.ai modeli ile üretilen özetler için de aşağıdaki şu sorguya başvurulmuştur:

“Makale başlığı” başlığa sahip ve anahtar kelimeler : “anahtar kelimeler olan bir makale özeti yazar mısın?”.

Bu sorguların sonucu olarak makale içerikleri birkaç başlık olarak döndürülmüştür. Bu sebeple sorgu aşağıdaki şekilde geliştirilmiştir:

“Tek bir başlık altında (özet) ve bir paragraf yaklaşık 150 - 200 kelimelik bir özet yazar mısın?”.

Claude.ai ile üretilen 100 adet makale özeti, v1 veri setinde bulunan Sağlık, Hukuk, Mühendislik ve Sosyoloji bilim dallarından eşit sayıda 25'er toplamda 100 insan özetinin aynı anahtar kelime ve başlığa sahiptir. Sonuç olarak v2 veri setinde her biri karşılığındaki özetle aynı makale için yazılmış olmak üzere 100 insan, 100 claude.ai, 100 ghatgpt olmak üzere 300 adet makale özeti bulunmaktadır. Şekil 3.3' de v2 veri setinden bir örnek paylaşılmıştır.

metin	etiket
<p>İnternet altyapısının güvenliği üzerine denemeler ve eğitimler sağlamak için siber güvenlik deneylerinin ve tatbikatlarının yapılması gereklidir. Bu çalışmada, bu deneyler için kullanılacak ağ benzetici, öykünüm ve sınama ortamlarının kullanımına dair bir ön inceleme sunulmuştur. Ağ sınama ortamları tanımlanmış; Emulab ve Epic sınama ortamı ayrıntılı olarak tanıtılmıştır. Ağ sınama ortamı kurulması için gereklilikler çalışmada önerilmiştir.</p>	insan
<p>Bu çalışma, siber güvenlik eğitimi ve araştırmalarında kullanılan ağ benzetim sistemleri ve sınama ortamlarının etkinliğini değerlendiren bir ön inceleme sunmaktadır. Siber güvenlik deneyleri, gerçek ağ altyapılarında risk oluşturmadan güvenlik açıklarının test edilmesi ve savunma stratejilerinin geliştirilmesi için kritik önem taşır. Çalışma, sanal ağ ortamlarının siber güvenlik tatbikatlarındaki rolünü ve Emulab gibi gelişmiş benzetim platformlarının sağladığı avantajları analiz etmektedir. Ağ benzetim teknolojileri, karmaşık ağ topolojilerinin sanal ortamda yeniden oluşturulmasını ve çeşitli saldırı senaryolarının güvenli bir şekilde test edilmesini mümkün kılar. Sistem, penetrasyon testleri, olay müdahale senaryoları ve güvenlik açığı analizlerinin gerçekçi ortamlarda gerçekleştirilmesini sağlar. Emulab ve benzeri platformlar, fiziksel donanım kaynaklarının sanallaştırılması yoluyla maliyet etkin ve ölçeklenebilir deneyim ortamları sunar. Çalışma, farklı benzetim araçlarının performans, gerçekçilik ve kullanım kolaylığı açısından karşılaştırmalı analizini yaparak siber güvenlik eğitimi için en uygun platform seçiminde rehberlik sağlar. Bu yaklaşım, siber güvenlik uzmanlarının yetiştirilmesi ve kurumsal güvenlik kapasitelerinin geliştirilmesi için önemli bir metodolojik çerçeve sunmaktadır.</p>	claude_ai
<p>Siber tehditlerin giderek karmaşık hale gelmesi, siber güvenlik deneyleri ve tatbikatlarının daha gerçekçi ve kontrollü ortamlarda gerçekleştirilmesini zorunlu kılmaktadır. Bu bağlamda, ağ benzeticileri ve ağ sınama ortamları, güvenlik açıklarını analiz etmek, savunma stratejilerini test etmek ve yeni saldırı senaryoları oluşturmak için kritik bir rol oynamaktadır. Bu çalışmada, siber güvenlik testleri için kullanılan ağ benzetim ve sınama ortamları incelenmiş, bu sistemlerin avantajları ve sınırlamaları değerlendirilmiştir. Özellikle, Emulab gibi ağ benzeticilerinin gerçekçi ağ koşullarını modelleme kapasitesi, farklı saldırı senaryolarının test edilmesi açısından ele alınmıştır. Çalışmanın sonuçları, ağ güvenliği testleri için uygun benzetim araçlarının seçilmesinin, siber güvenlik deneylerinin doğruluğunu ve etkinliğini artırmada kritik bir faktör olduğunu göstermektedir.</p>	chatgpt

Şekil 3.3: v2 veri setinden örnek bir satır görülmektedir, aynı konulu makalenin insan, Chatgpt ve claud.ai tarafından yazılmış içeriği. Toplamda 300 satırlık veri setinden 3 satırı örnek olarak gösterilmiştir.

3.1.1. Metin Ön İşleme (Pre-processing)

Veri kalitesi, makine öğrenimi modellerinin performansını önemli ölçüde etkilemektedir. Yetersiz veya düşük kaliteli veriler, modelin doğruluğunun, etkinliğinin ve güvenilirliğinin azalmasına neden olabilir. Genel olarak doğal dilden türetilen metin verileri yapılandırılmamış (işlenmemiş) ve gürültülüdür. Dolayısıyla metin ön işleme dağınık, ham ve yapılandırılmamış metin verilerini, makine öğrenimi modellerini eğitmek için etkili bir şekilde kullanılacak bir forma dönüştürmek ve daha iyi sonuçlar ile içgörüler sağlamak için kritik bir adımdır. Metin ön işleme ham metin verilerini temizlemek, dönüştürmek ve NLP veya ML görevlerine uygun bir formata hazırlamak için kullanılan bir dizi tekniği ifade eder. Metin ön işlemenin amacı sonraki analiz veya modelleme için metin verilerinin kalitesini ve kullanılabilirliğini arttırmaktır. Veri ön işleme için genellikle yapılan işlemler aşağıda verildiği gibidir (Aydın, t.y.):

a) Etkisiz Kelimeler (StopWords)

- Türkçe için: ['en', 'fakat', 'lakin', 'ancak', 'hiç', 'ama', 'bu', 'daha', 'ile', 'için', 'acaba', 'hep', 'birkaç', 'defa', 'çünkü', 'aslında', 'da', 'gibi', 'bazı', 'hem', 'eğer', 'diye', 'belki', 'biz', 'her', 'çok', 'de', 'biri', 'hepsi', ,, vb.]
- İngilizce dili için : ['i', 'me', 'my', 'myself', 'we', 'can', 'few', 'most', 'the', 'to', 'will', ..., etc.]

gibi kelimeler çok az bilgi taşıyan dolgu kelimelerdir. Bu kelimeler kullanıldıkları metinlerde gramatik olarak gerekli olurken, metnin anlamsal olarak ayrıştırılması açısından çoğunlukla önemli bir görev üstlenmemektedirler.

Çoğu durumda, doğal dil işleme modellerinin programlanması ve bilgi çıkarımı yapılması söz konusu olduğunda, bilgisayarlara bu kelimeleri göz ardı etmelerinin söylenmesi eğitilen modelin performansını artırıcı etki yaratmaktadır. Bilgi vermeyen, içerik sağlamayan bu sözcüklere literatürde “StopWords”, Türkçe karşılığı olarak “Dolgu Kelimeleri/Etkisiz Kelimeler” denilmektedir. Bir metinde StopWords kelimelerin varlığı özellikle kelimelerin görünme sayıları üzerine çalışan algoritmaların kullanımında veriden anlamlı bilgi çıkarmayı zorlaştırabilir. Çünkü bu kelimeler metin içerisinde çoğu zaman en çok görünme oranlarına (occurrence count) sahip olurlar. StopWords listesinin önceden belirlenmesi genellikle modelin “gürültüyü” azaltmasını kolaylaştırır. NLP bu bilgilendirici olmayan kelimeleri

görmezden gelebilir ve sonuç olarak içgörülerini gerçekleştirmek için daha büyük, daha çeşitli miktardaki veriler arasında daha hızlı hareket edebilir (“What are Stop Words”, t.y.). Bu kelimeleri veri setinden temizlemek modelin önemli kelimelere odaklanmasına yardımcı olabilir.

b) Küçük Harflere Çevirme (Lowercasing)

Doğal dil işleme süreçlerinde çoğunlukla tercih edilen temel bir ön işleme adıdır. Bu işlem sayesinde aynı anlama sahip olan ancak farklı boyutlardaki harf kullanım durumlarında (örneğin “Bilgi” ve “bilgi”) metin içerisinde yer alan kelimelerin görünümü birbirleriyle eşitlenir ve modelin bu kelimeleri farklı terimler gibi algılamasının önüne geçilir. Böylece, veri setindeki gereksiz varyasyonlar azaltılarak hem işlem karmaşıklığı düşürülür hem de algoritmanın daha tutarlı ve etkili öğrenme gerçekleştirmesi sağlanır (Aydın, t.y.).

c) Kelimenin Kökünü Bulma (Stemming)

Köklendirme, doğal dil işlemede kelimeleri kök veya temel biçimlerine indirgemek için kullanılan bir metin ön işleme tekniğidir. Kök çıkarmanın amacı tıpkı küçük harflere çevirme tekniğinde olduğu gibi kelimelerin farklı varyasyonlarını basite indirgeyerek aynı anlamdaki kelimelerle modeli doğru eğitmektir. Köklendirme ile ulaşılmak istenen sonuç kelimeleri basitleştirmek ve standartlaştırmaktır; bu da bilgi çıkarımına, metin sınıflandırmaya ve diğer NLP görevlerinin performansını artırmaya yardımcı olur (Saturn Cloud, t.y.). Örnek : [‘running’, ‘runner’, ‘runs’] kelimelerine köklendirme tekniği uygulandığında: [‘run’, ‘run’, ‘run’] şekline dönüşmekte ve hepsi için ortak kök olan “run” kelimesi ortaya çıkmaktadır.

d) Anlamlı Kök Bulma (Lemmatization)

Kelimenin anlamı ve bağlamı göz önünde bulundurularak kelimeleri “lemma” adı verilen temel biçimlerine dönüştüren daha karmaşık bir tekniktir. Köklendirme işleminden farklı olarak, lemmatizasyon süreci, elde edilen sonucun dilde gerçekten var olan geçerli bir kelime olmasını garanti eder. Bu yöntem genellikle, kelimeleri temel hâline indirgemek için sözlük ya da kelime dağarcığı gibi kaynaklardan yararlanmayı içerir. Örnek: [‘running’, ‘runner’, ‘runs’] şeklinde bir kelime vektörü lemmatizasyondan sonra konuşmanın ilgili bölümlerine ve bağlamlarına göre farklı lemmalara indirgenmekte ve [‘run’, ‘runner’, ‘run’] kelimeleri ortaya çıkmaktadır (Roop, t.y.).

e) Etiket Kodlama (Label Encoding)

Python programlama dilini için tanımlanmış scikit-learn kütüphanesinde bulunan bir işlem/sınıftır ve genellikle kategorik sütunlardaki metin tabanlı etiketleri sayısal değerlere dönüştürmek için kullanılır. Bu dönüşüm özellikle makine öğrenimi algoritmalarının girdi olarak sayısal verilere ihtiyaç duyduğu durumlarda yapılır. Metin etiketi olan alfabetik verileri sayısal değerlere dönüştürerek makine öğrenmesi tarafından işlenebilir bir forma getirmek için metin ön işleme olarak gerçekleştirilir (Scikit-learn, t.y.).

3.2. Metin Temsili (Word Embedding)

Doğal dil işleme (NLP) alanında metin temsili, kelimelerin bilgisayar tarafından işlenebilir sayısal yapılara dönüştürülmesini sağlayan bir yöntemdir. Genellikle çok boyutlu sürekli uzayda konumlanan sayısal vektörler aracılığıyla ifade edilen kelimeler arasındaki anlamsal ve sözdizimsel ilişkileri modeller. Böylece benzer anlamlara sahip kelimelerin vektör uzayında birbirine yakın yerleşmesini mümkün kılar. Makine öğrenmesi ve derin öğrenme tabanlı algoritmaların metin verisini daha etkin ve anlamlı biçimde işleyebilmesine imkan sağlar. Literatürde yaygın olarak kullanılan metin temsili teknikleri arasında Word2Vec, TF-IDF, FastText ve bağlama duyarlı dil modellerinden BERT gibi yöntemler öne çıkmaktadır. Bu yöntemler, dilin yapısal ve anlamsal özelliklerini farklı düzeylerde yakalayıp metin madenciliği, bilgi çıkarımı ve otomatik anlama sistemleri gibi çeşitli uygulama alanlarında kritik bir rol oynamaktadır.

3.2.1. Word2Vec

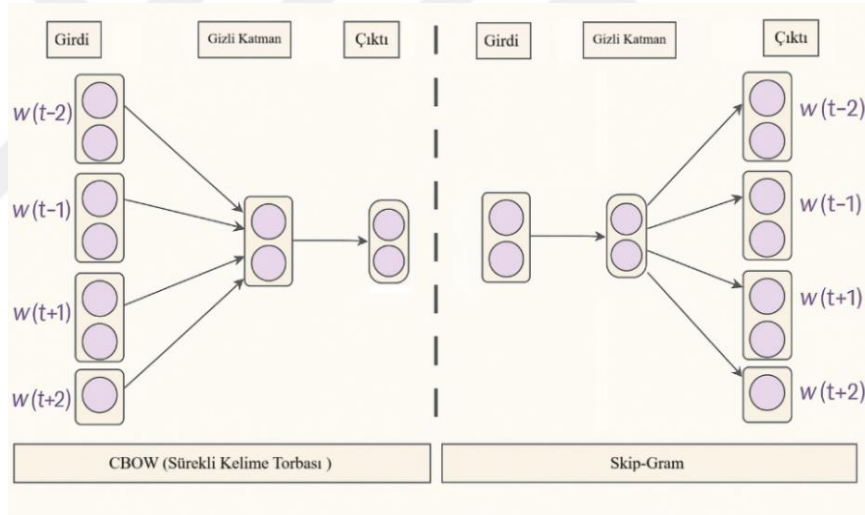
Word2Vec aracı girdi olarak bir metin gövdesini alır ve kelime vektörlerini bu girdiye çıktı olarak üretir. İlk olarak eğitim metni verilerinden bir kelime havuzu oluşturur ve buradan kelimelerin vektörel gösterimini öğrenir (Google, t.y.). Word2Vec kelimeleri sayısal vektörler olarak temsil eden eğitilmiş, sığ ve 2 katmanlı bir sinir ağıdır. Giriş eğitim setindeki tüm belgeleri/metinleri içerir. Ağın bu metinleri işlemesi için kelimeler one-hot encoding (her kelimenin yalnızca tek bir “1” değeri ve diğer tüm konumlarda “0” olacak şekilde, vektör biçiminde temsil edilmesidir.) kodlamasında temsil edilirler. Gizli katmanda bulunan nöron sayısı istenilen yerleştirmenin uzunluğuna eşittir. Tüm kelimelerin 300 uzunluğunda vektörler olması istenirse, gizli katman 300 nöron içerecektir. 2 temel öğrenme algoritması bulunmaktadır, CBOW ve

Skip Gram. Bu iki yöntem de bir kelimenin temsilini öğrenerek cümle içerisindeki diğer kelimeleri tahmin etmede kullanılabilir anlamlı bağlamlar üretmeyi amaçlar.

CBOW hedef kelimeyi çevresindeki kelimelere göre tahmin eder. Örneğin, “Kedi süt içer” cümlesi incelendiğinde model daha sonra bu cümle üzerinde farklı hedef kelimeler için yineleme yapacaktır. Örneğin: “Kedi ... içer” girdiler olurken, “süt” ise hedef kelimedir. CBOW tüm bağlamı tek bir gözlem olarak ele aldığı için bilginin dağıtımını yumuşatır.

Skip-Gram ise CBOW'un tam tersi şekilde çalışır. Burada, bir girdi sözcüğü alınır ve bu kelimedenden yola çıkarak çevresinde yer alan kelimeleri tahmin etmesi beklenir. Örneğin, “Kedi” kelimesini modellemek için, çevresindeki kelimeleri (“süt”, “içer”) tahmin etmeye çalışır (Suri, t.y.).

CBOW, Skip-Gram'dan daha hızlı bir algoritmadır ve sık kullanılan kelimelerle daha iyi bir performans göstererek çalışır.



Şekil 3.4: Word2Vec mimarisinde kullanılan CBOW ve Skip-gram modelleri.

CBOW, bağlam kelimelerden hedef kelimeyi tahmin ederken; Skip-gram, hedef kelimedenden bağlam kelimeleri tahmin etmeye çalışır.

Kaynak : Sangole, 2024

3.2.2. FastText

Kelimeleri ve metinleri daha verimli bir şekilde temsil etmek için geliştirilmiş bir kelime vektör (word embedding) modelidir. Facebook AI Research araştırmacıları tarafından geliştirilen FastText, Word2Vec fikrini temel alır ve n-gramların alt kelime

düzeyindeki özelliklerini modele entegre ederek bunu geliştirir (Karadeniz, Tokdemir ve Maraş, 2024). FastText ile Word2Vec arasındaki en belirgin fark, FastText'in kelimeleri tek bir bütün olarak değil, n-gram adı verilen alt birimlere ayırarak işlemesidir. Bu yaklaşım sayesinde, model kelimelere ait daha isabetli vektör temsilleri öğrenebilir (Salsabiila, Irmanda ve Arista, 2023). FastText kütüphanesi, skipgram yöntemini kullanarak kelime özelliklerini kendisi çıkarır. Örneğin, 'fantastik' kelimesi fan, fant, fanta, fantas, fantast, fantasti, fantastic vb. olarak kabul edilir. Burada "n" değeri 1 ile başlayıp, kelimenin uzunluğuna kadar değişebilir. Bu çeşitlilik veri kümesindeki daha nadir veya hiç görülmemiş kelimeler için sıfır vektör gösterimi döndürülmemesini sağlar. Böylelikle daha az görülen kelimeler için model daha da güçlü hale getirilir ki bu katkı da konular arasında sınıflandırma problemine katkıda bulunur (Tahsin, Mozumder, Shahriyar ve Mollah, 2020).

Başka bir örnek olarak 3 harfli n-gram kümelemesini ele alırsak "learning" kelimesi için "lea", "ear", "arn", "rni", "ing" gibi 3 harfli alt kelimelerden vektörler oluşturulur ve bu vektörler ile model eğitilebilir (Pratiwi, Budi ve Alfina, 2018). Ancak, üretilen n-gram sayısının, corpustaki (sözlükteki) toplam kelime sayısından çok daha fazla olması, eğitim süresini uzatmaktadır. Bununla birlikte, dokümanlarda nadir bulunan kelimeleri, Word2Vec'e kıyasla daha iyi temsil edebilmektedir. Word2Vec gibi yöntemler, kelime dağarcığında yer alan her kelime için ayırt edici vektör gösterimleri üretir. Fakat bu durum dilin iç yapısının öğrenilmesine herhangi bir katkı sağlamamaktadır. Bu sebeple vektörel gösterimler morfolojik olarak zengin diller için bir sınırlamadır. Çünkü vektörel gösterimde kelimelerin sözdizimsel ilişkisi göz ardı edilmektedir. Morfolojik açıdan zengin dillere özgü kurallar birçok kelimenin yapısında etkili olduğundan, bu tür dillerde karakter düzeyinde bilgi kullanımıyla vektör temsillerinin kalitesi artırılabilir. Bu bağlamda, FastText modeli, karakter n-gramlarını dikkate alarak, kelime vektörlerini bu alt birimlerin ortalaması şeklinde oluşturur. Bu yaklaşım yalnızca kelime yapısını daha doğru yansıtmakla kalmaz, aynı zamanda sözlükte yer almayan yeni kelimelerin de temsil edilebilmesini mümkün kılar. Yani model eğitimi sırasında metin içerisinde mevcut olmayan kelime yerleştirmelerini de bulabilir (Bojanowski vd., 2017).

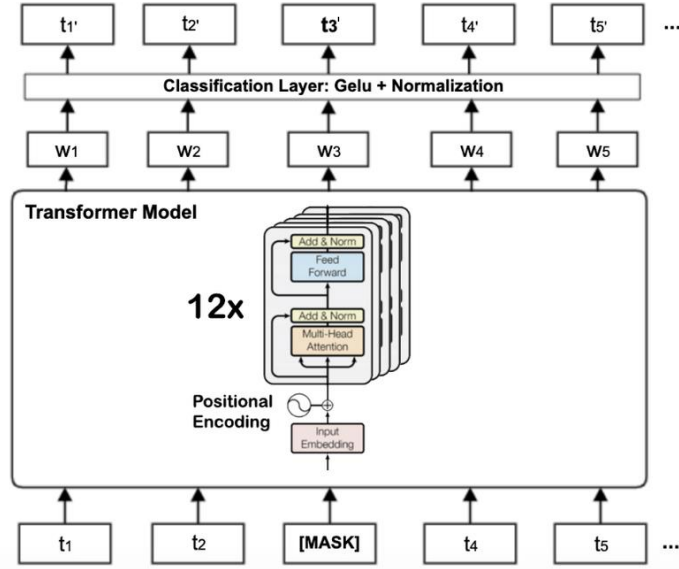
3.2.3. Bidirectional Encoder Representations from Transformers (BERT)

BERT, Google tarafından 2018 sonlarında geliştirilen modern büyük dil modellerinden biridir (Devlin vd., 2019). Kelimelerin bağlamsal anlamlarını öğrenen

ve bu anlamları vektörlerle temsil eden bir modeldir. BERT kelime gömme vektörü, bir kelimenin yalnızca anlamını değil, cümle içindeki bağlamına göre değişen anlamını da yakalamayı amaçlar. Bu özellik, onu geleneksel yöntemlerden (Word2Vec, FastText gibi) ayırır ve daha güçlü bir kelime temsil yöntemi haline getirir. BERT gömme, çevreleyen kelimelerden etkilenen kelime vektörleri üretir. BERT, çeşitli görevler için en son modelleri üretmek üzere ince ayarlanabilen bir dil gösterim modelidir (Ramakrishnan ve Babu, 2023).

BERT, kelimelerin hem solundan hem de sağından gelen bağlamsal bilgiyi aynı anda öğrenir. Örneğin: “Bahçede kırmızı bir gül vardı.” ve “Onun şakasına gül.” şeklinde verilen iki cümledeki ‘gül’ kelimesinin anlamları farklıdır. BERT, bu farkı bağlamdan öğrenir ve her biri için farklı embedding vektörü üretir. Ayrıca BERT, kelimeleri daha küçük birimlere ayırarak (subword tokenization) bu kelimeler için daha esnek temsil imkânı sağlar. Örneğin, ‘anlaşılır’ kelimesi ‘an’, ‘##laşıl’, ‘##ır’ şeklinde alt birimlere bölünebilir. Bu yöntem daha önce görülmeyen kelimeleri eğitilecek modelin anlaması hususunda yardımcı olur. BERT kelime yerleştirmelerinin özelliği, bağlamı anlamalarıdır. Bağlam, bir kelimenin yerleştirilmesinin bir cümlede nasıl kullanıldığına bağlı olarak değişebileceği anlamına gelir. Kelime yerleştirmeleri için diğer yöntemler genellikle bir kelime cümlede nerede görünürse görünsün aynı yerleştirmeyi oluşturur (Analytics Vidhya, 2023).

BERT, öz-dikkat mekanizmalarıyla birlikte ileri beslemeli sinir ağlarını etkin şekilde kullanan bir transformatör mimarisi üzerine inşa edilmiştir. Bu mimari yapı, metin içindeki uzun mesafeli bağımlılıkları yakalama konusunda oldukça etkilidir. Daha eski modeller metni yalnızca tek bir yönde (çoğunlukla soldan sağa) okurken BERT, metnin her iki yönden, yani soldan sağa ve sağdan sola işlenmesini sağlayan çift yönlülük özelliğine sahiptir. Bu sayede BERT, bir cümledeki tüm kelimeleri aynı anda değerlendirerek her bir kelimenin bağlamını derinlemesine analiz edebilir. Geniş kapsamlı bir metin korpusu üzerinde önceden eğitilmiş olması, modelin zengin ve bağlamsal kelime temsilleri öğrenmesini mümkün kılar. Ayrıca, bu ön eğitim sonrası ince ayar yapılabilmesi, BERT’ in farklı doğal dil işleme görevlerine kolaylıkla uyarlanmasını sağlar. Çok katmanlı bir transformatör yapısına sahip olması da karmaşık bağlamsal ilişkileri daha etkili şekilde öğrenebilmesine imkan tanır (Devlin vd., 2019).



Şekil 3.5: BERT modelinin pre-training (öğretim) sürecini özetlemektedir

Kaynak : Nvidia, 2019

Girdi cümlesindeki kelimeler (örneğin t_1, t_2, \dots, t_5) önce gömme (embedding) katmanında sayısal vektörlere dönüştürülür. Ardından, pozisyon bilgileri eklenerek bu vektörler 12 katmanlı bir Transformer mimarisine aktarılır. Bu yapı içinde çok başlı dikkat mekanizması ve ileri beslemeli (feed-forward) katmanlar aracılığıyla, her kelimenin bağlamsal ilişkilerini yansıtan gizil temsiller (W_1, W_2, \dots) elde edilir. Elde edilen bu temsiller, GELU aktivasyon fonksiyonu ve katman normalizasyon süreçlerini içeren bir sınıflandırma katmanında işlenir. Bu süreçte özellikle [MASK] ile gizlenmiş kelimenin (t_3') doğru bir şekilde tahmin edilmesi hedeflenir. Söz konusu yapı, BERT' in dil anlama yetisinin temelini oluşturmaktadır.

3.2.4. TF-IDF (Term Frequency-Inverse Document Frequency, Terim Frekans-Ters Belge Frekansı)

TF-IDF, yani “terim sıklığı ve ters belge sıklığı”, bir kelimenin belirli bir belge içindeki önemini ölçmeye yarayan bir istatistiksel yöntemdir. Genellikle makine öğrenmesi ve bilgi erişimi alanlarında kullanılan bu yöntem, bir doküman kümesindeki (ya da gövdesindeki) sözcük, ifade ya da lemma gibi metin öğelerinin, ilgili belge açısından ne kadar anlam taşıdığını sayısal olarak değerlendirmeye yardımcı olur. TF-IDF, bir terimin benzersizliğini ve özgünlüğünü, bir belgedeki ortaya çıkış sıklığını tüm korpustaki yaygınlığına karşı koyarak değerlendirir. Yüksek TF-IDF' e sahip terimler, belgeyi önemli ölçüde etkiler (Yang ve Long, 2023). TF (terim sıklığı) ve

IDF (ters belge sıklığı) olmak üzere iki bölüme ayrılabilir.

TF, bir terimin (t) bir belge (d) içindeki göreceli sıklığı olarak tanımlanabilir. Terimin belgede geçtiği sayının $f(t, d)$ belgedeki toplam terim sayısına ($\sum f(w, d)$) bölünmesiyle Denklem 3.1'deki gibi hesaplanır. w ise d belgesinde bulunan herhangi bir kelime/terimi ifade eder.

$$TF(t, d) = \frac{f(t,d)}{\sum f(w,d)} \quad (3.1)$$

IDF, bir kelimenin belge koleksiyonunda ne sıklıkta geçtiğini ya da ne kadar seyrek kullanıldığına bakar. Bir kelimenin sıkça geçtiği belgelerin önemi düşürülürken, nadiren geçen kelimelere daha fazla ağırlık verilir. Denklem 3.2'deki verildiği şekliyle hesaplanır. Toplam belge sayısının (N), t terimi içeren belge sayısına ($df(t)$) bölünerek logaritması alınır (Semrush, t.y.).

$$IDF(t, d) = \log\left(\frac{N}{df(t)}\right) \quad (3.2)$$

TF-IDF'in temel mantığı, bir terimin önemi ile belgeler genelinde ne kadar yaygın olduğu arasında ters orantı kurmasına dayanır. TF bir terimin bir belgede ne sıklıkla görüldüğü hakkında bilgi verirken, IDF bir terimin belgeler koleksiyonundaki göreceli nadirliği hakkında bilgi verir. Bu değerleri çarparak nihai TF-IDF değeri elde edilir. TF-IDF için hesaplama formülü aşağıda Denklem 3.3'te verildiği şekilde ifade edilmektedir (Capital One, t.y.).

$$TF - IDF(t, d) = TF(t, d) \times IDF(t, d) \quad (3.3)$$

3.3. Sınıflandırma Yöntemleri

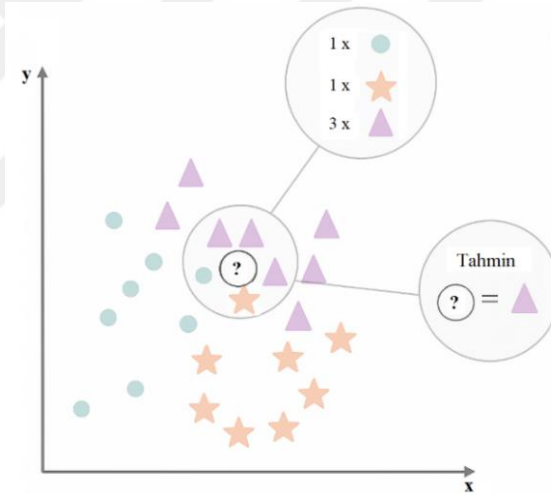
Sınıflandırma yöntemleri olarak en yaygın kullanılan makine öğrenmesi, derin öğrenme ve dönüştürücü tabanlı modeller kullanılmıştır.

3.3.1. Makine Öğrenmesi Algoritmaları

Makine öğrenmesi, verilerdeki karmaşık ilişkileri istatistiksel ve hesaplamalı yöntemlerle modelleyerek gelecekteki gözlemler için tahminler üreten yöntemlerin bütünüdür. Bu çalışmada, en sık kullanılan algoritmalar arasında yer alan Destek Vektör Makineleri, Gauss (Normal) Naif Bayes, k-En Yakın Komşu, XGBoost (Aşırı Gradyan Artırma), Rastgele Orman ve Lojistik Regresyon yöntemleri kullanılmıştır.

a. KNN (K-En Yakın Komşu - K-Nearest Neighbors)

"En basit" denetlenen makine öğrenimi algoritmaları arasında olan KNN bir zamanlar olduğu kadar popüler olmasa da hala yaygın olarak kullanılmaktadır ve daha karmaşık modeller geliştirmeye çalışırken, sınıflandırma projelerinde en azından k-En Yakın Komşu algoritmasını bir tahmin performansı ölçütü olarak değerlendirilmesi tavsiye edilmiştir (Raschka, 2018). KNN, veri noktalarının benzerliklerine dayanarak, hangi gruba ait olduklarını tahmin eder. Sınıflandırma problemlerinde, bir veri noktasına ait sınıf etiketini, komşularının sahip olduğu etiketlerin çoğunluğuna göre belirler. Ancak burada dikkat edilmesi gereken bir ayrım vardır: "Çoğunluk oyu" ifadesi teknik anlamda %50'den fazla oy gerektirirken, bu yalnızca iki sınıfın bulunduğu durumlar için geçerlidir. Eğer sistem birden fazla sınıf içeriyorsa –örneğin beş sınıf varsa– bir sınıfa ait karar verebilmek için %50 oy çoğunluğu gerekmez; %20'nin üzerinde oy alan sınıf etiketi atanabilir (IBM, t.y.). Bu durum Şekil 3.6'da görselleştirilmiştir.



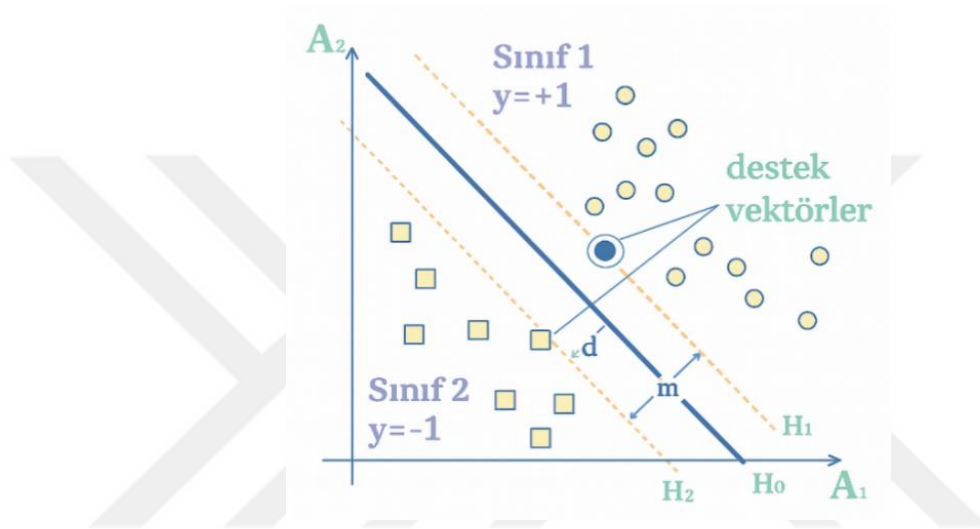
Şekil 3.6 k=5 iken 3-sınıflı bir problem için Knn gösterimi

Kaynak : Raschka, 2018

b. SVM (Support Vector Machine, Destek Vektör Makineleri)

SVM, N boyutlu bir uzayda her sınıf arasındaki mesafeyi en yüksek seviyeye çıkaran optimum bir çizgi veya hiper düzlem bularak verileri sınıflandıran bir gözetimli makine öğrenme algoritmasıdır. Vladimir N. Vapnik ve meslektaşları tarafından 1990'lı yıllarında geliştirilmiş (Vapnik, Golowich, ve Smola, 1996) olan SVM'ler sınıflandırma problemlerinde yaygın olarak kullanılır. Destek Vektör Makineleri (SVM), farklı sınıflara ait veri noktaları arasındaki en yakın mesafeyi temsil eden marjı

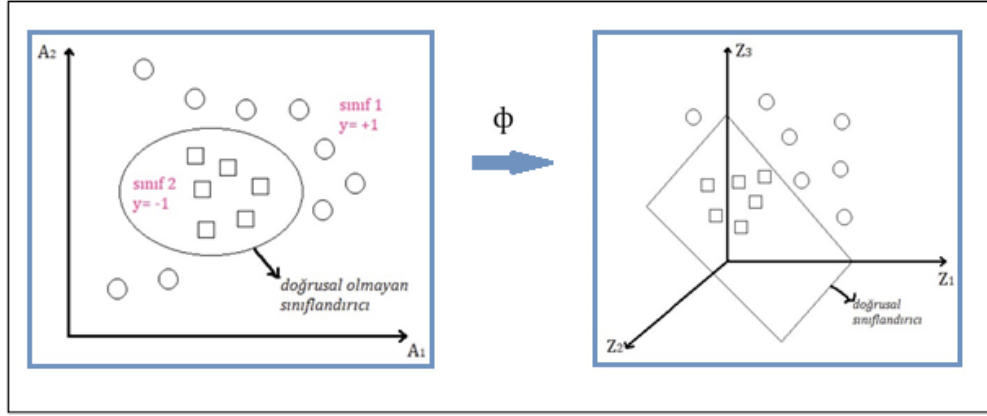
maksimize eden en uygun hiper düzlemi bularak sınıflar arasında ayrım yapar. Veri kümesindeki öznelik sayısı, bu hiper düzlemin iki boyutlu bir ortamda bir doğru mu yoksa daha yüksek boyutlu bir uzayda bir düzlem mi olacağını belirler. Aynı veri kümesi için birden fazla hiper düzlem olabileceğinden, en geniş marjı sağlayan sınır, algoritmanın en iyi ayrımı yapmasına olanak tanır. Bu strateji, modelin yeni veriler üzerinde daha iyi genelleme yapmasına ve daha isabetli tahminlerde bulunmasına katkı sağlar. Marjı belirleyen ve hiper düzleme olabilecek en yakın noktada yer alan veri noktalarına destek vektörleri adı verilir.



Şekil 3.7 Doğrusal ayrılabilme durumu

Kaynak : IBM, t.y.

SVM algoritması, hem doğrusal (Şekil 3.7) hem de doğrusal olmayan sınıflandırma görevlerini işleyebildiği için makine öğreniminde yaygın olarak kullanılır. Ancak, veriler arasında doğrudan bir ayrım yok ise çekirdek fonksiyonları kullanılarak bu veriler daha yüksek boyutlu bir uzaya dönüştürülür ve böylece doğrusal ayrım mümkün hale gelir (Şekil 3.8). Bu yönteme literatürde "çekirdek hilesi" adı verilmektedir (IBM, 2023).



Şekil 3.8: Doğrusal olarak ayırlamama durumu

Kaynak : Liu, 2015

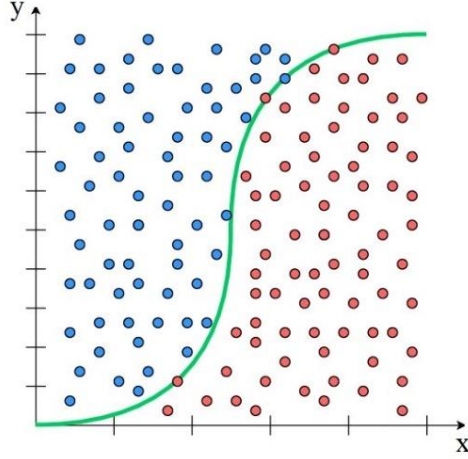
c. Lojistik Regresyon (Logistic Regression)

Lojistik Regresyon, bağımsız belirlenen değişkenler ile bağımlı olarak belirlenen değişkenler arasındaki ilişkiyi modelleyen, ikili (binary) ya da çoklu (multiclass) sınıflandırma problemlerinde kullanılan bir algoritmadır. Bu yöntemin temel amacı, belirli bir olayın hangi sınıfa ait olabileceğine dair olasılığı tahmin etmektir. Elde edilen tahmin değeri her zaman 0 ile 1 arasında bir olasılık skoru olarak ifade edilir. Lineer regresyonda olduğu gibi, bağımsız değişkenlerin doğrusal bir birleşimi kullanılarak hedef değişkenin değeri öngörülmeğe çalışılır. Ancak lineer regresyonun aksine, tahmin edilen değerlerin olasılık olarak yorumlanabilmesi için tahmin değeri sigmoid fonksiyonu kullanılarak Denklem 3.4’ deki gibi 0 ile 1 arasına indirgenir.

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (3.4)$$

$$z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b \quad (3.5)$$

Giriş değişkenlerinin belirli ağırlıklarla çarpılması ve bir sabit değer eklenmesiyle lineer bir model oluşturur. Bu lineer model, Denklem 3.5’ deki gibi ifade edilebilir. Elde edilen z değeri, sigmoid fonksiyonu kullanılarak bir olasılığa dönüştürülür. Bu olasılık değeri, verinin hangi sınıfa ait olacağını belirler. Eğer olasılık 0.5'ten büyükse, model veriyi sınıf 1 olarak sınıflandırır, aksi takdirde sınıf 0 olarak sınıflandırır. Lojistik Regresyon, hızlı bir model olup karmaşık olmayan veriler üzerinde oldukça etkili çalışır (Techtarget, 2024).



Şekil 3.9: 2 Sınıf için lojistik regresyonun temel bir diyagramı görülmektedir.

Kaynak : Hasib vd., 2023

d. GaussianNB (Gaussian Naive Bayes, Gauss (Normal) Naif Bayes)

Bayes sınıflandırması ifadesi, istatistiksel değerlerin koşullu olasılıkları arasındaki bağlantıyı ifade eden bir denklem olan Bayes yöntemine dayalı bir kategorizasyon tekniğini ifade eder (Pushpakumar vd., 2022). Bayes teoremi sayesinde, öncül varsayımlar ile mevcut gözlemler birleştirilerek anlamlı ve etkili çıkarımlar yapmak mümkün hale gelir. Denklem 3.6, Bayes teoremi olarak bilinir. Bu teorem, bir olayın olasılığını, diğer bir olayın gerçekleştiği bilgisi ışığında hesaplamaya yardımcı olur.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3.6)$$

$P(A|B)$: Olay B gerçekleştiğinde olay A'nın gerçekleşme olasılığıdır ki buna koşullu olasılık denir. Yani, B'nin gerçekleştiği durumlarda A'nın olma olasılığını ifade eder.

$P(B|A)$: Olay A gerçekleştiğinde olay B'nin olasılığıdır. Bu da bir başka koşullu olasılık ifadesidir. $P(A)$: henüz B hakkında bir bilgi yokken A'nın gerçekleşme olasılığıdır, $P(B)$: B olayının gerçekleşme olasılığıdır (Naiem, Khedr, Idrees ve Marie (2023)).

Naive Bayes yöntemleri, öznitelikler arasında bağımsızlık varsayımına dayanır. Bu sınıflandırıcılar, bir özellik grubunda herhangi bir nitelik mevcutsa, bunun diğer özelliklerin varlığı veya yokluğu üzerinde herhangi bir etkisi olmadığı kabul eder. Gerçekte bazı öznitelikler birbirine bağlı ya da bir diğerinin varlığına bağlı olsa bile, her bir özellik olasılık hesaplamalarında bağımsız bir katkı sağlar. Sınıflandırma

sürecinde, sınıf etiketleri ilgili özelliklerin etkisiyle belirlenir. Modelin çıkarım gücü geliştikçe, eğitim verileri bu süreci daha da iyileştirmek için yeniden kullanılabilir. Eğitim verilerinin birbirini dışlayan ve eşit büyüklükte alt kümeleri oluşturulur (Pushpakumar vd., 2022).

Gaussian Naive Bayes ise, özellikle sürekli verilerle çalışırken etkinliği ve sadeliğiyle öne çıkan bir makine öğrenme algoritmasıdır. Bu model, Naive Bayes sınıflandırıcıları ailesine aittir ve özelliklerin Gaussian (normal) dağılıma sahip olduğunu varsayar. Bu varsayım hem hesaplamaları kolaylaştırır hem de güvenilir sonuçlar üretmeye yardımcı olur. Gaussian Naive Bayes, Bayes teoremine dayalı olarak, bir olayın gerçekleşme olasılığını, o olayla ilgili mevcut bilgileri kullanarak tahmin eder. Modelin “naive” yani saf olarak adlandırılmasının nedeni, tüm özelliklerin birbirinden bağımsız olduğu varsayımdır; bu da olasılıkların daha basit hesaplanmasına olanak tanır (Kashishdafa, 2024).

e. Rastgele Orman (Random Forest)

Veri bilimindeki sınıflandırma algoritmaları arasında, sınıflandırıcı hiyerarşisinin en üstünde yer alan (Turing, 2022) Random Forest, L. Breiman tarafından 2001 yılında önerilmiş olup hem sınıflandırma hem de regresyon problemlerinde genel amaçlı ve oldukça başarılı bir yöntem olarak kabul görmüştür. Bu yaklaşım, birkaç rastgele karar ağacını bir araya getirerek tahminlerini ortalama alma yoluyla birleştirir ve değişken sayısının gözlem sayısından fazla olduğu durumlarda son derece etkili sonuçlar verir. Ayrıca, geniş ölçekli problemlerde kullanılabilir kadar esnek olan bu yöntem, farklı türdeki özel öğrenme görevlerine rahatlıkla adapte edilebilir ve değişkenlerin önem derecelerine ilişkin anlamlı ölçümler sunar (Biau ve Scornet, 2016).

Algoritma, temelinde “bagging” (bootstrap aggregating) yaklaşımını kullanmaktadır. Bu yöntemde, her karar ağacı farklı bir bootstrap örneği ile eğitilir ve sonuçlar çoğunluk oyu veya ortalama yöntemiyle birleştirilerek modelin varyansı azaltılır. Bu sayede modelin aşırı öğrenmeye karşı direnci artar ve genelleme performansı güçlenir. Bu yöntemde veri seti ve özellikler farklı rastgele alt kümelere ayrılarak çok sayıda karar ağacı oluşturulur. Her bir karar ağacı, verileri sınıflandırma konusunda kendi değerlendirmesini yapan bağımsız bir uzman gibi işlev görür. Nihai tahmin ise her ağacın ürettiği sonuçlar arasında en sık tekrar eden (en çok oyu alan) sınıf seçilerek belirlenir. Ayrıca her bir ağacın düğüm bölme işlemi sırasında tüm özellikler yerine

rastgele seçilmiş bir özellik alt kümesinin kullanılması, ağaçlar arasındaki korelasyonu azaltır. Bu çeşitlilik, toplu kararın doğruluğunu artıran temel etkenlerden biridir. Karar ağaçları gözetimli öğrenmede sıkça kullanılan algoritmalar arasında yer alsa da zaman zaman önyargı ve aşırı öğrenme gibi sorunlara açık olabilirler. Bununla birlikte, rastgele orman algoritmasında birden çok karar ağacının bir araya gelerek oluşturduğu topluluk yapısı, özellikle ağaçlar birbirinden bağımsız olduğunda, daha isabetli tahminler yapılmasını sağlar (Shafi, 2024).

f. XGBoost (Extreme Gradient Boosting, Aşırı Gradyan Artırma)

Büyük veri kümelerinde hız, kullanım kolaylığı ve performans için tasarlanmış olan XGBoost, parametrelerin optimizasyonunu veya ayarlanmasını gerektirmez. Yüksek yürütme hızı ve model performansı sebebiyle XGBoost kullanıldığında veri kümesinin boyutuyla ilgili hiçbir kısıtlama yoktur, bu nedenle diğer algoritmalarla mümkün olabilecek olandan daha büyük veri kümeleri ile çalışabilir. Ayrıca, diğer modellere kıyasla daha üstün performans sergileyebilen modellerin oluşturulmasına olanak tanıdığı için de dikkate değerdir. XGBoost algoritması; rastgele orman (RF), gradyan artırma makineleri ve gradyan artırmalı karar ağaçları gibi yöntemlerle karşılaştırıldığında hem işlem süresi hem de model başarımı açısından genellikle daha iyi sonuçlar vermektedir. "Extreme Gradient Boosting" ifadesinin kısaltması olan XGBoost, birden fazla zayıf tahminleyicinin çıktısını birleştirerek daha güçlü bir sonuç üreten topluluk öğrenme (ensemble learning) yöntemidir. Özellikle büyük veri setleriyle çalışma kapasitesi ve yüksek doğruluk oranı sayesinde, sınıflandırma ve regresyon gibi pek çok makine öğrenmesi uygulamasında yaygın şekilde tercih edilmektedir (Chen ve Guestrin, 2016).

3.3.2. Derin Öğrenme Modelleri

Verilerin farklı soyutlama düzeylerinde temsillerini otomatik olarak öğrenebilen ve çok sayıda işlem katmanından oluşan hesaplama modellerini ifade eder. Bu yaklaşımlar, konuşma-nesne tanıma, nesne tespiti ve ilaç keşfi araştırmalar gibi bir çok çalışma alanında önemli ilerlemeler sağlamıştır. Büyük ölçekli veri kümelerindeki karmaşık yapıları ortaya çıkarabilmek için derin öğrenme modelleri, her katmanda önceki temsilden yeni ve daha üst düzey temsillerin elde edilmesini sağlayan geri yayılım (*backpropagation*) algoritmasını kullanır. Uygulama alanına göre farklı mimariler tercih edilmektedir: Görüntü, video, ses ve konuşma işleme gibi alanlarda

derin evriřimli sinir ađları (*Convolutional Neural Networks – CNN*) öne çıkarken; metin ve konuşma gibi sıralı verilerin işlenmesinde tekrarlayan sinir ađları (*Recurrent Neural Networks – RNN*) yaygın olarak kullanılmaktadır (LeCun, Bengio ve Hinton, 2015).

a. Evriřimli Sinir Ađı (CNN - Convolutional Neural Network)

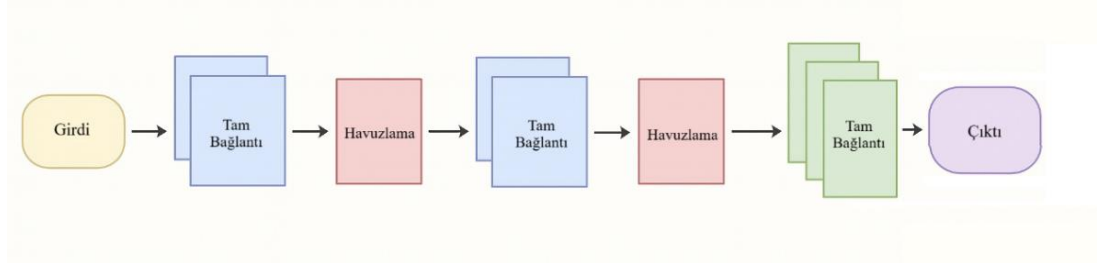
Evriřimli sinir ađı (CNN), bir makine öğrenme modeli kategorisidir. CNN' ler, nesne tespiti konusunda oldukça başarılı oldukları için görüntü tanıma ve bilgisayarla görme alanındaki birçok görevde yaygın şekilde kullanılmaktadır. Görüntü işleme amacıyla geliştirilmiş olsalar da bu mimariler ses verileri veya diđer sinyal tabanlı verilerle çalışacak şekilde de uyarlanabilir. CNN mimarisinin katmanları genellikle 3 temel katmandan oluşur; convolutional (evriřim), pooling (havuzlama) ve fully connected (tam bađlantılı) katmanlarıdır. Girdi verisi bu katmanlardan geçtikçe, ađın öğrenme kapasitesi ve desen tanıma becerisi giderek artar. Her katman, giriş görüntüsünün farklı özelliklerini analiz eder. Modelin çözmeye çalıştığı görevin karmařıklığına bađlı olarak, CNN mimarisi onlarca, hatta bazı durumlarda yüzlerce ya da binlerce katmandan oluşabilir. Bu katmanların her biri, bir önceki katmanın çıktısını temel alarak daha karmařık desenleri algılamaya çalışır (Gillis, Craig & Awati, 2024).

Evriřim katmanı, ađın giriş görüntüsünden önemli özellikleri çıkarmaya başladığı ilk adımdır. Görüntü, filtre (çekirdek olarak da bilinir) adı verilen küçük bir matrisle matematiksel olarak işlenir ve bu işlem sayesinde görüntüdeki temel desenler belirlenir. Evriřim katmanından sonra uygulanan doğrusal olmayan bir aktivasyon fonksiyonu olan ReLU (Rectified Linear Unit - Doğrultulmuş Doğrusal Birim) uygulanır. Negatif deđerleri sıfıra çevirir ve böylece ađın karmařık (dođrusal olmayan) verileri daha etkili şekilde öğrenmesini sađlar.

Havuzlama katmanı, özellik haritalarının boyutunu küçültürken en önemli bilgileri korur. Örneđin, maksimum havuzlama yöntemi, belirli bir bölgedeki en yüksek deđeri seçerek hem veri miktarını azaltır hem de ađın daha verimli çalışmasını sađlar.

Tam bađlantılı katmanı, evriřim ve havuzlama katmanlarından sonra elde edilen özellik haritaları, tek bir vektör haline getirilir. Bu vektör, tam bađlantılı katmanlara aktarılır ve burada öğrenilen özellikler birleřtirilerek son tahmin (sınıflandırma gibi) üretilir (Nerdnodes2023, t.y.). Derin öğrenme modelleri, özellikle CNN'ler, yüksek

karmaşıklık kapasiteleri ve büyük ölçekli verilerdeki ayrıntılı örüntüleri öğrenme yetenekleri nedeniyle aşırı uyuma karşı hassastır (DataCamp, 2023).



Şekil 3.10 Cnn modeli mimarisi

Kaynak: Lopez-Martin vd., 2017

b. LSTM (Uzun Kısa Süreli Bellek - Long Short Term Memory)

Geleneksel yinelenen sinir ağlarında (RNN – Recurrent Neural Network) sıkça karşılaşılan gradyan kaybı (vanishing gradient) sorununu aşmak amacıyla özel olarak geliştirilmiş bir RNN türüdür (Khan vd., 2023). LSTM'ler, sıralı verilerle çalışırken önceki zaman adımlarından gelen bilgileri hafızada tutabilme yeteneğine sahiptir ve bu sayede, gelecekteki adımların daha doğru şekilde tahmin edilmesine olanak sağlar. Bu özellik onları uzun vadeli bağımlılıklar içeren görevler için oldukça uygun hale getirir LSTM'ler, bellek hücresine giren ve çıkan bilgi akışını düzenlemek için bir dizi kapı kullanır. Sonuç olarak, LSTM'ler, daha önceki zaman adımlarından gelen bilgilerin daha sonraki zaman adımlarını tahmin etmek için önemli olduğu uzun vadeli bağımlılıkları öğrenme konusunda uzmandır. 1997 yılında Hochreiter ve Schmidhuber tarafından tanıtıldıklarından bu yana konuşma tanımadan makine arıza tahminine kadar çeşitli sıralı veri görevlerinde yaygın olarak kullanılmış olup oldukça etkili hale gelmiştir. Çeşitli görevlerde diğer RNN türlerinden ve geleneksel ML yöntemlerinden daha iyi performans gösterdikleri gösterilmiştir. LSTM'lerin uzun vadeli bağımlılıkları yakalayabilme ve sıralı verileri etkili biçimde modelleyebilme kabiliyeti, onları doğal dil işleme alanındaki birçok uygulama için vazgeçilmez hâle getirmiştir. Örneğin, bir metnin duygusal içeriğini analiz etmek amacıyla, kelime dizisinin tamamını göz önünde bulundurarak duygu sınıflandırması yapmak için kullanılabilirler. Ayrıca metni, haber makaleleri ve film eleştirileri gibi farklı kategorilere ayırmak için de kullanılabilir. Bununla beraber LSTM'ler çoğunlukla doğal sahne görüntüsü ve video karelerinde varlık tanıma, alaycılık tanımlama ve senaryo tanımlama adı verilen soru cevap sistemlerinde kullanılmaktadır. LSTM'lerin

metinlerdeki bağlamsal ilişkileri algılayıp analiz edebilme yetisi, onları doğal dil işleme (NLP) alanındaki gelişmelerin önemli yapı taşlarından biri hâline getirmiştir (Al-Selwi et al., 2024).

c. Kapılı Tekrarlayan Birim (GRU - Gated Recurrent Unit)

Geçitli Yinelemeli Birim (GRU), konuşma, metin veya sensör sinyalleri gibi ardışık girdileri işlemek için kullanılan bir tür yapay sinir ağı yapısıdır. Zaman adımları boyunca önemli ayrıntıları hatırlamasına ve bunları iletip iletmemeye veya atmaya karar vermesine imkan tanıyan bellek yeteneklerine sahiptir. GRU'lar, girdi öğeleri arasındaki uzun menzilli bağımlılıkları analiz etmede etkili olabilir ve bu da onları öngörücü bakım, kalite kontrolü ve anormallik tespiti gibi çeşitli endüstriyel uygulamalar için yararlı hale getirir (PlainEnglish.io, 2024). LSTM'lere benzer ancak basitleştirilmiş bir mimariye sahip, sıralı verilerdeki bağımlılıkları yakalamak için tasarlanmıştır. GRU yapısında güncelleme ve sıfırlama kapıları olmak üzere iki temel "kapısı (gate)" vardır. Hangi bilgilerin gizli duruma aktarılacağını belirleyen güncelleme kapısıdır (update gate). Sıfırlama kapısı (reset gate) ise yeni girdi ile önceki gizli bilgiyi ne kadar birleştireceğini belirler. LSTM'e göre (giriş, çıkış, unutma kapısı) daha az kapıya sahip olduğu için GRU'nun yapısı daha basit ve hesaplaması daha hızlıdır. LSTM'de "bellek hücresi (cell state)" ve "gizli durum (hidden state)" ayrı tutulur. GRU'da bu ikisi tek bir yapı içinde birleştirilmiştir ki bu da daha az parametre demektir. Daha hızlı öğrenme ve daha az bellek kullanımı sağlar. Ancak çok karmaşık dizilerde LSTM'ler kadar iyi performans göstermeyebilir (Idrees, t.y.).

3.3.3. Dönüştürücü (Transformer) Tabanlı Modeller

Dönüştürücü (Transformer) tabanlı modeller, dikkat (attention) mekanizmasını kullanarak değişken uzunluktaki girdi dizileri üzerinde yüksek düzeyde paralel işlem gerçekleştirebilen derin öğrenme mimarileridir. Başlangıçta makine çevirisi amacıyla geliştirilen bu mimari, encoder ve decoder olmak üzere iki temel bileşenden oluşmaktadır. Her bir bileşen, çoklu başlık içeren öz-dikkat (multi-head self-attention) mekanizmalarını ve konuma duyarlı ileri beslemeli katmanları (position-wise feed-forward layers) bünyesinde barındırır. Tekrarlayan veya evrimsel yapılara başvurmaksızın tamamen dikkat mekanizmasına dayanması, uzun menzilli bağımlılıkların etkili biçimde modellenmesini ve hesaplama sürecinin yüksek düzeyde paralelleştirilebilmesini mümkün kılar. Esnek ve genelleştirilebilir bu yapı, doğal dil

işleme başta olmak üzere görsel ve işitsel veri işleme gibi farklı alanlarda geniş çapta uygulama alanı bulmaktadır (Lin, Wang, Liu ve Qiu, 2022). ChatGPT-2 ve BERTurk gibi modeller de farklı amaç ve veri kümeleri için uyarlanmış Transformer tabanlı mimarilerden yararlanmaktadır.

d. BERTurk

Stefan Schweter, Türkçe için ilk geniş çaplı BERT tabanlı önceden eğitilmiş modellerden olan BERTurk-cased ve BERTurk-uncased'i Hugging Face üzerinde yayımlamıştır. BERTurk, Transformer mimarisi üzerine kuruludur ve bu mimaride metin girişini okumak için encoder katmanı bulunur. BERTurk modelinde enkodere verilen giriş, önce sayısal vektörlere dönüştürülen bir dizi tokenden oluşur ve bu vektörler sinir ağı tarafından işlenir (Savci ve Das, 2023). BERTurk , Türkçe diline yönelik topluluk odaklı bir BERT modelidir. Ön eğitimi büyük bir Türkçe korpusu aracılığıyla yapılır. Türkçe bir metin içindeki özel isimleri (kişiler, yerler, kuruluşlar, tarih, para birimi vb.) tanıma ve etiketleme işlemi olan NER konusunda resmi BERT modelinden daha iyi performans gösterir (Abdullahyayik, t.y.). BERTurk, BERT-base mimarisini temel alır ve 12 Transformer katmanına (enkoder) sahiptir. Gizli Katman Boyutu: Her katmanda 768 boyutlu gizli vektörler kullanılır. Kendine Dikkat Mekanizması: Her katmanda 12 adet çok başlı (multi-head) kendine dikkat (self-attention) mekanizması bulunur. Kelime Dağarcığı: Model, 128.000 alt kelime (subword) içeren bir kelime dağarcığı ile eğitilmiştir (Toprakucar, t.y.). Modelin mevcut versiyonu, Türkçe OSCAR korpusunun filtrelenmiş ve cümle bölümlendirilmiş hali, yakın zamanda yayınlanmış bir Wikipedia dökümü, çeşitli OPUS korpusları ve Kemal Oflazer tarafından sağlanan özel bir korpus üzerinde eğitilmiştir. Son eğitim gövdesinin boyutu 35 GB olup, 4.404.976.662 adet token bulunmaktadır (Stefan-it, t.y.).

e. GPT-2 Tabanlı Sınıflandırma

GPT-2 (Generative Pre-trained Transformer 2), OpenAI tarafından geliştirilmiş, Transformer mimarisine dayanan, büyük ölçekli bir dil modelidir. Önceden geniş bir metin korpusu üzerinde eğitilmiş bir modeldir. Bu sayede birçok NLP görevine “fine-tuning” (ince ayar) ile kolayca uyarlanabilir (OpenAI, 2019). GPT-2, bir kelime dizisinde sıradaki kelimeyi tahmin edebilmek için 40 GB’lık büyük bir veri kümesiyle önceden eğitilmiştir. Bu yöntem sayesinde model, özellikle metin üretme gibi

görevlerde, hiçbir ek örnek verilmeden bile birçok farklı görevi başarıyla yerine getirebilmektedir. Normalde GPT-2 bir metin üretici modeldir; yani “verilen kısmı tamamla” tarzında işler. Ancak sınıflandırma gibi görevler için GPT-2 modelinin üzerine küçük bir sınıflandırma başlığı (classifier head) ekleyerek, modelin metin sınıflandırma görevlerine uyarlanmasını sağlar (Hugging Face, t.y.). GPT-2, yalnızca dekoder yapısını kullanan bir modeldir ve bu nedenle özellikle metin üretme (text generation) görevlerinde etkilidir. Buna karşılık, BERT yalnızca enkoder bileşenini kullanır ve anlama odaklı görevlerde öne çıkar. İki model arasındaki bu fark, Transformer mimarisindeki enkoder ve dekoder yapılarının, modelin görevine göre nasıl tercih edildiğini ortaya koyar. GPT-2, bir cümleyi üretirken kelime kelime (token token) ilerler; her adımda sıradaki kelimeyi tahmin eder. Örneğin, “Bugün hava çok...” ifadesinden sonra, “güzel” mi yoksa “soğuk” mu gibi bir kelime geleceğini tahmin etmeye çalışır. Bu yapı, yalnızca önceki kelimeleri dikkate alan auto-regressive (otomatik regresyon) bir yaklaşımdır ve geleneksel dil modelleri (örneğin RNN’ler) ile benzerlik gösterir. Öte yandan, BERT auto-regressive bir model değildir. Bu, model tasarımında yapılan bilinçli bir tercihtir: BERT, metin üretme kabiliyetinden feragat etmiş; bunun karşılığında bir kelimeyi tahmin ederken hem önceki hem de sonraki bağlamı aynı anda değerlendirme yeteneği kazanmıştır. Bu çift yönlü (bidirectional) yaklaşım, bağlamsal anlayışı derinleştirir ve anlama görevlerinde yüksek performans sağlar (Yal, t.y.).

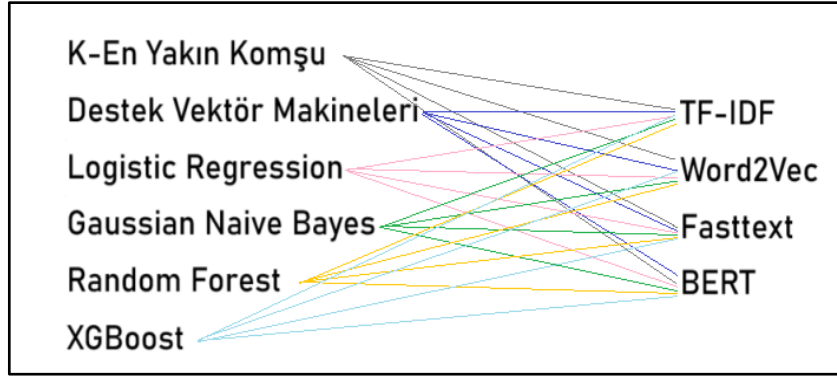
DÖRDÜNCÜ BÖLÜM

DENEYSEL ÇALIŞMALAR

Bu tez çalışması içerisinde Türkçe akademik makale özetleri ve yapay özetlerle oluşturmuş olduğumuz üç farklı veri seti üzerinde, birkaç farklı kelime gömme yöntemi kullanılarak metin vektörleri oluşturulmuş, farklı derin öğrenme ve büyük dil modelleri kelime vektörleri üzerinde eğitilerek birbirinden farklı testler gerçekleştirilmiş ve model performansları farklı metrikler aracılığıyla karşılaştırılmıştır.

4.1. Kelime Gömme Yöntemlerinin Belirlenmesi (Embedding)

Çalışmamızın ilk kısmında v0 veri setimize K-En Yakın Komşu (KNN), Gaussian Naive Bayes (GaussianNB), Destek Vektör Makineleri (SVM), Lojistik Regresyon (LR), Rastgele Orman (RF) ve XGBoost makine öğrenme modelleri uygulanmıştır. Her bir model sırayla TF-IDF, Word2Vec, FastText ve BERT kelime gömme (embedding) yöntemleriyle birleştirilerek sınıflandırma işlemi gerçekleştirilmiştir (Şekil. 4.1).

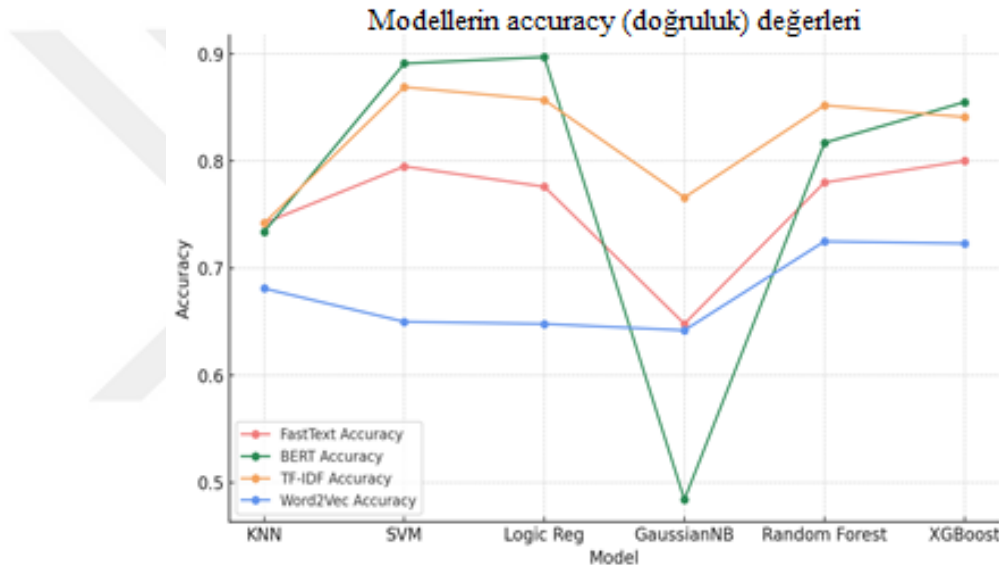


Şekil 4.1: Her bir makine öğrenmesi algoritması sırasıyla tüm embedding yöntemleriyle birlikte çalıştırılmıştır

Her bir makine öğrenmesi algoritmasının veri seti üzerindeki performansı karşılaştırılmış olmakla beraber embedding yöntemlerinin bu performanslara olan etkisi de gözlemlenmiştir. Dolayısıyla embedding yöntemlerinin de başarımı ortaya çıkmıştır. Doğruluk, çalışma zamanı ve f1-skoru gibi metriklere bahsedilen makine öğrenmesi ve embedding yöntemlerin başarımı ölçmek için başvurulmuştur.

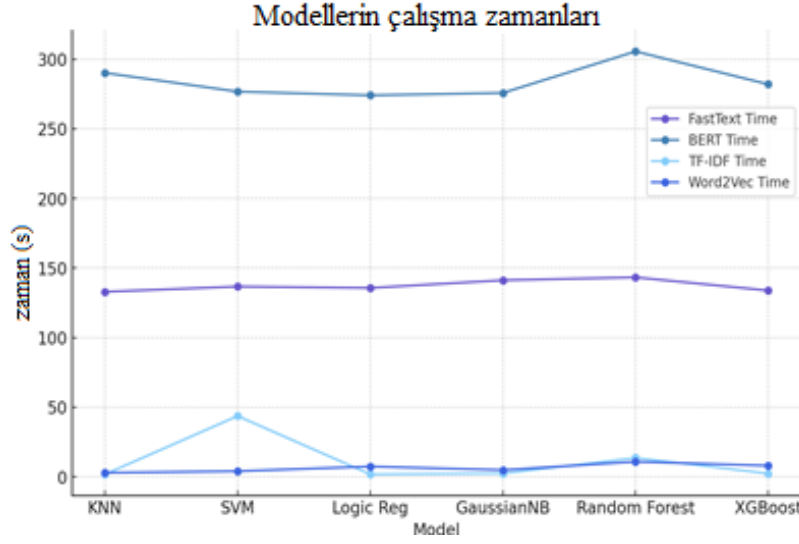
Şekil 4.2 incelendiğinde, BERT tabanlı öznelik çıkarım yönteminin özellikle SVM

ve Lojistik Regresyon algoritmalarıyla birlikte en yüksek doğruluk oranlarına (%89 – %90 aralığında) ulaştığı görülmektedir. Buna karşın, Gaussian Naive Bayes algoritması ile kullanıldığında performans ciddi biçimde düşmekte, başarı oranı %50'nin altına inmektedir. Bu durum, BERT'ten elde edilen vektörlerin dağılımının Gaussian varsayımlarını karşılamadığına işaret etmektedir. Öte yandan, TF-IDF yöntemi, genellikle kararlı ve yüksek doğruluk değerleri sunmuş; en iyi performansını yine SVM ile çalıştırıldığında göstermiştir. FastText genellikle %75 – %80 bandında doğruluk sağlamış, ancak GaussianNB ile birlikte performansı belirgin şekilde düşmüştür. Word2Vec ise doğruluk açısından en düşük sonuçları veren yöntem olmuş, hiçbir modelde %70'in üzerine çıkamadığı gözlemlenmiştir.



Şekil 4.2: Gömme yöntemlerinin makine öğrenmesi algoritmalarıyla birlikte çalıştırıldıklarında elde edilen doğruluk değerleri

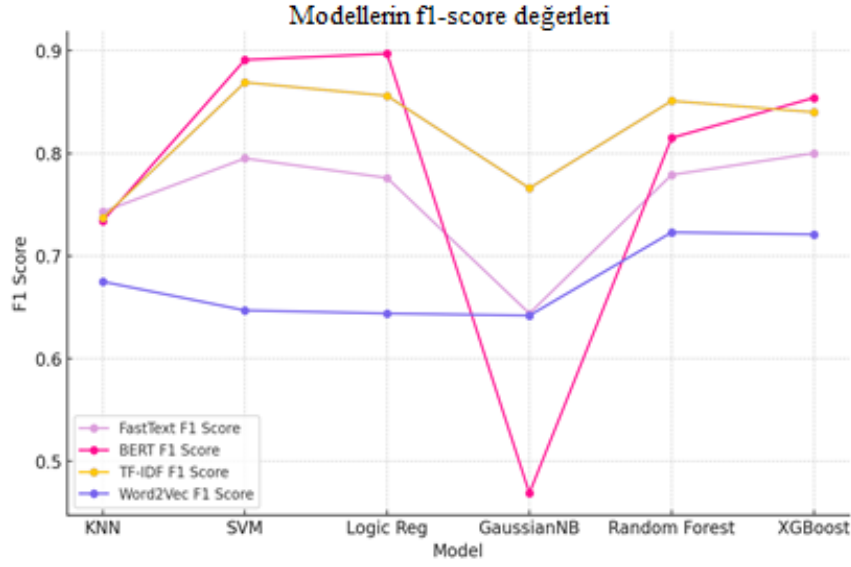
Şekil 4.3'te sunulan çalışma süreleri grafiği, farklı kelime gömme yöntemlerinin sınıflandırma algoritmaları üzerindeki zaman maliyetlerini ortaya koymaktadır. En yüksek zaman maliyeti BERT yönteminde ortaya çıkmış olup, tüm modellerde yaklaşık 280 - 300 saniyelik bir süreye ulaşılmıştır. Bu, BERT'in açık ara en maliyetli yöntem olduğunu göstermektedir. FastText ise ortalama 135 - 145 saniyelik daha makul ve sabit bir süre sunmaktadır. Pek çok modelde çalışma süresi 5 ile 20 saniye aralığında kalan TF-IDF performans açısından en verimli olan yöntemlerden olmuştur. En kısa süreyle çalışan yöntem Word2Vec olsa da düşük doğruluk değerleri göz önünde bulundurulduğunda, hız/performans dengesi dikkatle değerlendirilmelidir.



Şekil 4.3: Gömme yöntemlerinin makine öğrenmesi algoritmalarıyla beraber çalıştırıldığında geçen sürelerin karşılaştırılması

Şekil 4.4’ de yer alan f1-skoru grafiğinde görüldüğü gibi BERT kelime gömme yöntemi SVM ve Lojistik Regresyon algoritmalarıyla birlikte en yüksek f1-skoru değerlerine (~ 0.90) ulaşmıştır. Ancak BERT GaussianNB ile kullanıldığında bu değerlerde ciddi bir düşüş yaşanmıştır. Elde edilen sonuç oluşturulan bu kombinasyonun sınıflandırma başarısının oldukça zayıf olduğunu göstermektedir. TF-IDF yöntemi genelde istikrarlı bir performans sergilemiş olup, SVM ile çalıştırıldığında yaklaşık 0.87 değerinde bir f1-skora ulaşmıştır. FastText ise TF-IDF’ e benzer sonuçlar vermekle birlikte, f1-skoru değerleri biraz daha düşük kalmıştır. Word2Vec, f1-skoru açısından da en zayıf sonucu veren yöntem olmuştur.

Elde edilen süreye ilişkin bulgular, yöntemlerin hesaplama karmaşıklıklarıyla doğrudan bağlantılıdır. BERT, çok katmanlı ve parametre bakımından yoğun Transformer mimarisi nedeniyle öznetelik çıkarımı aşamasında en yüksek zaman maliyetine (~280 - 300 sn) sahiptir. FastText, Word2Vec temelli bir yaklaşım olmakla birlikte karakter n-gram yapısı nedeniyle ek işlem adımları gerektirir; bu nedenle BERT’ten daha hızlı, ancak TF-IDF’ten daha yavaş çalışmaktadır (~135 - 145 sn). TF-IDF, yalnızca frekans temelli matris işlemlerine dayandığı için oldukça kısa sürede sonuç üretmektedir (5 - 20 sn). Word2Vec ise önceden eğitilmiş vektörlerin doğrudan kullanılabilmesi sayesinde en hızlı çalışan yöntem olmuşsa da bağlamsal bilgi kapasitesinin sınırlı olması, doğruluk en düşük seviyede kalmasına sebep olmuştur.



Şekil 4.4: Gömme yöntemlerinin makine öğrenmesi modelleri ile çalıştırıldıklarında elde edilen f1-skoru değerleri

Elde edilen bulgular, farklı kelime gömme ve sınıflandırma yaklaşımlarının, veri setinin dilsel ve anlamsal özelliklerini yakalama biçimlerinin, model performansında belirleyici bir rol oynadığını ortaya koymaktadır. BERT tabanlı öznetelik çıkarım yöntemi, kelimeleri bağlam içinde temsil etme yeteneği sayesinde özellikle SVM ve Lojistik Regresyon algoritmalarıyla birlikte yüksek doğruluk ve f1-skoru sağlamıştır. Buna karşılık, Gaussian Naive Bayes algoritmasının normal dağılım varsayımı, BERT'in yüksek boyutlu ve normal dağılmayan vektörleriyle uyumsuzluk göstermiş, bu nedenle performans kaybı yaşanmıştır. TF-IDF yöntemi bağlam bilgisini içermemesine rağmen istikrarlı sonuçlar üretmiş; FastText, morfolojik açıdan sunduğu avantajlara karşın, derin bağlamsal ilişkileri BERT kadar başarılı şekilde yakalayamamıştır. Word2Vec ise sabit bağlam penceresi ve tek yönlü yapısından ötürü en düşük performansı göstermiştir. Bu değerlendirmeler, çalışmamız kapsamında yapılan deneysel gözlemlerden elde edilen çıkarımlara dayanmaktadır.

Tüm bu değerler incelendiğinde en başarılı kelime gömme yönteminin BERT olduğu görülmektedir. Doğruluk ve f1-skoru açısından en yüksek performansı göstererek, sınıflandırma problemi için açık farkla en başarılı gömme/vektörleme yöntemi olmuştur. Zaman açısından maliyetli olsa da doğruluk ve sınıflandırma başarısı açısından tercih sebebidir. Sonuç olarak, insan ve yapay zekâ metinlerinin ayırımında esas veri seti üzerinde kullanılacak temel gömme yöntemi BERT olmuş ve bu yöntem, ileri düzey sınıflandırma modelleriyle entegre şekilde kullanılmak üzere seçilmiştir.

4.2. Modeller ile Sınıflandırma

v1 veri setini sınıflandırmak için CNN, LSTM, GRU, GPT-2 ve BERTurk gibi modeller kullanılmıştır. Modellerin doğrulukları grafikler ve karmaşıklık matrisleri aracılığıyla görselleştirilmiş, ayrıca her model için kesinlik (kesinlik), duyarlılık (duyarlılık), f1-skoru ve doğruluk (accuracy) gibi performans metrikleri hesaplanmıştır. Eğitim süresi, bellek kullanımı (RAM ve GPU) ve işlemci kullanımı gibi kaynak tüketim bilgileri de her model için ayrı olarak değerlendirilmiş olup sırasıyla Tablo 4.1, Tablo 4.6, Tablo 4.10, Tablo 4.15 ve Tablo 4.20'de gösterilmiştir.

4.2.1. CNN Modeli

CNN modeli 5, 10, 16, 20 boyutlarındaki döngüler (epoch) ile erken durdurma (early stopping) yöntemi de eklenerek eğitilmiştir. Şekil 4.5.a, Şekil 4.5.b, Şekil 4.7.a, Şekil 4.7.b, Şekil 4.9.a, Şekil 4.9.b, Şekil 4.11.a ve Şekil 4.11.b de sırasıyla 5, 10, 16 ve 20 epoch değerleri için doğruluk - loss grafikleri yer almaktadır. Tablo 4.2, Tablo 4.3, Tablo 4.4, ve Tablo 4.5'te aynı sıradaki epoch değerleri için doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri yer almaktadır. Ayrıca Şekil 4.6, Şekil 4.8, Şekil 4.10 ve Şekil 4.12' de aynı sıradaki 4 farklı epoch değeri için karmaşıklık matrisleri yer almaktadır. Modelin Colab Notebook ortamında çalıştırıldığındaki kaynak tüketimi Tablo 4.1'de gösterilmiştir.

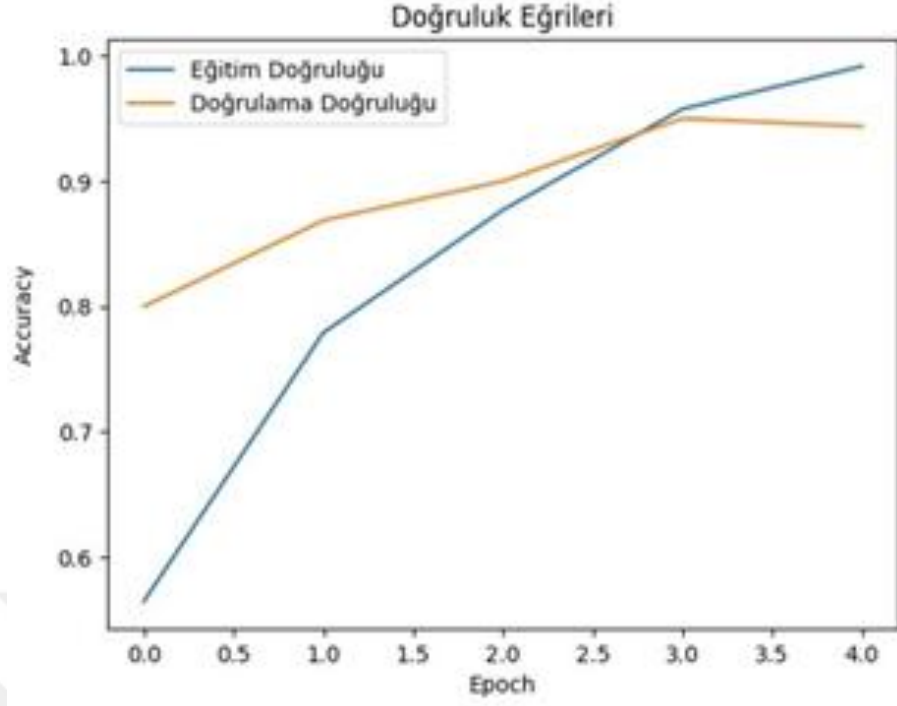
Modelin kaynak kullanımı Tablo 4.1'deki gibidir. Colab Notebook ortamında CNN modeli 53 saniyede eğitilmiş, maksimum GPU bellek kullanımı 0.15 GB, bellek rezervasyonu ise 0.02 GB olarak ölçülmüştür. RAM kullanım oranı %3,6 (2.23/83.48 GB), CPU kullanım oranı ise %10,9'dur. Bu sonuçlar, modelin eğitim süresi ve donanım kullanımı açısından yüksek verimliliğe sahip olduğunu göstermektedir.

Tablo 4.1: Colab Notebook ortamında çalıştırılan CNN modelinin kaynak kullanımı

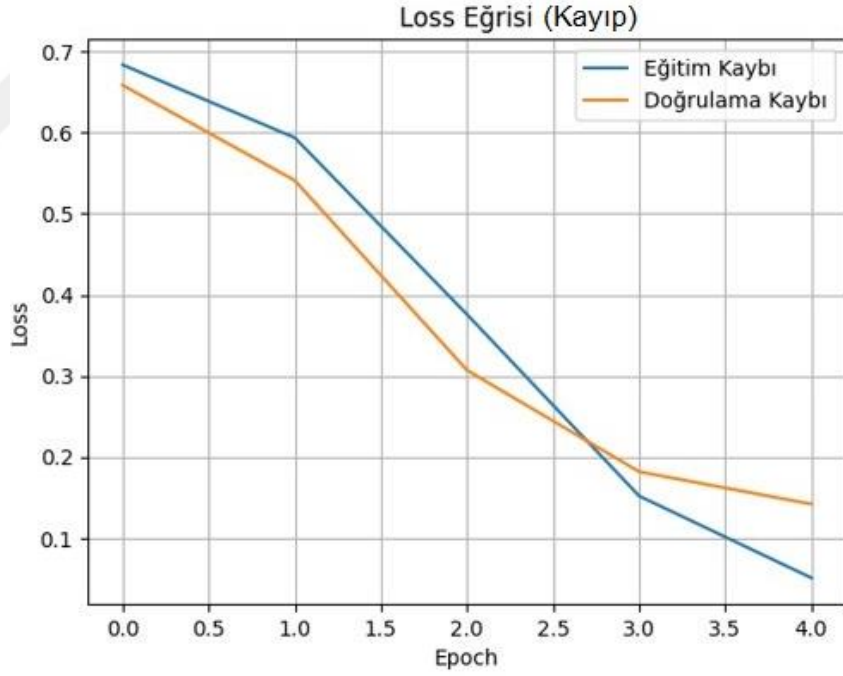
CNN Kaynak kullanımı	
Eğitim Süresi:	53 saniye
Maksimum GPU bellek kullanımı (allocated):	0.15 GB
Maksimum GPU bellek rezervasyonu (reserved):	0.02 GB
RAM Kullanımı:	3.6%
Toplam RAM:	83.48 GB
Kullanılan RAM:	2.23 GB
CPU Kullanımı:	10.9%

Şekil 4.5.a incelendiğinde, eğitim doğruluğunun (mavi çizgi) ilk epoch'tan itibaren sürekli artış göstererek 5. epoch sonunda %99 seviyesine ulaştığı görülmektedir. Doğrulama doğruluğu (turuncu çizgi) ise başlangıçta yaklaşık %95 düzeyinde olup, sonraki epoch'larda hafif bir düşüş eğilimi sergilemiştir. Bu durum, modelin son epoch'larda eğitim verisine daha fazla uyum sağlamasıyla ilişkili hafif bir overfitting belirtisi olarak değerlendirilebilir.

Şekil 4.5.b'de, eğitim kaybının (mavi çizgi) her epoch'ta düzenli biçimde azalarak 5. epoch sonunda neredeyse sifıra yaklaştığı gözlenmektedir. Doğrulama kaybı (turuncu çizgi) da benzer bir düşüş eğilimi göstermekte, ancak 3. epoch'tan itibaren azalma hızının yavaşladığı görülmektedir. Eğitim ve doğrulama kayıpları arasındaki farkın düşük olması, belirgin bir aşırı uyumun oluşmadığını düşündürmektedir.



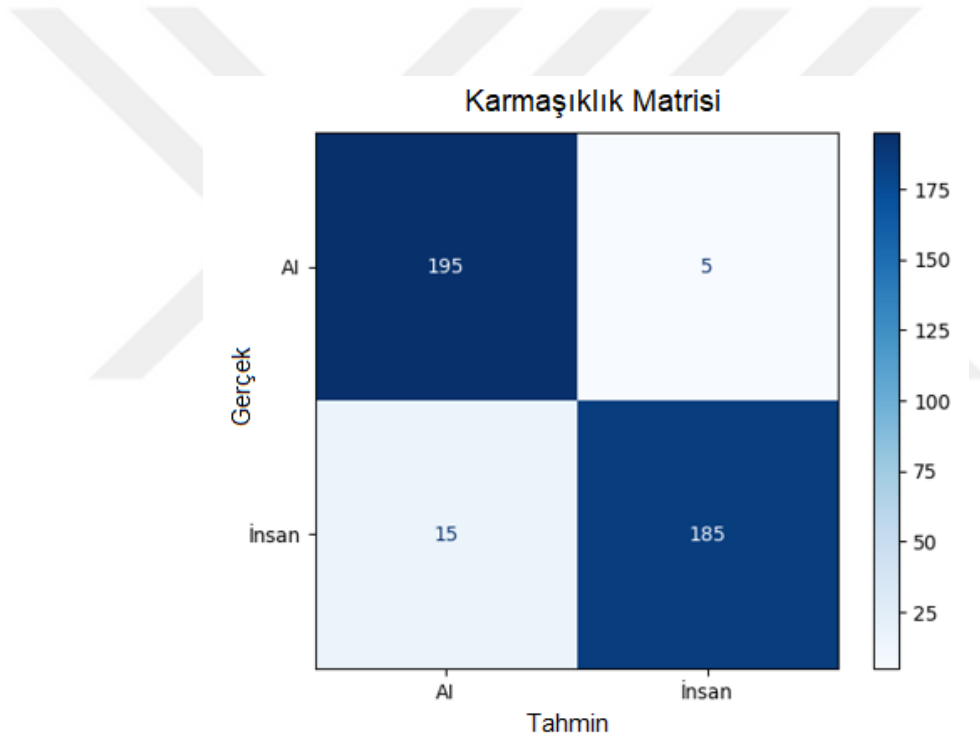
Şekil 4.5.a: epoch = 5 için CNN modelinin accuracy (doğruluk) değeri grafiği



Şekil 4.5.b: epoch = 5 için CNN modelinin kayıp (loss) değeri grafiği

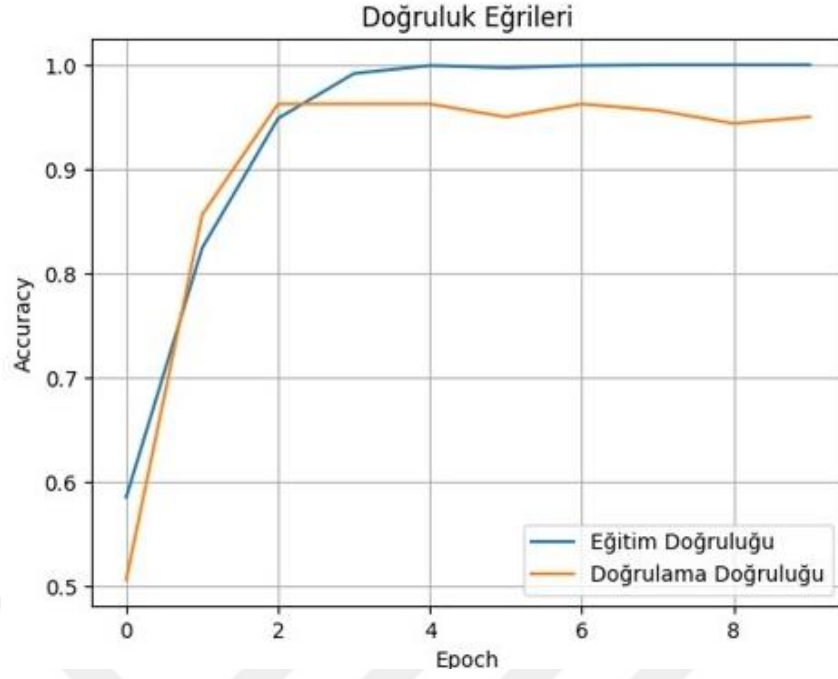
Tablo 4.2: epoch = 5 değeri için CNN modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

CNN modeli epoch=5 için doğruluk: 0.95				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.93	0.97	0.95	200
insan	0.97	0.93	0.95	200
doğruluk			0.95	400
makro ort.	0.95	0.95	0.95	400
ağırlıklı ort.	0.95	0.95	0.95	400

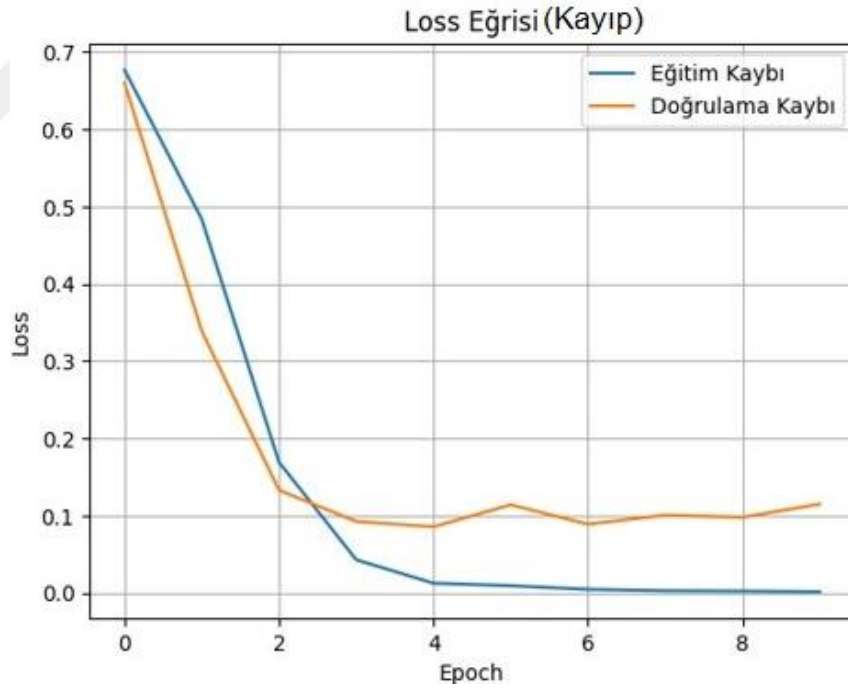


Şekil 4.6: CNN modelinin epoch = 5 için karmaşıklık matrisi

Şekil 4.6' da AI sınıfı: 200 örneğin 195'i doğru, yalnızca 5'i insan olarak yanlış sınıflandırılmış. İnsan sınıfı: 200 örneğin 185'i doğru, 15'i AI olarak yanlış sınıflandırılmış. İnsan sınıfında hata sayısı biraz daha fazla, bu da modelin AI sınıfını ayırt etmede daha başarılı olduğunu gösteriyor.



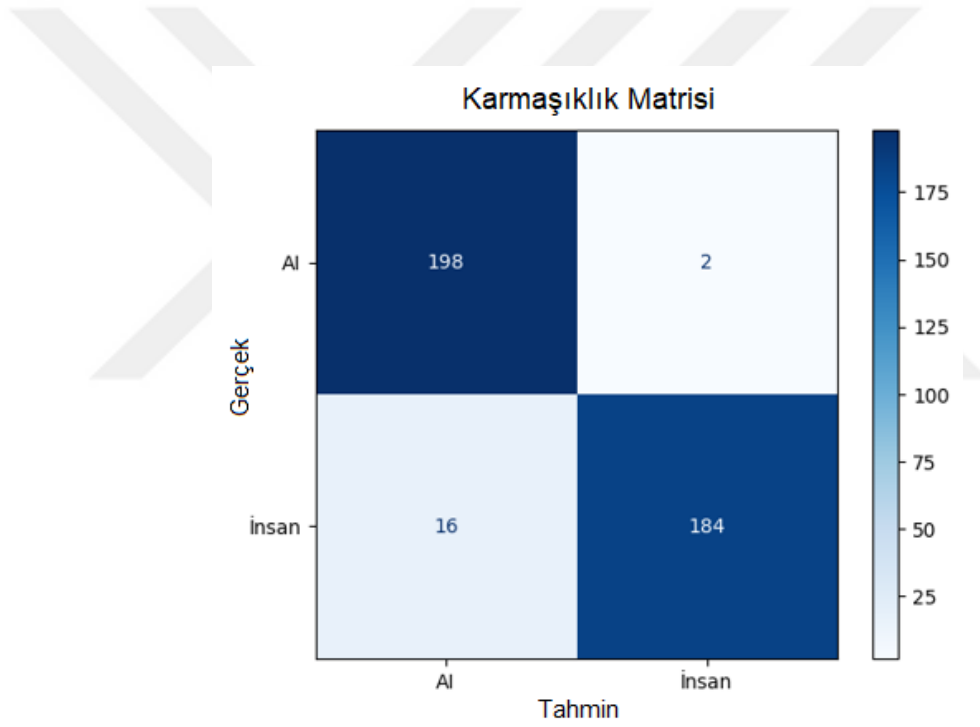
Şekil 4.7.a: epoch = 10 için CNN modelinin accuracy (doğruluk) değeri grafiği



Şekil 4.7.b: epoch = 10 için CNN modelinin kayıp (loss) değeri grafiği

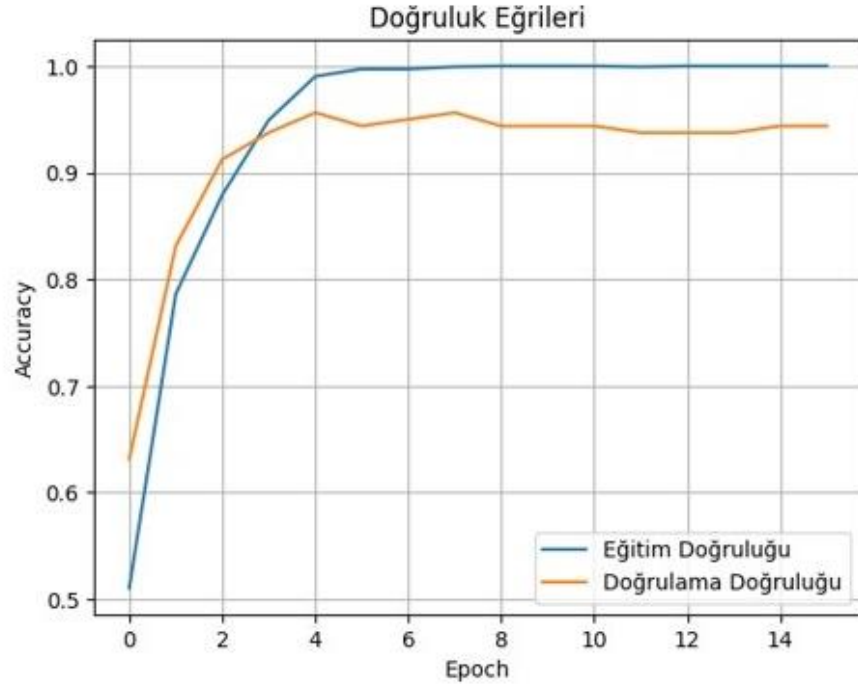
Tablo 4.3: epoch =10 için CNN modelinin doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri

CNN modeli epoch=10 için doğruluk: 0.955				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.93	0.99	0.96	200
insan	0.99	0.92	0.95	200
doğruluk			0.95	400
makro ort.	0.96	0.96	0.95	400
ağırlıklı ort.	0.96	0.95	0.95	400

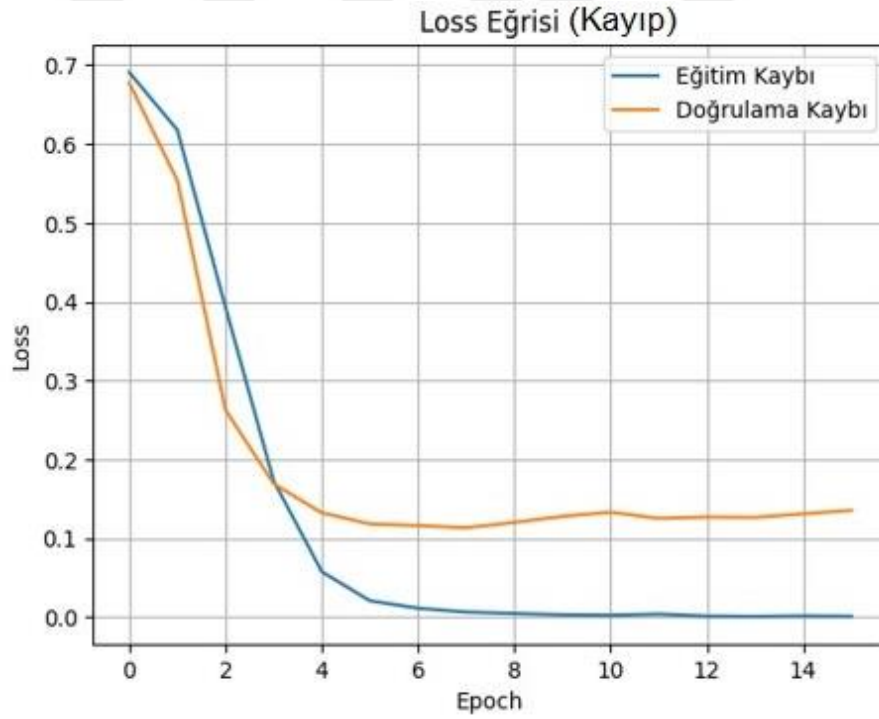


Şekil 4.8: epoch =10 için CNN modelinin Karmaşıklık matrisi

Şekil 4.8’de AI sınıfında 0.99 gibi yüksek bir duyarlılık (recall) değeriyle yapay zekâ tarafından üretilen metinleri başarıyla tanımlamıştır. Ancak insan sınıfı için bu değerinin 0.92’ ye düşmesi bazı insan metinlerinin AI olarak sınıflandırıldığını göstermektedir. Şekil 4.7.a ve Şekil 4.7.b de incelendiğinde eğitim doğruluğu 0.5’ten 1.0’a, doğrulama doğruluğu ise 0.5’ten 0.95’e yükselmiştir. Bu sonuçlar, modelin yaklaşık 0.95-0.97 doğruluk aralığında genelleme kapasitesini koruduğunu ve epoch değeri 10 iken dengeli bir performansa ulaştığını göstermektedir.



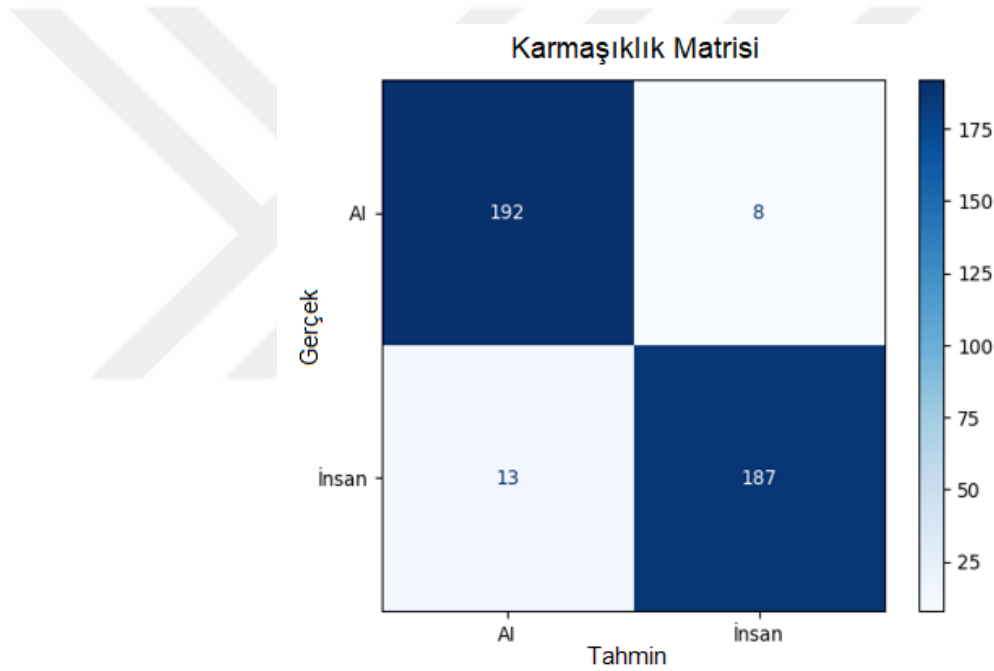
Şekil 4.9.a: epoch =16 için CNN modelinin accuracy (doğruluk) grafiği



Şekil 4.9.b: - epoch =16 için CNN modelinin kayıp (loss) grafiği

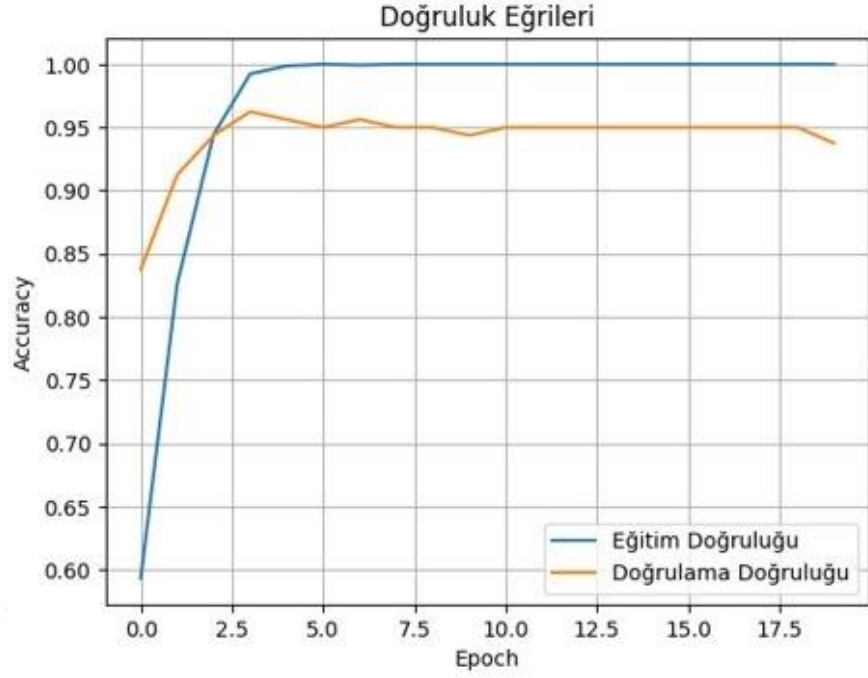
Tablo 4.4: epoch = 16 değeri için CNN modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

CNN modeli epoch=16 için doğruluk: 0.947				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.94	0.96	0.95	200
insan	0.96	0.94	0.95	200
doğruluk			0.95	400
makro ort.	0.95	0.95	0.95	400
ağırlıklı ort.	0.95	0.95	0.95	400

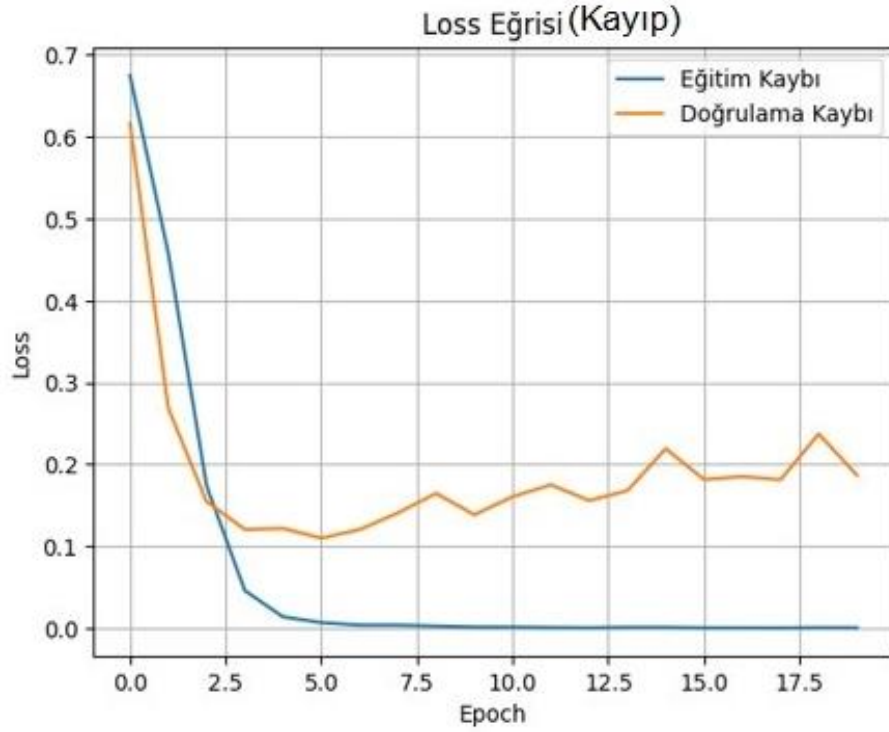


Şekil 4.10: epoch =16 için CNN modelinin Karmaşıklık matrisi

Model, 16. epoch'ta genel doğruluk oranı olarak 0.947' ye ulaşmıştır. AI sınıfında 0.96 doğrulukla yüksek performans gösterirken, insan sınıfında bu oran 0.93 olarak gerçekleşmiştir. Model, AI metinlerini başarılı biçimde tanımlamakta; ancak insan metinlerinde kısmi hatalar yapmaktadır. 10. epoch'a kıyasla doğrulukta küçük bir düşüş gözlenmiş olup modelin 16. epoch itibarıyla öğrenme platosuna ulaştığını göstermektedir. Eğitim doğruluğu 1.0'a yaklaşmış, doğrulama doğruluğu yaklaşık 0.95 seviyesinde sabitlenmiş, loss eğrileri ise minimum düzeyde dengede kalmıştır.



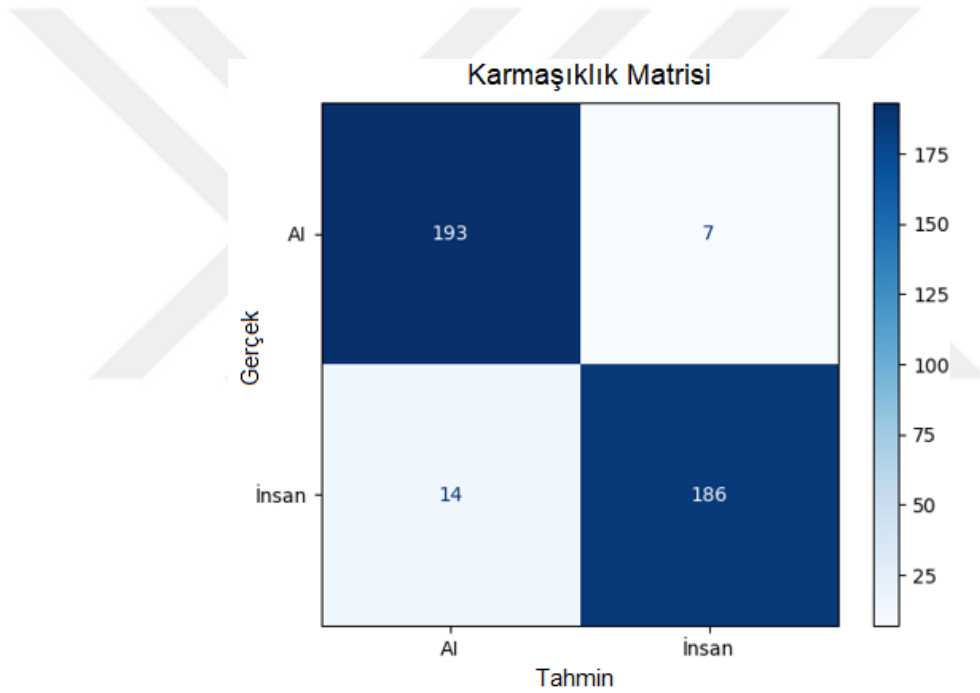
Şekil 4.11.a: epoch =20 için CNN modelinin accuracy (doğruluk) grafiği



Şekil 4.11.b: epoch =20 için CNN modelinin kayıp (loss) grafiği

Tablo 4.5: epoch = 20 değeri için CNN modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

CNN modeli epoch=20 için doğruluk: 0.947				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.93	0.96	0.95	200
insan	0.96	0.93	0.95	200
doğruluk			0.95	400
makro ort.	0.95	0.95	0.95	400
ağırlıklı ort.	0.95	0.95	0.95	400



Şekil 4.12: epoch =20 için CNN modelinin Karmaşıklık matrisi

Modelin 20. epoch itibarıyla eğitim doğruluğu 1.0 seviyesine ulaşırken, doğrulama doğruluğu yaklaşık 0.95 civarında sabit kalmış ve doğrulama kaybında hafif bir artış gözlenmiştir. Bu eğilim, modelin öğrenme platosuna ulaştığını ancak bu noktadan itibaren kısmi bir aşırı öğrenme eğilimi göstermeye başladığını ortaya koymaktadır. Karmaşıklık matrisi sonuçlarına göre model, AI metinlerini yaklaşık 0.96 oranında doğru sınıflandırmış; insan metinlerinde ise yaklaşık 0.93 doğruluk sağlamıştır. Genel doğruluk oranı 0.947 olmuştur.

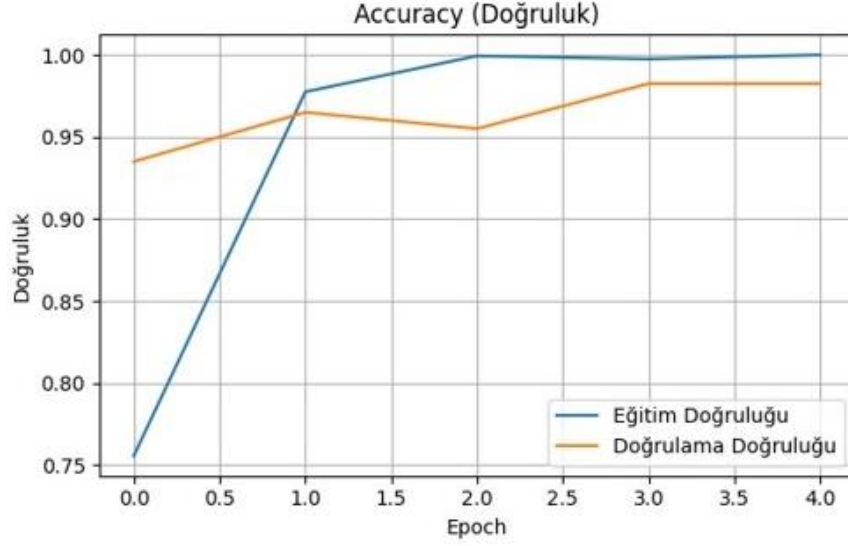
4.2.2. LSTM Modeli

LSTM modeli sıfırdan eğitilmiştir. Girdi metinleri tokenize edilip sabit uzunlukta hizalanmış, ardından embedding katmanıyla kelime vektörlerine dönüştürülmüştür. Modelde bir adet LSTM katmanı ve ardından dropout ve dense katmanlar kullanılmıştır. İki sınıf için softmax aktivasyon fonksiyonu uygulanarak sınıflandırma gerçekleştirilmiştir. Model, categorical_crossentropy kayıp fonksiyonu ve adam optimizasyon algoritmasıyla 10 epoch boyunca eğitilmiş; eğitim süresi, doğruluk, kayıp değerleri ve sistem kaynak kullanımı detaylı olarak izlenmiştir.

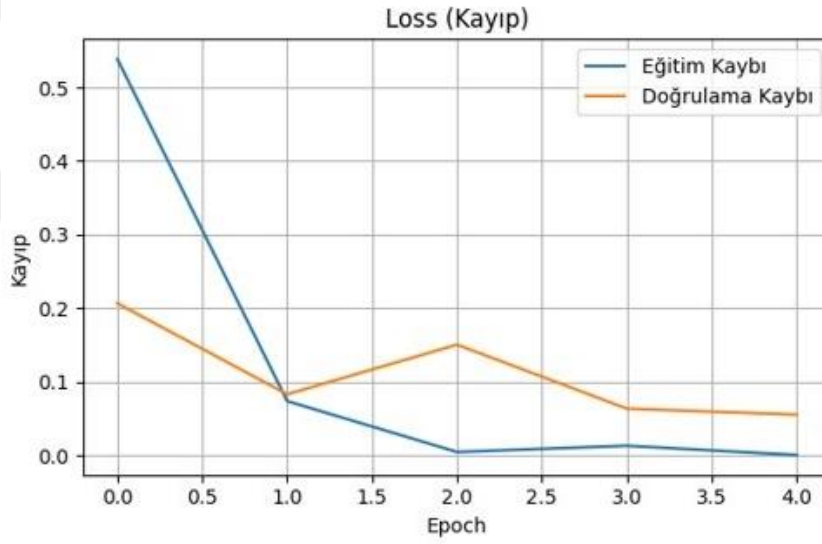
LSTM modeli 5, 10, 20 epoch değerleri ile eğitilmiş olup farklı epoch değerleri için eğitim süresi, doğruluk, kayıp değerleri tablolar ve grafiklerle gösterilmiştir. Ayrıca sistem kaynak kullanımı Tablo 4.6'da early stop değerine (epochs = 10) göre izlenmiştir. Şekil 4.13.a, Şekil 4.13.b, Şekil 4.15.a, Şekil 4.15.b, Şekil 4.17.a, Şekil 4.17.b de sırasıyla 5, 10 ve 20 epoch değerleri için doğruluk - kayıp grafikleri yer almaktadır. Tablo 4.7, Tablo 4.8, Tablo 4.9'da aynı sıradaki epoch değerleri için doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri yer almaktadır. Her bir epoch için sırasıyla karmaşıklık matrisi Şekil 4.14, Şekil 4.16 ve Şekil 4.18'de gösterilmiştir.

Tablo 4.6: Colab ortamında çalıştırılan LSTM modelinin kaynak kullanımı

LSTM Kaynak kullanımı	
Eğitim Süresi:	8 saniye
Maksimum GPU bellek kullanımı (allocated):	0.06 GB
Maksimum GPU bellek rezervasyonu (reserved):	0.02 GB
RAM Kullanımı:	% 3.2
Toplam RAM:	83.48 GB
Kullanılan RAM:	2.67 GB
CPU Kullanımı:	10.9%



Şekil 4.13.a: epoch =5 için LSTM modelinin accuracy (doğruluk) değeri grafiği

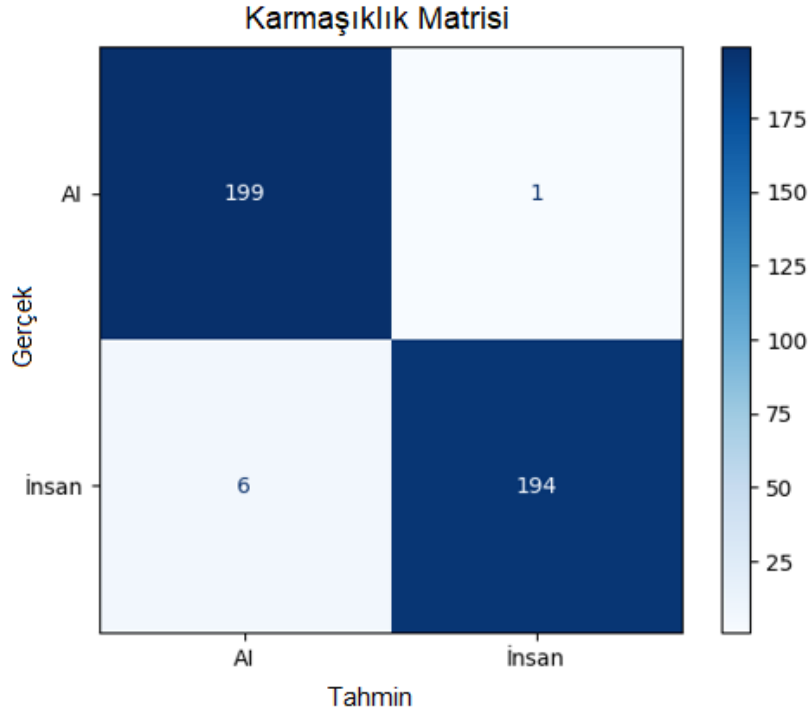


Şekil 4.13.b: epoch =5 için LSTM modelinin kayıp (loss) değeri grafiği

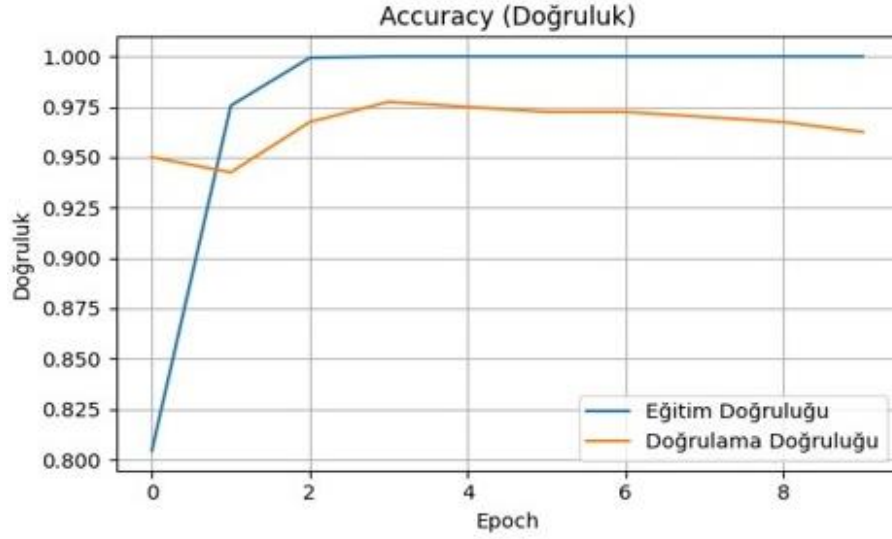
Şekil 4.14' de görüldüğü gibi model, 400 örnek üzerinden 0.98 doğruluk elde etmiş ve yalnızca 7 örnek yanlış sınıflandırılmıştır. AI ve insan sınıfları dengeli biçimde ayrılmış, f1-skoru 0.98 olarak ölçülmüştür. Yüksek doğruluk ve düşük kayıp değerleri, modelin genelleme başarısının yüksek ve overfitting riskinin düşük olduğunu göstermektedir. Eğitim doğruluğu 0.75'ten başlayarak hızla 1.0'e ulaşmış, doğrulama doğruluğu ise 0.95 civarında dengeli seyretmiştir. Eğitim kaybı hızla azalmış, doğrulama kaybı ise düşük düzeyde paralel ilerlemiştir.

Tablo 4.7: epoch = 5 değeri için LSTM modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

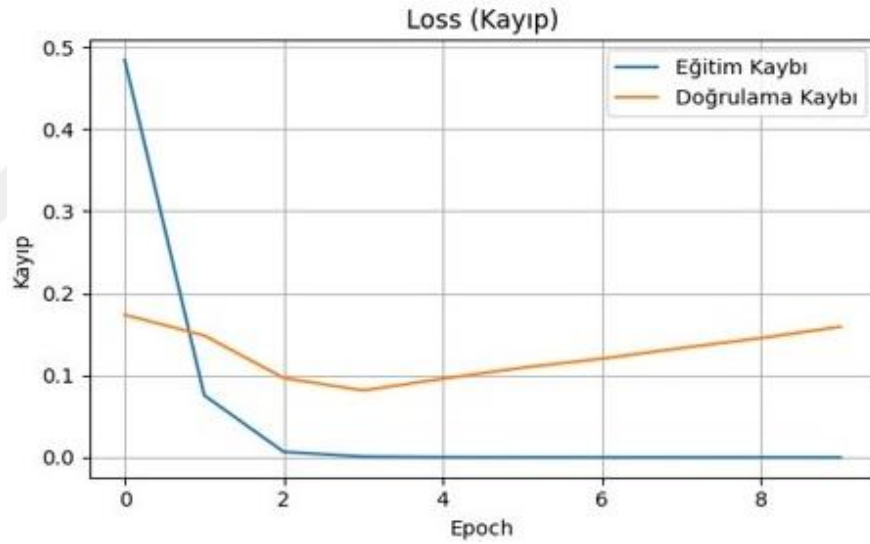
LSTM modeli epoch=5 için Doğruluk : 0.9825				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
Ai	0.97	0.99	0.98	200
insan	0.99	0.97	0.98	200
doğruluk			0.98	400
makro ort.	0.98	0.98	0.98	400
ağırlıklı ort.	0.98	0.98	0.98	400



Şekil 4.14 epoch =5 için LSTM modelinin Karmaşıklık matrisi



Şekil 4.15.a: epoch = 10 için LSTM modelinin accuracy (doğruluk) değeri grafiği

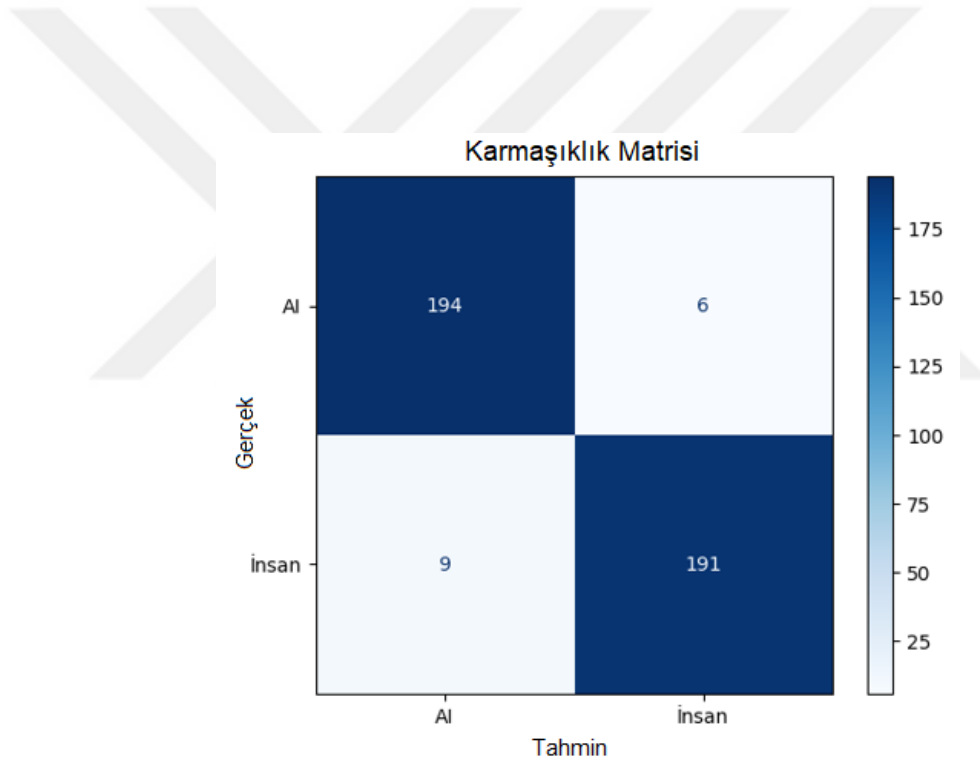


Şekil 4.15.b: epoch =10 için LSTM modelinin kayıp (loss) değeri grafiği

Şekil 4.15.a ve ekil 4.15.b' de görüldüğü gibi eğitim doğruluğu 1.0 seviyesine ulaşmış, doğrulama doğruluğu ise 0.97 civarında plato yaparak sabitlenmiştir. Eğitim kaybı sıfıra yaklaşırken, doğrulama kaybı 2. epoch'tan itibaren hafif bir artış eğilimi göstermiştir. Bu durum, modelin veriyi oldukça iyi öğrendiğini ancak genelleme yeteneğinin azalmaya başladığını ortaya koymaktadır.

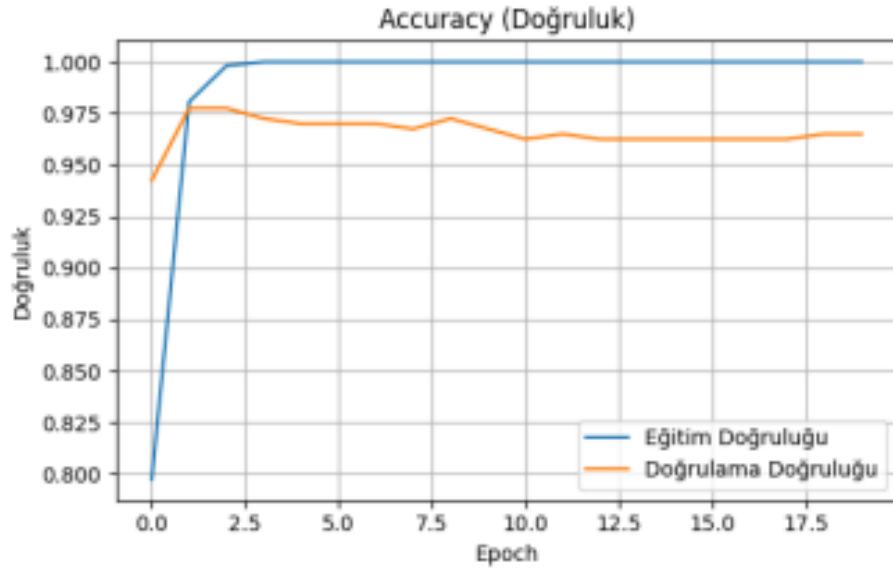
Tablo 4.8: epoch = 10 deęeri için LSTM modelinin doęruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı deęerleri

LSTM modeli epoch=10 için Doęruluk: 0.962				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.96	0.97	0.96	200
insan	0.97	0.95	0.96	200
doęruluk			0.96	400
makro ort.	0.96	0.96	0.96	400
aęırlıklı ort.	0.96	0.96	0.96	400

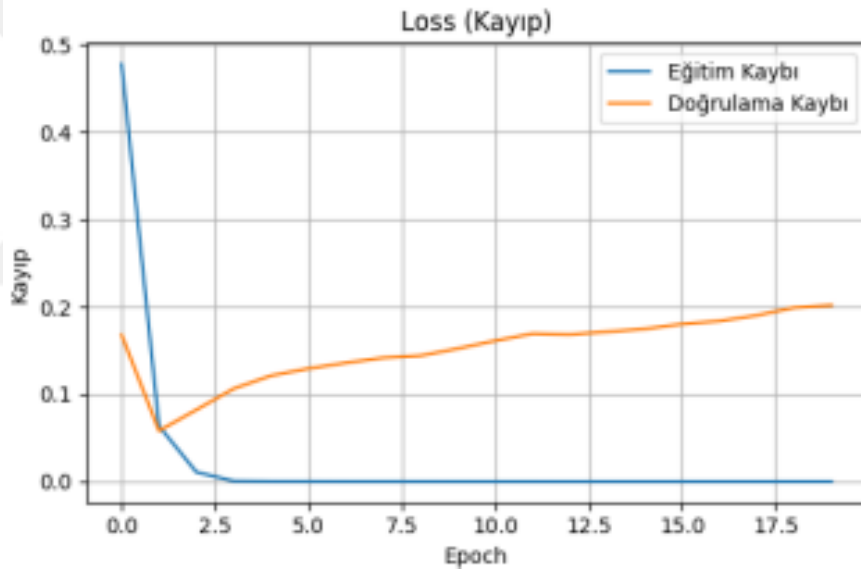


Şekil 4.16: epoch = 10 için LSTM modelinin Karmaşıklık matrisi

Epoch=10 için modelde yanlış sınıflandırma sayısı 15'e yükselmiş olup, AI sınıfında *recall* 0.97, insan sınıfında ise 0.95 olarak elde edilmiştir. Her iki sınıf için f1-skoru yaklaşık 0.96 düzeyindedir. Bu sonuçlar, modelin hâlâ güçlü bir performans sergilediğini ancak doęrulama kaybındaki artış nedeniyle aşırı öğrenme (overfitting) sürecine girmeye başladığını göstermektedir.



Şekil 4.17.a: epoch= 20 için LSTM modelinin accuracy (doğruluk) değeri grafiği

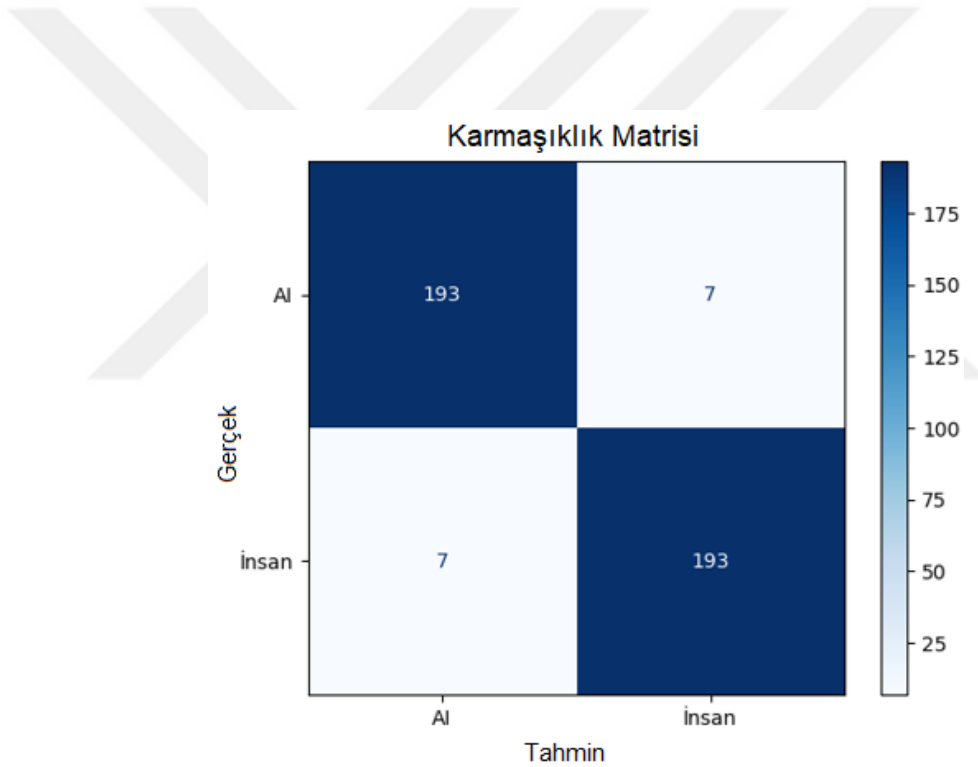


Şekil 4.17.b: epoch= 20 için LSTM modelinin kayıp (loss) değeri grafiği

Eğitim doğruluğunun 1.0 seviyesine ulaşmış olduğu ve doğrulama doğruluğunun ise yaklaşık 0.95 düzeyinde sabitlenmiş olduğu görülmektedir. Doğrulama kaybındaki artış, modelin ezberleme aşamasına geçtiğini ve genelleme yeteneğinin azaldığını göstermektedir. Bu nedenle, bu aşamadan sonraki epoch artışları modelin performansına anlamlı bir katkı sağlamayacağı düşünülmektedir.

Tablo 4.9: epoch = 20 değeri için LSTM modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

LSTM modeli epoch=20 için Doğruluk: 0.965				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.96	0.96	0.96	200
insan	0.96	0.96	0.96	200
doğruluk			0.96	400
makro ort.	0.96	0.96	0.96	400
ağırlıklı ort.	0.96	0.96	0.96	400



Şekil 4.18: epoch= 20 için LSTM modelinin Karmaşıklık matrisi

Tablo 4.9'daki sonuçlara göre LSTM modelinin 20. epoch'taki genel doğruluk oranı 0.965 olarak elde edilmiştir. Her iki sınıfta da (AI ve insan) kesinlik, duyarlılık ve f1-skoru değerleri 0.96 olup modelin performansı dengeli bir yapı göstermektedir. Şekil 4.18'de görüldüğü gibi model, her iki sınıfta da 193 örneği doğru sınıflandırmış ve 7'şer hatalı tahmin yapmıştır.

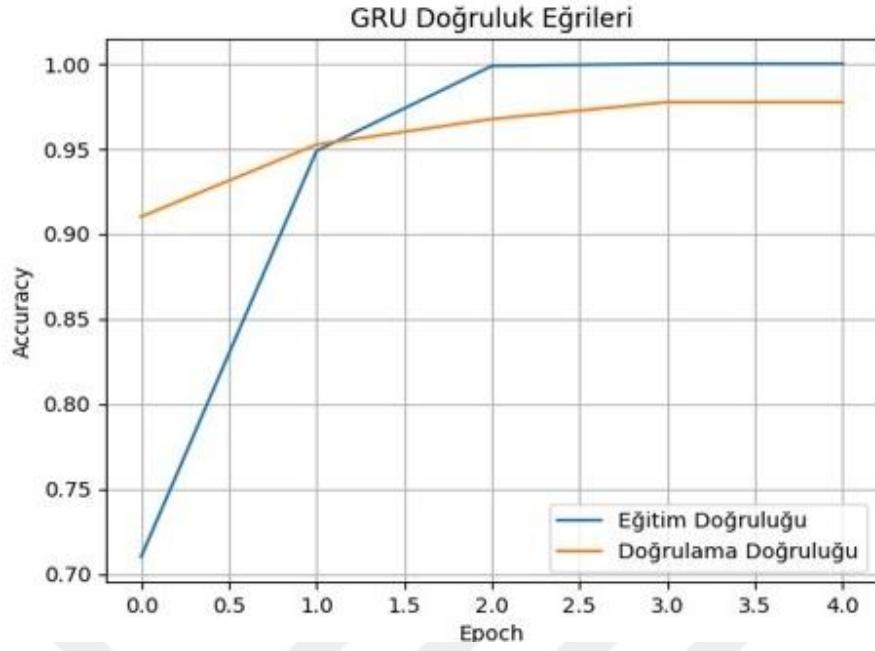
4.2.3. GRU Modeli

Metinler tokenizer aracılığıyla sayısallaştırılmış ve sabit uzunlukta hizalanmıştır. Modelde embedding katmanından sonra bir GRU katmanı, ardından dropout ve dense katmanlar yer almaktadır. Çıkış katmanında iki sınıf için softmax aktivasyon fonksiyonu kullanılarak ai ve insan yazımı metinler ayrıştırılmıştır. GRU modeli sıfırdan 5, 10, 12 ve 20 epoch değerleri ile eğitilmiştir. Sistem kaynak kullanımı Tablo 4.10'da yer almaktadır ve early stop değerine (epochs = 12) göre izlenmiştir. Şekil 4.19.a, Şekil 4.19.b, Şekil 4.21.a, Şekil 4.21.b, Şekil 4.23.a, Şekil 4.23.b, Şekil 4.25.a ve Şekil 4.25.b'de de sırasıyla 5, 10, 12 ve 20 epoch değerleri için doğruluk ve loss grafikleri verilmiştir. Tablo 4.11, Tablo 4.12, Tablo 4.13 ve Tablo 4.14' de aynı sıradaki epoch değerleri için doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri yer almaktadır. Ayrıca her bir epoch için sırasıyla Şekil 4.20, Şekil 4.22, Şekil 4.24 ve Şekil 4.26'de karmaşıklık matrisi bulunmaktadır.

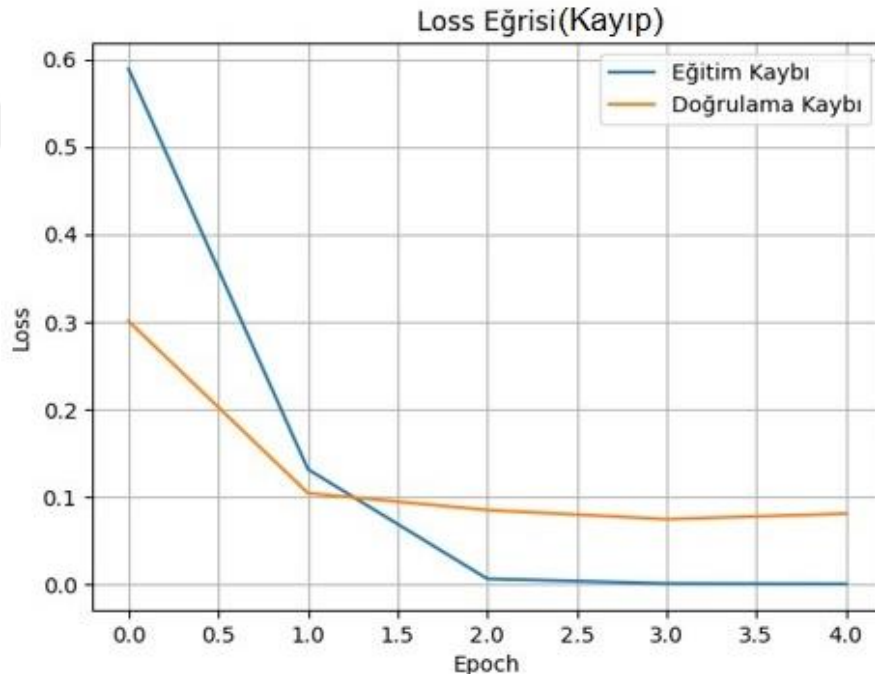
Tablo 4.10: Colab ortamında çalıştırılan GRU modelinin kaynak kullanımı

GRU Kaynak kullanımı	
Eğitim Süresi:	12 saniye
Maksimum GPU bellek kullanımı (allocated):	0.05 GB
Maksimum GPU bellek rezervasyonu (reserved):	0.02 GB
RAM Kullanımı:	% 3.3
Toplam RAM:	83.48 GB
Kullanılan RAM:	1.97 GB
CPU Kullanımı:	% 6.5

Şekil 4.19.a ve Şekil 4.19.b'de görüldüğü gibi, model 5. epoch'ta eğitim doğruluğunu hızla artırarak 1.0 seviyesine yaklaşmış, doğrulama doğruluğu ise yaklaşık 0.97 civarında seyredip artış eğilimi göstermiştir. Eğitim ve doğrulama kayıpları birlikte azalarak stabil bir düzeye ulaşmış, doğrulama kaybı düşük ve paralel ilerlemiştir. Bu sonuçlar, modelin kısa sürede dengeli bir öğrenme gerçekleştirdiğini, genelleme yeteneğinin yüksek olduğunu ve overfitting belirtilerinin bulunmadığını göstermektedir.



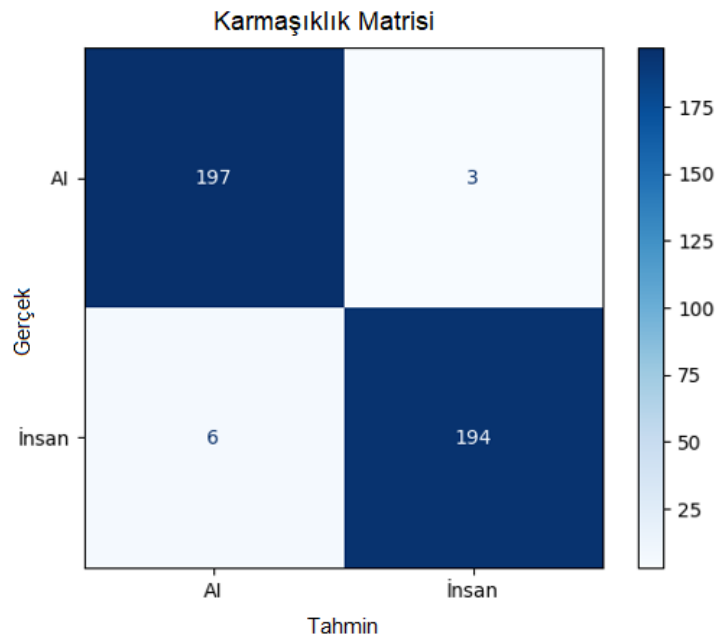
Şekil 4.19.a: epoch= 5 için GRU modelinin accuracy (doğruluk) değeri grafiği



Şekil 4.19.b: epoch= 5 için GRU modelinin kayıp (loss) değeri grafiği

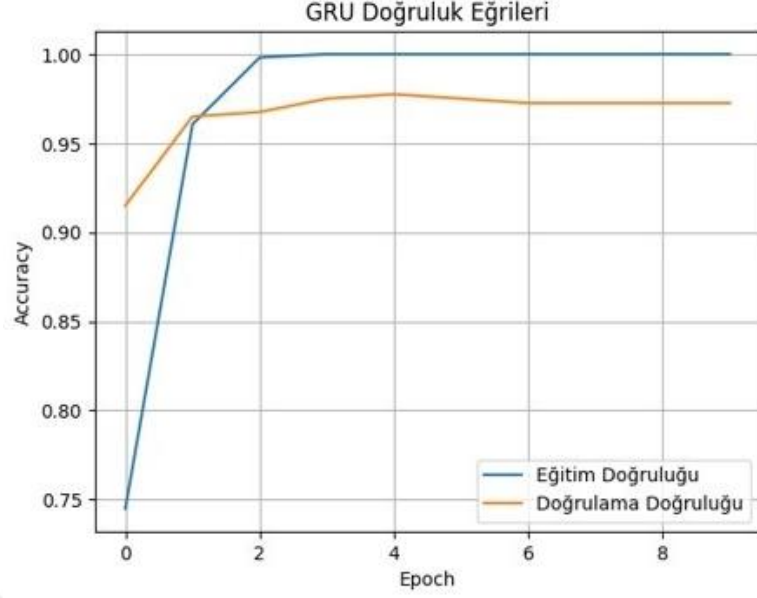
Tablo 4.11: epoch= 5 için GRU modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

GRU modeli epoch=5 için Doğruluk: 0.977				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.97	0.98	0.98	200
insan	0.98	0.99	0.98	200
doğruluk			0.98	400
makro ort.	0.98	0.98	0.98	400
ağırlıklı ort.	0.98	0.98	0.98	400

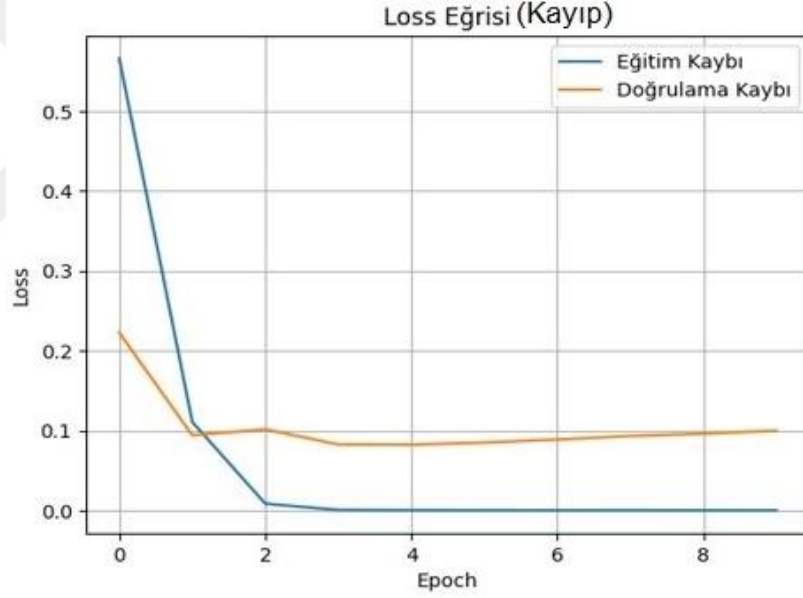


Şekil 4.20: epoch= 5 için GRU modelinin Karmaşıklık matrisi

Tablo 4.11'deki sonuçlara göre GRU modelinin 5. epoch'taki genel doğruluk oranı 0.97 olarak elde edilmiştir. AI sınıfında kesinlik 0.97, duyarlılık 0.98; insan sınıfında ise kesinlik 0.98, duyarlılık 0.99 olarak ölçülmüştür. Her iki sınıfta da f1-skoru 0.98 olup performans dengeli ve yüksek genelleme başarısı göstermektedir. Bu sonuçlar, modelin erken epoch değerlerinde hızlı ve etkili bir öğrenme süreci gerçekleştirdiğini gösterirken Şekil 4.20 de ise ai sınıfında 197 ve insan sınıfında 194 örneği doğru sınıflandırmış olduğu görülmektedir. Hatalı tahmin sayısının düşük olması ile modelin yüksek doğruluk ve güçlü genelleme performansı sergilediği kanısına varılmıştır.



Şekil 4.21.a: epoch= 10 için GRU modelinin accuracy (doğruluk) değeri grafiği

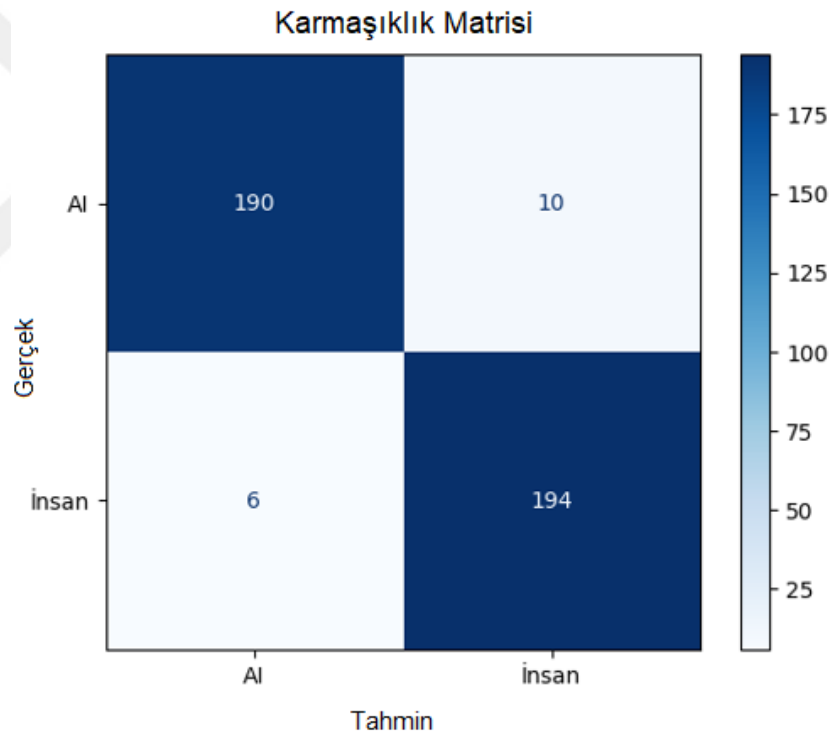


Şekil 4.21.b: epoch= 10 için GRU modelinin kayıp (loss) değeri grafiği

Şekil 4.20.a ve Şekil 4.20.b’de görüldüğü gibi, modelin 10. epoch’ta eğitim doğruluğu 1.0 seviyesine ulaşmış, doğrulama doğruluğu ise 0.97 düzeyinde sabitlenmiştir. Eğitim kaybı sıfıra yaklaşırken, doğrulama kaybı yaklaşık 0.1 seviyesinde hafif bir artış eğilimi göstermiştir. Bu durum, modelde hafif düzeyde overfitting belirtilerinin başladığını, ancak doğrulama doğruluğunun hâlâ yüksek seyretmesi nedeniyle modelin genel olarak güvenilir ve kararlı bir performans sergilediğini göstermektedir.

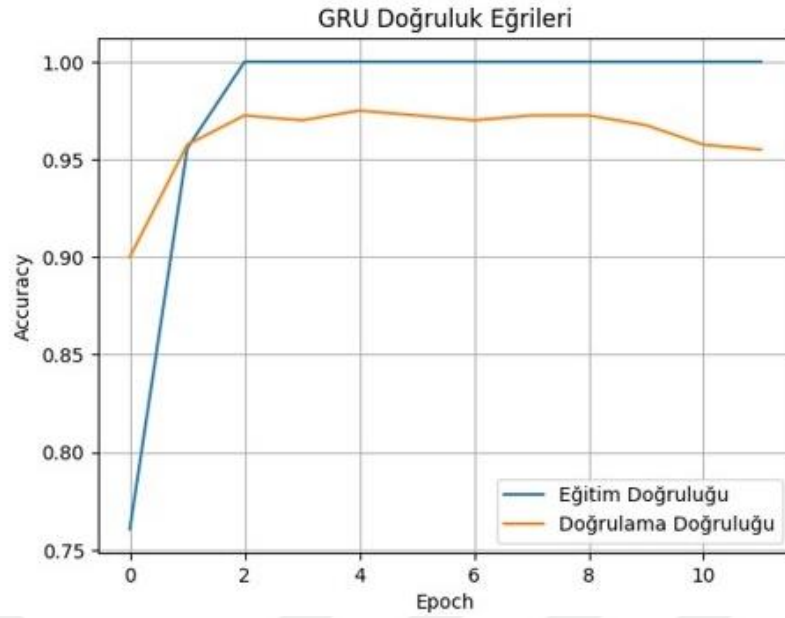
Tablo 4.12: GRU epoch= 10 için precision, duyarlılık, f1-skoru ve örnek sayısı değerleri

GRU modeli epoch=10 için Doğruluk: 0.972				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.97	0.97	0.97	200
insan	0.97	0.97	0.97	200
doğruluk			0.97	400
makro ort.	0.97	0.97	0.97	400
ağırlıklı ort.	0.97	0.97	0.97	400

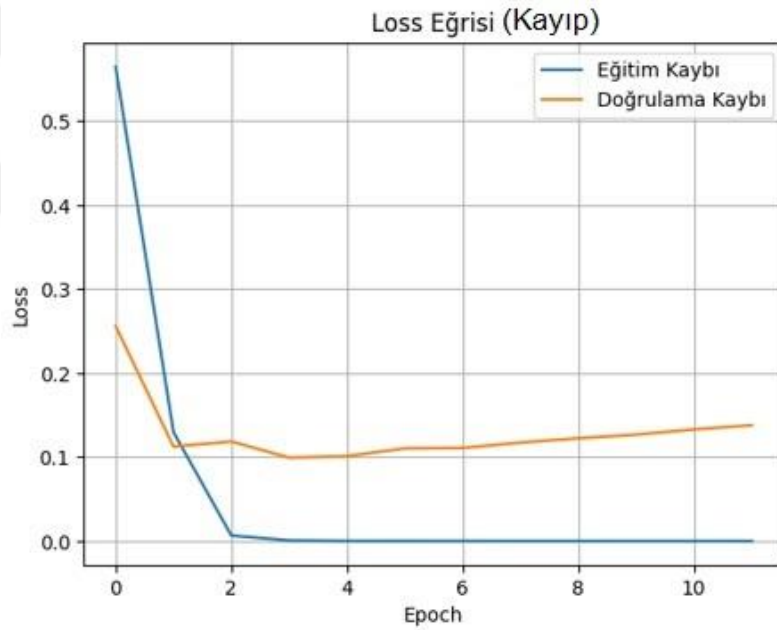


Şekil 4.22: epoch= 10 için GRU modelinin Karmaşıklık matrisi

Tablo 4.12'deki sonuçlara göre GRU modelinin 10. epoch'taki genel doğruluk oranı 0.972 olarak elde edilmiştir. Her iki sınıfta da (AI ve insan) kesinlik, duyarlılık ve f1-skoru değerleri 0.97 olup performans dağılımı dengeli bir yapı göstermektedir. Şekil 4.22'te görüldüğü gibi model, ai sınıfında 190 ve insan sınıfında 194 örneği doğru sınıflandırmıştır.



Şekil 4.23.a: epoch= 12 için GRU modelinin accuracy (doğruluk)değeri



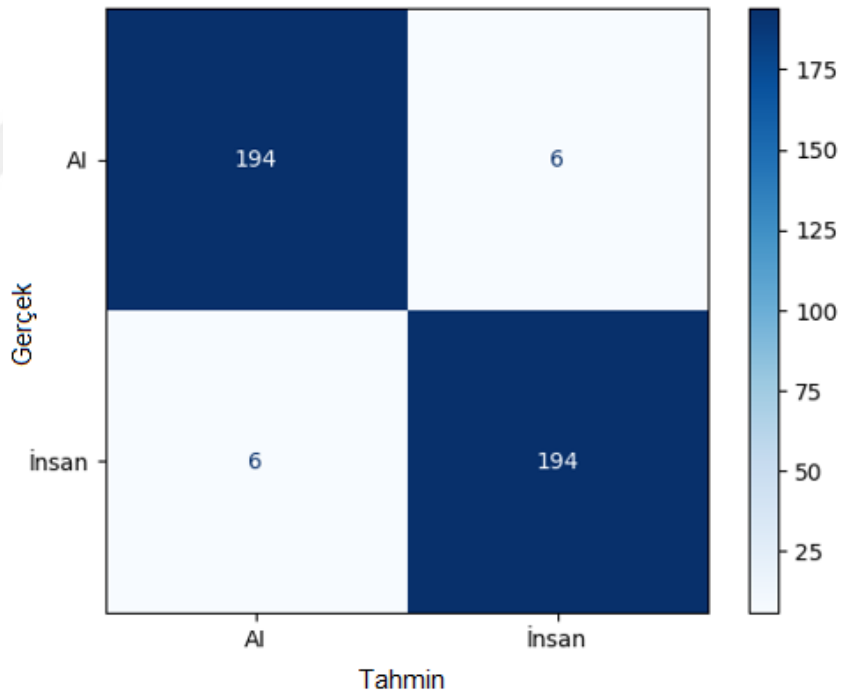
Şekil 4.23.b: epoch= 12 için GRU modelinin kayıp (loss) değeri

Şekil 4.23.a ve Şekil 4.23.b’de görüldüğü gibi, modelin 12. epoch’ ta eğitim doğruluğu 1.0’a ulaşmış, doğrulama doğruluğu ise yaklaşık 0.965 seviyesinde sabit kalmıştır. Eğitim kaybı oldukça düşük, doğrulama kaybı ise çok hafif bir artış eğilimindedir. Bu durum, hafif düzeyde overfitting olasılığına işaret etse de doğruluk oranı hâlâ yüksek kalmıştır.

Tablo 4.13. epoch = 12 değeri için GRU modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

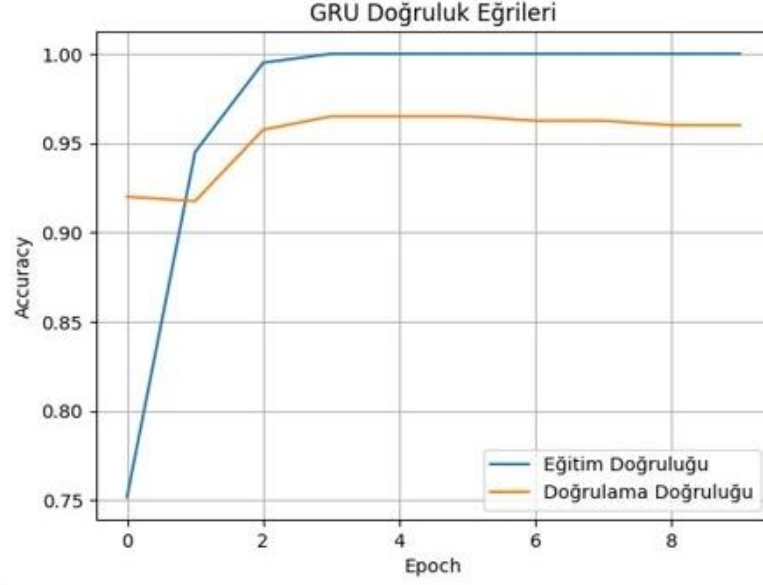
GRU modeli epoch=12 için Doğruluk: 0.97				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.97	0.97	0.97	200
insan	0.97	0.97	0.97	200
doğruluk			0.97	400
makro ort.	0.97	0.97	0.97	400
ağırlıklı ort.	0.97	0.97	0.97	400

Karmaşıklık Matrisi

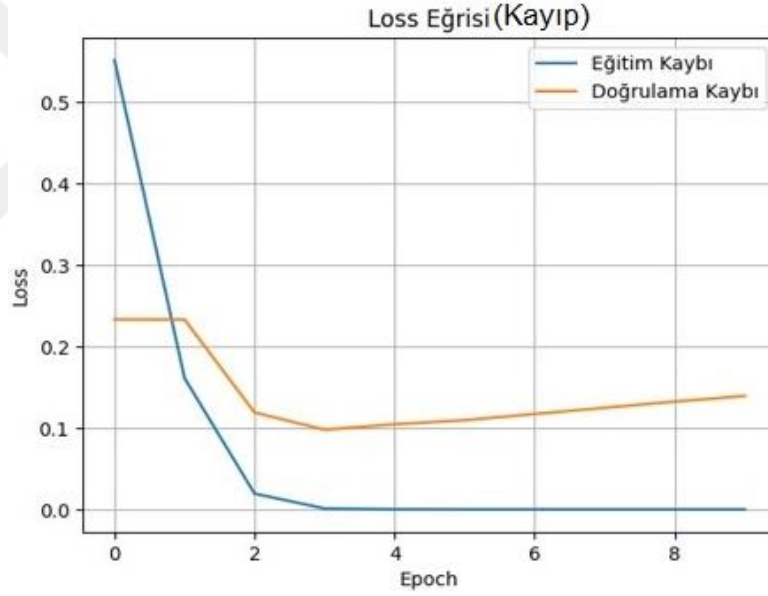


Şekil 4.24: epoch= 12 için GRU modelinin Karmaşıklık matrisi

Tablo 4.13'teki sonuçlara göre modelinin 12. epoch' taki genel doğruluk oranı 0.97 olarak elde edilmiştir. Her iki sınıfta da (ai ve insan) kesinlik, duyarlılık ve f1-skoru değerleri 0.97 olmuştur. Şekil 4.24'te görüldüğü gibi model, her iki sınıfta da 194 örneği doğru sınıflandırmış ve her iki sınıfta da 6 yanlış tahmin yapmıştır.



Şekil 4.25.a: epoch= 20 için GRU modelinin accuracy (doğruluk) değeri grafiği

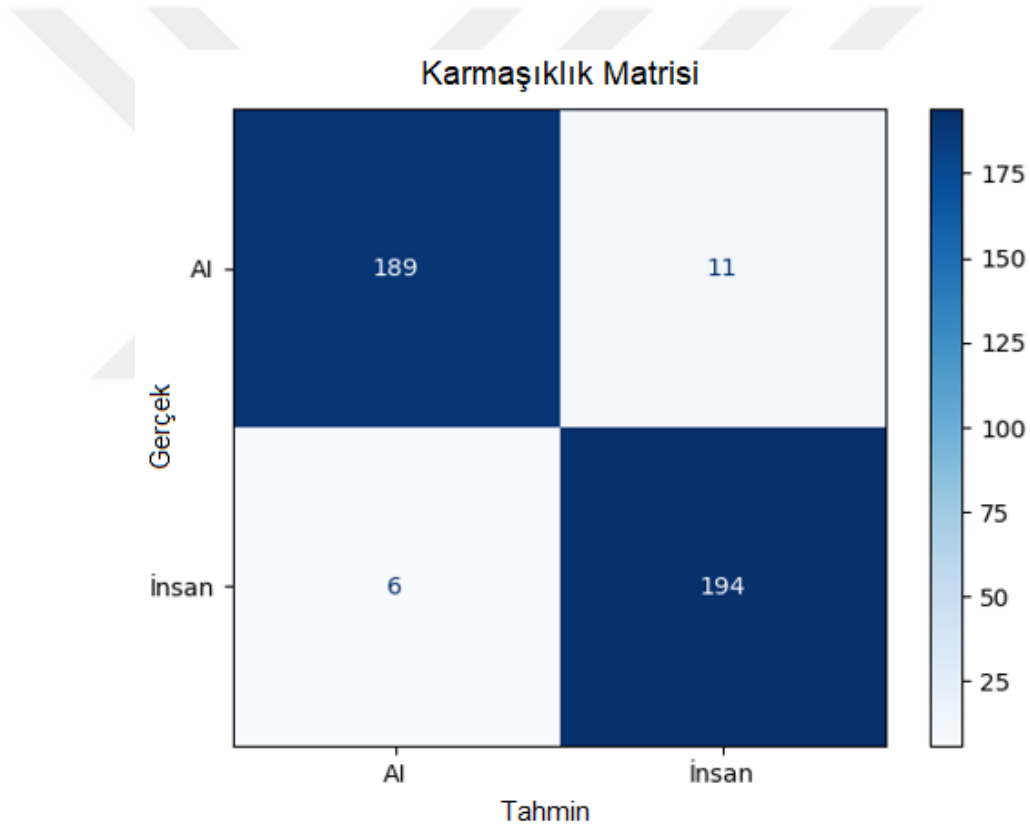


Şekil 4.25.b: epoch= 20 için GRU modelinin kayıp (loss) değeri

Şekil 4.25.a ve Şekil 4.25.b'de görüldüğü gibi, modelin 20. epoch'ta eğitim doğruluğu 1.0 seviyesinde sabitlenmiş, doğrulama doğruluğu ise dalgalı bir seyir izleyerek son aşamalarda hafif bir düşüş göstermiştir. Eğitim kaybı sıfıra ulaşırken, doğrulama kaybı belirgin biçimde artış eğilimi göstermiştir. Bu durum, modelin açık biçimde overfitting aşamasına geçtiğini ve eğitim verisini ezberlemeye başladığını ortaya koymaktadır. Dolayısıyla bu noktadan itibaren modelin genelleme yeteneği azalmış ve genel doğruluk oranı düşmeye başlamıştır.

Tablo 4.14: epoch = 20 değeri için GRU modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

GRU modeli epoch=20 için Doğruluk: 0.957				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.97	0.94	0.96	200
insan	0.95	0.97	0.96	200
doğruluk			0.96	400
makro ort.	0.96	0.96	0.96	400
ağırlıklı ort.	0.96	0.96	0.96	400



Şekil 4.26: epoch= 20 için modelinin Karmaşıklık matrisi

Tablo 4.14 ve Şekil 4.26’te görüldüğü gibi model, AI sınıfında 189, insan sınıfında ise 194 örneği doğru sınıflandırmıştır. Hatalı sınıflandırma sayısının düşük olması, modelin yüksek doğruluk ve dengeli performans sergilediğini göstermektedir.

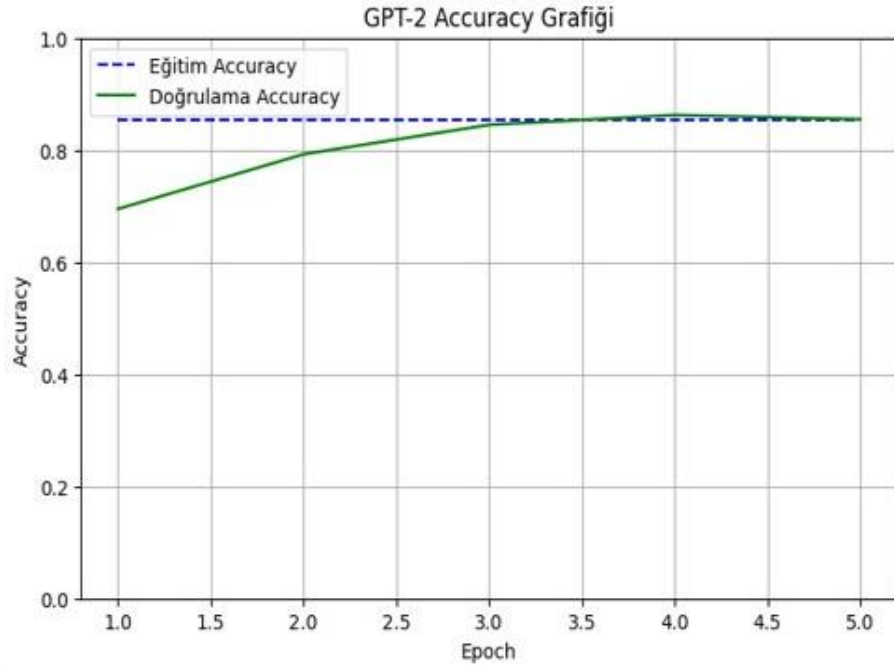
4.2.4. GPT-2 Modeli

124,441,344 parametreye sahip GPT-2 modeli, önceden eğitilmiş bir dil modeli olarak Hugging Face kütüphanesi üzerinden yüklenmiş ve bu çalışmaya özgü sınıflandırma görevi için yeniden eğitilmiştir (fine-tuning). Modelin son katmanı iki sınıflı sınıflandırma (ai/insan) için uyarlanmıştır. Ayrıca sistem kaynak kullanımı (Tablo 4.15) early stop değerine (epochs = 14) göre izlenmiştir. 5, 10, 14 ve 20 epoch değerleri için doğruluk - loss grafikleri Şekil 4.27.a, Şekil 4.27.b, Şekil 4.29.a, Şekil 4.29.b, Şekil 4.31.a, Şekil 4.31.b, Şekil 4.33.a ve Şekil 4.33.b' de sırasıyla yer almaktadır. Tablo 4.16, Tablo 4.17, Tablo 4.18 ve Tablo 4.19' de aynı sıradaki epoch değerleri için doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri yer almaktadır. Şekil 4.28, Şekil 4.30, Şekil 4.32 ve Şekil 4.34' de her bir epoch için sırasıyla karmaşıklık matrisi bulunmaktadır.

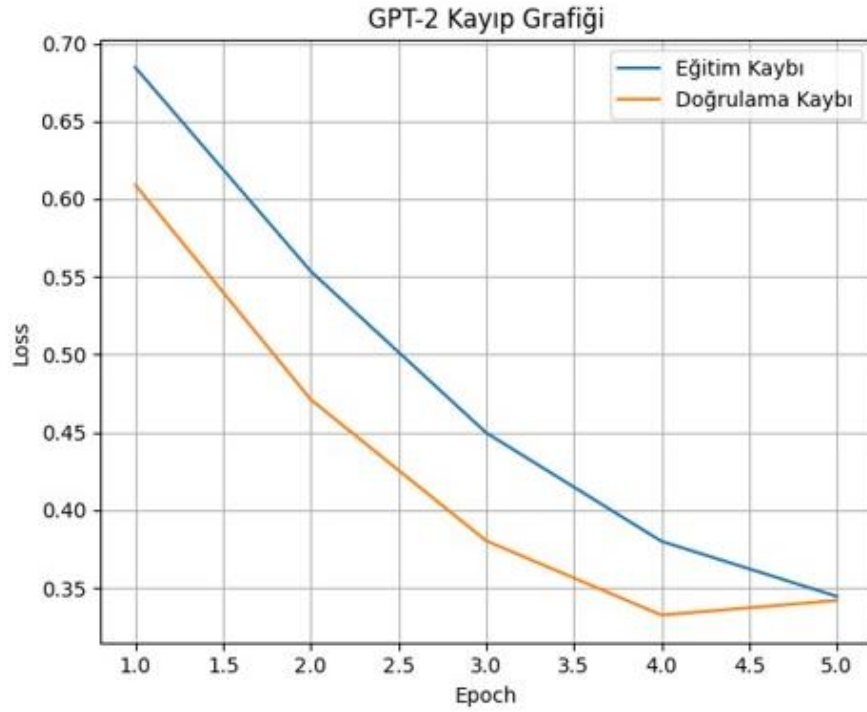
Tablo 4.15: Colab ortamında çalıştırılan GPT - 2 modelinin kaynak kullanımı

GPT -2 Kaynak kullanımı	
Eğitim Süresi:	6.47 dk
Maksimum GPU bellek kullanımı (allocated):	3.69 GB
Maksimum GPU bellek rezervasyonu (reserved):	4.03 GB
RAM Kullanımı:	% 5.4
Toplam RAM:	83.48 GB
Kullanılan RAM:	3.64 GB
CPU Kullanımı:	% 0.8

Şekil 4.27.a ve Şekil 4.27.b'de görüldüğü gibi, eğitim doğruluğu çok hızlı artarak yaklaşık 0.99 seviyesine ulaşmıştır. Doğrulama doğruluğu 2. epoch'ta 0.96 ile zirveye çıkmış, ardından kademeli bir düşüş göstererek yaklaşık 0.94.5 düzeyinde sabitlenmiştir. Eğitim kaybı hızla azalırken doğrulama kaybı 2. epoch'tan sonra artış eğilimi göstermiştir. Bu durum, modelin erken dönemde etkili bir öğrenme gerçekleştirdiğini ancak overfitting belirtilerinin ortaya çıkmaya başladığını göstermektedir.



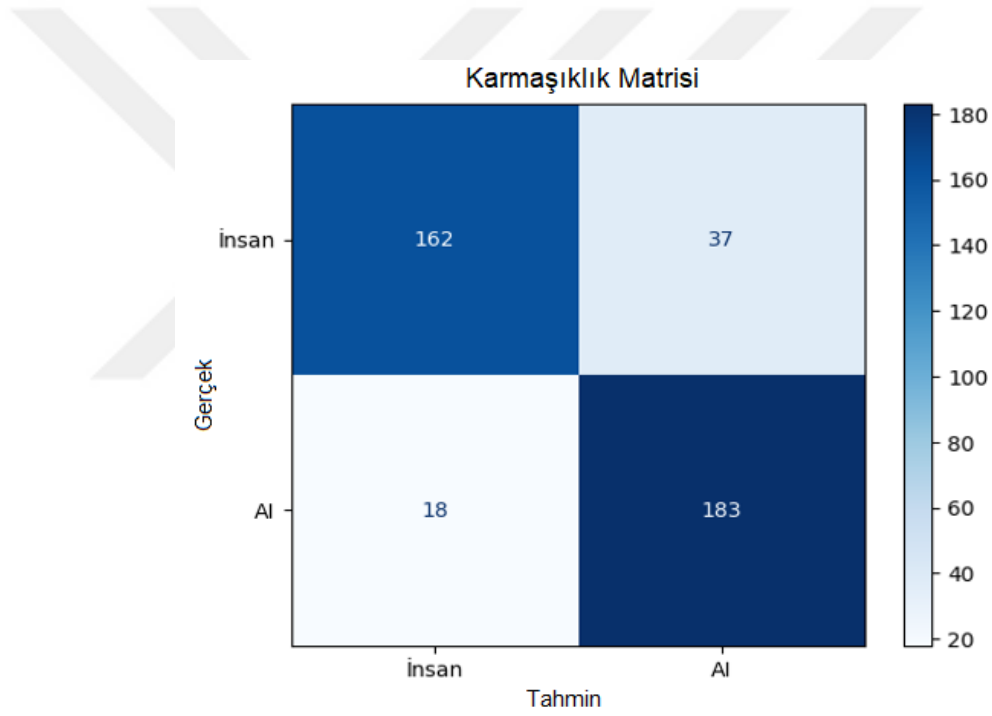
Şekil 4.27.a: GPT-2 modelinin epoch =5 için accuracy (doğruluk) değeri grafiği



Şekil 4.27.b: GPT - 2 modelinin epoch =5 için kayıp (loss) değeri grafiği

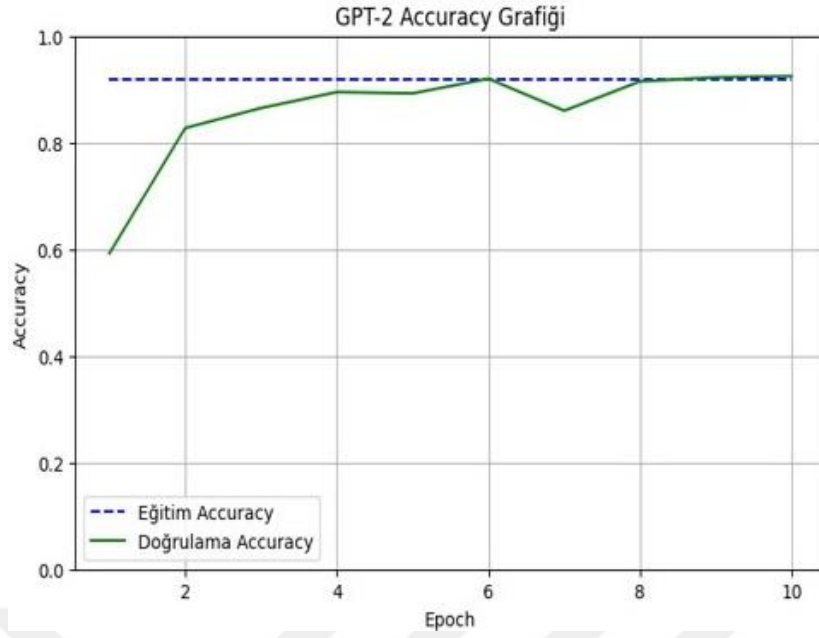
Tablo 4.16: epoch = 5 değeri için GPT-2 modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

GPT-2 modeli epoch=5 için doğruluk: 0.85				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.90	0.81	0.85	200
insan	0.83	0.91	0.87	200
doğruluk			0.86	400
makro ort.	0.87	0.86	0.86	400
ağırlıklı ort.	0.87	0.86	0.86	400

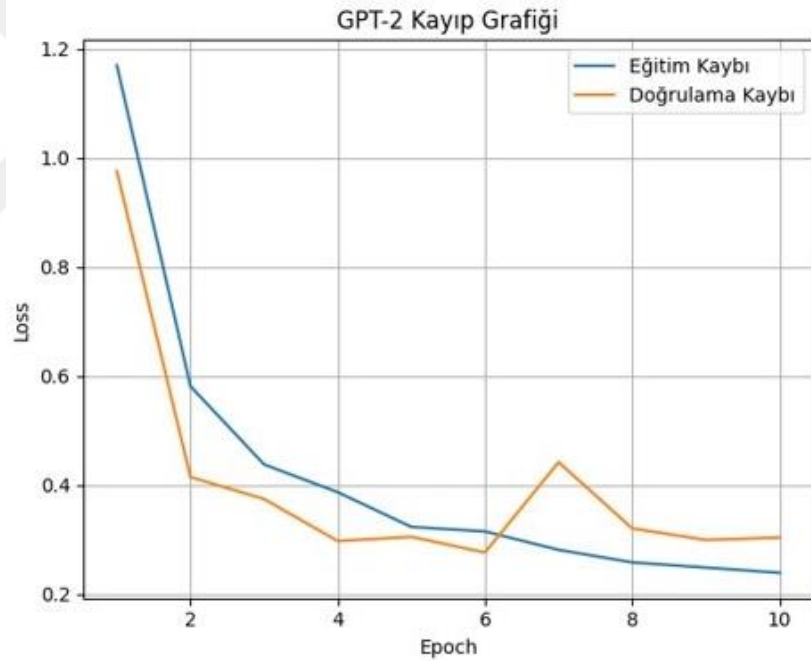


Şekil 4.28: GPT - 2 modelinin epoch =5 için karmaşıklık değeri

Şekil 4.28’ de görüldüğü gibi GPT-2 modeli, ai sınıfında 183 ve insan sınıfında 162 örneği doğru sınıflandırmıştır. Hatalı tahminlerin sayısı görece yüksek olup (37 insan, 18 ai), modelin genelleme performansının sınırlı ve doğruluğunun 0.85 seviyesinde kaldığını göstermektedir.



Şekil 4.29.a: GPT-2 modelinin epoch =10 için accuracy (doğruluk) değeri grafiği

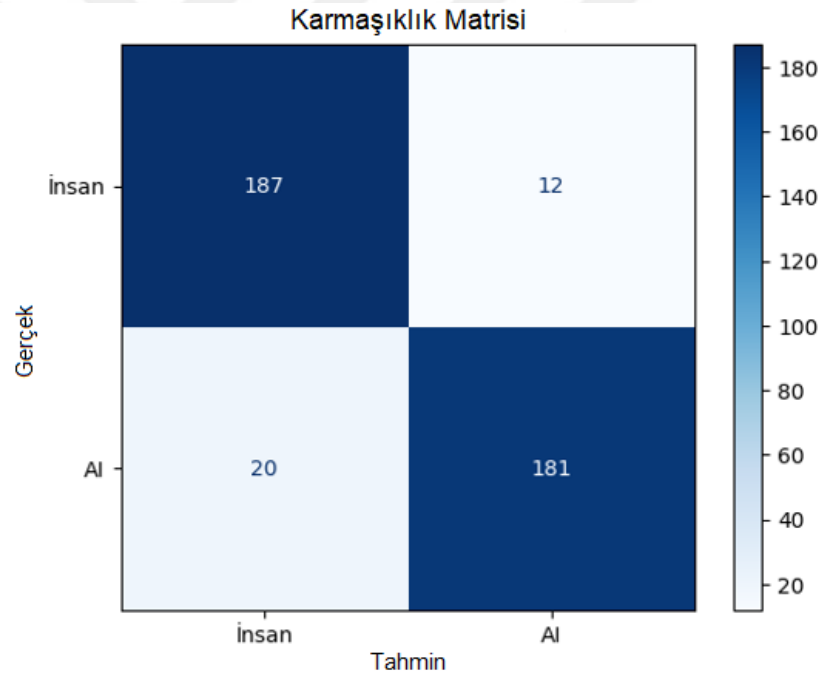


Şekil 4.29.b: GPT-2 modelinin epoch =10 için kayıp (loss) değeri grafiği

Şekil 4.29.a ve Şekil 4.29.b’de görüldüğü gibi, eğitim ve doğrulama kayıpları genel olarak düşüş eğilimindedir. Doğrulama doğruluğu 0.9 seviyelerine ulaşmış ve stabil seyretmiş, modelin dengeli fakat sınırlı bir öğrenme performansı gösterdiği gözlemlenmiştir.

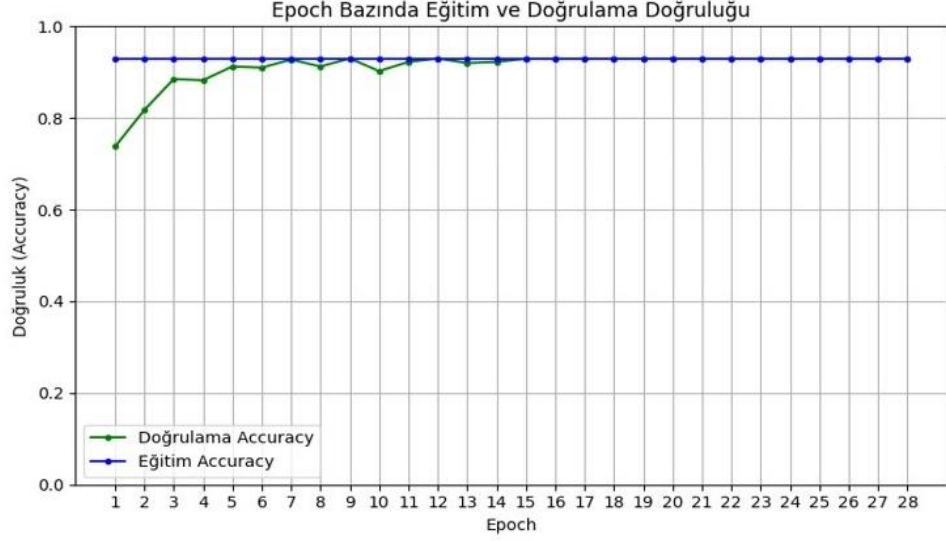
Tablo 4.17: epoch =10 için GPT-2 modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

GPT-2 modeli epoch=10 için doğruluk: 0.84				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.90	0.94	0.92	200
insan	0.94	0.90	0.92	200
doğruluk			0.92	400
makro ort.	0.92	0.92	0.92	400
ağırlıklı ort.	0.92	0.92	0.92	400

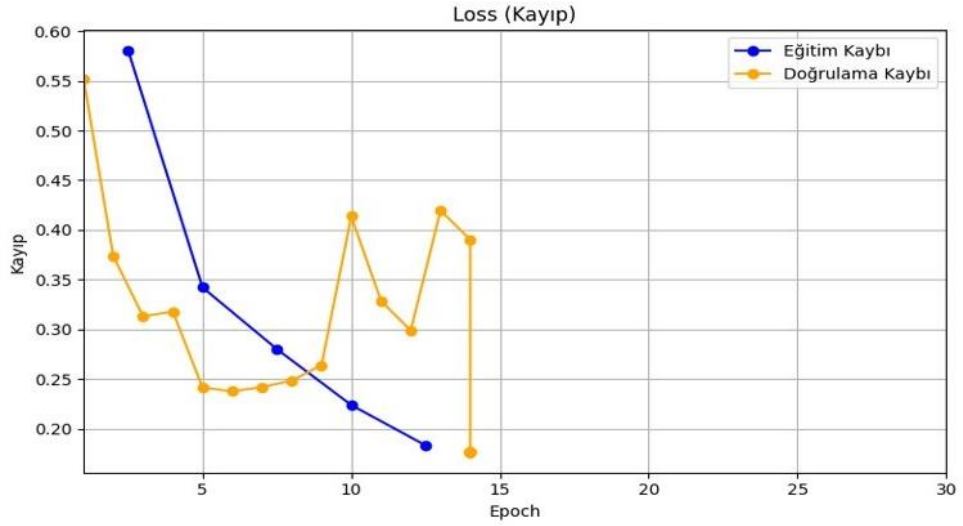


Şekil 4.30: GPT-2 modelinin epoch =10 için Karmaşıklık matrisi

Şekil 4.30’da görüldüğü gibi GPT-2 modeli, insan sınıfında 187 ve AI sınıfında 181 örneği doğru sınıflandırmıştır. Hatalı tahmin sayılarının (12 insan, 20 ai) benzer olması, modelin dengeli ancak sınırlı genelleme performansına sahip olduğunu ve yaklaşık 0.92 f1-score değerine ulaştığını göstermektedir.



Şekil 4.31.a: GPT-2 modelinin epoch =14 için accuracy (doğruluk)değerleri

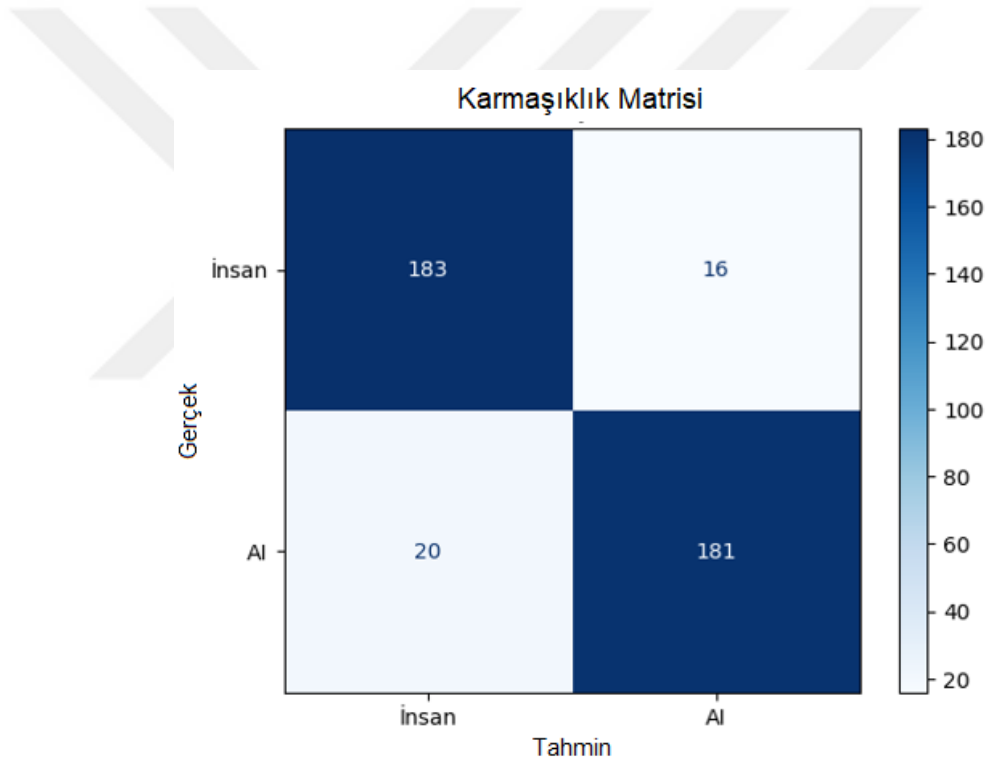


Şekil 4.31.b: GPT-2 modelinin epoch =14 için kayıp (loss) değerleri

Şekil 4.31.a ve Şekil 4.31.b’de görüldüğü gibi, eğitim doğruluğu 1.0 seviyesinde sabitlenirken doğrulama doğruluğu 0.90 civarında dengede kalmıştır. Eğitim kaybı sürekli azalırken doğrulama kaybı dalgalı bir seyir izlemiştir. Bu durum modelin yüksek öğrenme başarısı göstermesine rağmen genelleme performansında kararsızlık yaşadığını göstermektedir.

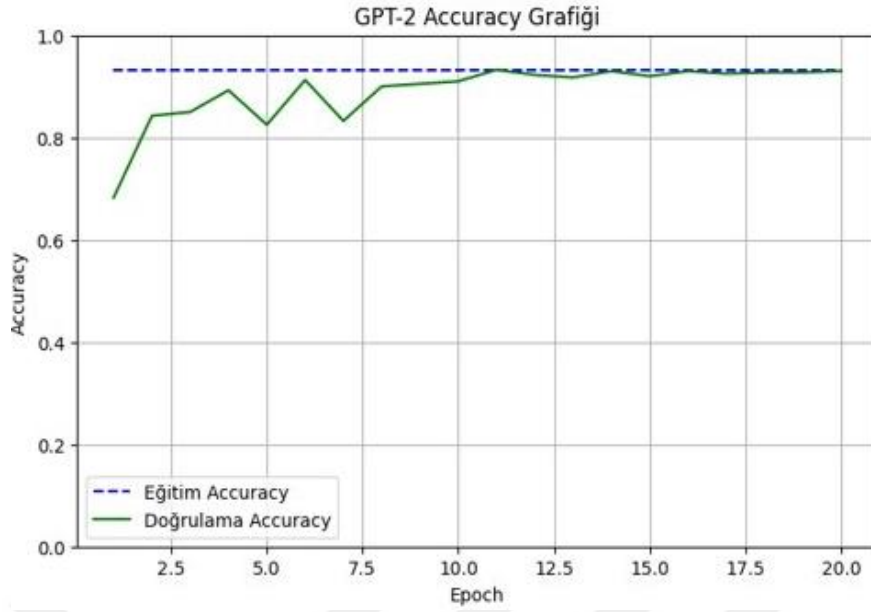
Tablo 4.18: epoch =14 için GPT-2 modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

GPT-2 modeli epoch=14 için doğruluk: 0.91				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.90	0.92	0.91	200
insan	0.92	0.90	0.91	200
doğruluk			0.91	400
makro ort.	0.91	0.91	0.91	400
ağırlıklı ort.	0.91	0.91	0.91	400

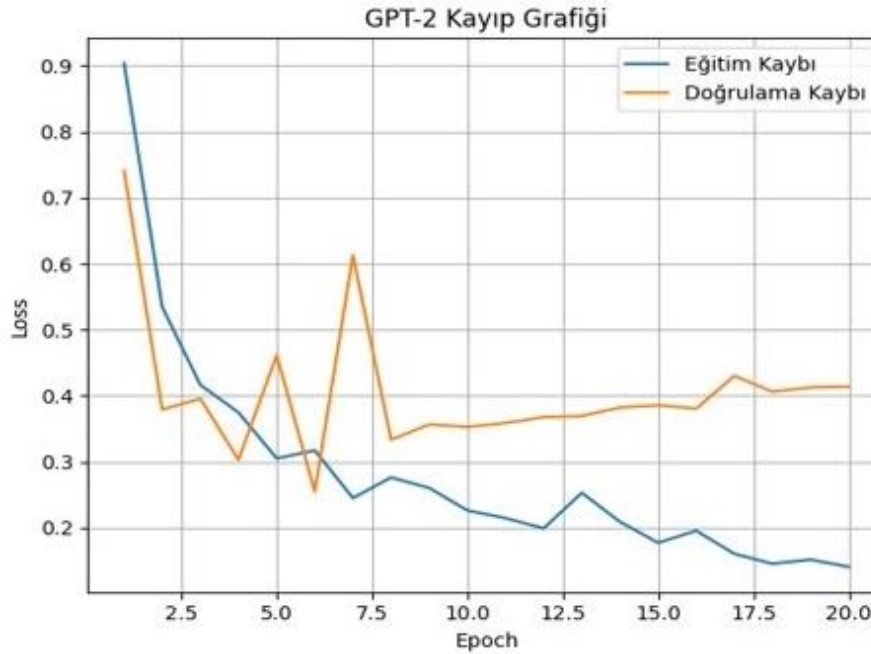


Şekil 4.32: GPT-2 modelinin epoch =14 için karmaşıklık matrisi

Şekil 4.32’de görüldüğü gibi GPT-2 modeli, AI sınıfında 181 ve insan sınıfında 183 örneği doğru sınıflandırmıştır. Hatalı tahminlerin benzer sayıda olması (20 ai, 16 insan), modelin dengeli ancak sınırlı genelleme performansı sergilediğini ve genel doğruluk oranınının 0.91 düzeyinde olduğunu göstermektedir.



Şekil 4.33.a: GPT-2 modelinin epoch =20 için doğruluk değeri grafiği

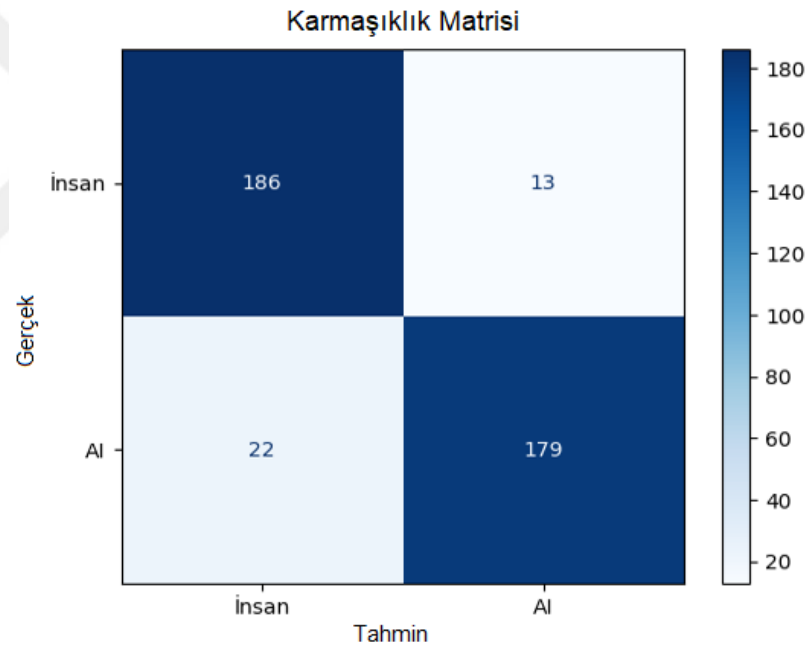


Şekil 4.33.b: GPT-2 modelinin epoch =20 için loss değeri grafiği

Şekil 4.33.a ve Şekil 4.33.b’de görüldüğü gibi, eğitim doğruluğu 1.0 seviyesinde sabitlenmiş olup doğrulama doğruluğu ise yaklaşık 0.90 civarında dengede kalmıştır. Eğitim kaybı azalma eğilimindeyken doğrulama kaybı dalgalı bir seyir izleyip artış eğilimi göstermiştir. Bu durum, modelin öğrenme kapasitesine ulaştığını ve kısmi overfitting belirtileri göstermeye başladığını ortaya koymaktadır.

Tablo 4.19: epoch = 20 için GPT-2 modelinin doğruluk, kesinlik, duyarlılık, f1-skoru ve örnek sayısı değerleri

GPT-2 modeli epoch=20 için doğruluk: 0.92				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.89	0.93	0.91	200
insan	0.93	0.98	0.91	200
doğruluk			0.91	400
makro ort.	0.91	0.91	0.91	400
ağırlıklı ort.	0.91	0.91	0.91	400



Şekil 4.34: GPT-2 modelinin epoch =20 için Karmaşıklık matrisi

Şekil 4.34'te görüldüğü gibi GPT-2 modeli, insan sınıfında 186 ve AI sınıfında 179 örneği doğru sınıflandırmıştır. Hatalı tahminlerin (13 insan, 22 ai) artmış olması, modelin genelleme performansının zayıfladığını ve overfitting etkisinin belirginleştiğini göstermektedir.

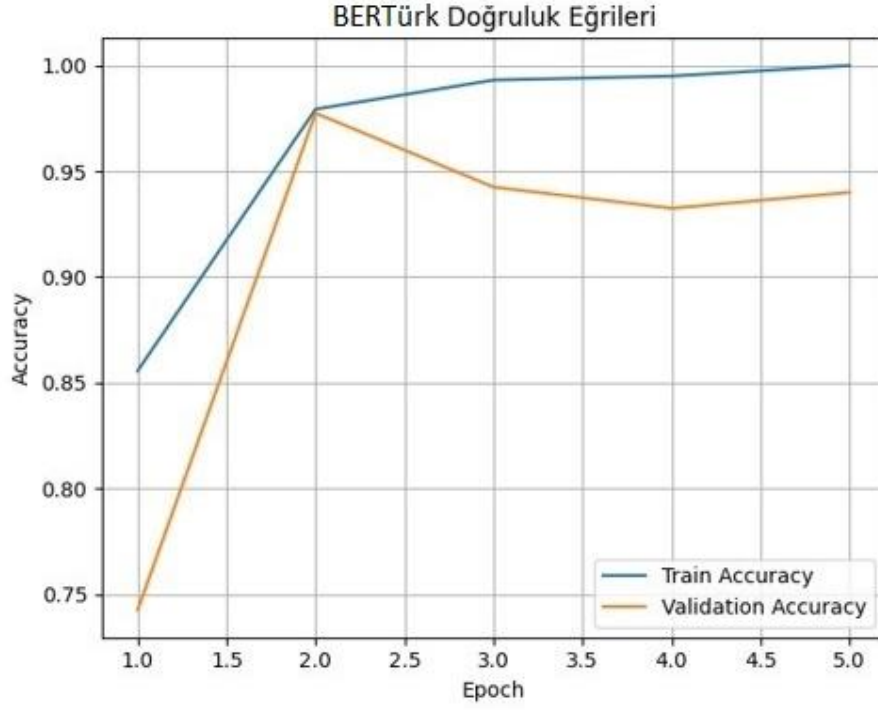
4.2.5. BERTurk Modeli

BERTurk modeli, önceden eğitilmiş bir Türkçe dil modeli olarak yüklenmiş ve bu çalışmaya özgü olarak iki sınıflı metin sınıflandırma görevi için yeniden eğitilmiştir. Eğitim sürecinde doğruluk, kayıp ve kaynak kullanımı (GPU, RAM) izlenmiş, model performansı doğrulama verisiyle değerlendirilmiştir. Ayrıca sistem kaynak kullanımı (Tablo 4.20) early stop değerine (epochs = 10) göre izlenmiştir. 5, 10 ve 20 epoch değerleri için doğruluk ve loss grafikleri sırasıyla Şekil 4.35.a, Şekil 4.35.b, Şekil 4.37.a, Şekil 4.37.b, Şekil 4.39.a ve Şekil 4.39.b’de yer alırken Tablo 4.21, Tablo 4.22 ve Tablo 4.23’ de aynı sıradaki epoch değerleri için doğruluk, precision, duyarlılık, f1-skoru ve örnek sayısı değerleri yer almaktadır. Şekil 4.36, Şekil 4.38 ve Şekil 4.40’ da her bir epoch için sırasıyla karmaşıklık matrisi bulunmaktadır.

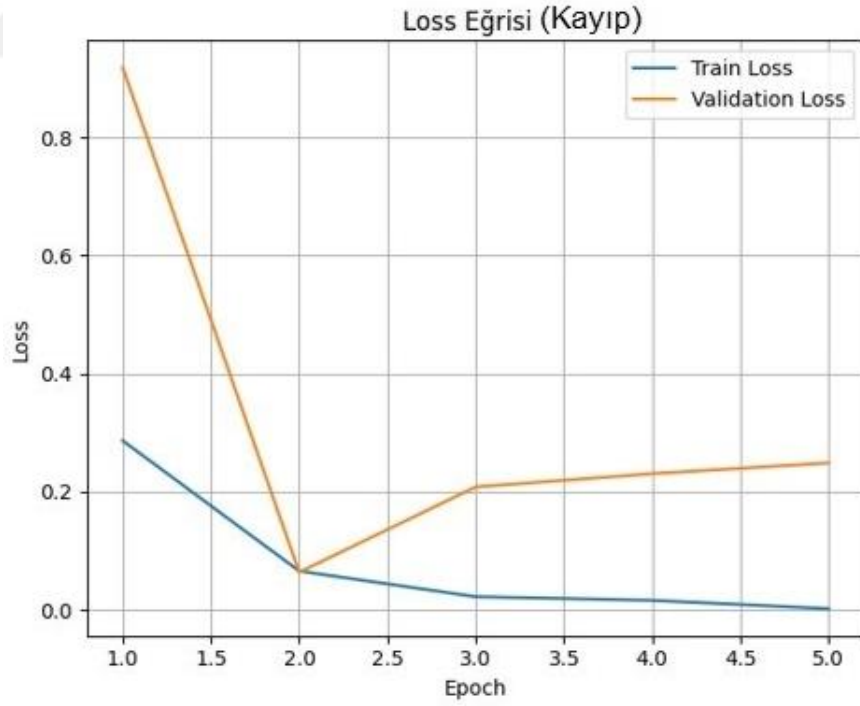
Tablo 4.20: Colab Notebook ortamında çalıştırılan BERTurk modelinin kaynak kullanımı

BERTurk Kaynak kullanımı	
Eğitim Süresi:	1 dk 43 sn
Maksimum GPU bellek kullanımı (allocated):	1.69 GB
Maksimum GPU bellek rezervasyonu (reserved):	2.29 GB
RAM Kullanımı:	% 4.5
Toplam RAM:	83.48 GB
Kullanılan RAM:	2.9 GB
CPU Kullanımı:	% 3.1

Şekil 4.35.a ve Şekil 4.35.b’de görüldüğü gibi, eğitim doğruluğu yaklaşık 1.0 seviyesine ulaşırken doğrulama doğruluğu 0.94 civarında sabitlenmiştir. Eğitim kaybı düşmeye devam ederken doğrulama kaybı artış eğilimi göstermiştir. Şekil 4.36’da ise modelin ai sınıfında oldukça başarılı olduğu (199 doğru tahmin) ancak insan sınıfında 23 örneği yanlış sınıflandırdığı görülmektedir.



Şekil 4.35.a: epoch =5 için BERTürk modelinin doğruluk grafiği

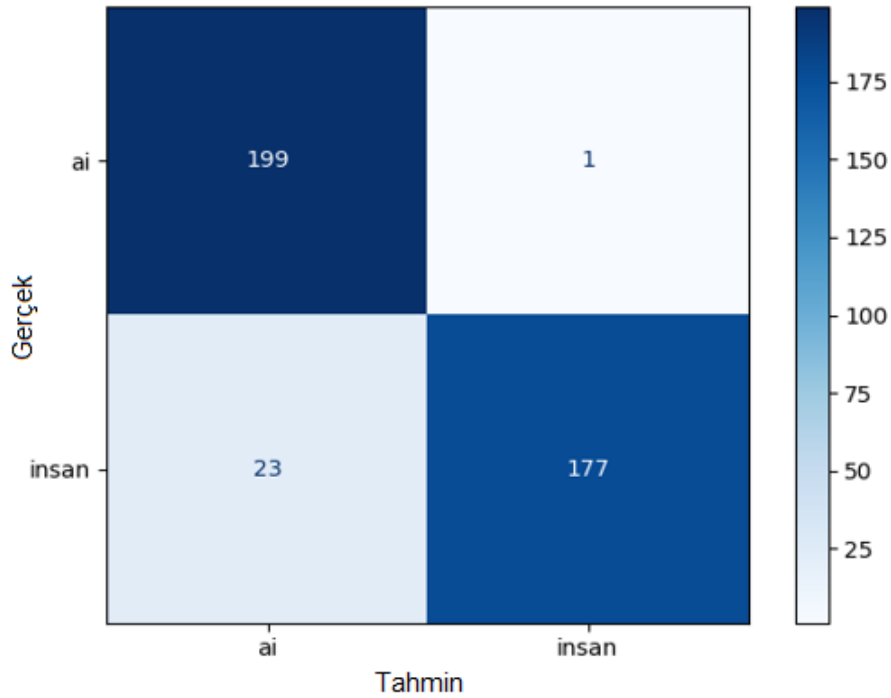


Şekil 4.35.b: epoch =5 için BERTürk modelinin kayıp grafiği

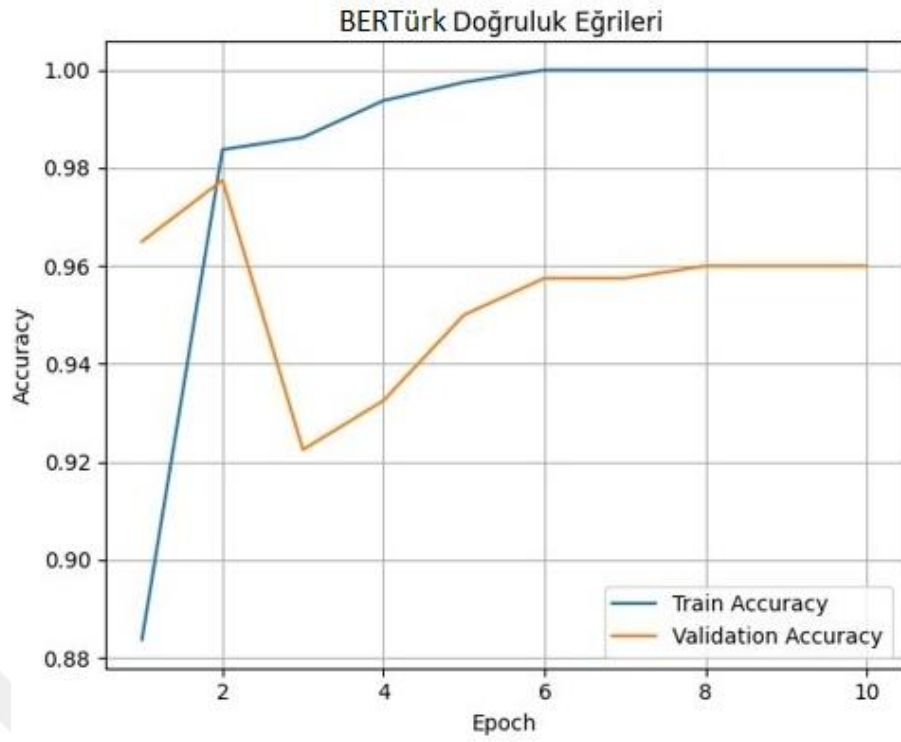
Tablo 4.21: epoch = 5 deęeri için BERTurk modelinin doęruluk, precision, duyarlılık, f1-skoru ve örnek sayısı deęerleri

epoch =5 için doęruluk: 0.94				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.90	0,99	0.94	200
insan	0,99	0.89	0.94	200
doęruluk			0.94	400
makro ort.	0.95	0.94	0.94	400
aęırlıklı ort.	0.95	0.94	0.94	400

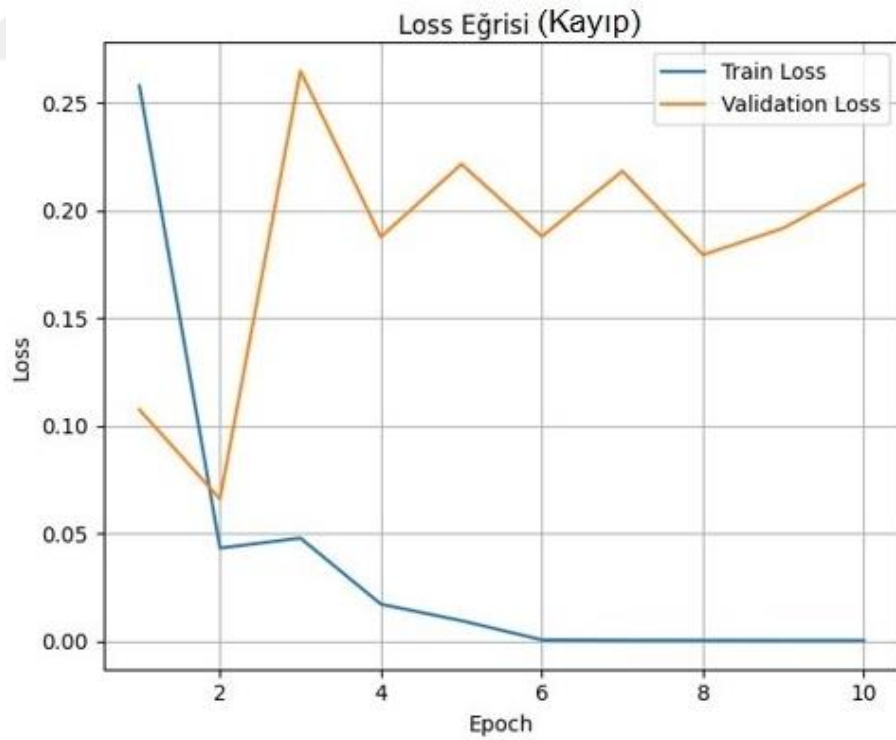
Karmaşıklık Matrisi



Şekil 4. 36: epoch =5 için BERTurk modelinin Karmaşıklık matrisi



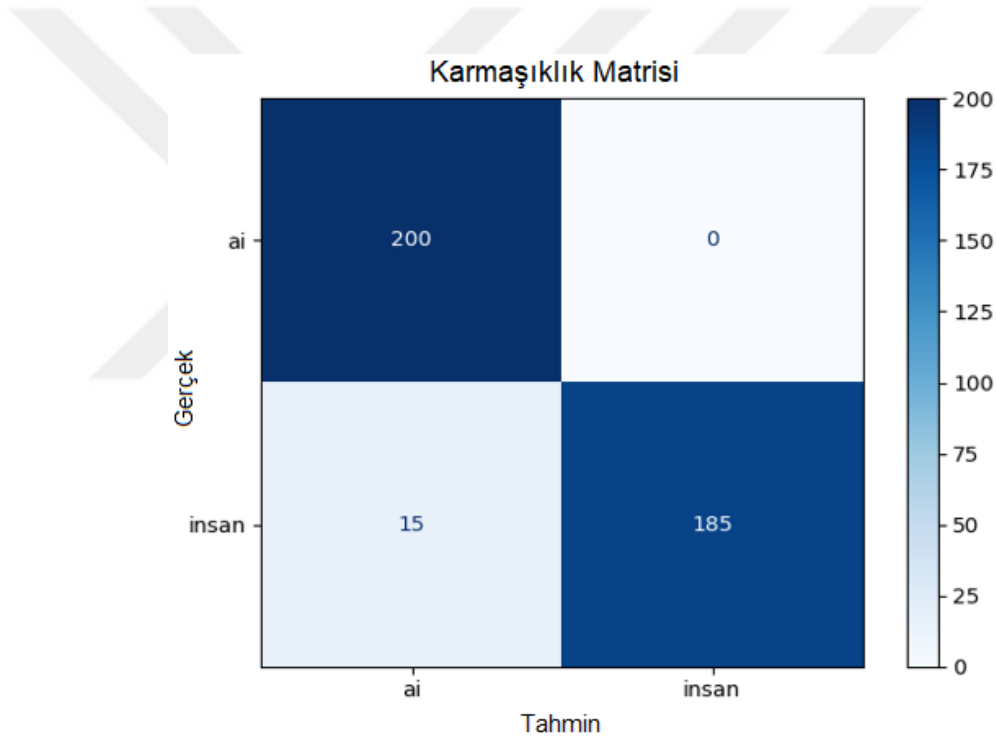
Şekil 4.37.a: epoch =10 için BERTürk modelinin doğruluk değeri grafiği



Şekil 4.37.b: epoch =10 için BERTürk modelinin kayıp değeri grafiği

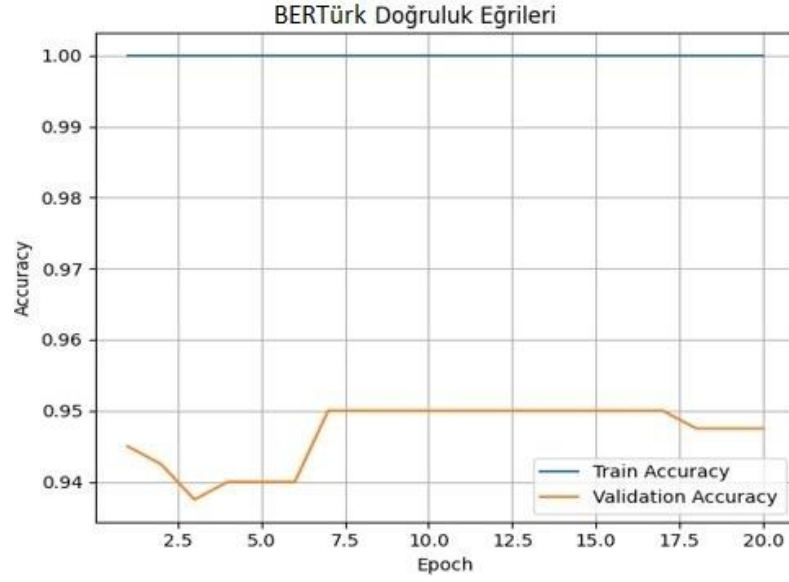
Tablo 4.22: BERTurk modelinde epoch =10 değeri için karmaşıklık matrisi

epoch = 10 için doğruluk: 0.96				
	kesinlik	duyarlılık	f1-skoru	örnek sayısı
ai	0.93	1.00	0.96	200
insan	1.00	0.92	0.96	200
doğruluk			0.96	400
makro ort.	0.96	0.96	0.96	400
ağırlıklı ort.	0.96	0.95	0.96	400

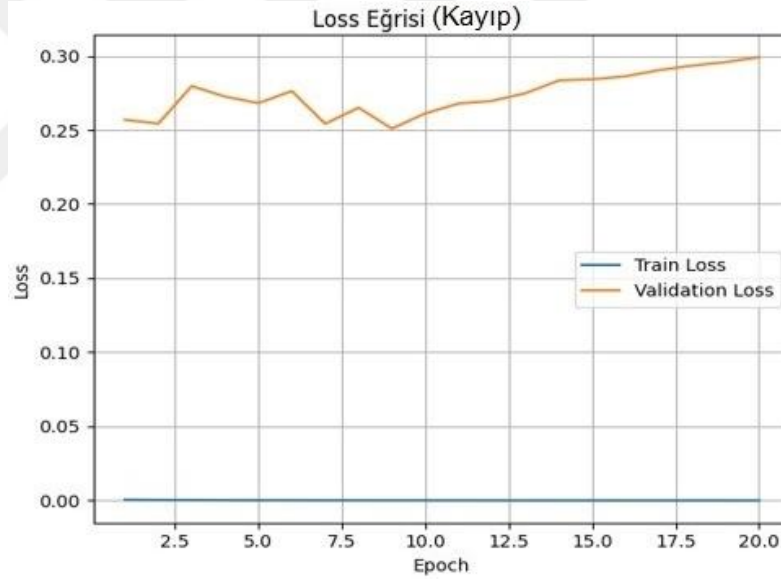


Şekil 4.38: BERTurk modelinde epoch =10 değeri için Karmaşıklık matrisi

Şekil 4.38’de görüldüğü üzere, BERTurk modeli epoch=10 için 0.96 doğruluk oranına ulaşmıştır. Model, ai sınıfındaki tüm örnekleri doğru sınıflandırırken (duyarlılık=1.0), insan sınıfında yalnızca 15 örneği hatalı tahmin etmiştir. Bu durum, modelin ai içeriklerini insan metinlerine göre daha doğru ayırt ettiğini göstermektedir. Genel olarak, kesinlik, duyarlılık ve f1-skoru değerlerinin 0.96 düzeyinde olmuştur.



Şekil 4.39.a: BERTürk modelinde epoch =20 değeri için doğruluk değeri grafiği

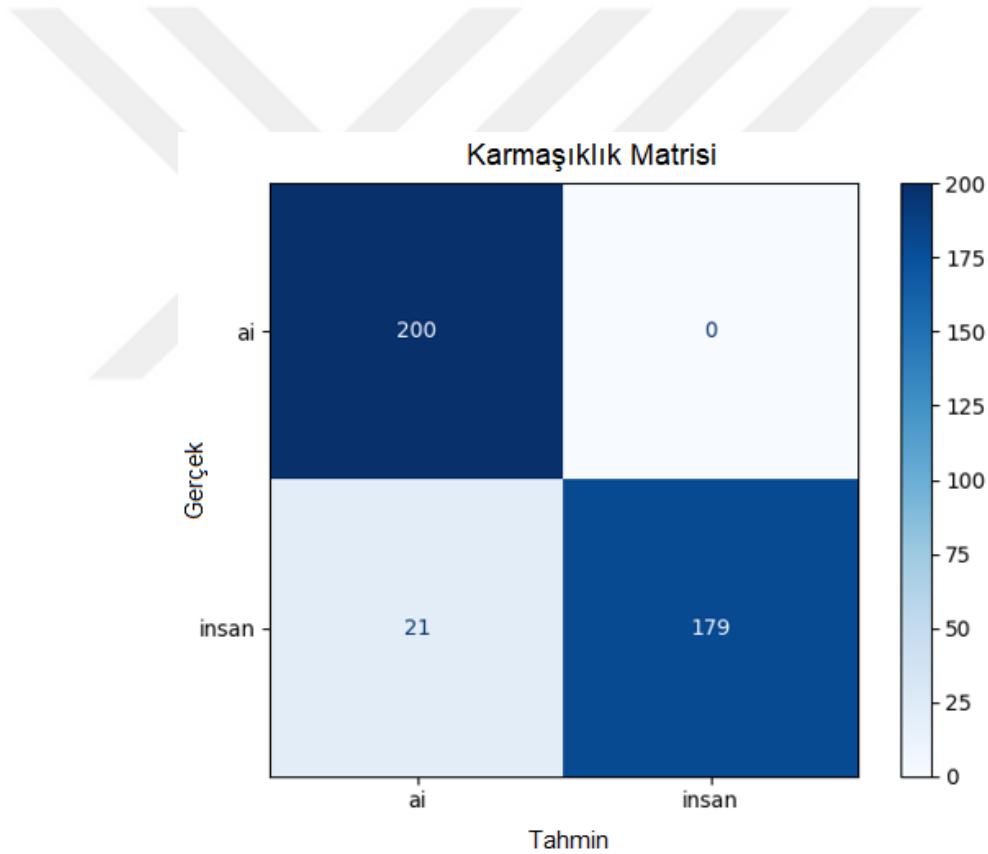


Şekil 4.39.b: BERTürk modelinde epoch =20 değeri için kayıp değeri grafiği

Şekil 4.39.a ve 4.39.b incelendiğinde, BERTürk modelinin eğitim doğruluğunun 1.0 düzeyinde sabit kaldığı, doğrulama doğruluğunun ise yaklaşık 0.94 - 0.95 aralığında seyrederek artık ilerleme göstermediği görülmektedir. Eğitim kaybı neredeyse sıfıra ulaşmışken, doğrulama kaybının 0.25 - 0.30 aralığında dalgalanması modelin öğrenmeyi tamamladığını ancak genelleme yeteneğinde sınırlı bir iyileşme kaldığını göstermektedir. Modelin 20. epoch itibarıyla öğrenme platosuna ulaştığını ve eğitim verisine aşırı uyum göstermeye başladığını işaret etmektedir.

Tablo 4.23: BERTurk modelinde epoch =20 deęeri için presicion duyarlılık, f1-skoru ve örnek sayısı deęerleri

epoch = 20 için doęruluk: 0.947				
	kesinlik	duyarlılık	f1-skor	örnek sayısı
ai	0.90	1.00	0.95	200
insan	1.00	0.90	0.94	200
doęruluk			0.95	400
makro ort.	0.965	0.95	0.95	400
aęırlıklı ort.	0.95	0.95	0.95	400

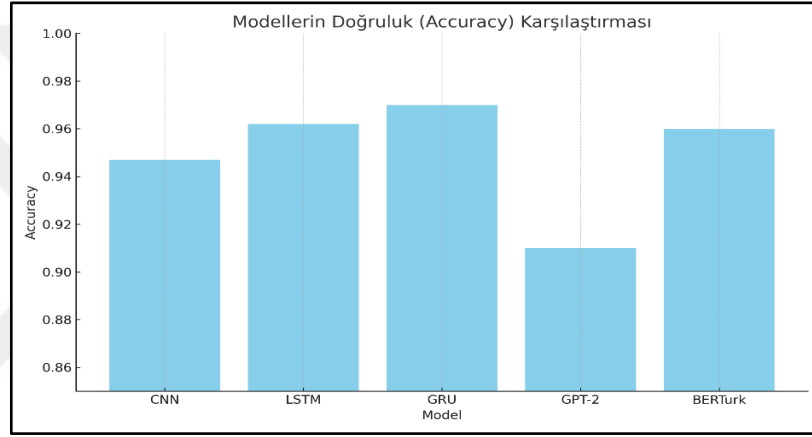


Şekil 4.40: BERTurk modelinde epoch =20 deęeri için karmaşıklık matrisi

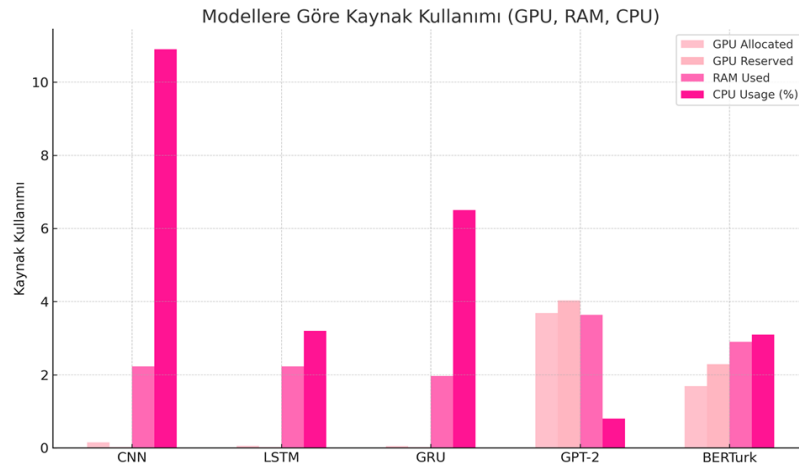
Şekil 4.40'ta görüldüęü üzere, BERTurk modeli epoch=20 için 0.95 doęruluk oranına ulaşmıştır. Model, ai sınıfındaki tüm örnekleri doęru sınıflandırırken, insan sınıfında 21 örneęi hatalı tahmin etmiştir.

4.3. Modellerin Performanslarının Karşılaştırılması

Modellerin eğitim süreçlerinde farklı epoch değerleri (5,10,20) ile birlikte early stopping uygulanmıştır. Her modelin doğrulama verisinde early stopping ile elde edilen epoch değeri referans alınmıştır. Bu bağlamda, GRU modeli 12. epoch' ta %97 doğrulukla en başarılı model olurken, onu sırasıyla LSTM (%96.2), BERTurk (%96), CNN (%94.7) ve GPT-2 (%91) takip etmiştir. Geleneksel derin öğrenme mimarileri (özellikle GRU ve LSTM) transformer tabanlı modellere göre sınıflandırma başarımı açısından daha yüksek doğruluk elde etmiş olup daha az kaynak (özellikle GPU belleği) tüketmişlerdir. Her bir modelin early stopping noktasındaki başarı metrikleri Şekil 4.41' de özetlenmiş olup kaynak kullanımlarını ise Şekil 4.42 de gösterilmiştir.



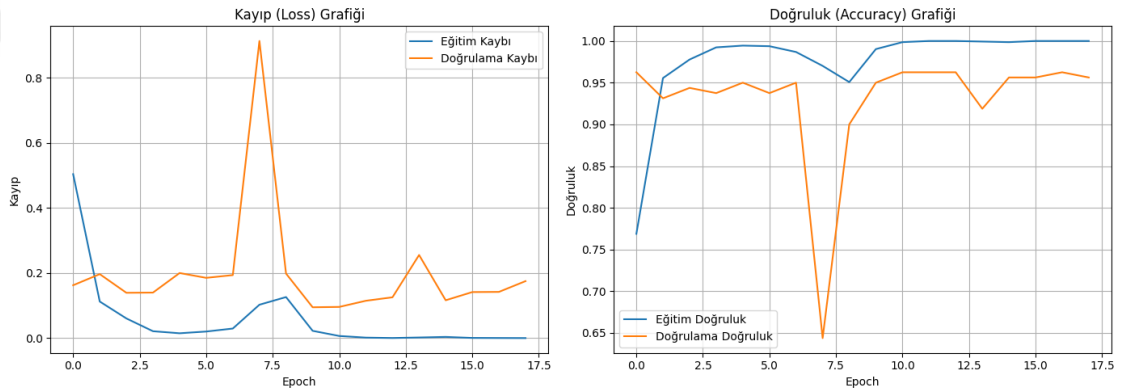
Şekil 4.41: Modellerin doğruluk değerlerinin karşılaştırılması



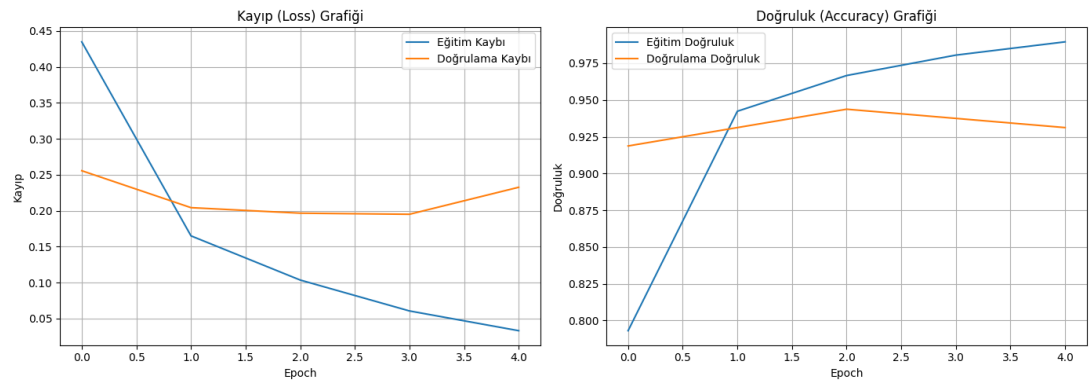
Şekil 4.42: Colab ortamında çalışan modellerin kaynak tüketimi

4.4. Bert + (CNN, LSTM , GRU) karşılaştırması

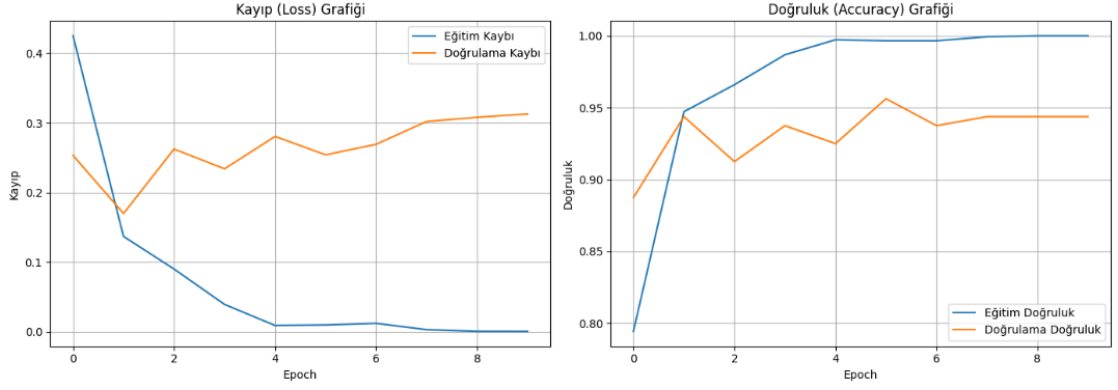
v0 veri seti kullanılarak makine öğrenmesi algoritmaları ile birlikte çalıştırılan kelime gömme yöntemlerinden en başarılı yönteminin (doğruluk değerinin en yüksek) BERT olmuştur. Derin öğrenme modellerinin sınıflandırma performansına katkısını gözlemlemek için BERT ile birlikte çalıştırılmış (v1 veri seti ile) ve sonuçları incelenmiştir. Şekil 5.3, Şekil 5.4, Şekil 5.5’ te sırasıyla BERT ile birlikte çalışan CNN, LSTM ve GRU modellerinin eğitim sürecindeki accuracy (doğruluk)ve kayıp (loss) grafikleri yer almaktadır. Her bir model için early stopping uygulandığı için grafiklerde farklı epoch değerleri görülmektedir.



Şekil 4.43: BERT + CNN birlikte çalışan modelin doğruluk ve kayıp değerleri



Şekil 4.44: BERT + LSTM birlikte çalışan modelin doğruluk ve kayıp değerleri

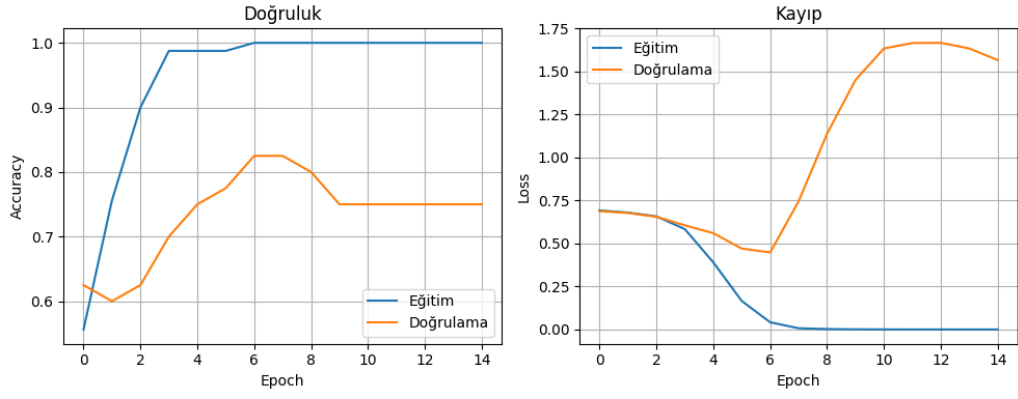


Şekil 4.45: BERT + GRU birlikte çalışan modelin doğruluk ve kayıp değerleri

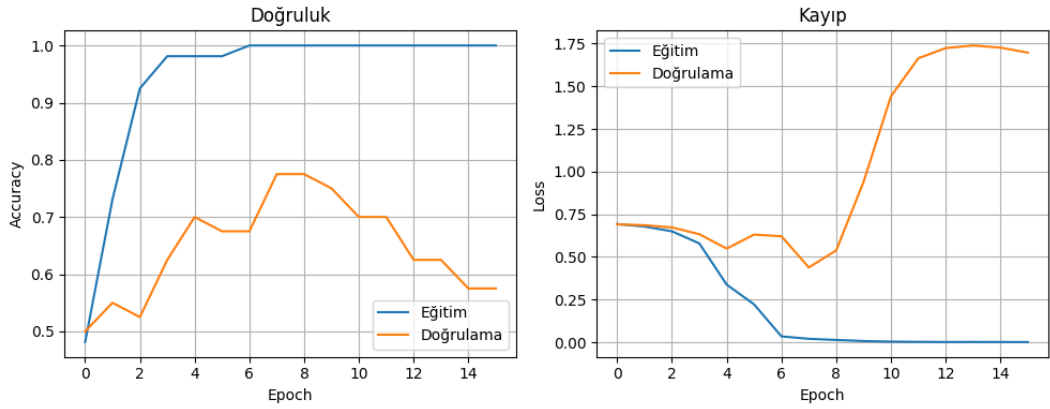
4.5. Caude ai ile modeli test

Modelimizin yalnızca ChatGPT metinlerini değil, genel anlamda farklı yapay zekâ sistemleri tarafından üretilmiş içerikleri tanıma kapasitesi değerlendirilmiştir. ChatGPT-4 tarafından üretilen v1 veri setimiz ile eğittiğimiz LSTM modelimizin farklı bir yapay zekâ aracı olan Claude.ai tarafından üretilen metinler üzerinde de sınıflandırma yeteneğine sahip olup olmadığı incelenmiş olup bu amaçla, mevcut veri setine ek olarak Claude.ai ile oluşturulmuş 100 adet yeni veri örneği (v2 veri seti) kullanılmıştır.

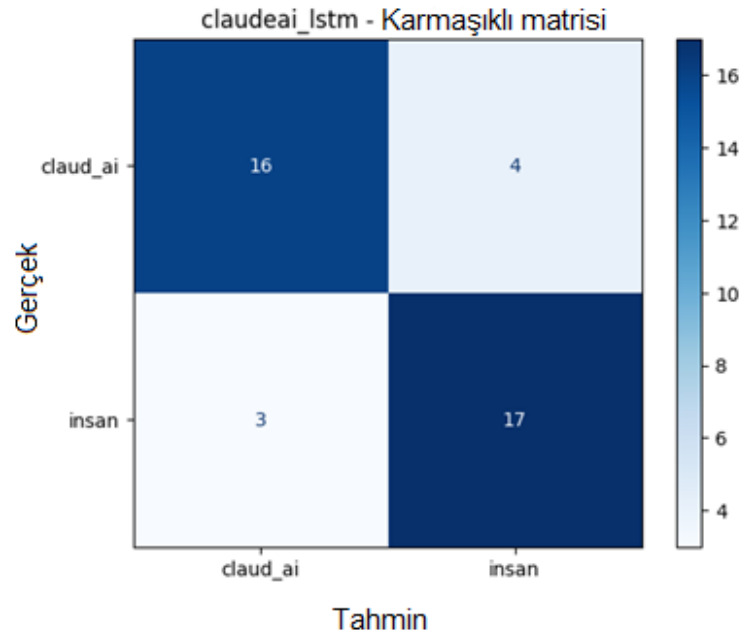
En başarılı modelin bu örnekler üzerindeki performansı ayrı ayrı değerlendirilmiştir. En başarılı model LSTM olarak karşımıza çıkmıştır. Yeni v2 veri seti ile sınıflandırma yapıldığında LSTM ağının özellikleri değiştirilmeden kullanılmıştır. Değerlendirme sürecinde hem Claude.ai hem de ChatGPT örnekleri için doğruluk (doğruluk), precision, duyarlılık ve f1-skoru gibi temel performans metrikleri hesaplanmış; ayrıca GPU, RAM ve CPU kaynak tüketimi bakımından da karşılaştırmalı analiz gerçekleştirilmiştir.



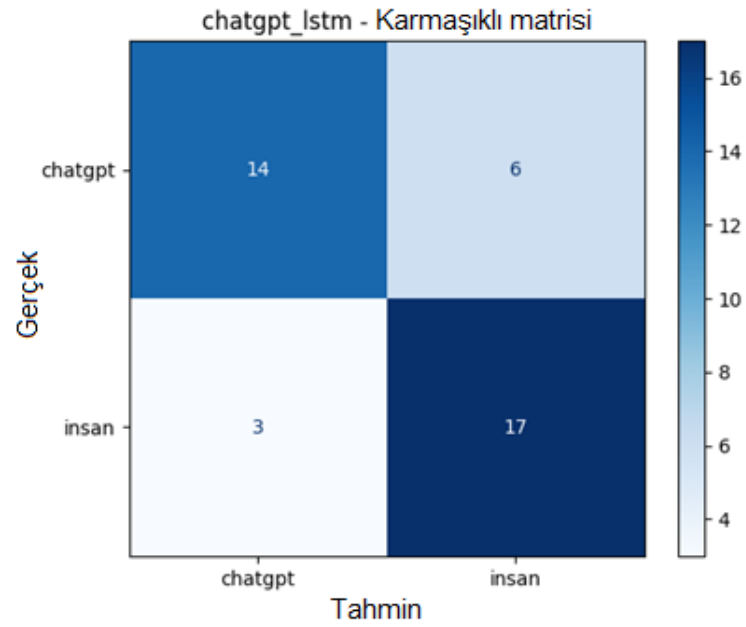
Şekil 4.46: 100 adet claude.ai ve 100 adet insan tarafından oluşturulan v2 veri seti ile sınıflandırma sonucunda doğruluk ve kayıp değerleri



Şekil 4.47: 100 adet chatgpt-4 ve 100 adet insan tarafından oluşturulan v2 veri seti ile sınıflandırma sonucunda doğruluk ve kayıp değerleri



Şekil 4.48: 100 adet claude.ai ve aynı başlık ve anahtar kelimeli 100 adet insan etiketli verilerin LSTM modeli ile sınıflandırma sonucu ortaya çıkan karmaşıklık matrisi



Şekil 4.49: 100 adet chatgpt-4 ve aynı başlık ve anahtar kelimeli 100 adet insan etiketli verilerin LSTM modeli ile sınıflandırma sonucu ortaya çıkan karmaşıklık matrisi

SONUÇ

Günümüzde, yapay zekâ teknolojilerinin oluşturduğu dökümanların doğruluğu ve kanıtlanabilirliği en büyük soru işaretlerinden biridir. Yapay zekâ sistemlerinin içerik üretme yetenekleri arttıkça, bu içeriklerin doğruluğu ve güvenilirliği önemli bir tartışma konusu haline gelmiştir. Yapay zekâ kaynaklı haberler, görseller veya ses kayıtları, doğrulanmadan paylaşıldıklarında toplumsal huzursuzluklara, ekonomik dalgalanmalara ve diplomatik krizlere neden olabilir. Sağlık, hukuk ve siyaset gibi önemli alanlarda yanlış bilgi yayılması büyük tehlikelere neden olabilir. Bu nedenle, yapay zekâ tarafından üretilen içeriklerin tespiti ve doğruluk analizi artık sadece akademik bir konu değil, aynı zamanda dünya çapında bir görevdir.

Bu çalışma, yapay zekâ tarafından üretilen metinleri insan yazımı içeriklerden ayırt etmeye yönelik olarak üç farklı veri seti (v0, v1, v2) üzerinde gerçekleştirilmiştir. İlk aşamada (v0), farklı kelime gömme teknikleri (TF-IDF, Word2Vec, FastText ve BERT) çeşitli makine öğrenmesi algoritmalarıyla birlikte test edilmiş ve BERT, en başarılı temsil yöntemi olmuştur. v1 veri setinde ise, BERT tabanlı ve BERT içermeyen derin öğrenme modelleri karşılaştırılmıştır. Daha sonra v2 veri seti üzerinden yapılan testlerde, v1 veri seti ile eğitilmiş en başarılı model olan LSTM' in, farklı bir yapay zekâ aracı olan Claude.ai tarafından oluşturulan metinleri de oldukça iyi seviyede bir doğruluk değeriyle ayırt edebildiği gözlemlenmiştir.

v1 veri seti ile eğitim aşamasında çeşitli epoch değerleri denenmiş ve erken durdurma stratejisi kullanılarak modellerin en verimli şekilde eğitildiği noktalar belirlenmiştir. Elde edilen sonuçlar, farklı model türlerinin doğruluk oranlarını ve kaynak kullanım verimliliğini karşılaştırmalı olarak ortaya koymaktadır. Ayrıca, bazı durumlarda geleneksel derin öğrenme mimarilerinin, transformer tabanlı modellere kıyasla daha yüksek doğruluk ve daha düşük donanım ihtiyacıyla öne çıkabildiği gözlemlenmiştir.

v1 veri seti üzerinde eğitilen modellerin eğitim süreçlerinde farklı epoch değerleri (5, 10, 16, 20) ile birlikte erken durdurma yöntemi uygulanmıştır. Her modelin doğrulama verisinde erken durdurma ile elde edilen epoch değerleri referans alınmıştır. Bu bağlamda, GRU modeli 12. epoch'ta 0.97 oranında en yüksek doğruluk değerine sahip model olurken, onu sırasıyla LSTM (0.96), BERTurk (0.96), CNN (0.94) ve GPT-2 (0.91) takip etmiştir. Geleneksel derin öğrenme mimarileri (özellikle GRU ve LSTM) transformer tabanlı modellere göre sınıflandırma başarımı açısından daha yüksek

doğruluk elde etmiş olup daha az kaynak (özellikle GPU belleği) tüketmişlerdir. Her bir modelin erken durdurma noktasındaki başarı metrikleri Tablo 5.1’de özetlenmiştir.

Tablo 5.1: Modellerin (early stop değeri) epoch değerleriyle çalıştırıldığındaki doğruluk, f1-skoru ve eğitim süresi

Model	epoch	doğruluk	f1-skoru	eğitim süresi dk : sn
CNN	16	0.947	0.95	0:53
LSTM	10	0.962	0.96	0:09
GRU	12	0.970	0.97	0:12
GPT-2	14	0.910	0.91	6:47
BERTurk	10	0.960	0.96	1:43

Tablo 5.1’e bakıldığında, eğitim süresi ve f1-skoru birlikte değerlendirildiğinde LSTM modeli hem kısa sürede eğitilmiş olması hem de yüksek performansı sayesinde en verimli seçenek olarak öne çıkmıştır. GRU da benzer bir başarı gösterse de eğitimi LSTM modeline göre biraz daha uzun sürmüştür. BERTurk ise yüksek doğruluk değeri üretmesine rağmen, eğitim süresi oldukça uzun sürmüştür. CNN, genel performans açısından diğer modellerin gerisinde kalırken; GPT-2, en uzun eğitim süresine sahip olması nedeniyle verimlilik açısından dezavantajlı bir konumda yer almıştır.

Yapılan deneysel çalışmalar sonucunda, her bir modelin öğrenme kapasitesine ve mimari yapısına bağlı olarak farklı epoch değerlerinde değişen öğrenme eğilimleri gözlemlenmiştir. Bazı modellerde belirli epoch aralıklarında aşırı öğrenme belirtileri gözlemlenmiş olsa da, birçok durumda yüksek doğrulukla beraber genelleme başarımı da korunmuştur. CNN modeli yaklaşık 10. epoch’ tan sonra eğitim doğruluğunun 0.99 oranına ulaştığı, doğrulama doğruluğunun ise 0.95 civarında sabitlendiği görülmüş ve 16. epoch’ ta overfitting belirtileri ortaya çıkmıştır. LSTM modeli 0.98 doğrulukla en yüksek performansı elde etmiş olup 8 - 12. epoch aralığında doğrulama kaybında artış gözlenmiştir. Hem eğitim hem doğrulama eğrilerinin paralel seyretmesiyle birlikte en kararlı ve güvenilir sonuçlardan birini epoch=5 civarında üretmiştir. GRU modeli ise 0.96 doğruluk ile istikrarlı bir genelleme başarımı sergileyerek LSTM’ e kıyasla daha az parametreye sahip olması nedeniyle overfitting eğilimi daha geç, yaklaşık 12. epoch

civarında ortaya çıkmıştır. GPT-2 modeli 0.91 doğruluk düzeyine ulaşmıştır. Yüksek parametrelili yapısı nedeniyle küçük veri setinde erken epoch değerlerinde aşırı öğrenme eğilimi göstermiştir. BERTurk modeli ise %96 doğruluk oranıyla başarılı bir sonuç elde etmişse de uzun süreli fine-tuning (10 - 20 epoch) işlemlerinde pretrained ağırlıkların bozulmasıyla küçük dalgalanmalar göstermiştir. Veri setinin dengeli fakat sınırlı boyutu ile modellerin farklı karmaşıklık düzeyleri birlikte değerlendirildiğinde, özellikle LSTM, GRU ve CNN modelleri belirli epoch aralıklarında overfitting oluşmadan yüksek doğruluk (0.95-0.98) ve güçlü genelleme kabiliyeti göstermiştir. Bu bulgular, uygun epoch seçimi, dropout oranı ve erken durdurma stratejilerinin derin öğrenme modellerinin güvenilir biçimde optimize edilmesinde kritik bir rol oynadığını ortaya çıkarmıştır.

İlerleyen çalışmalarda, overfitting etkilerini azaltmak amacıyla farklı yapay zekâ modellerinden üretilmiş metinlerle veri çeşitliliğinin artırılması ve regularization tekniklerinin (örneğin L2 veya weight decay) optimize edilmesi uygulanabilir bir yaklaşım olarak değerlendirilmektedir. Ayrıca, dropout oranlarının yeniden ayarlanması ve katman dondurma (layer freezing) stratejilerinin dikkatli biçimde kullanılmasıyla modellerin genelleme kapasitesinin daha da güçlendirilebileceği düşünülebilir.

Modelimizin doğruluk ve genelleme başarımını mevcut literatürdeki yaklaşımlar bağlamında değerlendirmek amacıyla, benzer çalışmalardan elde edilen sonuçlar Tablo 5.2' de karşılaştırmalı olarak verilmiştir.

Tablo 5.2 : Literatürdeki benzer çalışmalarla yapılan karşılaştırma tablosu

Çalışma	Yapay Zeka Aracı	Dil	Örnek Sayısı	Yöntem	Accuracy (test)	F1-score
(Schaaff, Schlippe ve Mindner 2024)	Chatgpt	İngilizce, Fransızca, Almanca, İspanyolca	300 300 300 300	XGBoost Random Forest MLP	0.90	0.90
					0.94	0.94
					0.94	0.93
					0.94	0.94
					0.98	0.98
					0.95	0.95
					0.97	0.97
					0.99	0.99
					0.87	0.87
					0.88	0.89
0.87	0.86					
0.90	0.90					
(Bafna ve ark 2024)	ChatGP,LLaMa, Cohere, Davinci, Dolly-v2,	İngilizce	125000	RoBERTa fine tuned RoBERTa +BiLSTM RoBERTa +GRU	0.88	0.88
					0.80	0.80
					0.80	0.84
(Wang, Li ve Li, 2024)	Belirtilmemiş	İngilizce	1378	BERT	0.977	-
(Wani ve ark., 2024)	ChatGPT, Bard	İngilizce	20000	BERT+CNN BERT+LSTM BERT+GRU	0.98	0.99
					0.98	0.99
					0.97	0.98
(Guo ve ark. 2023)	ChatGPT	İngilizce, Çince	12.575 11.683	RoBERTa fine tuned,	0.98	0.98
					0.96	0.96
(Er & Çoğalmış, 2024)	ChatGPT-3.5	Türkçe	1941	LSTM	0.88	-
Önerilen çalışma	ChatGPT-4	Türkçe	2000	CNN LSTM GRU BERTürk GPT-2	0.95	0.96
					0.98	0.98
					0.97	0.97
					0.96	0.96
					0.91	0.92

Çalışmalardaki dil farklılıkları, veri seti boyutları ve kullanılan modellerin çeşitliliği, doğrudan bir performans karşılaştırması yapmayı güçleştirmektedir. Ancak genel olarak derin öğrenme tabanlı modellerin (özellikle BERT tabanlı yapılar, LSTM ve GRU) metnin insan mı yoksa yapay zekâ tarafından mı üretildiğini yüksek doğrulukla ayırt edebildiği görülmektedir. Bununla beraber tabloda önerilen çalışma kısmında yer alan doğruluk değerleri, overfitting gözlenmeyen ve genelleme başarımı en yüksek olan epoch değerlerinden yola çıkarak tercih edilmiştir. Bu nedenle, sunulan sonuçların yalnızca yüksek doğruluk değil aynı zamanda modelin güvenilirliğini ve kararlılığını da yansıttığı düşünülmektedir. Son olarak önceki çalışmamızda (Er & Coğalmış, 2024) tablodaki yerini almıştır. ChatGPT-3.5 tarafından üretilen Türkçe akademik metinleri içermekte olup LSTM modeliyle 0.88 doğruluk elde edilmiştir. Bu çalışmada ise veri seti ChatGPT-4 ile güncellenmiş, örnek sayısı artırılmış ve CNN, LSTM, GRU, BERTürk, GPT-2 gibi farklı modellerle karşılaştırma yapılmıştır. Sonuçlar, özellikle LSTM modelinde 0.98 doğruluk ve 0.98 f1-skoru ile önceki çalışmaya kıyasla belirgin bir gelişme göstermiştir.

İnsan ve yapay zekâ tarafından üretilmiş Türkçe akademik makale özetlerinden elde edilen metinleri sınıflandırmaya yönelik beş farklı model incelenerek karşılaştırılmıştır. Uygulanan modeller arasında CNN, LSTM, GRU gibi klasik derin öğrenme mimarileri ile önceden GPT-2, BERTürk gibi eğitilmiş transformer tabanlı modeller yer almış olup, modellerin hem sınıflandırma başarımı hem de sistem kaynak kullanımları (GPU, RAM, CPU) detaylı şekilde analiz edilmiştir. Transformer tabanlı modeller yüksek doğruluk sağlasa da özellikle GPT-2'nin eğitim süresi ve GPU bellek kullanımı oldukça yüksektir. Buna karşılık GRU ve LSTM gibi modeller daha az kaynak kullanarak oldukça yüksek performans sunmuştur. Bu durum, kaynak kısıtlı sistemlerde geleneksel modellerin hâlâ güçlü bir alternatif olduğunu göstermektedir. Sonuç olarak, yapay zekâ destekli metin tespiti gibi görevlerde model başarımı kadar verimlilik ve kaynak kullanımı da dikkate alınmalıdır. LSTM modeli, bu çalışmada hem doğruluk hem de verimlilik açısından en dengeli sonuçları sunan model olmuştur. Geliştirilecek sistemlerin ölçeklenebilirliği açısından bu tür karşılaştırmalar önemli bir yol gösterici olacaktır.

KAYNAKÇA

- Abdullahayik, A. (t.y.). *BERTurk ile uçtan uca metin sınıflandırma*. Erişim adresi: <https://apdullahayik.medium.com/end-to-end-text-classification-with-berturk-a859d64aa265>
- Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A., Sumiea, E. H., Alqushaibi, A., & Ragab, M. G. (2024). RNN-LSTM: From applications to modeling techniques and beyond—Systematic review. *Journal of King Saud University-Computer and Information Sciences*, 102068.
- An, J., Ding, W. ve Lin, C. (2023). Correspondence: ChatGPT: Tackle the growing carbon footprint of generative AI. *Nature*, 615, 586. <https://doi.org/10.1038/d41586-023-00909-6>
- Analytics Vidhya. (2023, Ağustos). Creating BERT embeddings with Hugging Face Transformers. Erişim adresi: <https://www.analyticsvidhya.com/blog/2023/08/bert-embeddings/>
- Analytics Vidhya. (2023, Ocak). Introduction to FastText embeddings and its implication. Erişim adresi: <https://www.analyticsvidhya.com/blog/2023/01/introduction-to-fasttext-embeddings-and-its-implication/>
- Aydın, A. (t.y.). 1-Text preprocessing techniques for NLP. Medium. Erişim adresi: <https://ayselaydin.medium.com/1-text-preprocessing-techniques-for-nlp-37544483c007>
- Bafna, J. S., Mittal, H., Sethia, S., Shrivastava, M., & Mamidi, R. (2024). Mast kalendar at semeval-2024 task 8: On the trail of textual origins: Roberta-bilstm approach to detect ai-generated text. arXiv preprint arXiv:2407.02978.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5, 135-146.
- Bharadiya, J. (2023). A comprehensive survey of deep learning techniques natural language processing. *European Journal of Technology*, 7(1), 58-66.
- Bhattacharjee, A., & Liu, H. (2023). Fighting Fire with Fire: Can ChatGPT Detect

- AIgenerated Text? arXiv preprint arXiv:2308.01284.
- Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, 25(2), 197–227. <https://doi.org/10.1007/s11749-016-0481-7>
- Capital One. (t.y.). Understanding TF-IDF for machine learning. Erişim adresi: <https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/>
- Chaka, C. (2023). Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools. *Journal of Applied Learning and Teaching*, 6(2).
- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Coursera. (t.y.). What are stop words? Erişim adresi: <https://www.coursera.org/articles/what-are-stop-words>
- DataCamp. (2023, 14 Kasım). An Introduction to Convolutional Neural Networks (CNNs). Erişim adresi: <https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns>
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).
- Elkhatat, A. M., Elsaid, K., & Almeer, S. (2023). Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *International Journal for Educational Integrity*, 19(1), 17.
- Gehrmann, S., Strobelt, H., & Rush, A. M. (2019). Gltr: Statistical detection and visualization of generated text. arXiv preprint arXiv:1906.04043.
- George, A. S. ve George, A. H. (2023). A review of ChatGPT AI's impact on several business sectors. *Partners Universal International Innovation Journal*, 1(1), 9–23.

- Google. (t.y.). Word2Vec. Erişim adresi: <https://code.google.com/archive/p/word2vec/>
- Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., & Wu, Y. (2023). How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection. Hasib, K. M., Naseem, U., Keya, A. J., Maitra, S., Mithu, K., & Alam, M. G. R. (2023). A systematic review on airlines industries based on sentiment analysis and topic modeling.
- Hu, X., Chen, P. Y., & Ho, T. Y. (2023). Radar: Robust ai-text detection via adversarial learning. arXiv preprint arXiv:2307.03838.
- Hugging Face. (t.y.). GPT 2 [Model documentation]. Erişim adresi: https://huggingface.co/docs/transformers/model_doc/gpt2
- Hugging Face. (t.y.). GPT2ForSequenceClassification [Model documentation]. Erişim adresi: https://huggingface.co/docs/transformers/model_doc/gpt2#transformers.GPT2ForSequenceClassification
- IBM. (2023, Aralık 27). What are örnek sayısı vector machines (SVMs)? Erişim adresi: [https://www.ibm.com/topics/örnek sayısı-vector-machine](https://www.ibm.com/topics/örnek-sayısı-vector-machine)
- IBM. (t.y.). What is the k-nearest neighbors (KNN) algorithm? Retrieved August 7, 2025, Erişim adresi: <https://www.ibm.com/topics/knn>
- Idrees, H. (t.y.). RNN, LSTM ve GRU karşılaştırması: Sıralı veri modelleme için kapsamlı bir rehber. Erişim adresi: <https://medium.com/@hassaanidrees7/rnn-vs-lstm-vs-gru-a-comprehensive-guide-to-sequential-data-modeling-03aab16647bb>
- Kang, Y., Cai, Z., Tan, C. W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics*, 7(2), 139-172.
- Karadeniz, T., Tokdemir, G. ve Maraş, H. H. (2024, Ekim). Spam detection with FastText based features. 2024 Innovations in Intelligent Systems and Applications Conference (ASYU) (ss. 1–5). IEEE.
- Kashishdafa. (2024, March 23). Gaussian Naive Bayes: understanding the basics and applications. Erişim adresi: <https://medium.com/@kashishdafa0410/gaussian->

- Kashnitsky, Y., Herrmannova, D., de Waard, A., Tsatsaronis, G., Fennell, C., & Labbé, C. (2022, October). Overview of the DAGPap22 shared task on detecting automatically generated scientific papers. In *Third Workshop on Scholarly Document Processing*
- Khalil, M., & Er, E. (2023, June). Will ChatGPT Get You Caught? Rethinking of Plagiarism Detection. In *International Conference on Human-Computer Interaction* (pp. 475-487). Cham: Springer Nature Switzerland.
- Khan, A., Fouda, M. M., Do, D. T., Almaleh, A., & Rahman, A. U. (2023). Short-term traffic prediction using deep learning long short-term memory: taxonomy, applications, challenges, and future trends. *IEEE Access*
- Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia tools and applications*, 82(3), 3713-3744.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., & Zou, J. (2023). GPT detectors are biased against non-native English writers. *arXiv preprint arXiv:2304.02819*.
- Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI open*, 3, 111-132.
- Liu, X. Y. (2015, May). Private colleges teachers evaluation system based on support vector machine (svm). In *2015 International conference on Applied Science and Engineering Innovation* (pp. 1918-1921). Atlantis Press.
- Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., & Lloret, J. (2017). Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. *IEEE access*, 5, 18042-18050.
- Ma, Y., Liu, J., Yi, F., Cheng, Q., Huang, Y., Lu, W., & Liu, X. (2023). AI vs. human—differentiation analysis of scientific content generation. *arXiv*, 2301.
- Mitrović, S., Andreoletti, D., & Ayoub, O. (2023). Chatgpt or human? detect and explain. *explaining decisions of machine learning model for detecting short*

chatgptgenerated text. arXiv preprint arXiv:2301.13852.

Naiem, S., Khedr, A. E., Idrees, A. M., & Marie, M. I. (2023). Enhancing the efficiency of Gaussian naïve Bayes machine learning classifier in the detection of DDoS in cloud computing. *IEEE Access*, 11, 124597–124608. <https://doi.org/10.1109/ACCESS.2023.3328951>

Nerdnodes2023. (t.y.). Understanding Convolutional Neural Networks (CNN) —Deep Learning. Erişim adresi: <https://nerdnodes2023.medium.com/understanding-convolutional-neural-networks-cnn-deep-learning-d772a12c0397>

Nvidia. (2019, Kasım 22). Training and fine-tuning BERT using NVIDIA NGC. Nvidia Technical Blog. <https://developer.nvidia.com/blog/training-and-fine-tuning-bert-using-nvidia-ngc/>

Olsson, A., & Engelbrektsson, O. (2022). A thesis that writes itself: On the threat of AIgenerated essays within academia.

OpenAI. (2019, 14 Şubat). *Better language models and their implications*. Erişim adresi: <https://openai.com/index/better-language-models/>

PlainEnglish.io. (2024, 21 Mayıs). *GRU, RNN ve LSTM'ye giriş: Bu devrim niteliğindeki derin öğrenme modellerini anlamaya yönelik başlangıç rehberi*. Erişim adresi: <https://python.plainenglish.io/introducing-gru-rnn-and-lstm-a-beginners-guide-to-understanding-these-revolutionary-deep-35b509a34a5a>

Pratiwi, N. I., Budi, I. ve Alfina, I. (2018, Ekim). Hate speech detection on Indonesian Instagram comments using FastText approach. 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (ss. 447–450). IEEE.

Pushpakumar, R., Prabu, R., Priscilla, M., Renisha, P. S., Prabu, R. T., & Muthuraman, U. (2022, Haziran). A novel approach to identify dynamic deficiency in cell using gaussian nb classifier. In 2022 7th international conference on communication and electronics systems (ICCES) (pp. 31-37). IEEE.

Ramakrishnan, S. ve Babu, L. D. (2023, Ağustos). Enhancing Twitter sentiment analysis using attention-based BiLSTM and BERT embedding. 2023 9th International Conference on Smart Computing and Communications (ICSCC) (ss. 36–40). IEEE.

- Raschka, S. (2018). Stat 479: Machine learning lecture notes. https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/07_ensembles_notes.pdf
- Reshamwala, A., Mishra, D., & Pawar, P. (2013). Review on natural language processing. *IRACST Engineering Science and Technology: An International Journal (ESTIJ)*, 3(1), 113-116.
- Roop, T. (t.y.). Understanding the difference between stemming and lemmatization. Medium. Erişim adresi: <https://medium.com/@tejaswaroop2310/understanding-the-difference-between-stemming-and-lemmatization-dbfdfed98df0>
- Salsabiila, S. Z., Irmanda, H. N. ve Arista, A. (2023, Kasım). Comparison of FastText and Word2Vec weighting techniques for classification of multiclass emotions using the Conv-LSTM method. *2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS)* (ss. 125–130). IEEE.
- Sangole, S. (2024, Mayıs 17). Erişim adresi: words to wonders: The Word2Vec journey. Stackademic. <https://blog.stackademic.com/from-words-to-wonders-the-word2vec-journey-b6445c36c84d>
- Saturn Cloud. (t.y.). Stemming in natural language processing. Erişim adresi: <https://saturncloud.io/glossary/stemming/>
- Savci, P., & Das, B. (2023). Comparison of pre-trained language models in terms of carbon emissions, time and doğruluk in multi-label text classification using AutoML. *Heliyon*, 9(5).
- Schaaff, K., Schlippe, T., & Mindner, L. (2024). Classification of human-and AI-generated texts for different languages and domains. *International Journal of Speech Technology*, 27(4), 935-956.
- Scikit-learn. (t.y.). LabelEncoder. Erişim adresi: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>
- Semrush. (t.y.). An introduction to TF-IDF: What it is & how to use it. Erişim adresi: <https://www.semrush.com/blog/tf-idf/>
- Shafi, A. (2024, Ekim). Random Forest Classification with Scikit-Learn. Erişim adresi: <https://www.datacamp.com/tutorial/random-forests-classifier-python>

- Stefan-it. (t.y.). *TR BERTurk* [GitHub repository]. Erişim adresi:
<https://github.com/stefan-it/turkish-bert>
- Suri, M. (t.y.). A dummy's guide to Word2Vec. Medium. Erişim adresi:
<https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673>
- Tahsin, R., Mozumder, M. H., Shahriyar, S. A. ve Mollah, M. A. S. (2020, Haziran). A novel approach for e-mail classification using FastText. 2020 IEEE Region 10 Symposium (TENSYMP) (ss. 1392–1395). IEEE.
- Tang, R., Chuang, Y. N. ve Hu, X. (2023). The science of detecting LLM-generated texts. arXiv preprint arXiv:2303.07205. <https://arxiv.org/abs/2303.07205>
- TechTarget. (2024, Nisan 25). What is logistic regression? Erişim adresi:
<https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression/>
- TechTarget. (t.y.). Large Language Models (LLMs). Erişim adresi:
<https://www.techtarget.com/whatis/definition/large-language-model-LLM>
- TechTarget - CNN Tanımı
Gillis, A. S., Craig, L., & Awati, R. (2024, 25 Kasım). What is a convolutional neural network (CNN)? Erişim adresi:
<https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>
- Toprakucar, K. (t.y.). *BERT modeli ile Türkçe metinlerde sınıflandırma yapmak*. Erişim adresi: <https://medium.com/@ktoprakucar/bert-modeli-ile-t%C3%BCrk%C3%A7e-metinlerdes%C4%B1n%C4%B1fland%C4%B1rma-yapmak-260f15a65611>
- Turing. (2022, Haziran 28). Random Forest Algorithm : How It Works & Why It's So Effective. Erişim adresi: <https://www.turing.com/kb/random-forest-algorithm>
- Vapnik, V., Golowich, S., & Smola, A. (1996). Örnek sayısı vector method for function approximation, regression estimation and signal processing. In *Advances in Neural Information Processing Systems* (Vol. 9). MIT Press.
- Wang, H., Li, J., & Li, Z. (2024). AI-generated text detection and classification based on BERT deep learning algorithm. arXiv preprint arXiv:2405.16422.

- Wani, M. A., Abd El-Latif, A. A., ELAffendi, M., & Hussain, A. (2024). AI-based Framework for Discriminating Human-authored and AI-generated Text. *IEEE Transactions on Artificial Intelligence*.
- Yal, D. (t.y.). *GPT-2 transformator dil modellerinin gorsellestirilmesi*. Erişim adresi: <https://devrimdanyal.medium.com/gpt-2-transformat%C3%B6r-dil-modelleri-ning%C3%B6rselle%C5%9Ftirilmesi-fc4bfd510223>
- Yang, E. ve Long, Z. (2023, Eylül). Research on the weighting method based on TF-IDF and Apriori algorithm. 2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE) (ss. 1003–1005). IEEE.
- Yang, L., Jiang, F., & Li, H. (2023). Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text. arXiv preprint arXiv:2307.11380.
- Yu, X., Qi, Y., Chen, K., Chen, G., Yang, X., Zhu, P., ... & Yu, N. (2023). Gpt paternity test: Gpt generated text detection with gpt genetic inheritance. arXiv preprint arXiv:2305.12519.

ÖZGEÇMİŞ

Adı ve Soyadı: Rezzan Nisa Er

Derece	Bölüm/Program	Üniversite	Yıl
Lisans	Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü	Çanakkale Onsekiz Mart Üniversitesi	2010-2018
Y. Lisans	Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği (YL) (Tezli)	Çanakkale Onsekiz Mart Üniversitesi	2018-2021
Doktora	Lisansüstü Eğitim Enstitüsü Bilgisayar Bilimleri ve Mühendisliği (DR)	İstanbul Sabahattin Zaim Üniversitesi	2021-2025

Projeler:

Yapay Zekâ ile Üretilmiş Türkçe Bir Metni Tespit Etmek (İstanbul Sabahattin Zaim Üniversitesi Bilimsel Araştırma Projesi 2024-BAP-300-007 numaralı BAP300 projesi) (01/8/24 – Devam, Araştırmacı Bursiyer : Rezzan Nisa ER)

Yayınlar:

Uluslararası bilimsel toplantılarda sunulan ve bildiri kitabında (*Proceedings*) basılan bildiriler

Er, R. N. & Çoğalmış, K. N. (2024, April). Determining Whether a Turkish Text is Produced by Artificial Intelligence or Human. In 2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI) (pp. 1-5). IEEE.

Ulusal hakemli dergilerde yayımlanan makaleler:

Er, R. N., Uğurlu, B., & Bayram, U. (2021). Akıllı Evlerdeki Cihazların Enerji Tüketimine Göre Kaynağının Belirlenmesi. Mühendislik Bilimleri ve Araştırmaları Dergisi, 3(1), 47-58.

Ulusal bilimsel toplantılarda sunulan ve bildiri kitabında basılan bildiriler

Er, R. N. & Çoğalmış, K. N. (2025, July). A comparative evaluation of BERT, GloVe and CountVectorizer methods in academic text classification. Paper presented at the

16th International Istanbul Scientific Research Congress, Beykoz University, Istanbul, Türkiye.

Uğurlu, B., Bayram, U., & Er, R. N. Elektrik Enerjisi Kaynaklarının Güç Tüketimine Göre Otomatik Olarak Belirlenmesi. 7. Uluslararası Bilimsel Araştırmalar Kongresi (13–14 Mart 2020/Ankara)(UBAK), 271.

