



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



UÇUŞ VERİLERİNİN ANALİZİ VE TAHMİNİ
İÇİN DERİN ÖĞRENME YAKLAŞIMLARI

Gökhan GÜLŞEN

YÜKSEK LİSANS TEZİ

Uçak Mühendisliği Anabilim Dalı

Ocak-2025
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Gökhan GÜLŞEN tarafından hazırlanan “Uçuş Verilerinin Analizi ve Tahmini için Derin Öğrenme Yaklaşımları” adlı tez çalışması 20/01/2025 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Uçak Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Mesut UYANER

.....

Danışman

Doç. Dr. Mehmet Emin ÇETİN

.....

Üye

Dr. Öğr. Üyesi Vahit TONGUR

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Havvanur UÇBEYİAY
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Gökhan GÜLŞEN
Tarih: 27.01.2025

ÖZET

YÜKSEK LİSANS TEZİ

UÇUŞ VERİLERİNİN ANALİZİ VE TAHMİNİ İÇİN DERİN ÖĞRENME YAKLAŞIMLARI

Gökhan GÜLŞEN

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Uçak Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mehmet Emin ÇETİN

2025, 70 Sayfa

Jüri

Prof. Dr. Mesut UYANER
Doç. Dr. Mehmet Emin ÇETİN
Dr. Öğr. Üyesi Vahit TONGUR

Bu çalışma, uçuş verilerinin analizi ve tahmini için derin öğrenme yöntemlerinin uygulanmasını incelemektedir. Hızla büyüyen havacılık sektörü, uçuş güvenliğini artırmak, operasyonel verimliliği optimize etmek ve yolcu memnuniyetini sağlamak amacıyla uçuş rotası ve uçuş gecikmesi tahmini gibi konulara odaklanmayı gerektirmektedir. Tezde, uçuş rotası ve gecikmesi tahminleri için CNN ve LSTM gibi derin öğrenme modelleri kullanılmıştır. Çalışmada, ADS-B verileri ve BTS On-Time Performance verileri birleştirilerek hem zamansal hem de uzamsal analizler gerçekleştirilmiştir. Uçuş rotası tahminlerinde LSTM ile düşük hata oranları elde edilirken, gecikme tahminleri için CNN modeli başarıyla uygulanmıştır. Sonuçlar, uçuş rotası ve gecikmelerinin birbirine olan etkilerini ortaya koymakta ve havacılık operasyonlarının optimizasyonu için pratik çözüm önerileri sunmaktadır.

Anahtar Kelimeler: ADS-B, CNN, derin öğrenme, LSTM, rota tahmini, uçuş gecikmesi

ABSTRACT

MS THESIS

**DEEP LEARNING APPROACHES FOR ANALYSIS AND PREDICTION OF
FLIGHT DATA**

Gökhan GÜLŞEN

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN AERONAUTICAL ENGINEERING**

Advisor: Assoc. Prof. Dr. Mehmet Emin ÇETİN

2025, 70 Pages

Jury

Prof. Dr. Mesut UYANER

Assoc. Prof. Dr. Mehmet Emin ÇETİN

Asst. Prof. Dr. Vahit TONGUR

This study explores the application of deep learning methods for the analysis and prediction of flight data. The rapidly growing aviation sector necessitates a focus on flight route and delay prediction to enhance safety, optimize operational efficiency, and improve passenger satisfaction. In this thesis, deep learning models such as CNN and LSTM were employed for flight route and delay predictions. By integrating ADS-B data with BTS On-Time Performance datasets, both temporal and spatial analyses were conducted. The LSTM model achieved low error rates in flight route predictions, while the CNN model was effectively applied for delay estimations. The results reveal the interdependencies between flight routes and delays, offering practical solutions for the optimization of aviation operations.

Keywords: ADS-B, CNN, deep learning, flight delay prediction, flight route prediction, LSTM

ÖNSÖZ

Bu tez çalışması, uçuş verilerinin analizi ve tahmini üzerine derin öğrenme yaklaşımlarını incelemek ve bu alandaki bilimsel literatüre katkıda bulunmak amacıyla gerçekleştirilmiştir. Çalışma sürecinde elde edilen bilgi birikimi, yalnızca akademik hedeflerime ulaşmamı sağlamakla kalmayıp, aynı zamanda bu zorlu yolculukta bana çok değerli bir hayat deneyimi de kazandırmıştır.

Bu süreçte, bilgi ve deneyimleriyle yolumu aydınlatan, çalışmamın her aşamasında sabırla rehberlik eden değerli danışman hocam Doç. Dr. Mehmet Emin ÇETİN'e en içten teşekkürlerimi sunarım. Kendisinin rehberliği ve desteği olmasaydı bu çalışmayı tamamlamak benim için çok daha zor olurdu.

Ayrıca, bu süreçte her zaman yanımda olan, sevgisi, sabrı ve anlayışıyla bana güç veren eşim Uzm. Dr. Yasemin KOÇAK GÜLŞEN'e sonsuz teşekkür ederim

Son olarak, bu çalışmayı gerçekleştirmemde emeği geçen tüm arkadaşlarıma, aileme ve katkıda bulunan herkese teşekkür ederim. Bu çalışmanın, gelecekte bu alanda yapılacak araştırmalara ilham verebilmesi ve fayda sağlayabilmesi en büyük temennimdir.

Tez çalışmamın bilim dünyasına ve havacılık alanına faydalı olmasını dilerim.

Gökhan GÜLŞEN
KONYA-2025

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ	1
1.1. Motivasyon	2
1.2. Problem	3
1.3. Literatüre Katkısı	4
1.4. Tezin Organizasyonu	5
2. KAYNAK ARAŞTIRMASI	7
2.1. Uçuş Rotası Tahmininde Kaynak Araştırması.....	7
2.2. Uçuş Gecikmesi Tahmininde Kaynak Araştırması.....	9
3. METERYAL VE YÖNTEM.....	11
3.1. Python	11
3.2. Python Kütüphaneleri	12
3.2.1. Pandas	13
3.2.2. Numpy	13
3.2.3. Matplotlib.....	14
3.2.4. Scikit-learn.....	14
3.3. Yapay Zeka	14
3.3.1. Yapay sinir ağları.....	17
3.3.2. Makine öğrenmesi.....	19
3.4. Derin Öğrenme	23
3.4.1. Derin öğrenmenin kullanıldığı çalışmalar	25
3.4.2. Derin öğrenme kütüphaneleri	27
3.4.3. Derin öğrenme mimarileri	29
3.5. Derin Öğrenmede Kullanılan Önemli Parametreler	36
3.5.1. Aktivasyon fonksiyonları.....	36
3.5.2. Optimizasyon algoritmaları	39
3.5.3. Maliyet (Kayıp) fonksiyonları	40
3.6. Veri Ön İşleme.....	41
3.6.1. Veri temizleme.....	42
3.6.2. Özellik seçimi	42
3.6.3. Veri dönüşümü.....	43
3.6.4. Veri bölme	45
3.7. Literatürde Yapılan Çalışmalar.....	45
4. DENEYSEL ÇALIŞMALAR.....	51

4.1. Deney Ortamı ve Altyapı	51
4.2. Veri Seti ve Hazırlık Süreci	51
4.2.1. Uçuş gecikme tahmini için kullanılan veriler	51
4.2.2. Uçuş rota tahmini için kullanılan veriler	54
4.3. Veri Ön İşleme Adımları	55
4.4. Model Kurulumu ve Hiperparametreler	56
4.5. Eğitim Süreci	58
4.6. Performans Ölçütleri.....	59
5. BULGULAR.....	60
5.1. Uçuş Gecikme Tahmini Bulgu Sonuçları	60
5.2. Uçuş Rota Tahmini Bulgu Sonuçları	62
6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	63
6.1. Sonuçlar	63
6.2. Öneriler	64
7. KAYNAKLAR	67

SİMGELER VE KISALTMALAR

Kısaltmalar

ADS-B	: Automatic Dependent Surveillance–Broadcast
AI	: Artificial Intelligence
ANN	: Artificial Neural Network
ATC	: Air Traffic Control
BP	: Backpropagation Neural Network
CNN	: Convolutional Neural Network
DL	: Deep Learning
LSTM	: Long Short Term Memory
ML	: Machine Learning
NAS	: National Airspace System
USDOT	: United States Department of Transportation



1. GİRİŞ

Giriş Havacılık sektörü, son yıllarda hızlı bir büyüme göstererek dünya genelinde taşınan yolcu ve yük miktarını önemli ölçüde artırmıştır. Havada her zamankinden daha fazla uçak varken, hava sahasının gözetiminin otomatikleştirilmesine ihtiyaç vardır. Farklı niyetlere sahip uçakların farklı uçuş düzenlerinde uçtukları yaygın olarak bilinmektedir. Bu büyüme, sektördeki şirketler ve havaalanları için daha fazla verimlilik, güvenlik ve maliyet optimizasyonu gerekliliğini de beraberinde getirmiştir. Amerika Birleşik Devletleri'nde 2008 yılında tüm ticari havacılık aksaklıklarının toplam maliyeti 33,5 milyar dolar olarak hesaplanmıştır (Khaksar & Sheikholeslami, 2019). Havayolu şirketlerinin gecikme istatistiklerine göre, 2019 yılında uçuşların yalnızca yüzde 79'u zamanında varmış, bu da havayolu yolcularına yapılan masraflar, talep kaybı ve diğer dolaylı masraflar da dahil olmak üzere on milyarlarca dolarlık kayba yol açmıştır (Kim et al., 2016). Avrupa hava sahasında ise, 28 Haziran 2019 tarihinde 37288 adet uçuş ile rekor kırılmıştır (Tanja Grobotek, 2019). Uçuş güvenliği, yakıt tüketimi, operasyonel maliyetler ve yolcu memnuniyeti gibi faktörlerin iyileştirilmesi, uçuş verilerinin etkin bir şekilde analiz edilmesine ve gelecekteki uçuş koşullarının doğru bir şekilde tahmin edilmesine bağlıdır. Bu bağlamda, uçuş rotası ve uçuş gecikmesi tahmini gibi konular, havacılık sektörü için stratejik bir öneme sahiptir.

Günümüzde uçuş rotası tahmini, hem güvenlik hem de yakıt tüketimi açısından kritik bir role sahiptir. Özellikle yoğun hava trafiği olan rotalarda, doğru bir rota tahmini yapılarak uçuş güvenliği artırılabilir ve optimum rota seçimi sayesinde yakıt tüketimi azaltılabilir. Aynı zamanda, uçuş rotasının doğru tahmini, havaalanı yoğunluk yönetimi ve hava trafik kontrolü için de önemli bir avantaj sağlar.

Diğer yandan, uçuş gecikmesi tahmini, yolcu memnuniyeti ve operasyonel maliyetler açısından büyük önem taşımaktadır. Gecikmeler, havayolu şirketleri için ciddi mali kayıplara yol açmakta ve yolcuların seyahat planlarını olumsuz etkileyerek müşteri memnuniyetsizliğine neden olmaktadır. Bu nedenle, geçmiş uçuş verileri kullanılarak potansiyel gecikmelerin tahmin edilmesi, havayolu şirketlerine daha etkili bir planlama yapma ve müşterilere doğru bilgilendirme sağlama imkanı tanır.

Bu tez çalışmasında, derin öğrenme yöntemleri kullanılarak uçuş rotası ve uçuş gecikmesi tahminleri yapılması amaçlanmaktadır. Özellikle CNN (Lecun & Bengio, 1995) ve LSTM (Hochreiter & Schmidhuber, 1997) gibi zaman serisi verileri işleyebilen modeller kullanılarak uçuş verilerinin analizi ve tahmini gerçekleştirilecektir. Bu kapsamda yapılacak çalışmalar, hem uçuş operasyonlarının verimliliğini artırmaya katkı sunacak hem de havacılık sektöründeki çeşitli paydaşlar için faydalı bilgiler sağlayacaktır.

1.1. Motivasyon

Havacılık sektörü, operasyonel verimlilik, uçuş güvenliği ve yolcu memnuniyetini artırmak için sürekli iyileştirme arayışındadır. Ancak, uçuş planlaması ve operasyonel süreçler, karmaşık ve dinamik bir yapı sergiler. Özellikle uçuş rotası ve gecikme tahmini, bu süreçlerin en kritik parçalarını oluşturur. Bu iki problemi ayrı ayrı ele almak yerine, entegre bir yaklaşımla analiz etmek ve çözmek, daha bütünsel ve etkili sonuçlar elde edilmesini sağlayabilir.

Uçuş rotası tahmini, uçakların hava sahasındaki güvenliğini sağlarken, yakıt tüketimini ve karbon emisyonlarını minimize etmek için büyük bir fırsat sunar. Öte yandan, uçuş gecikmelerinin tahmini, operasyonel verimliliği artırmak, maliyetleri azaltmak ve yolcu memnuniyetini en üst düzeye çıkarmak açısından önemlidir. Ancak bu iki problemin birbiriyle ilişkili olduğu gerçeği sıklıkla göz ardı edilmektedir. Örneğin, optimum bir uçuş rotası belirlemek gecikme risklerini doğrudan etkileyebilir, ya da hava sahasındaki trafik yoğunluğu nedeniyle oluşabilecek bir gecikme, rotanın yeniden optimize edilmesini gerektirebilir.

Bu çalışma, uçuş rotası ve gecikme tahminini birlikte ele alarak, bu iki kritik problemin bir arada çözülmesine odaklanmaktadır. ADS-B gibi gerçek zamanlı uçuş verilerinden elde edilen kapsamlı bilgiler, uçuş rotası ve gecikme arasındaki karmaşık ilişkilerin derinlemesine incelenmesine olanak tanır. Bu bağlamda, derin öğrenme yöntemleri, hem zaman serisi verilerini (LSTM) hem de uzamsal ilişkileri (CNN) etkili bir şekilde modelleyerek, bu ilişkilerin analizi için güçlü bir araç sunmaktadır.

Bu entegre yaklaşımın motivasyonu, havacılık sektöründeki mevcut yaklaşımların ötesine geçerek, uçuş güvenliğini artıran, operasyonel verimliliği destekleyen ve çevresel etkileri azaltan çözümler sunmaktır. Araştırma, yalnızca akademik anlamda yenilikçi bir katkı sağlamakla kalmayıp, hava yolları, hava trafik kontrol birimleri ve yolcular için somut faydalar üretmeyi hedeflemektedir.

1.2. Problem

Bu tez çalışmasında iki soruya cevap bulmaya çalışıyoruz:

- I. Uçuş Rotası ve Güvenliği:** ADS-B verileri (*Automatic Dependent Surveillance-Broadcast (ADS-B)*, n.d.), uçakların anlık konum, yükseklik, hız ve yön bilgilerini içerdiğinden, uçuş rotalarının doğru bir şekilde izlenmesi ve tahmin edilmesi açısından oldukça önemlidir. Özellikle yoğun trafiğe sahip hava sahalarında uçakların güvenli bir mesafede seyretmeleri ve rotalarını belirlemeleri için doğru verilere ihtiyaç duyulmaktadır. Bu veriler, rotanın tahmin edilmesi ve hava trafik yoğunluğuna göre optimize edilmesine olanak sağlar. ADS-B verileri kullanılarak yapılan analizler sayesinde, hava trafiği yönetimi ve hava sahası güvenliği artırılabilir.
- II. Uçuş Gecikmesi Tespit Etmek ve Yolcu Memnuniyeti:** Uçuş sürecindeki anlık değişikliklerin gözlemlenmesine ve geçmiş uçuş verilerinin analizine imkan tanır. Gecikmelerin sebeplerinin analiz edilmesi, bu sorunların önceden tahmin edilmesini sağlayarak hem yolcu memnuniyetini artırabilir hem de havaalanı operasyonlarının düzenlenmesine yardımcı olur. Uçuş verileri kullanılarak hava durumu, trafik yoğunluğu veya teknik sebeplerle oluşabilecek gecikmeler öngörülebilir. Bu tahminler, hava yollarının yolcuları bilgilendirme ve alternatif çözümler sunma konusundaki etkinliğini artıracaktır.

1.3. Literatüre Katkısı

Bu çalışma, uçuş rotası tahmini ve uçuş gecikmesi tahminini ortak bir şekilde ele alarak, havacılık verilerinin analizi ve tahmini alanında önemli bir yenilik sunmaktadır. Literatürde, uçuş rotası ve gecikme tahmini genellikle birbirinden bağımsız olarak ele alınmış ve bu iki problemin ilişkileri üzerinde yeterince durulmamıştır. Ancak, bu iki problemin birbiriyle güçlü bir şekilde bağlantılı olduğu ve entegre bir çözümün daha etkin sonuçlar sunabileceği açıktır. Bu bağlamda, bu çalışma, uçuş rotası ve gecikme tahminine yönelik bütüncül bir yaklaşım sunarak, literatürdeki bu boşluğu doldurmayı amaçlamaktadır.

Bu çalışmanın literatüre sağladığı başlıca katkılar şu şekilde özetlenebilir:

- I. **Entegre Tahmin Yaklaşımı:** Çalışma, uçuş rotası ve gecikme tahminini bir arada ele alarak, bu iki problemin birbirine olan etkilerini analiz etmektedir. Bu entegre yaklaşım, mevcut tahmin yöntemlerinin ötesine geçerek, daha bütüncül bir çözüm sunmaktadır.
- II. **ADS-B Verilerinin Kapsamlı Kullanımı:** ADS-B verilerinin uçuş rotası ve gecikme tahmini bağlamında birlikte kullanılması, bu veri setinin sunduğu potansiyelin daha geniş bir perspektiften değerlendirilmesine olanak tanımaktadır. Çalışma, ADS-B verilerinin hem zaman serisi analizine hem de uzamsal ilişkilere yönelik kullanımını bir arada ele alarak yenilikçi bir analiz yöntemi sunmaktadır.
- III. **Derin Öğrenme Yöntemlerinin Uygulanması:** CNN ve LSTM gibi modern derin öğrenme modellerinin birlikte kullanılması, uçuş verilerindeki hem zamansal hem de uzamsal ilişkilerin yüksek doğrulukla modellenmesine olanak tanımaktadır. Bu bağlamda, çalışma, derin öğrenme yöntemlerinin havacılık verilerindeki etkinliğini göstermesi açısından literatüre katkı sağlamaktadır.
- IV. **Havacılık Sektörüne Yönelik Pratik Çözümler:** Çalışma, uçuş güvenliği, operasyonel verimlilik ve yolcu memnuniyeti gibi sektör için kritik öneme

sahip alanlarda uygulanabilir çözümler sunmaktadır. Bu bağlamda, araştırmanın çıktıları, hem teorik hem de pratik açıdan değerli bir kaynak teşkil etmektedir.

Sonuç olarak, bu tez çalışması ile uçuş verilerinin analizi ve tahmini alanında literatüre yenilikçi bir bakış açısı kazandırmakta ve havacılık sektörü için entegre çözümler geliştirilmesi adına önemli bir referans noktası oluşturmaktadır.

1.4. Tezin Organizasyonu

Yapılan tez çalışmasında uçuş rotası tahmini ve uçuş gecikmesi tahmini konularını ele alarak, derin öğrenme yöntemleriyle bu iki problemi entegre bir şekilde çözmeyi amaçlamaktadır. Çalışma, yedi başlıkta ele alınmıştır. İlk başlık olarak tez çalışmasının kapsamı tanıtılmaktadır. Motivasyon, bu çalışmanın neden gerekli olduğuna dair açıklamalarla desteklenmiştir. Ardından, ele alınan problemlerin detaylı tanımları yapılmış, literatüre katkıları açıklanmıştır. Ayrıca, çalışmanın genel organizasyonu hakkında bilgi verilmiştir. İkinci bölümde, literatür taramasına yer verilmiştir. Bu bağlamda, uçuş rotası tahmini ve uçuş gecikmesi tahmini ile ilgili literatürde yer alan çalışmalar incelenmiş, bu alandaki mevcut yaklaşımlar, kullanılan yöntemler ve karşılaşılan zorluklar tartışılmıştır. Üçüncü bölümde ise tezde kullanılan materyal ve yöntemler detaylandırılmıştır. Python programlama dili ve ilgili kütüphaneler (Pandas, Numpy, Matplotlib, Scikit-learn) açıklanmıştır. Ayrıca, Google Colab platformunun kullanımına değinilmiş, yapay zeka, makine öğrenmesi ve derin öğrenme yöntemleri tanıtılmıştır. Bu kapsamda, yapay sinir ağları, derin öğrenme modelleri (CNN, LSTM) ve mimarileri ile bu modellerde kullanılan önemli parametreler (aktivasyon fonksiyonları, optimizasyon algoritmaları, kayıp fonksiyonları) detaylı şekilde ele alınmıştır. Dördüncü bölümde, deneysel çalışmalara yer verilmiştir. Bu bölümde, uçuş rotası ve gecikmesi tahminine yönelik geliştirilen modellerin veri hazırlama, model eğitimi ve test süreçleri anlatılmıştır. ADS-B verilerinin işlenmesi, seçilen derin öğrenme mimarilerinin uygulanması ve kullanılan hibrit veri hazırlama yöntemleri detaylandırılmıştır. Beşinci bölümde, deneysel çalışmalardan elde edilen bulgular sunulmaktadır. Tahmin sonuçları, doğruluk oranları ve performans ölçütleri incelenmiş, uçuş rotası tahmini ve gecikme tahmini sonuçları karşılaştırmalı olarak ele alınmıştır. Altıncı bölümde, araştırmanın genel sonuçları ve bulgulara dayalı tartışmalar yapılmıştır. Elde edilen sonuçların

literatürle uyumu değerlendirilmiş ve bu çalışmanın havacılık sektörüne sağladığı potansiyel katkılar açıklanmıştır. Ayrıca, gelecekte yapılabilecek çalışmalara yönelik öneriler sunulmuştur. Son bölümde ise bu çalışmada faydalanılan kaynaklara yer verilmiştir.



2. KAYNAK ARAŞTIRMASI

2.1. Uçuş Rotası Tahmininde Kaynak Araştırması

Günümüzde, çoğu büyük uçak konumlarını uydu navigasyonu aracılığıyla tespit etmektedir. Uçak, ardından bu bilgiyi, alıcıya sahip olan herkesin erişebileceği belirli sabit frekanslarda iletir. Bu bilgi aktarımı gerçek zamanlı olarak yapılır. Her ADS-B mesajı, belirli bir uçakla ilgili çağrı işareti, hızı ve yönü gibi bilgileri içermektedir. Böylece uçağın tüm izleyiciler, örneğin diğer uçaklar, yer kontrolörü, askeri birlikler veya amatörler uçucular, tarafından gözlemlenmesini sağlar. Bu süreç hava sahası güvenliğini artırmak amacıyla hayata geçirilmiştir. Gerçek zamanlı olarak iletilen ADS-B verilerin örneği Çizelge 2.1'de yer alan bilgileri kapsar.

Çizelge 2.1. ADS-B mesajında bulunan bilgiler

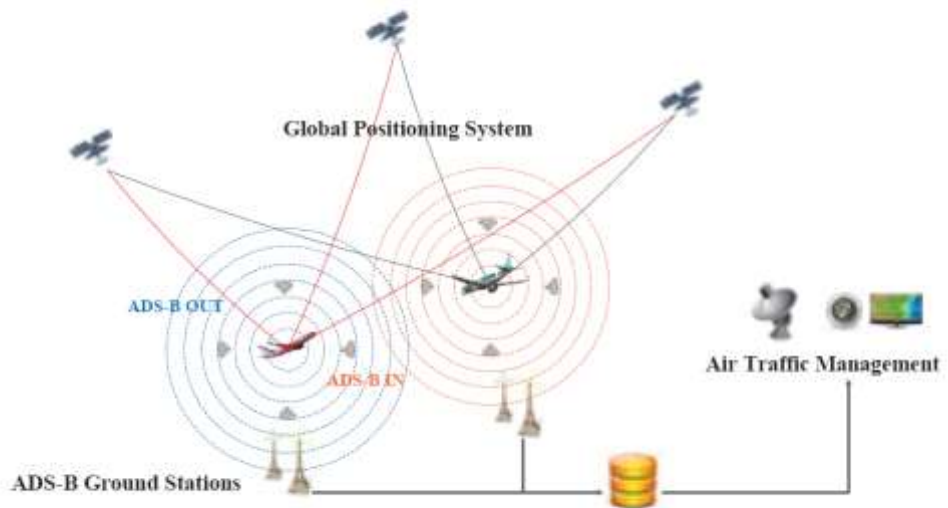
Parametre	Yörünge Noktası
Timehex	1730317727
ICAO Address Code	780E59
Flight Number	780E59
Longitude (°)	120.2188
Latitude (°)	37.6415
Altitude (Feet)	33000
Velocity (Knot)	256
Heading (°)	398.399

Radar sistemi, hâlâ yer kontrolü için birincil gözetleme yöntemi olmasına rağmen, 2020 yılından itibaren Avrupa Birliği, askeri uçaklar da dâhil olmak üzere tüm uçakların ADS-B transponderleri ile donatılmasını zorunlu kılmıştır (Laying down Requirements for the Performance and the Interoperability of surveillance for the Single European Sky Text with Eea Relevance., 2022). Diğer ülkeler benzer politikaları icra etmektedir. Bu nedenle Şekil 2.1'de de görüldüğü gibi her gün her saniye büyük miktarda ADS-B verisi oluşturulmaktadır.



Şekil 2.1. ADS-B değişimi takip edilen uçuşlar (*ADS-B Exchange.*, 2024)

ADS-B verileri, radar verilerinin aksine açıkça iletilmektedir. Erişilebilirlik ve veri çeşitliliği nedeniyle ADS-B verilerinin spesifik uçuş desenlerini tanımlamak için uygun bir seçenek haline getirmiştir. Pozisyon, yön, çağrı işareti, kalkış ve varış istasyonları saniyede üç ila dört kez yayınlar. Mesajlar, belirli bir zaman diliminde, Şekil 2.2'de gösterildiği gibi, yerden havaya ve havadan havaya veri bağlantıları aracılığıyla iletilir (Shi et al., 2018).



Şekil 2.2. ADS-B çalışma sistemi (Shi et al., 2018)

Birden fazla yer istasyonu, gözetim alanını genişletmek için kurulmuştur. Ancak bu durum, çelişkiye veya gereksiz verilere yol açtı. Ayrıca, ölçüm ve sistem hataları aynı

tür sensörler arasında bile farklılık gösterebilir. Farklı istasyonlar tarafından kaydedilen ham verilerin gürültüden de arındırılması gerekiyor.

Büyük bir veri seti ile gerçek ADS-B yörünge noktası bilgileri yüksek lisans tezimizin analiz süresince kullanılmıştır.

2.2. Uçuş Gecikmesi Tahmininde Kaynak Araştırması

Uçuş gecikmeleri, ekonomik ve çevresel etkileri nedeniyle literatürde önemli bir konu olarak incelenmektedir. Yolcularla doğrudan ilgili sonuçların yanı sıra, gecikme tahmini hava taşımacılığı sistemindeki karar verme sürecinde çok önemlidir. Uçuş rötalarının yolcular, havayolları ve havalimanları için başta ekonomik olmak üzere olumsuz etkileri vardır. Rötaların meydana gelme belirsizliği göz önüne alındığında, yolcular genellikle randevuları için saatler öncesinden seyahat etmeyi planlamakta ve zamanında varışlarını sağlamak için seyahat maliyetlerini artırmaktadırlar (Balakrishna et al., 2010).

Amerika Birleşik Devletleri'nde uçuş rötaları ciddi ve yaygın bir sorundur. 2007 yılında, neredeyse her dört havayolu uçuşları varış noktasına 15 dakikadan fazla geç ulaşmıştır (*Bureau of Transportation Statistics (BTS) (2009) Airline on-Time Statistics and Delay Causes: On-Time Arrival Performance.*, n.d.). Uçuş gecikmeleri, hava durumu, uçak bakım sorunları, hava taşıyıcısının mali durumu, havaalanı altyapısının kısıtlamaları, NAS, ATC, diğer uçuşlar ve uçaklar tarafından önceki gecikmeler, havaalanı güvenliği veya havayollarının belirli bir havaalanında uçuşları nasıl planladığı gibi faktörlerden kaynaklanabilir.

Bu tez çalışmamızda, Ocak-Temmuz 2024 döneminde uçuşlar ele alınmıştır. LAX – JFK, LAS-LAX ve ATL-MCO rotalarında gerçekleştirilen uçuş sayıları ve bu rotalarda ele alınan dönem boyunca gecikmeli gerçekleşen(15 dakikadan fazla geciken uçuşlar) uçuş sayıları Çizelge 2.2'de gösterilmiştir.

Çizelge 2.2. Ocak-Temmuz 2024 dönemindeki LAX-JFK, LAS-LAX VE ATL-MCO rotalarındaki uçuş sayıları ve gecikmeli uçuş sayıları

Rota	ICAO	Uçuş Sayısı	Gecikmeli Uçuş
Los Angeles - New York	LAX-JFK	7311	1363
Las Vegas - Los Angeles	LAS-LAX	7071	1715
Atalanta - Orlando	ATL-MCO	6772	1940

Bu veriler , ABD Ulaştırma Bakanlığı Ulaştırma İstatistikleri Bürosu tarafından her ay raporlanarak ayrıntılı bir şekilde sunulmaktadır. ABD Ulaştırma Bakanlığı Ulaştırma İstatistikleri Bürosu, 15 dakikadan fazla geciken ve Çizelge 2.3’de gösterilen nedenlerden biri meydana geldiğinde hava yolu uçuş gecikmesi olarak tanımlar (Yablonsky et al., 2014).

Çizelge 2.3. Uçuş gecikmesi nedenleri

Gecikme Sebebi	Açıklama
Hava Yolu Şirketi	İptal veya gecikmenin nedeni hava yolu şirketinin kontrolündeki koşullardan kaynaklanan sebepler. (uçak bakımı veya kabin ekibi sorunları, uçak içi temizliği, yakıt alma, bagaj işlemleri vs.).
Aşırı Hava Koşulları	Uçuş ekibinin değerlendirmesine göre, hortum, kar fırtınası veya kasırga gibi bir uçuştan gecikmelere veya engellere neden olan (gerçek veya tahmin edilen) önemli meteorolojik koşullar.
Ulusal Havacılık Sistemi (NAS)	Ulusal havacılık sistemine atfedilebilecek gecikmeler ve iptaller, aşırı olmayan hava koşulları, havaalanı operasyonları, yoğun trafik hacmi ve hava trafiği kontrolü gibi geniş bir koşul setini ifade eder.
Geciken Uçak	Aynı uçakla yapılan önceki uçuşun gecikmesi ile mevcut uçuşun geç kalkmasına neden olması.
Güvenlik	Bir terminalin veya peronun tahliyesi, güvenlik ihlali nedeniyle uçakların yeniden binilmesi, çalışmayan tarama ekipmanları ve/veya tarama alanlarında 29 dakikayı aşan uzun kuyruklar nedeniyle oluşan gecikmeler veya iptaller.

3. METERYAL VE YÖNTEM

Bu bölümde, Python programlama dili ve Python ekosistemindeki çeşitli kütüphaneler hakkında ayrıntılı bilgi verilecektir. Ayrıca, Python kodlarının çalıştırılması ve derlenmesi için yaygın olarak kullanılan ortam olan Google Colab üzerinde yapılan işlemler de ele alınacaktır. Yapay zeka (YZ) ve makine öğrenimi (ML) alanlarıyla ilgili temel kavramlar ve bu teknolojilerin uygulama alanları da incelenecektir. Bunun yanı sıra, literatürdeki benzer çalışmaların bir özeti sunulacak ve bu çalışmaların konuyla olan ilişkisi ve katkıları üzerinde durulacaktır. Bu bölüm, Python'un yapay zeka uygulamalarındaki rolünü ve bu alandaki gelişmeleri anlamaya yönelik temel bir çerçeve sunmayı amaçlamaktadır.

3.1. Python

Python, öğrenmesi kolay ve dinamik olarak türetilmiş yüksek seviyeli, genel amaçlı bir programlama dilidir. Python 1980 yılında Guido Van Rossum tarafından geliştirilmeye başlanıp 1991 yılında ilk sürümü yayınlanan kullanımı kolay ve yüksek seviyeli bir programlama dilidir. Python öğrenme açısından diğer dillere göre daha kolaydır. Ayrıca geniş kütüphanesi ve dinamik yapısı nedeniyle veri biliminden web geliştirmeye kadar farklı alanlarda sıklıkla kullanılmaktadır. CERN, NASA gibi birçok şirket Python'ı kullanmaktadır. İçerisinde hayatı kolaylaştırıcı çok fazla kütüphane ve framework buldurmasında dolayı, birçok alanda kullanılabilir. Aşağıda Python'ın kullanıldığı alanlar verilmektedir.

- Web geliştirme alanı
- Oyun geliştirme alanı
- Yapay zeka ve makine öğrenimi
- Görüntü işleme
- Veri bilimi
- Sağlık alanı
- Masaüstü çalışmaları
- Crawling
- Ses teknolojisi

3.2. Python Kütüphaneleri

Python, açık kaynaklı ve geniş bir kütüphane ekosistemine sahip bir programlama dilidir. Bu kütüphaneler, kullanıcıların yazılım geliştirme süreçlerini hızlandırmalarına, karmaşık işlemleri daha kolay hale getirmelerine ve çeşitli alanlarda derinlemesine analiz yapmalarına olanak tanır. Python kütüphaneleri, genellikle belirli bir işlevi yerine getiren, kullanıcıların işlerini kolaylaştıran araçlar ve modüller içerir. Bu kütüphaneler, Python'un esnekliğini ve gücünü büyük ölçüde artırır, kullanıcıların daha az kod yazarak daha verimli çözümler üretmelerine yardımcı olur.

Çizelge 3.1’de çeşitli Python kütüphanelerinin GitHub üzerindeki popülerliğini ve aktivitelerini göstermektedir. Kütüphanelerin, kullanıcılar tarafından beğenilen ve takip edilme sayısı (starts), her bir kütüphanenin kaç kez üzerinde çalışıldığı (forked), katkıda bulunan kişi sayısı ve son bir ayda yapılan değişikliklerin (contributors) sayısı gibi faaliyetler de yer almaktadır. Örneğin, NumPy ve scikit-learn gibi popüler kütüphaneler yüksek sayıda yıldız ve katkı ile oldukça aktifken, daha niş kütüphaneler (örneğin, Hadoopy veya Pydoop) daha az aktivite ve katkı almış. Bu veriler, her kütüphanenin ne kadar yaygın kullanıldığını ve aktif olarak geliştirildiğini anlamak için faydalıdır.

Çizelge 3.1. Python kütüphaneleri GitHub verileri(2024-Ekim)

Kütüphane	Starts	Forked	Contributors
NumPy	28.200	10.200	1701
SciPy	13.200	5.200	1549
Cython	9.600	1.500	476
pandas	43.900	18.00	3.372
PyTables	13.00	274	114
h5py	2.100	532	200
Matplotlib	20.400	7.700	1489
seaborn	12.600	1.900	200
Plotly	16.400	2.600	257
Bokeh	19.400	4.200	610
scikit-learn	60.300	25.400	2959
Shogun	3.000	1.000	175
mlxtend	4.900	873	106
TensorFlow	187.00	74.400	3623
Keras	62.100	19.500	1277
PyTorch	84.700	22.800	3623
Caffe	34.100	18.700	267
Hadoopy	243	59	3
Pydoop	237	59	12
Spark	40.100	28.400	2135

3.2.1. Pandas

Pandas, veri analizi ve veri bilimi için NumPy ve Matplotlib ile birlikte en yaygın ve en çok kullanılan Python kütüphanesidir. Github'da yaklaşık 2 milyondan fazla kullanımla ve 3.372 katkı sağlayıcısından oluşan aktif bir toplulukla, veri analizi ve temizliği için yaygın olarak kullanılmaktadır. Pandas, yapılandırılmış verilerin verimli ve sezgisel bir şekilde işlenmesi için özel olarak tasarlanmış kütüphaneler gibi hızlı ve uyumlu veri yapıları sunar. Pandas'ın temel özellikleri arasında, eksik verileri yönetme esnekliği sağlayan, veri setleri arasında yürütme için özel fonksiyonların oluşturulmasını kolaylaştıran, yüksek soyutlama sunan ve gelişmiş veri yapıları ile manipülasyon araçlarını kapsayan, açık ve kapsamlı bir sözdizimi ve işlevsellik sunar. CSV dosyalarını veri koleksiyonu formatına aktarmak için sağlam destek sunar. Bu da istatistik ve finans gibi çeşitli akademik ve ticari alanlarda uygulanabilir. Ayrıca, tarih aralığı oluşturma, doğrusal regresyon ve tarih kaydırma gibi zaman serisine özgü işlevsellikler de sunar (Saabith et al., n.d.).

3.2.2. Numpy

NumPy, Numerical Python'un kısaltmasıdır ve Python'da sayısal hesaplama için temel bir pakettir. Bilimsel yetenekler sunan birçok hesaplama aracı, veri alışverişi için standart bir arayüz olarak NumPy'nin dizi nesnelere kullanır. NumPy ile ilgili bilginin önemli bir kısmı pandas için de geçerlidir (McKinney & Safari, 2017).

Numerik Python'un birincil amacı dizileri ve matrisleri yönetmektir. Diziler ve matrisler kullanarak matematiksel hesaplamalar yapar. NumPy'nin kütüphanesi Python uygulama geliştirme için aşağıdaki temel özellikleri sunar (Sheela & Meena, n.d.):

NumPy, dizi nesnelere için sağlam bir N-boyutlu araç sunar. Dizi nesnesi, matris hesaplamaları için kullanılır. Python'da NumPy, yeniden kullanılabilir görevleri yerine getirmek için gelişmiş yöntemleri destekler. NumPy, lineer cebir fonksiyonları için temel paketi tanıtır. Matematiksel hesaplamalar için son derece faydalıdır.

3.2.3. Matplotlib

Matplotlib, Python'da en yaygın kullanılan veri görselleştirme paketlerinden biridir. Bu kütüphane, dünya çapında bilim insanları ve filozoflar tarafından kullanılan bu yazılımı oluşturmak için önemli çalışmalar yapan John Hunter ve birçok katkıci tarafından geliştirildi (Hafeez & Sial, 2021). Matplotlib, Python'da veri görselleştirme için bir grafik kütüphanesidir ve Python veri bilimi yığının temel bileşenlerinden biri olarak NumPy, Pandas ve diğer ilgili kütüphanelerle sorunsuz bir şekilde entegre olur.

3.2.4. Scikit-learn

Scikit-learn (Sklearn), Python'da makine öğrenimi için en değerli ve dayanıklı kütüphanedir. Python'da tutarlı bir arayüz aracılığıyla sınıflandırma, regresyon, kümeleme ve boyut indirgeme dahil olmak üzere makine öğrenimi ve istatistiksel modelleme için bir dizi etkili araç sunar. Bu kütüphane, esasen Python'da geliştirilmiş olup NumPy, SciPy ve Matplotlib üzerine kuruludur. Sklearn kütüphanesinin amacı, çoğunlukla veri modelleme için kullanılır ve birçok türde öngörücü veri analizi için etkili, kullanıcı dostu araçlar sunar. Bu kütüphanenin ana kullanım durumları altı gruba şu şekilde sınıflandırılabilir:

- Veri Ön İşleme
- Regresyon
- Sınıflandırma
- Kümeleme
- Model Seçimi
- Boyut Azaltma

3.3. Yapay Zeka

Yapay Zeka (AI), makinelerin geleneksel olarak insan zekası gerektiren görevleri yerine getirmesini sağlayarak modern teknolojideki en dönüştürücü ilerlemelerden birini temsil eder. Bu ilerlemeler, akıl yürütme, öğrenme, karar verme ve görsel algılama gibi süreçleri içerir. Yapay zeka uygulamaları, sağlık hizmetlerinden finans sektörüne,

otonom araçlardan ulusal güvenliğe kadar birçok sektörü kapsar ve bu da onu geleneksel iş akışlarını devrim niteliğinde değiştirebilen genel amaçlı bir teknoloji haline getirir.

Yapay Zeka oluşum itibari ile aşağıdakiler de dahil olmak üzere geniş bir teknoloji ve metodoloji yelpazesini kapsar :

- **Makine Öğrenimi** : Sistemlerin verilerden açık programlama olmadan öğrenmesini ve gelişmesini sağlayan algoritmalar. Bu, denetimli, denetimsiz ve pekiştirmeli öğrenmeyi içerir.
- **Derin Öğrenme**: Büyük veri setlerini analiz etmek için birden fazla katmana sahip yapay sinir ağlarını kullanan ML'nin bir alt kümesi, görüntü tanıma, doğal dil işleme ve tahmine dayalı analiz gibi alanlarda mükemmel sonuçlar verir.
- **Doğal Dil İşleme** : Makinaların insan dilini anlamasını, yorumlamasını ve üretmesini sağlar, bu da sanal asistanlar ve sohbet botları gibi uygulamalar için kritik öneme sahiptir.

Yapay zeka, büyük miktarda veriyi hızlı bir şekilde analiz edebilmesi ve insanların ayırt etmesi zor veya imkansız olan desenleri ve içgörülerini ortaya çıkarabilmesi nedeniyle özellikle değerlidir. Örneğin, tahmine dayalı bakımda, yapay zeka makinelerden gelen sensör verilerini analiz ederek arızaları tahmin edebilir ve böylece duruş sürelerini en aza indirebilir. Benzer şekilde, sağlık hizmetlerinde, AI araçları tıbbi görüntülerin veya hasta geçmişlerinin analizi yoluyla erken hastalık teşhisinde yardımcı olmak için kullanılabilir.

Yapay zeka algoritmalarının benimsenmesini hızlandıran üç temel faktör, bu teknolojilerin hızlı gelişimi ve yaygın kullanımını sağlamıştır:

- **Veri Çeşitliliği ve Hacminin Artışı**: Son yıllarda, dijitalleşmenin hız kazanmasıyla birlikte büyük ve çeşitli veri setlerine erişim artmıştır. Özellikle sensörler, IoT cihazları, sosyal medya platformları ve işletme sistemleri gibi kaynaklardan toplanan yapılandırılmış ve yapılandırılmamış veriler, yapay

zeka modellerinin eğitilmesi için gerekli olan temel yapı taşlarını oluşturmuştur. Büyük veri setleri, makine öğrenimi ve derin öğrenme algoritmalarının karmaşık ilişkileri öğrenebilmesini sağlamıştır.

- **Hesaplama Gücündeki İlerlemeler:** Modern yapay zeka, büyük ölçüde grafik işleme birimleri (GPU) ve özel yapay zeka hızlandırıcıları gibi yüksek performanslı donanımlar sayesinde mümkün olmuştur. Bu donanımlar, özellikle derin öğrenme mimarileri için gereken yoğun matris işlemlerini paralel olarak gerçekleştirebilmektedir. Bu teknolojik sıçrama, daha büyük modellerin eğitilmesini ve daha kısa sürede sonuç alınmasını sağlamıştır.
- **Algoritmik Gelişmeler:** Son yıllarda, yapay zeka algoritmalarındaki ilerlemeler (örneğin, derin öğrenme mimarileri olan Convolutional Neural Networks (CNN) ve Recurrent Neural Networks (RNN) gibi modeller), karmaşık problem çözme ve genelleştirme kapasitelerini büyük ölçüde artırmıştır. Özellikle Transformer tabanlı modellerin gelişimi, doğal dil işleme ve bilgisayar görüşü gibi alanlarda çığır açmıştır. Bu algoritmalar, çok boyutlu veri ilişkilerini daha etkin bir şekilde modelleyebilmektedir.

Bu üç faktör, yapay zeka çözümlerinin hızla endüstriyel, bilimsel ve günlük hayat uygulamalarında benimsenmesine olanak tanımış, aynı zamanda yüksek doğruluk ve verimlilik sağlayarak geniş bir etki alanı yaratmıştır (Cockburn et al., 2018; Reynolds et al., 2019).

Yapay zeka, sağladığı büyük potansiyele rağmen, bazı önemli zorluklarla karşı karşıyadır. Veri önyargıları, açıklanabilirlik eksikliği ve etik kaygılar, yapay zeka sistemlerinin geliştirilmesi ve uygulanmasında başlıca sorunlar arasında yer alır. AI sistemlerinin adil, şeffaf ve güvenli olması, hem araştırma hem de politika geliştirme süreçlerinde sürekli ele alınması gereken konulardır. Özellikle sağlık veya ulusal güvenlik gibi kritik alanlarda kullanılan yapay zeka sistemlerinde güvenilirlik çok büyük önem taşır. Bu güveni sağlamak için, kapsamlı doğrulama, yönetim çerçeveleri ve kullanıcı güvenini artırmaya yönelik açıklanabilir yapay zeka yöntemleri geliştirilmelidir (Cockburn et al., 2018; Reynolds et al., 2019).

Yapay zekanın "genel amaçlı bir teknoloji" olarak işlev görebilmesi, farklı sektörlerde inovasyonu teşvik etmektedir. Yapay zeka sadece mevcut alanlarda verimliliği artırmakla kalmaz, aynı zamanda mikroişlemci veya internetin icadına benzer şekilde tamamen yeni araştırma alanlarının kapılarını aralar. Bu dönüşümsel potansiyel, otonom sistemler, ileri düzey robotik ve kişiselleştirilmiş tıp gibi öncü alanlarda yoğun bir şekilde kullanılmasını teşvik etmiştir (Reynolds et al., 2019).

Yapay zeka sayesinde araştırmacılar ve kurumlar, karmaşık problemleri çözme, operasyonları optimize etme ve karar verme süreçlerini geliştirme konusunda büyük adımlar atmaktadır. Bu teknoloji, endüstriden bilime birçok alanda yenilikçi çözümler sunarak, geleceği şekillendiren önemli bir araç olmaya devam etmektedir (Reynolds et al., 2019).

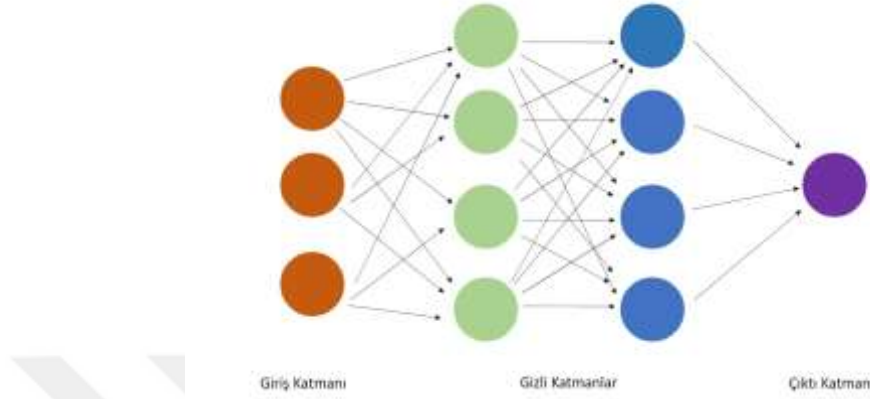
3.3.1. Yapay sinir ağları

Yapay Sinir Ağları (ANN), insan beynindeki biyolojik sinir ağlarını temel alarak geliştirilen bir yapay zeka modeli sınıfıdır. İlk olarak 1943 yılında Warren McCulloch ve Walter Pitts tarafından ortaya atılan bu fikir, 1980'lerden itibaren veri işleme ve modelleme alanında devrim yaratmıştır. Günümüzde ANN, görüntü işleme, doğal dil işleme, ses tanıma ve tahmin modelleri gibi birçok alanda kullanılmaktadır (Goodfellow et al., n.d.; McCulloch & Pitts, 1943).

Yapay Sinir Ağları, birbiriyle bağlantılı birçok yapay nöron veya "düğümünden" oluşur. Bu nöronlar, belirli bir girdiyi alır, üzerinde işlem yapar ve bir çıktı üretir. Temel yapılar Şekil 3.1'deki yapıyı göz önünde bulundurarak şu şekilde özetlenebilir:

- **Giriş Katmanı (Input Layer):** Verilerin modele girdiği katmandır. Örneğin, bir görüntü işleme modelinde pikseller, bu katmanın girdisini oluşturur.
- **Gizli Katmanlar (Hidden Layers):** Girdi verilerinin özelliklerinin çıkarıldığı ara katmanlardır. Bu katmanların sayısı ve nöronlarının düzenlenişi, modelin karmaşıklığını ve öğrenme kapasitesini belirler.

- **Çıkış Katmanı (Output Layer):** İşlenen verilerin son çıktısını üretir, örneğin bir sınıflandırma modelinde kategorilerden biri seçilir (Goodfellow et al., n.d.; LeCun et al., 2015).



Şekil 3.1. Yapay sinir ağı modeli

ANN, ağırlıklar ve aktivasyon fonksiyonları kullanarak veriyi işler. Her bağlantı bir ağırlık ile çarpılır ve her nöronun çıktısı, bir aktivasyon fonksiyonu aracılığıyla hesaplanır. Yaygın aktivasyon fonksiyonları arasında sigmoid, ReLU ve tanh yer alır. Öğrenme işlemi sırasında, modelin ağırlıkları, hata fonksiyonuna (örneğin, ortalama kare hata veya çapraz entropi) dayalı olarak optimize edilir (LeCun et al., 2015).

Yapay zeka, bilgisayarların veya makinelerin insana özgü öğrenme kabiliyetini Yapay Sinir Ağları aracılığıyla kazanmalarını sağlar. İnsanlar, öğrenme süreciyle edindikleri bilgi ve deneyimleri kullanarak farklı konular ve sorunlarla ilgili tahminlerde bulunabilirler. Ancak Yapay Sinir Ağları, karmaşık sorunları daha hızlı ve etkili bir şekilde çözme kapasitesine sahiptir. Bunu, insan beynini model alarak ve onun çalışma prensiplerini taklit ederek başarır. Bu sistem yalnızca tahminleme değil, aynı zamanda sınıflandırma, kümeleme, analiz ve veriler arasındaki ilişkileri belirleme gibi yeteneklere de sahiptir. Bu üstün yeteneklerle kontrol ve sistem tanımlama, görüntü ve ses tanıma, tahmin ve kestirim, arıza analizi, tıp, haberleşme, trafik ve üretim yönetimi gibi bir çok alanda kullanımı mevcuttur (Pirim, n.d.). Temel olarak kullanım alanları şöyle özetlenebilir :

- **Görüntü İşleme:** Derin öğrenme temelli sinir ağları, yüz tanıma ve tıbbi görüntü analizi gibi alanlarda yaygın olarak kullanılmaktadır (Krizhevsky et al., n.d.).
- **Doğal Dil İşleme :** Dil modelleri (GPT, BERT gibi) YSA'nın güçlü uygulamalarına örnek teşkil eder. Bu modeller, dil anlama ve üretiminde son derece başarılıdır (Vaswani et al., 2017).
- **Tahmin ve Optimizasyon:** Finansal veri analizi, enerji tüketimi tahmini ve lojistik optimizasyon gibi alanlarda da ANN kullanılmaktadır.

Karmaşık ve büyük verilerle çalışmada ANN kullanarak güçlü avantajlara sahip olunacaktır. Özellikle, derin öğrenme mimarileriyle birlikte büyük veri setlerinde ilişkileri ve desenleri etkili bir şekilde öğrenebilir. Ancak yüksek hesaplama gücü gerektirmesi ve açıklanabilirlik eksikliği, uygulama alanında güven sorunlarına yol açabilir. Büyük miktarda veri gerektirmesi de veri hazırlama sürecini zaman alıcı hale getirir (LeCun et al., 2015; Vaswani et al., 2017).

Yapay sinir ağları, özellikle derin öğrenme yaklaşımlarıyla günümüz teknolojisinin temel taşlarından biri haline gelmiştir. Görüntü işleme, dil işleme, ve tahmin sistemlerinde sunduğu çözümlerle, modern endüstriyel ve bilimsel problemleri çözmede etkili bir araçtır. Ancak, bu modellerin güçlü yanları kadar sınırlamaları da göz önüne alınmalı ve açıklanabilirlik gibi alanlarda ilerleme kaydedilmelidir.

3.3.2. Makine öğrenmesi

Makine öğrenmesi (ML), verilerden kendi başına "öğrenebilen" geniş bir algoritma yelpazesini içerir (Lindholm et al., 2022). Bilgisayarların belirli bir görevi açıkça programlanmadan öğrenmesini sağlayan bir yapay zeka dalıdır. Bu teknoloji, verilerden otomatik olarak desenler çıkararak ve bu desenleri öğrenerek tahminler ve çıkarımlar yapar. Arthur Samuel tarafından 1959'da tanımlanan makine öğrenmesi, günümüzde veri işleme, sınıflandırma, kümeleme, regresyon ve öneri sistemleri gibi alanlarda yaygın olarak kullanılmaktadır (Goodfellow et al., n.d.; Lindholm et al., 2022).

Makine öğrenmesi üç temel bileşene dayanır:

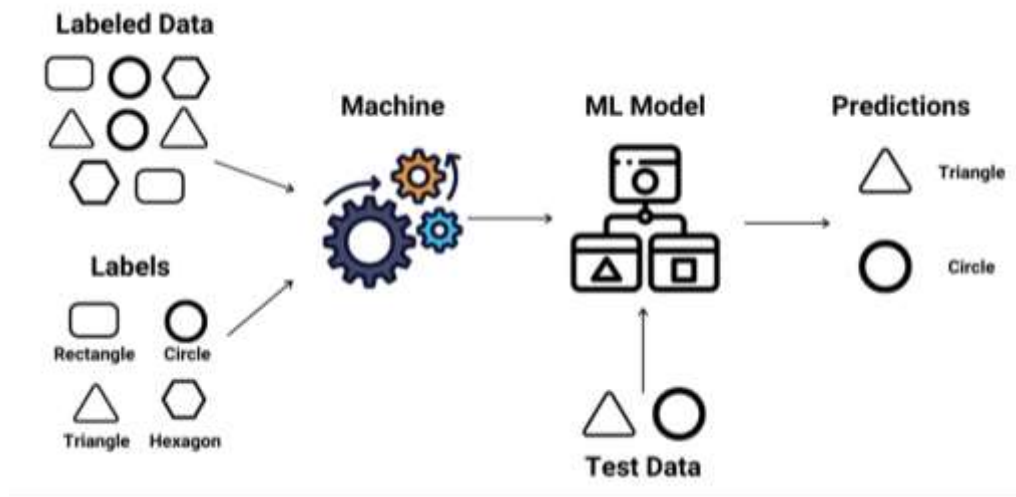
Veri: Modelin eğitilmesi için gerekli olan en önemli kaynaktır. Büyük ve kaliteli veri setleri, modelin doğruluğunu doğrudan etkiler.

Algoritmalar: Verilerden öğrenmeyi sağlayan matematiksel ve istatistiksel yöntemlerdir. Örneğin, doğrusal regresyon, karar ağaçları ve destek vektör makineleri gibi algoritmalar, farklı türde problemlere çözüm sunar.

Hedef Fonksiyonu: Modelin çıktısını değerlendiren ve hata oranını minimize etmeyi hedefleyen fonksiyondur. Bu fonksiyon, makine öğrenmesi sisteminin performansını ölçmek için kullanılır (Goodfellow et al., n.d.; Jordan et al., n.d.).

Makine öğrenmesi, verilerin yapısına ve kullanılan öğrenme türüne göre üç ana kategoriye ayrılır: **denetimli öğrenme**, **denetimsiz öğrenme** ve **pekiştirmeli öğrenme**. Her bir tür, farklı veri türleri ve problemler için uygun çözümler sunar.

Denetimli Öğrenme (Supervised Learning), etiketlenmiş bir veri seti (labeled dataset) kullanılarak model eğitiminin gerçekleştirildiği bir makine öğrenmesi yöntemidir. Bu yöntemde, her girdiye karşılık bir doğru çıktı (etiket) verilir ve model, bu ilişkiyi öğrenerek, Şekil 3.2, yeni verilere doğru tahminler yapmayı hedefler.



Şekil 3.2. Denetimli Öğrenme Modeli (Supervised Learning Model) (Ravish Raj, n.d.)

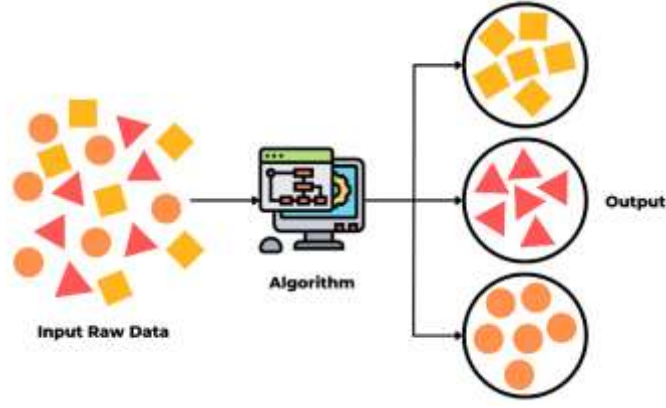
Denetimli öğrenme genellikle iki ana gruba ayrılır:

- **Sınıflandırma (Classification):** Veriler, önceden tanımlanmış kategorilere atanır. Örneğin, bir e-posta sisteminde gelen postaların "spam" veya "spam değil" olarak sınıflandırılabilir. Algoritmalar arasında lojistik regresyon, karar ağaçları ve destek vektör makineleri (SVM) bulunur.
- **Regresyon (Regression):** Sürekli bir değer tahmin edilir. Örneğin, ev fiyatlarının tahmini ya da hisse senedi değerlerinin öngörülmesi bu alan için değerlendirilebilir. Lineer regresyon ve çoklu doğrusal regresyon bu kategoride sıkça kullanılır.

Denetimli öğrenmenin başarısı, büyük ve kaliteli etiketlenmiş veri setlerine dayanır. Ancak, bu tür veri setlerinin oluşturulması oldukça zaman alıcı ve maliyetlidir. Günümüzde sağlık (ör. görüntü analizi), finans (ör. kredi riski değerlendirmesi) ve pazarlama (ör. müşteri segmentasyonu) gibi alanlarda yaygın olarak kullanılır (Goodfellow et al., n.d.; Jordan et al., n.d.).

Denetimsiz Öğrenme (Unsupervised Learning), etiketlenmemiş verilerle çalışır ve bu veriler arasındaki gizli yapıları veya desenleri keşfetmeyi amaçlar. Eğitim sırasında veri setinde Şekil 3.3'de görüldüğü gibi belirli bir çıktı etiketi bulunmaz; bunun yerine, model kendi içindeki ilişkileri öğrenir. Temel uygulama alanları şunlardır:

- **Kümeleme (Clustering):** Veri noktalarını, belirli özelliklerine göre gruplara ayırır. Örneğin, bir müşteri segmentasyonu analizi, farklı alışveriş alışkanlıklarına sahip müşterileri gruplandırmak için kullanılabilir. K-means, DBSCAN ve hiyerarşik kümeleme algoritmaları popülerdir.
- **Boyut İndirgeme (Dimensionality Reduction):** Büyük veri setlerindeki değişken sayısını azaltarak görselleştirme ve işlemeyi kolaylaştırır. Örneğin, PCA (Ana Bileşen Analizi) bu tür bir algoritmadır ve veri analitiğinde yaygın olarak kullanılır.



Şekil 3.3. Denetimsiz Öğrenme Modeli (Unsupervised Learning Model) (What Is Unsupervised Learning?, 2024)

Denetimsiz öğrenme, büyük veri kümelerinin analizinde güçlü bir araçtır, ancak doğru sonuçlar elde etmek için algoritmaların çıktılarını doğru şekilde yorumlamak kritik öneme sahiptir. Örneğin, biyoloji alanında genetik verilerin analizi veya sosyal medya platformlarında kullanıcı davranışlarını anlamak gibi uygulamalar bu yöntemeye dayalıdır (Hastie et al., 2009).

Pekiştirmeli Öğrenme (Reinforcement Learning), bir ajan (yapay zeka sistemi) ile çevresi arasındaki etkileşimlere dayanır. Ajan, çevresinden aldığı ödül (reward) veya ceza (penalty) sinyallerine göre bir strateji geliştirir ve bu strateji, uzun vadede en yüksek ödülü elde etmek için optimize edilir. Bu tür öğrenmede, veri seti yerine bir ortam (environment) bulunur ve ajan bu ortamda kararlar alır.

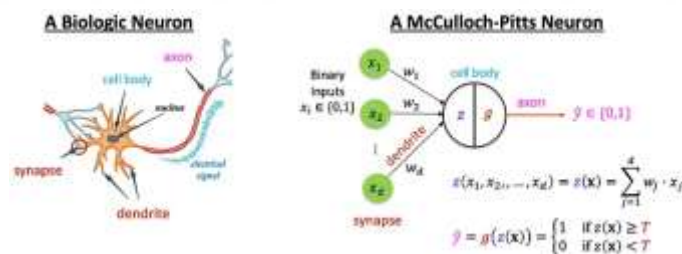
- **Ödül Tabanlı Sistem:** Ajanın eylemleri, her adımda bir ödül veya ceza ile değerlendirilir. Amaç, toplam ödülü maksimuma çıkarmaktır.
- **Markov Karar Süreçleri (Markov Decision Processes):** Pekiştirmeli öğrenmenin temelini oluşturur. Bu süreç, bir dizi durum (state), eylem (action) ve olasılık geçişleri üzerinden çalışır.

Pekiştirmeli öğrenme, özellikle oyun geliştirme (ör. AlphaGo) ve robotik (ör. insansız hava araçlarının kontrolü) gibi alanlarda büyük başarılar elde etmiştir. Ayrıca, otonom araçların rotasını belirlemek ve enerji sistemlerini optimize etmek gibi kritik uygulamalarda da kullanılır (Silver et al., 2016; Sutton & Barto, 1999).

Makine öğrenmesi, veri analitiği ve otomasyon süreçlerinde devrim yaratarak işletmelerin ve bilim insanlarının karmaşık problemleri çözmesine olanak tanır. Denetimli, denetimsiz ve pekiştirmeli öğrenme yaklaşımlarıyla, farklı alanlardaki uygulamaları çeşitlenmekte ve derinleşmektedir. Ancak, etik ve önyargı sorunları gibi zorluklar, makine öğrenmesinin daha geniş çaplı benimsenmesi için çözülmesi gereken önemli konulardır.

3.4. Derin Öğrenme

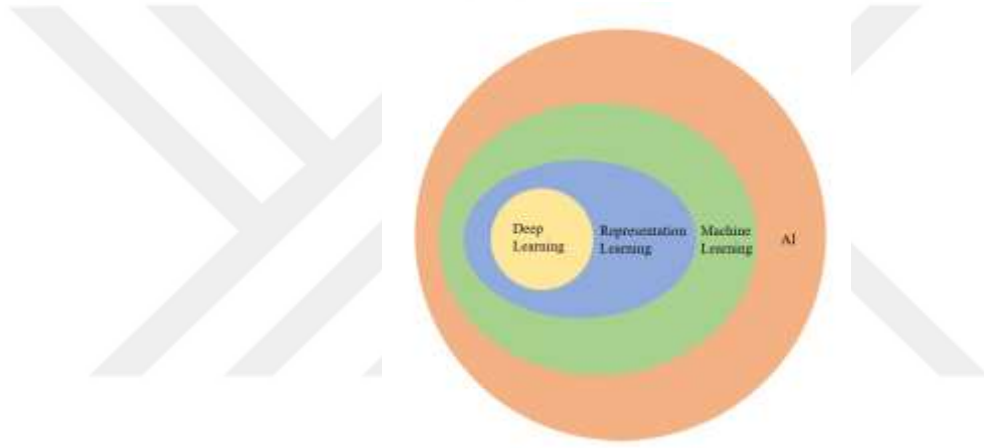
Derin öğrenme, yapay zekânın bir alt dalı olarak uzun bir geçmişe sahiptir. İlk temelleri, McCulloch ve Pitts tarafından 1943 yılında tanıtılan yapay nöron modeliyle atılmıştır. Bu model, insan beynindeki nöronların matematiksel olarak modellenmesini Şekil 3.4'teki gibi kurulumuştur (McCulloch & Pitts, 1990). Daha sonra, 1958 yılında Frank Rosenblatt, ilk öğrenen sinir ağı olan Perceptron'u geliştirmiştir. Ancak, Minsky ve Papert'in 1969 tarihli çalışmasında, tek katmanlı Perceptron ağlarının sınırlamaları gösterilmiştir ve bu durum, yapay sinir ağlarına olan ilgiyi bir süre azaltmıştır (LM Po, 2024)



Şekil 3.4. McCulloch ve Pitts tarafından oluşturulan insan beynindeki nöronların matematiksel olarak modellenmesi (Rosenblatt, n.d.)

1980'lerde, geri yayılım algoritmasının (backpropagation) tanıtılmasıyla sinir ağlarına olan ilgi yeniden canlanmıştır. Bu yöntem, sinir ağlarının daha etkin bir şekilde eğitilmesini sağlamıştır (D. E. Rumelhart et al., 1986). 2000'li yıllarda ise, büyük veri (big data) ve yüksek işlem gücüne sahip grafik işlem birimlerinin (GPU) yaygınlaşması sayesinde derin öğrenme, karmaşık problemlerin çözümünde etkili bir araç haline gelmiştir (D. Rumelhart et al., 2002).

Makine öğrenmesi yapay zekânın bir alt dalıdır ve verilerden öğrenerek otomatik tahmin ve kararlar almayı hedefler. Derin öğrenme (DL) ise, makine öğrenmesinin bir alt kümesidir ve çok katmanlı yapay sinir ağları kullanarak daha karmaşık problemlere çözüm sunar (Goodfellow et al., n.d.). Ancak son yıllarda derin öğrenme algoritmalarının popülaritesindeki artışın sebebi veriye erişimin artması ve daha iyi hesaplama donanımlarından kaynaklanmaktadır. Goodfellow ve diğerleri (Goodfellow et al., n.d.) derin öğrenmeyi Şekil 3.5'teki Venn diyagramı ile açıklamaktadır. Derin öğrenme modellerinin geleneksel makine öğrenimi yöntemlerinden daha iyi performans göstermesi için genellikle büyük miktarda veriye ihtiyaç duyar ve hesaplama açısından da daha maliyetlidir.



Şekil 3.5. Derin öğrenmenin yapay zekânın sadece bir alt kümesi olduğunu gösteren Venn şeması

Derin öğrenmenin yükselişi, özellikle üç ana faktörle ilişkilidir:

- I. **Büyük Veri:** Günümüzde sosyal medya, sensörler ve IoT cihazlarından elde edilen verilerin hacmi büyük oranda artmıştır (D. Rumelhart et al., 2002).
- II. **İleri Donanım Teknolojisi:** GPU ve TPU gibi yüksek performanslı donanımlar, derin öğrenme modellerinin eğitimini hızlandırmıştır (Krizhevsky et al., n.d.).

III. **Gelişmiş Algoritmalar:** Konvolüsyonel Sinir Ağları (CNN), Tekrarlayan Sinir Ağları (RNN) ve Transformer modelleri gibi yeni mimariler, daha etkili ve hızlı çözümler sunmaktadır (Vaswani et al., 2017).

Derin öğrenme, büyük veri ve güçlü algoritmalar sayesinde bilim ve teknoloji dünyasında devrim yaratmaktadır. Gelecekte, derin öğrenme modellerinin daha verimli ve daha etik bir şekilde kullanılmasını sağlamak için açıklanabilir yapay zeka (Explainable AI) gibi yeni yaklaşımların geliştirilmesi beklenmektedir.

3.4.1. Derin öğrenmenin kullanıldığı çalışmalar

Derin öğrenme, yapay zekâ ve makine öğrenmesinin önemli bir dalı olarak birçok sektörde devrim yaratmış ve çeşitli karmaşık problemlerin çözümüne katkı sağlamıştır. Bu bölümde, derin öğrenmenin uygulama alanlarını, kullanılan yöntemleri ve elde edilen sonuçları geniş bir şekilde ele alacağız.

Görüntü İşleme ve Bilgisayarlı Görü, derin öğrenme, özellikle görüntü işleme ve bilgisayarlı görü (computer vision) alanında büyük ilerlemeler sağlamıştır. Konvolüsyonel Sinir Ağları (CNN), görüntüdeki özellikleri otomatik olarak çıkararak sınıflandırma, nesne tespiti ve segmentasyon gibi görevlerde kullanılır.

- **Tıbbi Görüntüleme:** Derin öğrenme, X-ray, MRI ve CT taramaları gibi tıbbi görüntülerde anormallik tespiti için yaygın olarak kullanılmaktadır. Örneğin, bir çalışmada göğüs röntgenlerindeki kanser belirtilerini tespit etmek için CNN kullanılarak doktorlarla eşdeğer doğrulukta sonuçlar elde edilmiştir (Rajpurkar et al., 2017).
- **Otonom Araçlar:** Otonom sürüşte kullanılan derin öğrenme algoritmaları, trafik işaretlerini tanıma, şerit tespiti ve nesne algılama gibi kritik görevleri yerine getirmektedir (Bojarski et al., 2016).

Doğal Dil İşleme (NLP), metin işleme ve doğal dil anlama görevlerinde etkili sonuçlar sunmuştur. Özellikle Transformer mimarisi, dil modellemeyi ve metin tabanlı görevleri büyük ölçüde geliştirmiştir.

- **Makine Çevirisi:** Google Translate gibi araçlarda, derin öğrenme tabanlı modeller kullanılarak çok dilli çeviri işlemleri geliştirilmiştir (Vaswani et al., n.d.).
- **Metin Analizi:** Duygu analizi, spam tespiti ve müşteri geri bildirimlerinin sınıflandırılmasında derin öğrenme modelleri başarıyla uygulanmıştır (Devlin et al., 2018).

Sağlık ve Biyoinformatik, derin öğrenme sayesinde tanı, tedavi planlaması ve ilaç geliştirme süreçlerinde büyük ilerlemeler kaydetmiştir.

- **Hastalık Tahmini:** Elektronik sağlık kayıtlarından hastalık tahmini yapılmasında RNN tabanlı modeller kullanılarak yüksek doğruluk oranları elde edilmiştir (Choi et al., n.d.).
- **Genetik Veri Analizi:** DNA sekanslarının analizinde, derin öğrenme modelleri genetik hastalıkları önceden tahmin etmek için kullanılmaktadır

Finans ve Ekonomi, derin öğrenme algoritmaları veri analitiği, risk yönetimi ve dolandırıcılık tespiti gibi alanlarda kullanılmıştır.

- **Dolandırıcılık Tespiti:** Kredi kartı işlemlerindeki sahtekarlıkları tespit etmek için derin öğrenme tabanlı modeller geliştirilmiştir.
- **Portföy Optimizasyonu:** Büyük veri analitiği ve derin öğrenme yöntemleri, yatırım stratejilerinin belirlenmesinde kullanılmaktadır.

Eğitim ve İnsan Bilgisayar Etkileşimi, bireysel öğrenme deneyimlerini optimize etmek ve öğrenme materyallerini kişiselleştirmek için kullanılır.

- **Akıllı Öğrenme Sistemleri:** Öğrencilerin öğrenme hızını ve anlayış düzeyini analiz ederek özel eğitim programları öneren sistemler geliştirilmiştir (Piech et al., 2015).

Oyun ve Simülasyon, karmaşık stratejik oyunlarda ve simülasyonlarda üstün performans göstermiştir.

- **AlphaGo:** Derin Q-öğrenme algoritması, Go oyununda insan şampiyonlarına karşı üstünlük sağlamıştır (Silver et al., 2016).

Enerji ve Çevre, enerji talebini tahmin etmek, yenilenebilir enerji kaynaklarının verimliliğini artırmak ve iklim değişikliği analizinde kullanılmaktadır.

- **Enerji Optimizasyonu:** Akıllı şebeke sistemlerinde enerji talep tahmini ve optimizasyonu için uygulanmaktadır (Massaoudi et al., 2021).

Sanat ve Yaratıcılık, sanat ve tasarım alanında derin öğrenme, görsel sanatlar, müzik bestesi ve film yapımında yenilikçi yaklaşımlar sunmaktadır.

- **Sanatsal Stil Transferi:** Görüntülerin bir sanat stiline dönüştürülmesinde kullanılan algoritmalar, yaratıcı endüstrilerde büyük bir etki yaratmıştır (Gatys et al., 2015).

3.4.2. Derin öğrenme kütüphaneleri

Derin öğrenmenin insan beyninin bilişsel süreçlerini taklit eden makine öğrenimi ve yapay zekanın bir alt kümesi olduğuna yukarıda değinmiştik. Veri bilimi, istatistik ve tahmine dayalı modelleme gibi önemli alanları içerir. Birçok derin öğrenme kütüphanesi, veri yükleme ve verimli model eğitimi için kullanıcı dostu araçlar ve komutlar sunarak derin öğrenme modellerinin geliştirilmesini ve dağıtımını kolaylaştırır. Python'daki önde gelen derin öğrenme kütüphaneleri, hassas ve doğru tahminler yapmayı daha erişilebilir hale getiren araçlar sağlar. TensorFlow, Keras ve PyTorch gibi önemli derin öğrenme kütüphaneleri derin öğrenme analizlerinde kullanılan önemli kütüphanelerdir.

Caffe, Berkeley Vision and Learning Center tarafından geliştirilmiş son derece ileri düzey bir araç kutusudur. Modüler yapısı, verimliliği ve minimal ek çaba ile birden fazla GPU'yu destekleme özelliği ile dikkat çeker. Ağ mimarisini ve çözücü metodolojilerini belirlemek için JSON benzeri bir metin dosyası kullanır.

TensorFlow: Google tarafından geliştirilen, makine öğrenimi için uçtan uca açık kaynak bir platformdur. Makine öğrenimi uygulamaları için kapsamlı, esnek araçlar, kütüphaneler ve topluluk markalarından oluşan bir ekosisteme sahiptir. Birçok soyutlama seviyesi bulunduğundan, çözülmesi gereken probleme uygun olanın seçilmesine olanak tanır. Keras API'si, birçok derin öğrenme probleminde yaygın olarak kullanılmaktadır. Google, Intel, Twitter, CocaCola, Airbnb, DeepMind, GE Healthcare, NeRSC gibi büyük şirketler, TensorFlow ile derin öğrenme uygulamaları geliştirmektedir.

Keras, Python dilinde yazılmış bir paket olup, arka planda Theano veya TensorFlow kullanır. Kapsamlı çözümler oluşturarak daha basit kullanım ve okunabilirlik sağlar. Bu paket, optimizasyon, normalizasyon ve aktivasyon fonksiyonları gibi en ileri düzey algoritmalara sahiptir. Keras, hem Theano hem de TensorFlow'ü destekler; ancak, giriş verisi boyutlarıyla ilgili varsayımlar farklılık gösterdiğinden, her iki paket kullanımı ile uyumluluğu sağlamak için dikkatli bir tasarım gereklidir.

Theano, çok boyutlu diziler içeren matematiksel formülleri verimli bir şekilde tanımlayan, bunların değerlendirilmesini ve optimize edilmesini kolaylaştıran bir Python kütüphanesidir. Python'da geliştirilen genel amaçlı bir kütüphane olup, derin öğrenme için doğrudan kullanılabilmesi gibi, Keras veya Lasagne gibi frameworklerle de kullanılabilir. 2007 yılından bu yana geliştirilmiş ve birçok makine öğrenimi probleminin çözümünde kullanılmıştır. 2017 yılında Yoshua Bengio, Theano kütüphanesinin geliştirilmesinin sonlandırıldığını duyurmuştur.

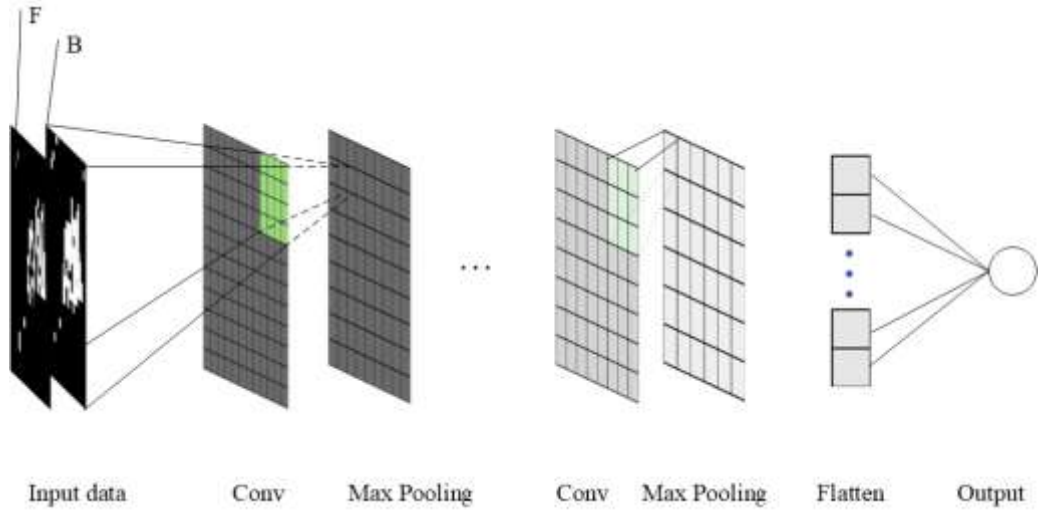
PyTorch, ekran kartlarını kullanabilme yeteneği sayesinde hız açısından oldukça popüler bir kütüphanedir. PyTorch'un başarısının sebeplerinden biri, sinir ağı modellerini kolayca oluşturabilmesidir. Ayrıca, PyTorch, arka planda CPU ve GPU gibi farklı donanım kaynaklarını aynı anda kullanabilme yeteneğine sahiptir. PyTorch, Numpy kütüphanesine benzer bir Tensor yapısı sunar ve bu yapı, veri manipülasyonunu daha verimli hale getirir. Tensor yapısı, sayısal hesaplamalar ve derin öğrenme için oldukça uygundur ve GPU desteği sayesinde büyük veri kümeleriyle çalışırken önemli bir hız avantajı sağlar.

3.4.3. Derin öğrenme mimarileri

Makine öğrenimi ailesinin bir alt kümesi olan derin öğrenme, görev odaklı yöntemler yerine veri setlerini öğrenmeye odaklanır. Geleneksel makine öğrenimi yöntemlerinin aksine, derin öğrenme daha soyut bilgilerin öğrenilmesine olanak tanıyan daha derin mimariler oluşturma avantajına sahiptir. Derin öğrenmenin temel özelliklerinden biri makine öğrenimi modelleri için veriden anlamlı ve faydalı özelliklerin elle seçilmesi ihtiyacını ortadan kaldırarak verilerden otomatik olarak anlamlı özellikler öğrenme yeteneğidir. Geleneksel makine öğrenimi yöntemlerinde, modelin verilerden doğru sonuçlar çıkarabilmesi için uzmanlar, veriye ait önemli özellikleri manuel olarak seçip tasarlamak zorundadır. Ancak derin öğrenme, bu süreci otomatikleştirir. Yani, model, verilerdeki karmaşık desenleri ve ilişkileri keşfederek, en uygun özellikleri kendi başına öğrenir. Bu, özellikle büyük ve karmaşık veri setlerinde büyük bir avantaj sağlar. Çünkü insan müdahalesine gerek kalmadan daha doğru ve verimli sonuçlar elde edilebilir. Derin öğrenme, CNN, RNN ve LSTM gibi algoritmalar aracılığıyla çeşitli alanlarda pratik uygulamalar bulmuştur (Coşkun et al., 2017) ve her biri sonraki bölümlerde ayrıntılı olarak tartışılacaktır.

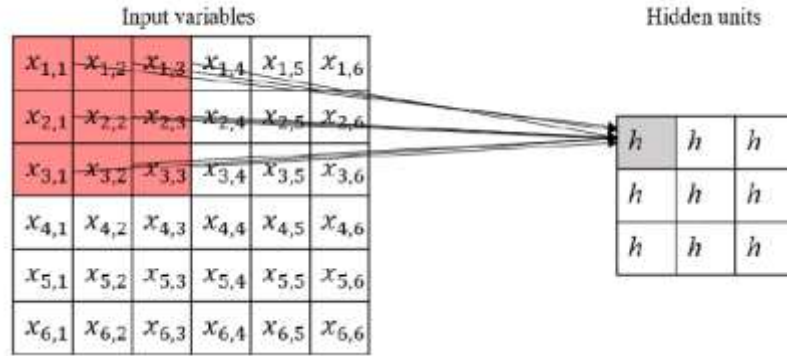
3.4.3.1. Evrişimli sinir ağları (Convolutional Neural Networks)

Evrişimli Sinir Ağı, görüntüler, ses ve zaman serileri gibi iki veya üç boyutlu verilerdeki yerel özellikleri tanıyabilen bir derin öğrenme modelidir (Lecun & Bengio, 1995). CNN'ler, özellikle görüntü işleme ve doğal dil işleme gibi alanlarda yaygın olarak kullanılmaktadır. Temelde, giriş verilerinden belirli desenler veya özellikler çıkaran konvolüsyon katmanları aracılığıyla verileri işler ve pooling katmanları ile, Şekil 3.6, boyutları azaltarak önemli bilgileri korur.



Şekil 3.6. CNN Model'inin yapısı (Li & Jing, 2022)

Konvolüsyon Katmanları (convolution layers), CNN modelinin temel bileşenidir. Veriyi filtreler aracılığıyla Şekil 3.7'deki gibi küçük parçalara bölerek işler. Her filtre, 2 boyutlu konum verilerinin (örneğin, enlem-boylam çiftlerinin) tahmininde, konvolüsyon katmanları kullanılarak verinin çevresel ilişkileri ve yerel bağımlılıkları öğrenilebilir. Bu yapı, özellikle konum verilerinin geçmiş hareketlerinden gelecekteki pozisyonu tahmin etmede yararlıdır.



Şekil 3.7. Konvolüsyonel katmanların diagramı (Goodfellow et al., n.d.)

Pooling (havuzlama), CNN'in işlenen veri boyutunu küçültmek ve modelin işlem hızını artırmak için kullandığı bir yöntemdir. Pooling işlemi, verinin küçük bölgelerinin ortalamasını veya maksimum değerini alarak önemli bilgileri korur. Bu şekilde, modelin karmaşıklığı azalır ve model daha hızlı çalışır.

Convolutional Neural Networks (CNN), derin öğrenme ve yapay zeka alanında devrim yaratan güçlü bir modeldir. Bu güçlü modelin diğer derin öğrenme modellerine karşı sağlamış olduğu üstünlük sağladığı alanlar vardır. Bu üstün alanları şöyle açıklayabiliriz:

Yerel Bağımlılıkları Öğrenme Yeteneği ile CNN'ler, katmanlarında bulunan filtreler sayesinde verilerdeki yerel kalıpları yakalamakta başarılıdır. Uçuş rotası ve gecikme verilerinde hız, konum, yükseklik gibi özelliklerin yerel ve kısa vadeli değişiklikleri, gelecekteki tahminler için önemlidir. CNN'in bu kısa vadeli korelasyonları yakalayabilmesi, tahmin performansını artırır.

Zaman Serisindeki Uzun Vadeli Özellikleri Öğrenme, uçuş verisi zaman serisi verilerden olduğundan dolayı CNN katmanları bu serilerdeki uzun vadeli desenleri tanıyarak tahminlerde doğruluk sağlayabilir. Filtrelerin konvülyasyon işlemi, uçuşun zaman içindeki yön değişikliklerini veya hızdaki değişimleri yakalamakta başarılıdır, böylece uçuş rotası ve gecikme tahminleri yapılırken önemli bilgileri öğrenir.

Karmaşıklığı Azaltarak Daha Hızlı İşlem Yapabilme, CNN modelleri, veri boyutunu azaltmak için kullanılan "pooling" işlemi sayesinde gereksiz bilgileri filtreleyerek veri boyutunu küçültür. Bu özellik, hem işlem hızını artırır hem de daha küçük veri boyutuyla analiz yapılmasını sağlar. Uçuş tahmini gibi zaman hassasiyeti taşıyan işlemlerde bu hız avantajı önemli bir üstünlük sağlar.

Görsel Verilerin Kullanılabilme Potansiyeli, uçuş tahmini yalnızca metinsel veya sayısal verileri değil, uçuş rotasının 2D/3D görselleştirilmesiyle oluşan harita veya radar görüntülerini de kullanabilir. CNN, bu tip görsel verilerle çalışmak için idealdir. Özellikle uçuş rotasının görsel olarak takip edilmesi gereken durumlarda CNN modelleri daha başarılı sonuçlar verebilir.

Kullanıcı Hatalarını Azaltmak İçin Veri İşlemeye Gerek Duymaması, CNN özellikle ham verilerle de çalışabilme özelliği sayesinde, çok fazla veri ön işleme adımına gerek duymadan öğrenme yeteneğine sahiptir. Bu, uçuş rotası ve gecikme tahmini verilerinin modelde hızlı ve etkili bir şekilde kullanılmasını sağlar.

RNN(Recurrent Neural Network-Tekrarlayan Sinir Ağı) ve LSTM(Long Short Term Memory-Uzun Kısa Süreli Bellek) gibi modeller, uzun vadeli bağımlılıkları öğrenmede başarılı olsa da, CNN'in sağladığı yerel özelliklerin ve kısa vadeli kalıpların hızlıca öğrenilmesi gibi bir avantaja sahip değildir. CNN'ler, uçuş rotasındaki ani değişiklikleri ve gecikmelere neden olan kısa süreli anormallikleri tanımlamada başarılıdır.

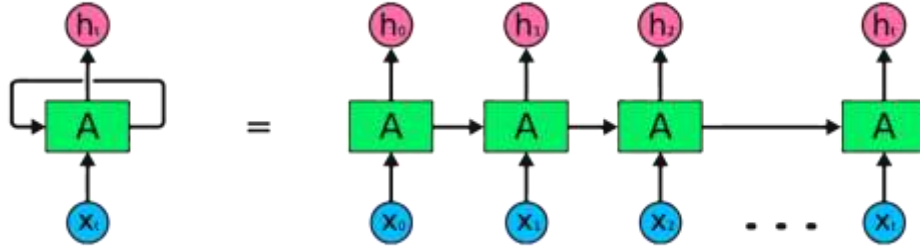
Karar Ağaçları veya Regresyon Modelleri gibi geleneksel yöntemler, veri içindeki uzun vadeli ve kısa vadeli bağımlılıkları aynı anda yakalamada yetersiz kalabilir. Bu nedenle CNN'in, uçuş tahmininde, geleneksel yöntemlere göre daha yüksek doğruluk sunduğu gözlemlenmektedir.

3.4.3.2. Tekrarlayan sinir ağı (Recurrent Neural Network)

Sıralı verileri işlemek için tasarlanmış bir tür sinir ağına geri beslemeli sinir ağı modelidir (D. Rumelhart et al., 2002). Zaman serisi, konuşma ve metin gibi zamansal bir boyuta sahip verileri analiz edebilirler. RNN'lerin temel özellikleri arasında, önceki verileri hatırlama ve bu bilgileri yeni verilerle birleştirerek tahminlerde bulunma yeteneği bulunur. Bu özellikler, RNN'leri doğal dil işleme (NLP) ve diğer sıralı veri analiz görevlerinde oldukça önemli ve tercih edilen bir model haline getirir.

RNN, katmanlardaki nöronlara gelen bilgilere matematiksel işlemler uygulayarak çıktı üreten bir yapıda çalışır. Bu yapıda bilgi yalnızca ileri doğru işlenir. Temelde, giriş verileri ağıdan geçirilerek bir çıktı elde edilir. Bu çıktı, doğru değerlerle karşılaştırılarak hata hesaplanır. Ardından, ağıdaki ağırlıklar hataya bağlı olarak güncellenir ve böylece en doğru sonuçları veren bir model oluşturulur.

Yinelenen (recurrent) yapılarda ise, yalnızca o andaki giriş verisine değil, önceki girişlere de bağlı olarak elde edilir. Şekilde görülebileceği gibi, RNN'de t anındaki giriş verilerinin yanı sıra, $t-1$ anındaki gizli katman sonuçları da t anındaki gizli katmanın girdisi olarak kullanılır. Yani, $t-1$ anındaki girişe verilen karar, t anındaki kararı da etkiler. Bu ağlarda girişler, mevcut ve önceki bilgilerin birleşimiyle çıktı üretir.



Şekil 3.8. RNN Model'ini yapısı (Abid Ali Awan, 2022)

Ardışık verileri işlemek ve kısa vadeli bağımlılıkları tanıma konusunda başarılı bir modeldir. Ancak, vanishing gradient (kaybolan gradyan) problemi olarak bilinen bir sorunla karşı karşıya kalır. Bu problem, ağıın katman sayısı arttıkça gradyanların küçülmesi ve kaybolması anlamına gelir. RNN'ler bu nedenle uzun vadeli ilişkileri yakalayamaz, dolayısıyla geçmişteki önemli bilgileri kaybeder. Uçuş rotası ve gecikme tahmininde, belirli bir uçuşun daha önceki konumu, hızı, gecikme süreleri ve hava durumu gibi geçmiş verilerin uzun vadeli etkilerinin dikkate alınması gerekir. RNN'ler bu bilgileri koruyamadığından uçuş tahminlerindeki doğruluk düşer.

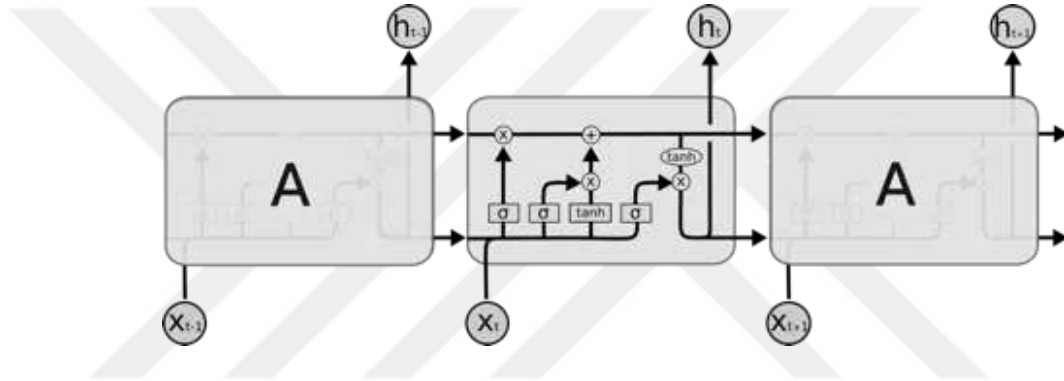
Recurrent Neural Networks (RNN), sıralı verilerle çalışırken geleneksel yapay sinir ağlarından önemli bir fark yaratır. Geleneksel ağlar, yalnızca mevcut giriş verilerini kullanarak çıktı üretirken, RNN'ler, hem mevcut girişleri hem de önceki zaman adımlarındaki bilgileri birleştirerek daha anlamlı ve doğru tahminlerde bulunabilir. Bu özellik, RNN'leri özellikle doğal dil işleme, zaman serisi analizleri ve diğer sıralı veri görevlerinde güçlü bir araç haline getirir. RNN'ler, önceki bilgileri hatırlayabilme ve gelecekteki tahminler için bu bilgileri kullanabilme yetenekleri sayesinde, daha karmaşık ve dinamik veri setlerine uyum sağlama konusunda önemli avantajlar sunar. Bu nedenle, sıralı verilerle yapılan modellemelerde RNN'ler, verimlilik ve doğruluk açısından vazgeçilmez bir model olmuştur.

3.4.3.3. Uzun-kısa süreli hafıza (LSTM-Long Short-Term Memory)

Bir tür RNN mimarisine sahip modeldir. Ardışık girdilerde uzun vadeli bağımlılıkları depolayabilme özelliğine sahiptir. Zamansal seriler, metin ve ses dahil olmak üzere ardışık verileri LSTM'ler kullanılarak işlenebilir ve analiz edilebilir. Uzun vadeli bağımlılıkları öğrenme konusunda üstün bir yeteneğe sahiptir. Bu, ağıın gizli

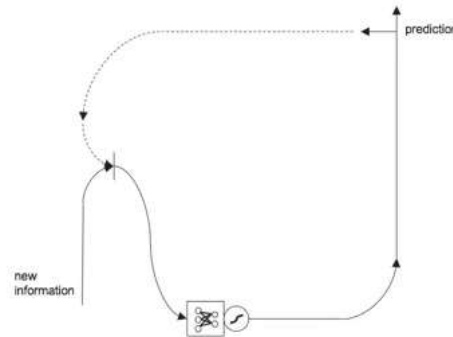
katmanlarında kullanılan özel "kapı" yapıları sayesinde mümkün olur. Bu kapılar, hangi bilgilerin saklanacağına ve hangi bilgilerin unutulacağına karar verir. Bu özellik, uzun dizilerdeki önemli bilgilerin korunmasını, hücreye yazılmasını ve okunmasını sağlar.

LSTM hücreleri, hangi bilgilerin depolanacağına, ne zaman okunacağına, yazılacağına veya silineceğine karar vermek için kapıları kullanır. Bu kapılar, bir ağ yapısına ve aktivasyon fonksiyonlarına sahiptir. Tıpkı nöronlar gibi, gelen veriyi ağırlıklarına göre geçirir ya da engeller. Bu ağırlıklar, ağın eğitim sürecinde öğrenilir. Bu yapı sayesinde hücre, hangi verileri kabul edip hangi verileri bırakacağına ya da sileceğine karar verir.

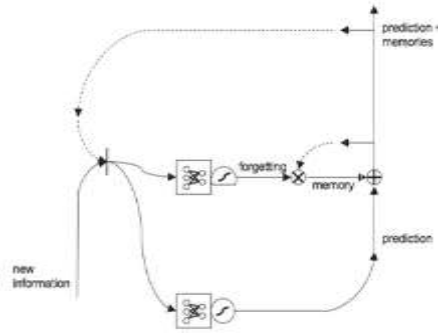


Şekil 3.9. LSTM Network yapısı

Şekil 3.10'da klasik bir RNN çalışma sistemi gösterilmiştir. Şekil 3.11'de ise Şekil 3.10'daki hücre yapısına bellek eklenmiştir. Bu şekilde, uzun zaman önceki veriler de dikkate alınarak kararlar verilmiş olur.

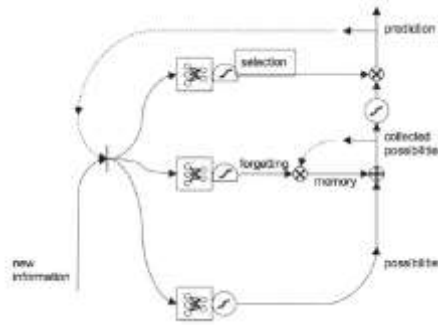


Şekil 3.10. Basit RNN yapısı

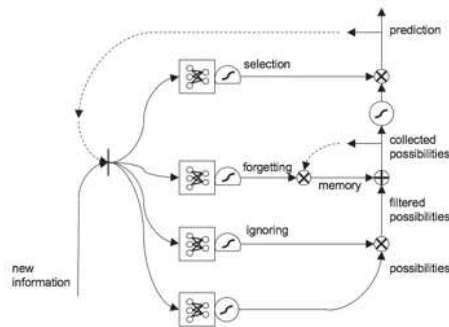


Şekil 3.11. Bellek yapısı

Şekil 3.11’de yer alan + işareti, eleman bazında yapılan toplama işlemini ifade ederken, x işareti ise eleman bazında çarpma işlemini göstermektedir. Çarpma işlemi, verinin hangi kısımlarının ne kadar etkili olacağını belirler ve bellekteki verilerin ağırlıklarla çarpılmasıyla bu hesaplama yapılır. Ardından, toplama işlemi ile bellekteki veriler ve olasılıklardan gelen bilgiler birleştirilerek tahmin üretilir.



Şekil 3.12. Selection birimi



Şekil 3.13. Ignoring birimi

LSTM'in yapısında yer alan kapılar, modelin hangi bilgiyi tutacağını, hangi bilgiyi unutacağını ve hangi bilgiyi bir sonraki aşamaya ileteceğini belirleyen filtrelerdir. Bu

kapılar, öğrenme sürecinde bellek hücresine giren ve çıkan bilgiyi kontrol eder ve her biri kendi içinde küçük bir sinir ağıyla donatılmıştır. Şekil 3.12’de toplama ve bellekte tutma kapısı eklenmiştir. Bu kapı, gelen bilgiyi toplar ve hangi verilerin bellekte tutulup hangilerinin tahmin işlemi için kullanılacağına karar verir. Kapı, gelen bilgileri bir filtre aracılığıyla değerlendirir ve bellek hücresinde yalnızca gerekli bilgileri saklar. Diğer bir kapı da filtreleme ve güncelleme kapısıdır. Şekil 3.13’de görüldüğü gibi belleğe eklenmeden önceki bilgileri filtrelemek için bir diğer kapı devreye girer. Bu kapı, hangi bilginin önemli olduğunu belirler ve önemsiz olanları eler. Bu işlem, giriş verilerini bellekte mevcut bilgilerle birleştirmeden önce uygulanır. Element bazında çarpma işlemi yapılarak gereksiz bilgiler filtrelenir.

LSTM'in bu kapı mekanizmaları, zaman serisi verilerindeki uzun vadeli bağımlılıkları etkili bir şekilde öğrenmesini sağlar. Ayrıca, bu kapılar farklı yapılandırmalarla değiştirilebilir ve modelin gereksinimlerine göre özelleştirilebilir. Bu esneklik, LSTM'i zaman serisi tahmininden doğal dil işleme ve diğer karmaşık görevlerde kullanıma uygun hale getirir.

3.5. Derin Öğrenmede Kullanılan Önemli Parametreler

Derin öğrenme, yapay sinir ağlarına dayanan bir yaklaşım olduğu için, yapay sinir ağlarında yer alan tüm terimler ve kavramlar, derin öğrenme mimarilerinde de geçerlidir. Bunlar şunlardır:

- Aktivasyon fonksiyonları
- Optimizasyon fonksiyonları
- Maliyet (Kayıp) fonksiyonları

3.5.1. Aktivasyon fonksiyonları

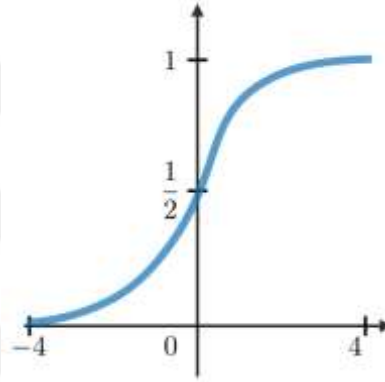
Aktivasyon fonksiyonları, derin öğrenme modelleme süreçlerinde ağın her bir katmanındaki nöronların ne kadar aktif olacağını belirleyen matematiksel fonksiyonlardır. Bu fonksiyonlar, modelin doğruluğunu artırmak, doğrusal olmayan özellikler öğrenebilmesini sağlamak ve karar sınırlarını daha karmaşık hale getirebilmek için kritik bir öneme sahiptir (LeCun et al., 2015). Sigmoid Fonksiyonu, ReLU

Fonksiyonu, Tanh Fonksiyonu, Softmax Fonksiyonu ve Leaky ReLU yaygın olarak kullanılan aktivasyon fonksiyonlarından bazılarıdır.

3.5.1.1. Sigmoid fonksiyonu

Sigmoid fonksiyonu, çıktıyı $[0, 1]$ aralığında sınırlar ve özellikle binary sınıflandırma problemlerinde tercih edilir. Bu fonksiyon, nöronların çıktısını olasılık olarak yorumlamaya olanak tanır.

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



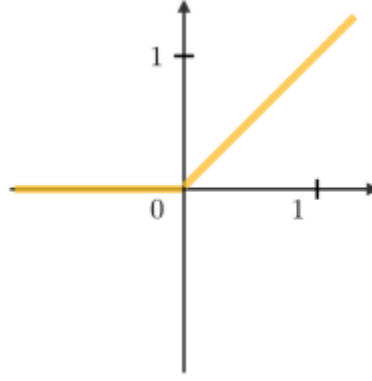
Şekil 3.14. Sigmoid fonksiyonu

Sigmoid fonksiyonu genellikle sınıflandırma problemlerinde, özellikle ikili sınıflandırma durumlarında çıktıların olasılık olarak yorumlanmasında kullanılır (Goodfellow et al., n.d.).

3.5.1.2. ReLU (Rectified Linear Unit) fonksiyonu

ReLU, derin öğrenme modellerinde sıklıkla kullanılan aktivasyon fonksiyonlarından biridir. ReLU, negatif değerleri sıfıra eşitleyip pozitif değerleri olduğu gibi bırakır. Bu, modelin daha hızlı öğrenmesini ve daha verimli çalışmasını sağlar.

$$ReLU(x) = \max(0, x) \quad (2)$$



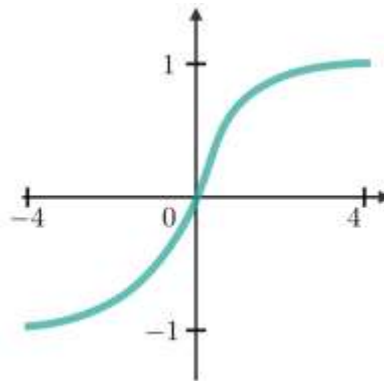
Şekil 3.15. ReLU fonksiyonu

ReLU'nun avantajı, çok hızlı hesaplanabilmesi ve sıfırların ortadan kaldırılmasıyla modelin daha derin katmanlarda öğrenme yapabilmesidir (Chollet, 2017).

3.5.1.3. Tanh (Hiperbolik Tanjen) fonksiyonu

Tanh fonksiyonu, sigmoid fonksiyonunun bir genellemesidir. Çıktıyı -1 ile 1 arasına sıkıştırır. Bu fonksiyon, özellikle veri setinde negatif değerlerin de önemli olduğu durumlarda tercih edilir.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$



Şekil 3.16. Tanh fonksiyonu

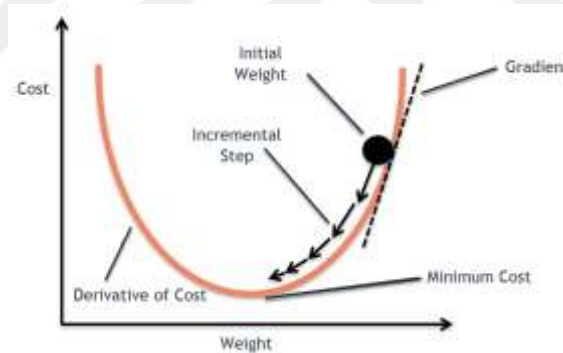
Tanh fonksiyonu, daha geniş bir aralıkta çıktı ürettiği için, sigmoid fonksiyonuna kıyasla öğrenme sürecinde daha iyi sonuçlar verebilir (LeCun et al., 2015).

3.5.2. Optimizasyon algoritmaları

Derin öğrenme modellerinin başarılı bir şekilde eğitilebilmesi için doğru optimizasyon algoritmalarının seçilmesi büyük önem taşır. Optimizasyon algoritmaları, modelin öğrenme sürecini yönlendirir ve modelin daha doğru tahminler yapabilmesi için ağırlıkları günceller. Bu algoritmalar, ağırlıkları güncellemek için gradyan iniş (gradient descent) ve türev tabanlı teknikler kullanır.

3.5.2.1. Gradyan iniş (Gradient Descent)

Derin öğrenme ve makine öğrenmesi alanlarında en yaygın kullanılan optimizasyon algoritmalarından biridir. Bu algoritma, her bir eğitim örneği için gradyan hesaplaması yaparak modelin ağırlıklarını günceller. Bu süreç, her bir iterasyonda kayıp fonksiyonunun türevini alarak, modelin ağırlıklarını negatif gradyana yönlendirerek, Şekil 3.13, ağırlıkları günceller.



Şekil 3.17. Gradient Descent algoritması

3.5.2.2. Adam optimizasyonu (Adam Optimization)

Adam algoritması, her parametre için adaptif öğrenme oranı kullanarak, modelin eğitilmesini daha verimli ve hızlı hale getiren bir optimizasyon yöntemidir. Bu algoritma, gradyanların ilk ve ikinci momentlerini (ortalama ve karelerin ortalaması) hesaba katarak ağırlık güncellemelerini gerçekleştirir.

3.5.3. Maliyet (Kayıp) fonksiyonları

Maliyet (Kayıp) fonksiyonları, derin öğrenme ve makine öğrenmesi modellerinin öğrenme süreçlerinin temel yapı taşlarından biridir. Bu fonksiyonlar, modelin tahmin ettiği çıktılar ile gerçek (beklenen) çıktılar arasındaki farkı ölçer. Bu fark, modelin ne kadar doğru çalıştığını belirler. Maliyet fonksiyonlarının amacı, bu farkı minimize etmektir, yani modelin tahminlerini mümkün olan en doğru hale getirmektir (Goodfellow et al., n.d.). Kayıp fonksiyonu, modelin eğitim verisi üzerinde ne kadar iyi performans gösterdiğini değerlendirmek için kullanılır ve genellikle "loss function" veya "error function" olarak adlandırılır (Bishop, 2006).

Kayıp fonksiyonları, çözülmek istenen problem türüne bağlı olarak farklılık gösterir. İki ana kategoriye ayrılabilirler (Chollet, 2017):

3.5.3.1. Regresyon problemleri için kayıp fonksiyonları

Ortalama Kare Hata (Mean Squared Error - MSE), MSE, tahmin edilen değer ile gerçek değer arasındaki farkların karelerinin ortalamasını alır. Bu, büyük hataların daha fazla cezalandırılmasına neden olur (Goodfellow et al., n.d.).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4)$$

Burada, y_i gerçek değer, \hat{y}_i tahmin edilen değerdir ve n örnek sayısını ifade eder.

Ortalama Mutlak Hata (Mean Absolute Error - MAE), MAE, tahmin edilen değer ile gerçek değer arasındaki farkların mutlak değerlerinin ortalamasını alır. MSE'ye göre daha az hassastır, çünkü büyük hataları küçük hatalarla aynı şekilde cezalandırır (Bishop, 2006).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5)$$

3.5.3.2. Sınıflandırma problemleri için kayıp fonksiyonları

Sınıflandırma problemlerinde, çıktı sınıf etiketleri gibi kategorik verilerdir. Bu tür problemlerde yaygın olarak kullanılan kayıp fonksiyonları şunlardır:

Çapraz Entropi Kayıp Fonksiyonu (Cross-Entropy Loss), Çapraz entropi, özellikle sınıflandırma problemlerinde kullanılır. Bu fonksiyon, gerçek sınıf etiketleri ile modelin tahmin ettiği olasılık dağılımı arasındaki farkı ölçer. Çapraz entropi kaybı, modelin doğru sınıfı seçme olasılığını artırarak, tahminlerin doğruluğunu maksimize etmeye çalışır (Goodfellow et al., n.d.).

$$H(p, q) = - \sum_{i=1}^c y_i \log(\hat{y}_i) \quad 6$$

Burada, y_i gerçek etiket, \hat{y}_i modelin tahmin ettiği olasılık ve C sınıf sayısını ifade eder.

Hinge Kayıp Fonksiyonu, Hinge kayıp fonksiyonu, özellikle doğrusal sınıflandırıcılar ve destek vektör makineleri (SVM) için kullanılır. Bu kayıp fonksiyonu, sınıfın doğru olup olmadığını belirlemek için kullanılan bir sınır oluşturur ve bu sınırın etrafında bir “marj” bırakır. Bu sayede, sınıflar arasındaki farkları optimize eder (Chollet, 2017).

$$\text{HingeLoss} = \max(0, 1 - y_i \cdot \hat{y}_i) \quad (7)$$

Burada, \hat{y}_i gerçek etiket, (1 veya -1) ve \hat{y}_i modelin tahmin ettiği değer (sınıf etiketinin doğruluğu) arasındaki farkı ölçer.

3.6. Veri Ön İşleme

Derin öğrenme ve makine öğrenimi çalışmalarında veri setlerini modellemeden önce uygulanan önemli bir adımdır. Bu süreç, veriyi analiz edilebilir ve modellerin öğrenebilmesi için uygun hale getirmeyi amaçlar. Ham veriler genellikle eksik, hatalı veya düzensiz olabilir, bu nedenle doğru ve temiz bir veri seti oluşturmak için çeşitli işlem

adımları gereklidir. Veri ön işleme, modelin doğruluğunu artırabilir, eğitim süresini kısaltabilir ve daha verimli sonuçlar elde edilmesini sağlayabilir.

3.6.1. Veri temizleme

Veri temizleme, verideki hatalı, eksik veya tutarsız verilerin düzeltilmesi sürecidir. Ham veriler genellikle yanlış etiketler, eksik değerler, tutarsız formatlar veya hatalı girişler içerebilir. Bu işlemlerle veri daha güvenilir ve doğru hale getirilir.

- **Eksik Veriler**, eğer veri setinde eksik değerler varsa, bu veriler ya ortalama, medyan veya mod ile doldurulabilir, ya da ilgili satırlar tamamen kaldırılabilir.
- **Hatalı Veriler**, verilerin hata içerip içermediği kontrol edilir. Yanlış etiketler veya yanlış formatlar düzeltilir.
- **Aykırı Değerler (Outliers)**, aykırı değerler tespit edilip, bunların model üzerindeki olumsuz etkileri giderilebilir.

3.6.2. Özellik seçimi

Özellik seçimi, modelin performansını iyileştirmek için veri setindeki gereksiz veya fazla ilişkili (redundant) özelliklerin kaldırılması işlemidir. Gereksiz özellikler, modelin karmaşıklığını artırarak overfitting (aşırı öğrenme) sorununa yol açabilir.

Özellik seçimi şu yollarla yapılabilir:

- **Korrelasyon Analizi**, özellikler arasındaki yüksek korelasyonlar belirlenir ve benzer bilgileri sağlayan özellikler kaldırılır.
- **Özelliklerin Önemi**, özelliklerin model için ne kadar önemli olduğuna karar verilerek en önemli özellikler seçilir.

3.6.3. Veri dönüşümü

Ham verilerin analiz ve modelleme süreçlerinde daha uygun hale getirilmesi için yapılan işlemler bütünüdür. Veri dönüşümü, verinin formatını veya yapısını değiştirerek, verinin daha anlamlı ve analiz edilebilir hale gelmesini sağlar. Bu işlem, modelin daha verimli çalışmasını sağlamak, daha iyi sonuçlar elde etmek ve veri setindeki doğruluğu artırmak için önemlidir.

Veri dönüşümü genellikle birkaç ana amaca hizmet eder:

- **Veriyi daha uygun hale getirmek**, modelin daha kolay öğrenilmesi için veriyi uygun bir formata dönüştürmek.
- **Verinin dağılımını düzleştirmek**, özellikle non-lineer veriler üzerinde daha iyi sonuçlar almak için veriyi daha simetrik bir hale getirmek.
- **Veri ilişkilerini daha iyi yansıtmak**, verinin anlamını kaybetmeden, bazı ilişkileri veya yapıları model için daha belirgin hale getirmek.

Veri dönüşüm işlemleri birkaç farklı şekilde yapılabilir. En yaygın veri dönüşümü teknikleri şunlardır:

3.6.3.1. Log dönüşümü

Log dönüşümü, genellikle verilerin long-tailed (uzun kuyruklu) dağılımlarını düzleştirmek için kullanılır. Birçok doğal ve finansal veri, bu tür uzun kuyruklu dağılımlar sergiler. Log dönüşümü, yüksek değerleri daha küçük bir aralığa çeker ve bu sayede modelin öğrenme sürecini daha verimli hale getirir.

Örneğin, gelir verisi gibi bazı veriler çok büyük sayılar içerebilir, ancak bu tür verilerin logaritması alındığında daha düzgün bir dağılım elde edilebilir.

$$y = \log(x) \quad (8)$$

Burada, gerçek etiket, (1 veya -1) ve \hat{y}_i modelin tahmin

3.6.3.2. Normalizasyon

Normalizasyon, genellikle verilerin belirli bir aralığa (örneğin, 0 ile 1 arasında) ölçeklenmesidir. Bu işlem, özellikle min-max scaling veya z-score standardizasyonu gibi yöntemlerle yapılır.

- **Min-Max Normalizasyonu**, verilerin 0 ile 1 arasına sıkıştırılması.

$$x_{scaled} = \frac{x - x_{min}}{x_{min} - x_{max}} \quad (9)$$

- **Z-Score Standardizasyonu**, verilerin ortalamasını sıfır ve standart sapmasını bir yapacak şekilde dönüştürülmesi.

$$x_{scaled} = \frac{x - x_{mean}}{x_{std}} \quad (10)$$

3.6.3.3. One-Hot Encoding

Kategorik verilerin model tarafından işlenebilmesi için sayısal verilere dönüştürülmesi gereklidir. Birçok model, sayısal verilerle çalışmak için optimize edilmiştir. One-Hot Encoding, kategorik verileri sayısal verilere dönüştürmenin yaygın bir yoludur.

Örneğin, bir "Renk" sütununda Kırmızı, Mavi, Yeşil değerleri varsa, her bir renk için ayrı bir sütun oluşturulur:

- Kırmızı → [1, 0, 0]
- Mavi → [0, 1, 0]
- Yeşil → [0, 0, 1]

3.6.3.4. Label Encoding

Kategorik verileri sayısal verilere dönüştüren bir araçtır. Her benzersiz kategoriye bir tamsayı değeri atar. Örneğin, ["Kırmızı", "Mavi", "Yeşil"] verisini [0, 1, 2] şeklinde dönüştürür.

3.6.3.5. Date/Time Transformation (Tarih/Zaman Dönüşümü)

Zaman serisi verileriyle çalışırken, tarih ve zaman özelliklerinin doğru bir şekilde dönüştürülmesi önemlidir. Tarihsel verilere dayalı özellikler genellikle ay, gün, hafta numarası, yıl gibi alt özelliklere dönüştürülerek daha anlamlı hale getirilir.

Örneğin, 2024-12-14 12:06:53 tarih verisini unix zaman damgası (Unix timestamp) formatına çevirerek 1734167213 şeklinde kullanılabilir.

3.6.4. Veri bölme

Veri bölme, veriyi eğitim (training), doğrulama (validation) ve test (test) setlerine ayırma işlemidir. Bu, modelin eğitim sırasında öğrendiği bilgileri, modelin gerçek dünya verilerine ne kadar iyi genellebileceğini ölçmek için gereklidir.

Veri bölme genellikle şu şekilde yapılır:

- **Eğitim Seti (Training Set)**, modelin öğrenme işlemi burada yapılır.
- **Doğrulama Seti (Validation Set)**, modelin hiperparametreleri ve diğer ayarları doğrulama için kullanılır.
- **Test Seti (Test Set)**, modelin performansı nihai test için bu veri ile ölçülür.

3.7. Literatürde Yapılan Çalışmalar

Son yıllarda derin öğrenme yaklaşımları, çeşitli karmaşık veri analizleri ve tahmin modellerinde önemli ilerlemeler sağlamıştır. Bu kapsamda, hava taşımacılığı gibi dinamik ve kritik bir alanda uçuş verilerinin analizi ve tahmini, büyük bir araştırma

konusu haline gelmiştir. Uçuş gecikmelerinin ve rotalarının öngörülmesi, havacılık sektörünün operasyonel verimliliğini artırmak, müşteri memnuniyetini iyileştirmek ve ekonomik kayıpları azaltmak için kritik bir rol oynamaktadır.

Gecikme ve rota tahmini problemleri, geleneksel istatistiksel yöntemlerden başlayarak makine öğrenimi ve derin öğrenme yöntemlerine evrilmiştir. Erken dönem çalışmalarda genellikle karar ağacı algoritmaları, Naive Bayes ve regresyon modelleri gibi yöntemler kullanılmış, ancak bu yaklaşımlar zaman serisi verilerdeki karmaşık bağımlılıkları yakalamakta sınırlı kalmıştır. Derin öğrenme, özellikle RNN, LSTM ve CNN gibi mimariler sayesinde bu eksikliklerin üstesinden gelmiştir.

Bu bağlamda yapılan çalışmalarda, CNN ve LSTM modelleri, uçuş verilerindeki zamansal bağımlılıkları öğrenmekte güçlü bir performans sergilemiştir. Örneğin, LSTM tabanlı yaklaşımlar, uçuş gecikmelerinin tahmininde doğruluğu %90'ın üzerine çıkarırken, karma modeller (ör. CNN-LSTM) hem zamansal hem de mekânsal özellikleri öğrenerek daha etkili sonuçlar sunmuştur. Bunun yanı sıra, hava durumu verilerinin modele entegrasyonu gibi ek faktörlerin kullanımı, tahmin performansını artırmada önemli bir etki yaratmıştır.

Uçuş gecikmelerinin derin öğrenme yöntemleri ile tahmini üzerine Y. J. Kim, S. Choi, S. Briceno, ve D. Mavris tarafından yapılan çalışmada kullanılan yöntemler, veri kaynakları, deney sonuçları ve gözlemleri ele aldığımızda, uçuş gecikmesi tahmini için günlük gecikme durumlarını analiz etmiş ve bireysel uçuş gecikmelerini daha kesin şekilde öngörmek için kullanılmıştır. Çalışmada ABD Ulaştırma Bakanlığı'nın Transtats veri tabanı kullanılarak Ocak 2010 ile Ağustos 2015 tarihleri arasında 10 büyük havaalanından elde edilen günlük Çizelge 3.2'deki veri setine dayanmıştır.

Çizelge 3.2. Uçuş gecikmesi modelinin girdileri ve çıktıları

Girdi-Çıktı Modeli	Veri Özelliği
Uçuş Verisi Özellikleri (Girdi)	Haftanın günü, Mevsim, Ay, Tarih
	Kalkış havaalanı, Varış havaalanı
	Planlanan kalkış - varış saati
	Kalkış havaalanının gecikme durumu
	Varış havalimanının gecikme durumu

Hava Durumu (Girdi)	Rüzgar yönü, Rüzgar hızı, Bulut yüksekliği, Görüş mesafesi, Yağış, Kar birikimi, Şiddet, Tanımlayıcı, Gözlem Kodu
Sınıflandırma (Çıktı)	Farklı eşik değerlerine sahip gecikme sınıfı (15 dakika, 30 dakika)

Günlük gecikmeler, gecikme eşik değerlerine göre sınıflandırılmıştır (örneğin 15 dakika ve 30 dakika). RNN ve LSTM modelleri farklı gün sekans uzunlukları ve derinlik seviyeleri (örneğin 7 gün ve 9 gün) ile test edilmiştir. Daha uzun sekans uzunlukları (ör. 9 gün) ve daha büyük eşik değerleri (30 dk), daha yüksek doğruluk elde edildiği gözlemlenmiştir. Daha fazla katman sayısı ve epoch değerleri uygulandığında %87.42 doğruluk değeri ile performansı artışı görülmüştür. Ayrıca yüksek uçuş trafiği verisine sahip Çizelge 3.3'deki havaalanlarında (ör. ATL ve LAS), modelin performansı daha yüksektir. Makalede yapılan çalışma ile LSTM ve derin RNN mimarilerinin uçuş gecikmesi tahmini için etkili olduğunu göstermiştir. Derin öğrenme, geleneksel yöntemlere kıyasla daha yüksek doğruluk oranları sunmakta ve özellikle büyük veri setlerinde daha iyi performans göstermektedir. Gelecekte, daha fazla veri ve farklı mimarilerle (ör. CNN) daha da iyi sonuçlar alınabileceği belirtilmiştir.

Çizelge 3.3. Farklı havalimanları için model doğruluğu

Havalimanı	Doğruluk Yüzdesi
Atlanta (ATL)	90.95
Los Angeles (LAX)	86.96
Chicago (ORD)	85.61
Dallas (DFW)	89.31
Denver (DEN)	89.82
New York (JFK)	86.51
San Francisco (SFO)	87.52
Charlotte (CLT)	91.80
Las Vegas (LAS)	91.81
Phoenix (PHX)	71.34

Yapılan bir başka çalışmada uçuş gecikmelerini tahmin etmek için ADS-B tabanlı büyük veri ve makine öğrenimi yöntemlerini birleştirerek kapsamlı bir yaklaşım sunmaktadır. Gui G., Liu F., Sun J., Yang J., Zhou Z. ve Zhao D'nin yapmış olduğu araştırmada, uçuş gecikmesini etkileyen geniş bir faktör yelpazesi ele alınmıştır. Veriler, ADS-B mesajları, hava durumu bilgileri, uçuş programları ve havaalanı trafiği gibi kaynaklardan toplanmış ve birleştirilmiştir. Bu entegre veri setini kullanılarak, gecikme

sürelerine göre Çizelge 3.4 ve Çizelge 3.5'deki gibi kategorilere ayırarak analiz etmeyi hedeflemiştir.

Çizelge 3.4. İkili sınıflandırma görevi (Binary Classification) (Gui et al., 2020)

Sınıf	Gecikme
Sınıf 0	Gecikme yok (uçuş planlanan zamanda veya 15 dk'dan az bir gecikme ile kalkış/yapılmış).
Sınıf 1	Gecikme var (uçuş 15 dk'dan fazla bir süreyle gecikmiş).

Çizelge 3.5. Dört kategoriye ayrılan sınıflandırma görevi (Four Categories Classification) (Gui et al., 2020)

Sınıf	Gecikme
Sınıf 0	Gecikme yok (0-15 dakika)
Sınıf 1	Bir saatten az gecikme (15-60 dk.)
Sınıf 2	Bir ila iki saat arası gecikme (60-120 dk)
Sınıf 3	İki saatin üzerinde gecikme (120+ dk)

Uçuş gecikmesi tahmini için LSTM ve Random Forest modelleri uygulanmıştır. Ayrıca, bu iki model sınıflandırma ve regresyon görevlerinde karşılaştırılmıştır. Çalışmada Aralık 2018 ile Mayıs 2019 arasında toplanan 5.761 öğeden oluşan bir veri seti kullanılmış olup 3.368 uçuşta gecikme olmamıştır. Random Forest tabanlı model, genelleştirme kabiliyeti sayesinde daha dengeli sonuçlar sunmuştur. İkili sınıflandırma görevinde %90.2 doğruluk oranına ulaşmıştır. Model, farklı kategorilerdeki uçuş gecikmelerini (ör. bir saatten az, bir-iki saat arası, iki saatin üzerinde) tahmin edebilmiştir, ancak dört kategoriye ayrılan sınıflandırma görevinde doğruluk oranı %70'e düşmüştür. Bu durum, daha karmaşık kategorilerin sınırlı veri ile modellenmesinin zorluğunu göstermektedir. Diğer taraftan LSTM tabanlı modelin zaman serisi verilerindeki güçlü performansı bu çalışmada vurgulanmaktadır. LSTM mimarisi, geçmiş verilere dayanarak gelecekteki gecikme durumlarını tahmin edebilme kapasitesine sahiptir. Ancak, sınırlı veri setleri nedeniyle bu modelde aşırı öğrenme (overfitting) problemi gözlemlenmiştir. Eğitim sırasında doğruluk oranı %99'a ulaşırken, test verilerinde doğruluk oranı belirgin şekilde düşmüştür. Bunun nedeni, modelin genelleştirme kabiliyetindeki sınırlamalardır. Dropout katmanı gibi düzenleme yöntemleriyle aşırı öğrenme sorunu bir miktar hafifletilmiştir, ancak tamamen ortadan kaldırılamamıştır. Bu sonuçlar bize gelecek çalışmalar için daha geniş veri setlerinin toplanması ve daha karmaşık model mimarilerinin tasarlanması önerilmiştir.

Uçuş rotalarının tahmin edilmesi, havacılık güvenliği ve operasyonel verimliliği artırmak için kritik öneme sahiptir. Bu bağlamda, makine öğrenimi ve derin öğrenme yöntemleri, uçakların gelecekteki hareketlerini yüksek doğrulukla tahmin edebilmek için etkili araçlar sunmaktadır. Özellikle CNN ve LSTM gibi modellerin birleştirilmesiyle oluşturulan hibrit yapılar, uçuş rotası tahmini gibi zamansal ve mekânsal özellikler içeren karmaşık problemlerin çözümünde öne çıkmaktadır. Ma L, ve Tian S.'nin yapmış olduğu çalışmada, ADS-B verileri gibi Çizelge 3.6'daki gerçek uçuş rotası bilgileri kullanılarak, uçakların enlem, boylam, yükseklik ve zaman gibi dört boyutlu (4D) hareketleri tahmin edilmektedir.

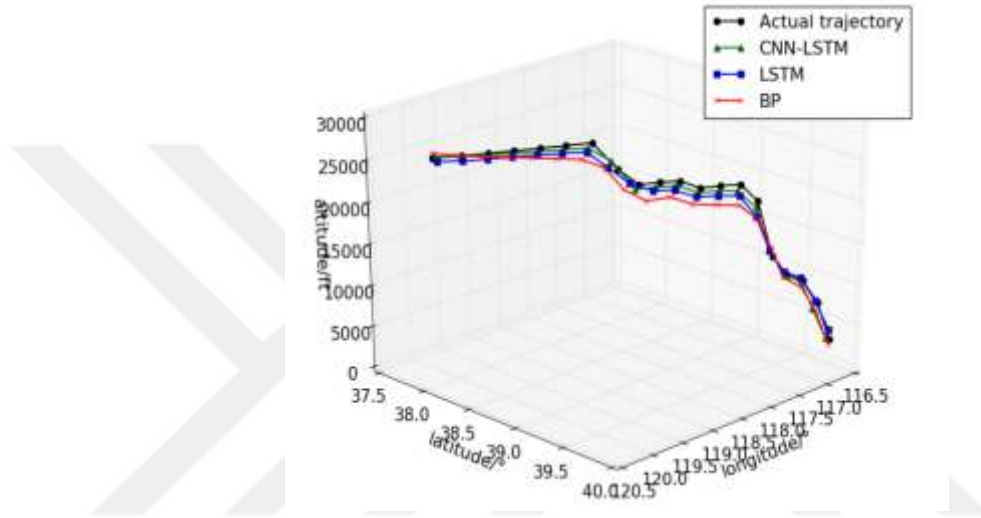
Çizelge 3.6. Yörünge noktasına ait özellikleri (Ma & Tian, 2020)

Parametre	Yörünge Noktası
Zaman	2017-05-06 07:35:10
ICAO Adres Kodu	780E59
Uçuş Numarası	XXX
Enlem /°	120.2188
Boylam /°	37.6415
Yükseklik /feet	25600
Hız /knot	398.399
Hücum Açısı /derece	49.071

Deneysel çalışmalara başlamadan önce verilerdeki kayıp veya tekrar eden noktalardan arındırılarak ve normalleştirme işlemleri uygulanarak modele giriş için hazırlanmıştır. Tahmin işlemleri, 6 ardışık zaman dilimi (zaman, enlem, boylam, yükseklik, hız ve yön) kullanılarak gelecekteki bir noktanın tahmin edilmeye çalışılmıştır.

Çalışma içerisinde BP(*Backpropagation Neural Network* - Geri Yayılım Sinir Ağı) temel öğrenme algoritmasıdır. Bu algoritmada, modelin tahmin ettiği sonuç ile gerçek sonuç arasındaki hatayı (loss) geriye doğru yayarak ağırlıkları günceller ve böylece modelin öğrenmesini sağlar. BP genellikle gradyan inişi (gradient descent) gibi optimizasyon yöntemleriyle birlikte çalışır. Buradaki model, tek adımlı (single-step) ve çok adımlı (multi-step) tahminler için tasarlanmıştır. Tek adımlı tahminde (bir sonraki zaman diliminde) CNN-LSTM modeli LSTM'ye göre %29.06, BP'ye göre ise %59.72 daha düşük hata oranına sahiptir. Üç adımlı tahminde (t+1, t+2 ve t+3 zaman dilimlerinde), CNN-LSTM'nin hata oranı LSTM'ye göre %24.31, BP'ye göre %50.42 daha azdır. Son olarak beş adımlı (t+1'den t+5'e kadar olan zaman dilimi) tahminde ise CNN-LSTM, LSTM'ye göre %11.50, BP'ye göre %47.20 daha düşük hata oranı

göstermiştir. Modelin tahmin performansı Kök Ortalama Kare Hatası (RMSE), Ortalama Mutlak Hata (MAE) ve Ortalama Mutlak Yüzde Hatası (MAPE) ölçütüyle değerlendirilmiştir. CNN-LSTM modelinin tüm ölçütlerde en düşük hata oranlarına sahip olduğunu göstermektedir. Modelin MAPE'ye göre doğruluk oranı %91 ile %95 arasında değişmektedir. CNN-LSTM modelinin mekânsal (longitude, latitude) ve zamansal (altitude, time) verilerdeki güçlü performansını gerçek verilerle olan korelasyonu açıkça Şekil 3.18'deki üç boyutlu yörünge tahmin grafiğinde de gösterilmiştir.



Şekil 3.18. Üç boyutlu yörünge tahmini (Ma & Tian, 2020)

Literatürdeki bu çalışmalar, hem derin öğrenme yöntemlerinin gücünü hem de uçuş verilerinin analizi ve tahmininde kullanılabilecek yenilikçi yaklaşımları ortaya koymaktadır. Tez çalışmamızda, mevcut yöntemlerin bir adım ötesine geçilerek, uçuş rotası tahmini ve gecikme türlerinin öngörülmesi için entegre derin öğrenme modellerinin geliştirilmesi hedeflenmektedir.

4. DENEYSEL ÇALIŞMALAR

Bu bölümde, tez kapsamında gerçekleştirdiğimiz deneysel çalışmaların ayrıntılı adımları sunulmaktadır. Öncelikle veri kaynakları ve özellikleri tanıtılacak, ardından veri ön işleme aşamaları, model kurulumları, eğitim süreçleri ve performans değerlendirmesi için kullanılan metrikler açıklanacaktır. Son olarak, elde edilen deney sonuçları yorumlanacak ve karşılaştırmalar yapılacaktır.

4.1. Deney Ortamı ve Altyapı

Bu çalışmadaki deneysel süreçler, yüksek hesaplama gücü gerektiren modellerin eğitimi ve tahmin işlemlerini kolaylaştırmak amacıyla Google Colab platformu üzerinden ve Victus HP Laptop Windows 11 Pro 13th Gen Intel(R) Core(TM) i7-13700H 2.40 GHz 16 GB RAM NVIDIA GeForce RTX 4050 Laptop GPU 6 Gbytes ile yürütülmüştür. Google Colab, araştırmacılara ücretsiz GPU (NVIDIA Tesla T4 veya P100) erişimi sağlayarak derin öğrenme modellerinin eğitim sürelerini önemli ölçüde kısaltmaktadır.

Çalışma sırasında Python 3.9 ve TensorFlow/Keras 2.x sürümleri, scikit-learn, pandas, numpy ve matplotlib gibi kütüphaneler kullanılmıştır. Bu ortam, gerekli tüm bağımlılıkların kolaylıkla kurulmasına, modellerin hızlı bir şekilde test edilmesine ve sonuçların tekrarlanabilir bir biçimde sunulmasına olanak tanımıştır.

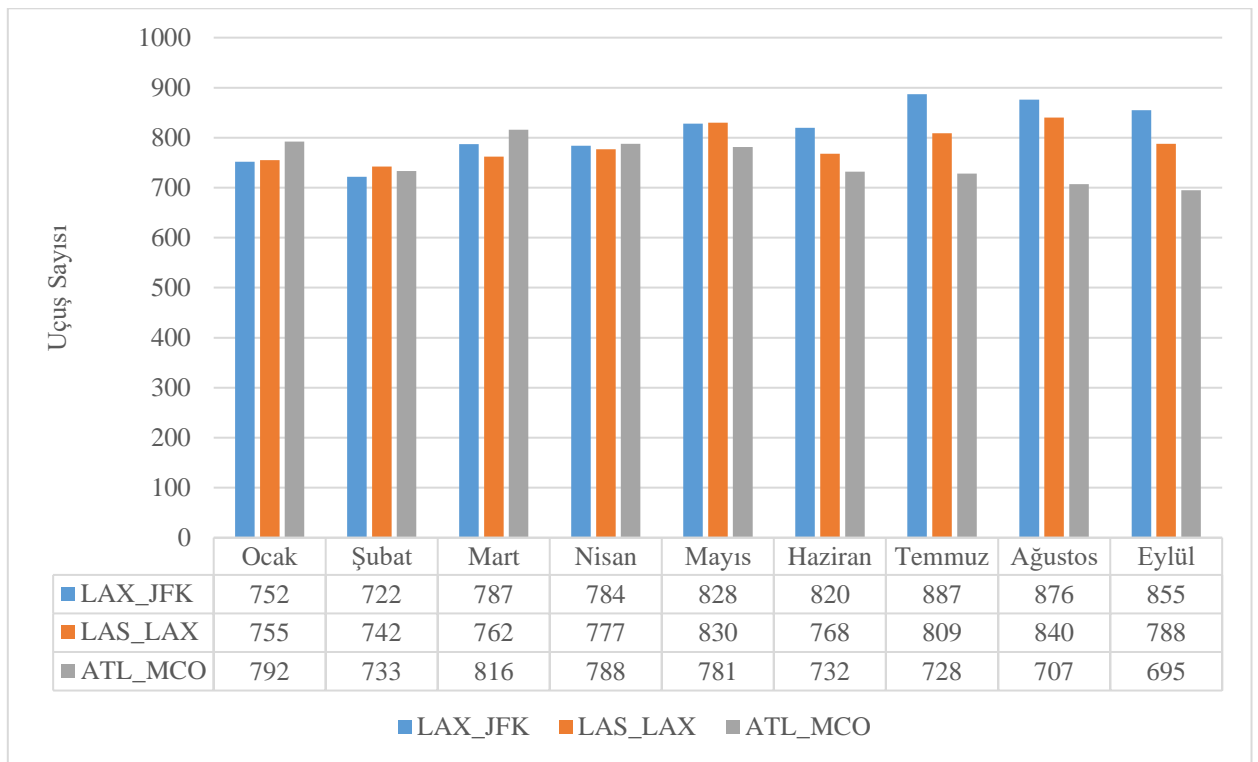
Bilgisayar altyapısının detayı, çalışmanın herhangi bir yerel donanım sınırlaması olmaksızın gerçekleştirilmesini sağlamış, bu sayede model mimarileri ve hiperparametre deneyleri hızlı ve pratik bir biçimde yürütülebilmiştir. Bu koşullar altında elde edilen sonuçlar, benzer donanım veya Google Colab gibi bulut tabanlı platformlar kullanılarak tekrar elde edilebilir özelliktedir.

4.2. Veri Seti ve Hazırlık Süreci

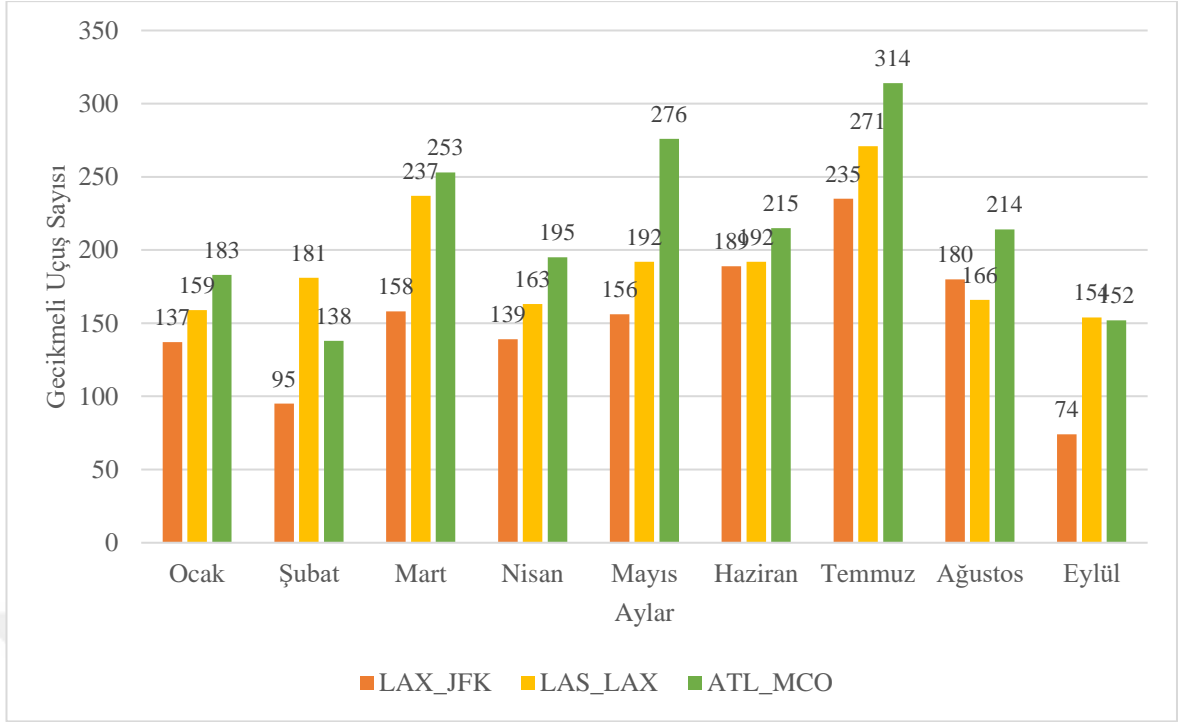
4.2.1. Uçuş gecikme tahmini için kullanılan veriler

Deneysel çalışmalar için ABD Ulaştırma Bakanlığı (USDOT) On-Time Performance veri seti kullanılmıştır. Bu veri seti, belirli bir dönem (Ocak-Eylül 2024)

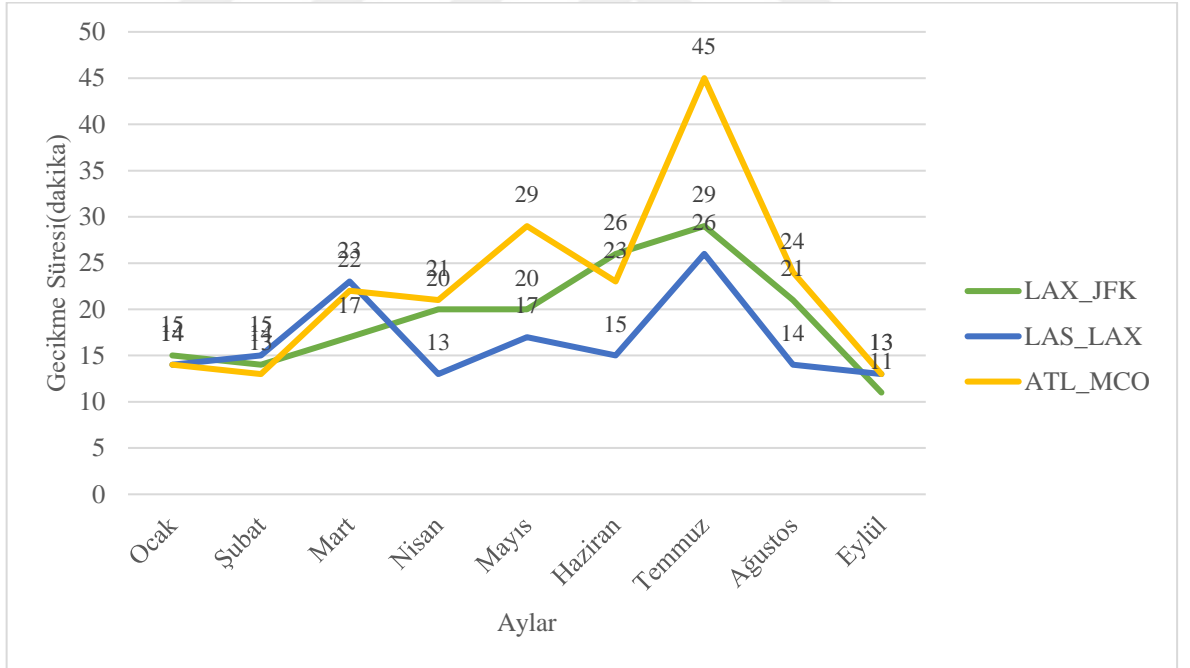
boyunca LAX – JFK, LAS-LAX ve ATL-MCO rotalarında gerçekleştirilen uçuşlara ait kalkış, varış zamanları, gecikme bilgileri ve havayolu gibi bilgileri içermektedir. Ham veriler farklı aylara ait CSV dosyalarından elde edilmiştir. Ele alınan rotalarda aylık bazda gerçekleşen uçuş sayıları Şekil 4.1’de gösterilmiştir. Yine bu rotalarda ay bazında gerçekleşen gecikmeli uçuşların sayısı Şekil 4.2’de gösterilmiştir. Analiz çalışmasının yapıldığı aylar içerisindeki rotalar boyunca yapılan uçuşlarda ortalama gecikme süreleri Şekil 4.3’te incelenmiştir. Üç rotada da en fazla uçuş gecikmesi Temmuz ayında gerçekleştiği gözlemlenmiştir. Gecikmelerin en fazla olduğu bu ay için rota tahmin çalışması yapılmıştır.



Şekil 4.1. Ay bazında gerçekleşen uçuş sayısı



Şekil 4.2. Ay bazında gerçekleşen gecikmeli uçuş sayısı



Şekil 4.3. Gerçekleşen uçuşların varış zamanlarına göre gecikme süreleri

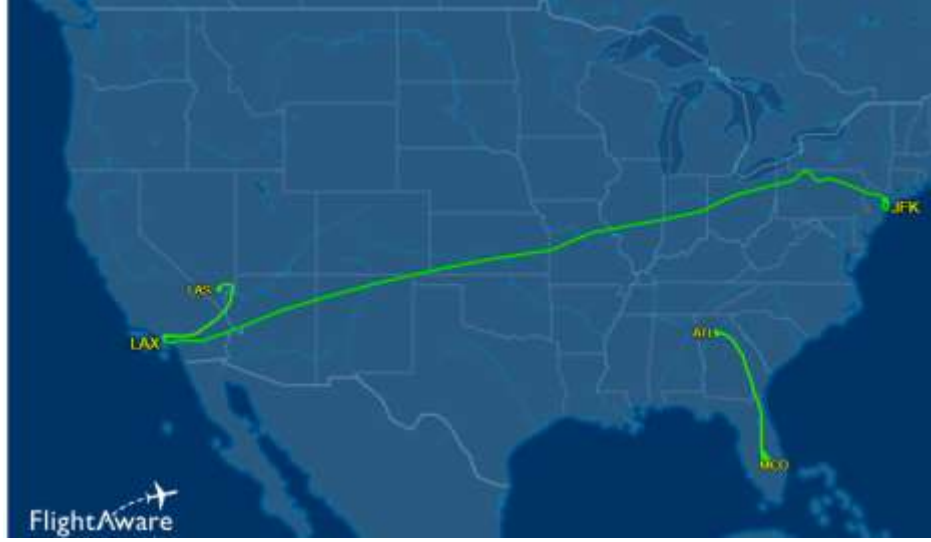
4.2.2. Uçuş rota tahmini için kullanılan veriler

Uçuş rota tahmini için gerekli olan veriler, ADS-B tabanlı uçuş bilgisi sağlayan çevrimiçi kaynaklardan elde edilmiştir. Bu çalışma kapsamında, gerçek ve geçmiş zamanlı uçuş izlerini dijital havacılık verileriyle sunan FlightAware'den (<https://www.flightaware.com>) elde edilmiştir. Uçuş ortalamasının ve ortalama gecikme süresinin en fazla olduğu dönem bakımından Temmuz 2024 içerisinde gerçekleşen uçuşlar ele alınmıştır. LAX-JFK, LAS-LAX ve ATL-MCO rotalarında Delta Airlines, American Airlines, Southwest Airlines ve United Airlines ile gerçekleşmiş uçuşların uçuş takip kayıtları veriseti olarak kullanılmıştır. Herbir rota için yaklaşık 12000 konum noktası verisi işlenmiştir.

Çizelge 4.1. ADS-B verisi

Parametre	Yörünge Noktası
Timehex	1731256930
Enlem (°)	33.7384
Boylam (°)	-116.6324
İrtifa (feet)	30.075
Hız (knot)	468
Yön (°)	90

Toplanan ADS-B verileri, ilgili uçuşun kalkıştan varışa kadar olan zaman diliminde düzenli aralıklarla (30 saniyelik aralıklar) alınan konum noktalarından oluşmaktadır. Her bir uçuş için elde edilen veri, Çizelge 4.1'de örneği verilen zaman serisi biçiminde enlem, boylam, irtifa, hız, yön gibi özellikler içerir. Bu zaman serisi verisi, uçuşun planlı rotasından fiilen nasıl saptığını anlamak ve gelecekteki konum bilgisini tahmin etmek üzere kullanılmıştır.



Şekil 4.5. LAX-JFK, LAS-LAX ve ATL-MCO rotalarında gerçekleştirilen uçuş grafiği

4.3. Veri Ön İşleme Adımları

Uçuş gecikme verileri ve ADS-B verileri ham verilerden okunarak zaman serisi bilgisine göre kronolojik sıraya göre okunur. Okunan değerlerden sütunlarda eksik değerlerin bulunduğu yerler incelenir. Burada gecikme değerleri (ARR_DELAY, DEP_DELAY) eksik olan satırlar çıkarıldı. Rota verilerinde eksik konum bilgilerinin (enlem, boylam) olduğu satırlar uçuş izindeki bu noktalar lineer enterpolasyon veya en yakın zamandaki geçerli konumun kullanımıyla dolduruldu. Ancak eksik verinin fazla olması durumunda ilgili uçuş analizden çıkarılmıştır. Gecikme değerleri çok büyük pozitif değerler (örneğin 100 dakikayı aşan gecikmeler) veya negatif değerlerin olduğu uçuşlar değerlendirmeye alınmamıştır. Rota verilerinde de coğrafi konumlar mantıksız noktalar (örneğin aniden çok uzakta bir konum) ise yine değerlendirmeye alınmamıştır. Gecikme tahmini için kullanılan verilerde havayolu (OP_CARRIER), havaalanı (ORIGIN, DEST) gibi kategorik özellikler Label Encoding yöntemiyle sayısal forma dönüştürülmüştür. Kullanılan her iki modelde (CNN, LSTM) veriler 0-1 aralığına (StandardScaler) ölçeklenmesi ile veriler eğitim süreci için istikrarlı hale getirilmiştir. Rota tahmini için her uçuşun verisi bir zaman serisi olarak ele alınır. Kullandığımız LSTM modeli gereği, sabit uzunlukta giriş penceresi (örneğin 10 zaman adımı) seçilerek her uçuş için $[t-(10-1) \dots t]$ aralığındaki gözlemler giriş, $t+1$ zaman adımındaki konum hedef olarak tanımlanmıştır. Bu işlemi tüm uçuşlar için yaparak model giriş-çıkış çiftleri elde edilmiştir. Genel kural verinin %80'i eğitim, %20'si test için ayrılmıştır. Rota

tahmini için zaman serisi yapısı olduğu için, geleceğe dair tahmin yapılması gerektiğinden veri kronolojik olarak eğitim ve test setine bölünmüştür. Son olarak temizlenmiş, encode edilmiş ve ölçeklenmiş veriler son bir kez inceleyerek ortalama, standart sapma, minimum ve maksimum değerleri kontrol edilmiştir.

4.4. Model Kurulumu ve Hiperparametreler

Bu bölümde uçuş gecikmesi ve rota tahmini için kullanılan modellerin seçimi, yapılandırılması ve hiperparametre optimizasyonu süreçleri açıklanmaktadır. Seçilen modellerin her biri, incelenen problem tipine ve verinin yapısına uygun olacak şekilde tasarlanmış, performansı artırmak amacıyla çeşitli hiperparametre denemeleri yapılmıştır.

Uçuş gecikmesi tahmini için CNN metodunun seçilmesinin temel nedeni, CNN'in verilerdeki yerel örüntüleri ve ilişkileri etkili bir şekilde öğrenme yeteneğidir. Özellikle, zaman serisi verilerindeki ardışık ilişkileri analiz edebilmesi ve birden fazla özelliğin olduğu uçuşların ayı, ayın günü, havayolu kodu, haftanın günü, kalkış ve varış havaalanları ile planlanan kalkış saatini içeren özelliklerden etkileşimlerini yakalayabilmesi, bu yöntemi uçuş gecikmesi gibi çok boyutlu bir problem için ideal hale getirmektedir.

CNN, uçuş verilerindeki karmaşık paternleri otomatik olarak öğrenerek, manuel özellik mühendisliği ihtiyacını azaltır. Ayrıca, CNN'in havacılık gibi operasyonel veri setlerinde büyük veri kümeleriyle çalışabilme kapasitesi, modeli hem ölçeklenebilir hem de güvenilir bir tahmin aracı haline getirir. Özellikle MaxPooling ve Dropout gibi düzenleme teknikleriyle birleştirildiğinde, CNN hem doğruluk hem de genelleme yeteneği açısından güçlü bir performans sergileyebilir.

Öncelikle, kategorik özellikler olan havayolu kodu, kalkış ve varış havaalanları Label Encoding yöntemiyle sayısal değerlere dönüştürülmüştür. Bu işlem, modelin kategorik verileri işleyebilmesini sağlamıştır. Ardından, tüm özellikler Min-Max Ölçeklendirme ile 0 ile 1 aralığına normalize edilmiştir. Normalizasyon, modelin daha hızlı ve kararlı bir şekilde öğrenmesine yardımcı olurken, özellikler arasındaki ölçek farklılıklarını ortadan kaldırmıştır. Modelin zaman serisi verilerini öğrenebilmesi için veriler 40 zaman adımı içeren dizilere bölünmüştür. Gizli katmanlarda öğrenmeyi

hızlandırarak, gradyan kaybolma sorununu azaltarak ve doğrusal olmayanlık kazandırarak modelin karmaşık ilişkileri öğrenebilmesi için aktivasyon fonksiyonu ReLU seçildi. Konvolüsyon, MaxPooling ve Dropout katmanları uygulanarak modelin performansı ve genelleme kabiliyeti desteklenmiştir. Konvolüsyon ve havuzlama işlemlerinin ardından veriler, Flatten katmanı ile düzleştirilerek tam bağlantılı katmanlara uygun hale getirilmiştir. 256 nöronlu Dense katmanı, modelin daha karmaşık ilişkileri öğrenmesine olanak tanırken, ReLU aktivasyon fonksiyonu kullanılarak doğrusal olmayan ilişkilerin modellenmesi sürdürülmüştür. Son olarak, linear aktivasyonlu tek bir nörondan oluşan çıkış katmanı, uçuş gecikme süresini tahmin etmektedir. Model, Adam optimizasyon algoritması kullanılarak derlenmiştir. Adam, adaptif öğrenme oranlarıyla modelin hızlı ve etkin bir şekilde öğrenmesini sağlamaktadır. Kayıp fonksiyonu olarak Ortalama Kare Hatası (MSE) seçilmiştir; bu, tahmin edilen değerlerle gerçek değerler arasındaki farkların karesinin ortalamasını alarak modelin hatasını minimize etmeye yardımcı olur. Ayrıca, Ortalama Mutlak Hata (MAE) metriği de modelin performansını değerlendirmek için kullanılmıştır.

Rota tahmininde LSTM mimarimizin girdi aşamasında veri olarak her örnek son 20 zaman adımına ($time_step=20$) ait unixtime, irtifa ve yer hızı özelliklerini içerir. LSTM modeli yapısı gereği oluşturduğumuz diziler 3 boyutlu [örnek sayısı, zaman adımı, özellik sayısı] tensör modeller oluşturulmuştur. Çıktı verisi olarak, model her bir girdi penceresi için enlem ve boylam değerlerini tahmin etmek üzere tasarlandı. Dolayısıyla çıkış katmanında 2 nöron bulunmaktadır (1 nöron enlem, 1 nöron boylam için). Modelimize tek bir LSTM katmanı eklendi ve bu katman 20 gizli birime (nöron) olarak ayarlanmıştır. Nöron sayısı arttıkça modelin zaman serisi paternlerini öğrenme kapasitesi artar, ancak hesaplama maliyeti de yükseleceğini belirtebiliriz. Ayrıca (overfitting) olma durumu da muhtemeldir. Daha az nöron, daha basit bir model demektir. Bu seçim deneme-yanılma veya hiperparametre optimizasyon teknikleriyle oluşturulmuştur. LSTM katmanının çıktı vektörü yani Dense katmanı 2 boyutlu bir çıktı elde etmektedir. Böylece 2 çıktılı dense katmanı, modelin tek bir zaman adımı sonrasındaki enlem ve boylam değerlerini tahmin etmesini sağlar.

4.5. Eğitim Süreci

Derin öğrenme modellerini bir probleme uyarlarken, en kritik aşamalardan biri eğitim (training) sürecidir. Bu süreçte, model belirli bir veri kümesinden (eğitim seti) yararlanarak hedef değişkeni en iyi şekilde tahmin etmeyi, sınıflandırmayı veya bir başka görevi yerine getirmeyi öğrenir. Eğitim işlemi, modelin başlangıçta rastgele seçilen ağırlıklarının, hatayı minimize edecek yönde sistematik olarak güncellenmesini içerir. Bu güncellemeler sonucunda model, veri kümesindeki örüntüleri (patterns) ve ilişkileri kavramaya başlar. Eğitim sürecinin amacı, modeli yalnızca eğitim verisini ezberletmek değil, aynı zamanda yeni, daha önce görülmemiş veriler üzerinde de yüksek performans gösterebilecek bir genelleme yeteneği kazandırmaktır.

Uçuş gecikmesi tahmininde kullandığımız CNN modelinin eğitimi sırasında kayıp fonksiyonu (loss function) olarak Ortalama Kare Hatası (MSE) kullanıldı. MSE ile tahmin edilen değer ile gerçek değer arasındaki farkın karesini alarak büyük hataların modele daha fazla ceza vermesini sağlandı. Bu sayede modelin tahminlerinin hedef değişkene daha yakın olması amaçlanmış oldu. Optimizasyon sürecinde Adam algoritması ile adaptif öğrenme sağlayarak model parametreleri daha hızlı ve kararlı bir şekilde güncellenmesi sağlandı. Model, tam eğitim verisi kümesini 100 kez dolaşarak (epoch sayısı), parametrelerini defalarca günceller. Yüksek sayıda epoch, modelin veriden daha fazla öğrenmesine olanak tanır, ancak aşırı öğrenme (overfitting) riski de taşır. Her güncelleme adımında (backpropagation), sadece 2 örnek üzerinden (batch_size=2) modelin ağırlıkları güncellendi. Küçük batch boyutu ile modelin sık sık ağırlık güncellemesi yapmasına olanak tanır, ancak eğitim süresini uzatabilir. Çoğu durumda 16, 32 veya 64 gibi daha büyük batch boyutları tercih edilir; burada seçilen 2 değerinin seçmemizin amacı modelin seçtiğimiz veri kümesinde hızlı güncellenmesini sağlamaktır.

LSTM modeli ile çalıştığımız yapıda da kayıp fonksiyonu MSE seçilmiştir. Adam optimizasyon algoritması ile eğitim aşaması oluşturulmuştur. Epoch sayısı 100 seçilerek hızlı güncelleme amaçlanmıştır.

4.6. Performans Ölçütleri

Modellerin eğitimi ve doğrulanması sürecinde ulaşılan son aşamadır. Elde edilen sonuçların geçerli ve güvenilir olduğunun gösterilmesi amacıyla performansın nicel olarak değerlendirilmesidir. Bu aşama, modelin yalnızca eğitim verisine uyum sağlamakla kalmayıp, yeni ve görülmemiş veri üzerinde de iyi bir genelleme kapasitesine sahip olup olmadığının ortaya konması açısından kritik önem taşır. Model sonuçları farklı modeller, özellik setleri veya hiperparametre ayarları ile karşılaştırılabilir ve hangi yaklaşımın en iyi sonucu verdiği objektif olarak belirlenebilir.

Bu çalışmada hem uçuş gecikmesi tahmini (CNN modeli) hem de uçuş rotası tahmini (LSTM modeli) için performans ölçütü olarak Ortalama Kare Hatası (Mean Squared Error , MSE), Ortalama Mutlak Hata (Mean Absolute Error, MAE) ve Kök Ortalama Kare(Root Mean Squared Error, RMSE) kullanılmıştır. MAE, tahmin edilen değerler ile gerçek değerler arasındaki farkların mutlak değerlerinin ortalamasını alarak elde edilir.

5. BULGULAR

Bu çalışmada uçuş gecikmesi ve rota tahminlerini ayrı modellerle analiz ederek iki farklı boyutta havacılık operasyon bilgisine elde ettik. Tez çalışması kapsamında veri setleri CNN ve LSTM algoritmaları ile eğitilmiştir. Bu sayede, uçuşların zaman boyutundaki performans sorunlarını (gecikme) ve mekânsal yöndeki varyasyonlarını (rota sapması) inceleyerek, karar vericilerin operasyonel planlamaları daha sağlıklı yapabilmeleri amaçlanmıştır.

Eğitim süreçlerinde, uygun veri ön işleme adımları, özellik seçimi ve hiperparametre ayarlamaları gerçekleştirilmiş, farklı dönemlerden elde edilen verilerle çalışarak modellerin genelleme yetenekleri test edilmiştir. Gecikme tahmini tarafında, uzun dönem verileri (Ocak-Eylül 2024), değişen mevsimsel koşullar ve operasyonel yoğunlukların modele tanıtılmasına imkân vermiştir. Rota tahmini tarafında ise, belirli bir döneme (Temmuz 2024) odaklanılarak mevsim geçişinin rota varyasyonlarına etkisi incelenmiştir.

5.1. Uçuş Gecikme Tahmini Bulgu Sonuçları

Havayolu uçuş gecikme tahmini modelinin her bir parametresi için en uygun hiperparametre değerleri, hiperparametre optimizasyonu süreciyle belirlenmiştir. Bu sürecin sonucunda elde edilen hiperparametre değerleri ve ilgili sonuçlar Çizelge 5.1 ve Çizelge 5.2’de sunulmaktadır. Çizelge 5.1 incelendiğinde yaklaşık 900 epoch civarında hata metrikleri en düşük seviyelere ulaşırken, 1000 epoch’ta hafif bir artış gözlenir. Bu durum modelin potansiyel olarak en iyi performansını elde etmek için çok uzun eğitime gerek olmayabileceğini, belirli bir noktada erken durdurma mekanizmasının kullanılarak gereksiz zaman ve kaynak harcamasının önlenebileceğini göstermektedir. Modelinizin farklı batch size değerlerine karşı gösterdiği Çizelge 5.2’deki performans metriklerinin oldukça benzer olduğunu ve genel olarak modelin batch size değişikliklerine karşı dayanıklı olduğunu göstermektedir. Ancak, batch_size 128 değeri, en düşük RMSE, MSE ve MAE değerlerine ulaşarak modelin en iyi performansını sağladığını görüyoruz.

Çizelge 5.1. Farklı epoch sayılarında model performans metrikleri

Epoch	RMSE	MSE	MAE
100	0.270	0.0730	0.201
200	0.270	0.0729	0.202
300	0.268	0.0722	0.201
400	0.269	0.0724	0.201
500	0.269	0.0726	0.202
600	0.266	0.0711	0.199
700	0.267	0.0713	0.199
800	0.265	0.0706	0.197
900	0.263	0.0695	0.195
1000	0.264	0.0697	0.197

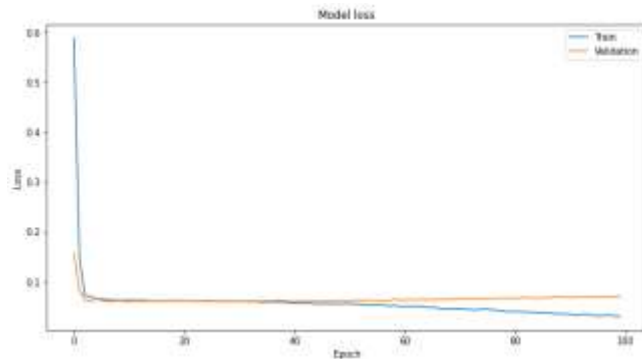
*batch_size=256

Çizelge 5.2. Batch Size değerlerine göre model performans metrikleri

batch_size	RMSE	MSE	MAE
8	0.265	0.070	0.2
16	0.265	0.070	0.198
32	0.266	0.071	0.198
64	0.266	0.071	0.199
128	0.262	0.069	0.197
256	0.263	0.069	0.196

*epoch=100

Modelimizde eğitim verisinin aşırı uyum sağladığını Şekil 5.1’de görebiliyoruz. Aynı zamanda doğrulama verisi üzerinde performansı artmamaktadır. Eğitim kaybının doğrulama kaybından daha düşük seyretmesi beklenen bir durumdur. Ancak aradaki fark büyümeye başladığında, modelin aşırı öğrenme(overfitting) yaptığı söylenebilir. Bu durumdan kaçınmak için veri artırma, dropout ve hipermetre optimizasyonu yapılmıştır. Bu grafikte doğrulama kaybının belirli bir noktadan sonra artmaması olumlu bir işarettir. Modelin eğitim performansı sürekli iyileşirken, doğrulama performansı belirli bir noktadan sonra sabit kalmıştır.

**Şekil 5.1** Model Loss Grafiği

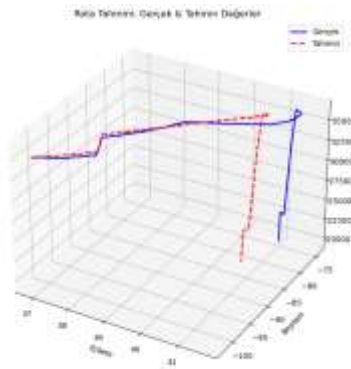
5.2. Uçuş Rota Tahmini Bulgu Sonuçları

Bu bölümde, uçuş rota tahmini çalışmalarına yönelik elde edilen bulgular ve sonuçlar detaylı bir şekilde ele alınmaktadır. Rota tahmini, uçakların ADS-B verilerine dayanarak geçmiş zaman adımlarındaki konum (enlem-boylam), hız ve irtifa gibi değişkenler kullanılarak gerçekleştirilmiştir. Bu süreçte, LSTM modeli tercih edilmiş ve zaman serisi verilerindeki ardışık bağımlılıkların yakalanması hedeflenmiştir. Model, uçuşun ilerleyen zaman adımlarında konum koordinatlarını tahmin ederek gerçek rotadan olan sapma oranını analiz etmeye yönelik olarak yapılandırılmıştır.

Modelin tahmin ettiği enlem (latitude) ve boylam (longitude) değerleri, gerçek değerlere oldukça yakın çıkmıştır. Bu durum, hem grafiksel görselleştirme Şekil 5.2 hem de tablo verileri Çizelge 5.3 üzerinden açıkça gözlemlenmiştir. Özellikle tahmin edilen koordinatlar ile gerçek koordinatlar arasındaki farkların oldukça küçük olduğu, tahmin sonuçlarının modelin yüksek doğrulukla çalıştığını gösterdiği söylenebilir. Tahmin edilen rota, genellikle gerçek rotaya büyük ölçüde uyum göstermiştir. Ortalama mutlak hata (MAE), tahmin edilen ve gerçek konumlar arasındaki farklar için 0.054 derece olarak hesaplanmıştır.

Çizelge 5.3. Gerçek ve tahmini konum bilgileri tablosu

Gerçek Konum		Tahmini Konum	
36.8866	-102.1193	36.8563	-102.4007
36.8931	-102.0864	36.859	-102.38296
36.8931	-102.0864	36.863	-102.3601
36.9005	-102.0587	36.8658	-102.3443
36.9105	-102.1677	36.9097	-102.2237



Şekil 5.2 3D Uçuş rotası: gerçek ve tahmin edilen koordinatlar

6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

6.1. Sonuçlar

Bu araştırmada, uçuş rotası tahmini ve uçuş gecikmesi tahmini üzerine derin öğrenme tabanlı yöntemler kullanılarak kapsamlı bir analiz gerçekleştirilmiştir. LAX-JFK rotası üzerindeki uçuş verileri, uçuş operasyonları ve hava trafiği kontrol süreçlerinde iki kritik unsuru, uçuş rotası ve gecikme tahminini anlamak ve modellemek için kullanılmıştır. Araştırma sonuçları, bu iki faktörün birbiriyle güçlü bir ilişkiye sahip olduğunu ve her iki unsurun optimize edilmesinin uçuş operasyonları açısından büyük önem taşıdığını ortaya koymaktadır.

Araştırma kapsamında, uçuş gecikmesi tahmini için CNN modeli, uçuş rotası tahmini için ise LSTM modeli kullanılmıştır. Bu modellerin performansları, doğruluk ve hata oranları ile değerlendirilmiştir. CNN modeli, uçuş gecikmelerini tahmin etmek için Ocak-Eylül 2024 tarihleri arasındaki uçuş verilerini analiz etmiş ve yaklaşık 20 dakikalık bir ortalama mutlak hata (MAE) ile tatmin edici bir performans göstermiştir. Uçuş gecikmesi incelenen dönem itibari ile trafik yoğunluğu ve uçuş saatleri gibi faktörlerden etkilenmiştir. Bu durum, uçuş rotası ve gecikme tahminlerinin birbirini destekleyici nitelikte olabileceğini göstermektedir. LSTM modeli ile de uçuş rotası tahmininde geçmiş zaman adımlarındaki enlem, boylam, irtifa ve hız bilgilerini kullanarak gelecekteki uçuş konumlarını oldukça başarılı bir şekilde öngörmüştür. Ortalama sapma oranı (MAE) 0.054 derece ile %5.4 oran olarak hesaplanmıştır.

Uçuş rotası analizi, modern havacılık endüstrisinde operasyonel verimlilik, yakıt tüketimi ve yolcu memnuniyetini etkileyecek olan uçuş gecikmesinin önlenmesi adına kritik bir öneme sahiptir. Rota tahmini, uçuş sürecinde meydana gelebilecek sapmaların öngörülmesi ve bu sapmaların etkilerinin minimize edilmesi açısından stratejik bir araç olarak kullanılabilir. Bu çalışmada elde edilen bulgular, uçuş rotası tahmininin aşağıdaki nedenlerle önemli olduğunu ortaya koymaktadır:

- **Yakıt ve Maliyet Optimizasyonu:** Rota sapmaları, uçuş süresinin uzamasına ve dolayısıyla yakıt tüketiminin artmasına yol açar. Tahmin edilen rota

bilgileri kullanılarak daha verimli uçuş rotaları oluşturulabilir ve operasyonel maliyetler azaltılabilir.

- **Uçuş Güvenliği:** Hava durumu, trafik yoğunluğu veya ani rota değişiklikleri gibi faktörler uçuş güvenliği açısından kritik öneme sahiptir. Rota tahmini modelleri, bu tür faktörlerin etkilerini önceden tahmin ederek uçuş ekiplerinin daha bilinçli kararlar almasına yardımcı olabilir.
- **Havayolu Operasyonlarının Verimliliği:** Rota sapmalarının azaltılması, havayolu operasyonlarında zamanında kalkış ve varış oranlarını artırabilir. Bu, havayolu şirketlerinin operasyonel güvenilirliğini ve müşteri memnuniyetini artırır.
- **Operasyonel Zorlukların Belirlenmesi:** Rota tahmini ve gecikme tahmini sonuçlarının birleştirilmesi, operasyonel darboğazların belirlenmesine yardımcı olabilir. Örneğin, belirli bir hava sahasında sıkça yaşanan rota sapmaları ve gecikmeler, havayolu şirketleri için potansiyel iyileştirme alanlarını işaret edebilir.
- **Hava Durumu ve Trafik Etkilerinin Modellenmesi:** Hava durumu ve hava trafik yoğunluğu gibi faktörler hem rotayı hem de gecikmeyi etkileyen temel unsurlardır. Bu çalışmada elde edilen bulgular, bu faktörlerin uçuş operasyonlarına olan etkilerinin detaylı bir şekilde modellenmesi gerektiğini ortaya koymaktadır.

6.2. Öneriler

Bu çalışma, uçuş rotası ve gecikme tahmini süreçlerinde derin öğrenme tabanlı modellerin uygulanabilirliğini ve etkinliğini ortaya koymuştur. Analizlerde gerçekleştirilen rota tahminleri, uçuş operasyonlarının optimize edilmesi ve havayolu endüstrisindeki verimliliğin artırılması açısından önemli bulgular sunmaktadır. Araştırma sonucunda, uçuş gecikmeleri ile uçuş rotası arasında önemli bir ilişki olduğu belirlenmiştir. Gecikmelerin büyük ölçüde trafik yoğunluğu ve operasyonel zorluklardan

kaynaklandığı, aynı faktörlerin uçuş rotasındaki sapmalara da neden olduğu gözlemlenmiştir. Özellikle yoğun hava trafik dönemlerinde meydana gelen rota sapmalarının, uçuş sürelerini uzatarak gecikmelere katkıda bulunduğu tespit edilmiştir. Elde edilen sonuçlar her iki modelin de uçuş operasyonlarının gelecekteki durumlarını öngörmeye güçlü araçlar olarak kullanılabileceğini göstermektedir. Özellikle rota tahminlerinin doğruluğu, uçuş operasyonlarının daha verimli bir şekilde yönetilmesine olanak tanıyacak niteliktedir.

Gecikme ve rota tahmini süreçlerinin entegre bir analizi ile rota tahmini sırasında ortaya çıkan sapmaların gecikme tahmin modellerine doğrudan girdi olarak eklenmesi, her iki modelin doğruluğunu artırabilir. Hava durumu, rüzgar hızı, hava trafik yoğunluğu gibi dinamik faktörler hem gecikme hem de rota tahminlerine dahil edilerek modellerin gerçek operasyon koşullarını daha iyi temsil etmesi sağlanabilir.

Rota sapmalarının ve gecikmelerin minimize edilmesi, yakıt tüketimi ve operasyonel maliyetlerin azaltılmasında önemli bir rol oynayabilir. Bu kapsamda, tahmin edilen rota bilgileri, daha kısa ve daha verimli hava yollarının belirlenmesinde kullanılabilir. Havayolu şirketleri, bu tahmin modellerini kullanarak kalkış ve varış planlamalarında daha kesin öngörülerde bulunabilir ve yolcu memnuniyetini artırabilir.

Daha büyük veri kümeleriyle çalışılarak modellerin genelleme yeteneği artırılabilir. Örneğin, LAX-JFK dışındaki rotalar için de benzer analizler yapılarak modelin farklı operasyonel koşullara adaptasyonu test edilebilir.

Hibrit modellerin (örneğin CNN-LSTM) geliştirilmesi, rota tahmini doğruluğunu daha da artırabilir ve bu doğrultuda daha karmaşık operasyonel örüntüler öğrenilebilir.

Bu çalışmanın sonuçları, modern havacılık operasyonlarının optimize edilmesinde uçuş gecikmesi ve rota tahmini analizlerinin kritik bir rol oynadığını göstermektedir. Çalışma kapsamında geliştirilen modeller, operasyonel süreçlerin iyileştirilmesine yönelik önemli bir adım teşkil etmektedir. Önerilen iyileştirmeler ve gelecekteki çalışmalar, bu modellerin daha geniş bir kapsamda uygulanabilirliğini artırabilir ve havayolu endüstrisinin karşılaştığı operasyonel zorlukların üstesinden gelinmesine katkıda bulunabilir. Bu bağlamda, uçuş rotası ve gecikme tahmini üzerine

yapılacak daha fazla çalışma, havacılık güvenliđi ve müşteri memnuniyetinin artırılmasında önemli bir rol oynayacaktır.



7. KAYNAKLAR

- Abid Ali Awan. (2022, March 16). *Recurrent Neural Network Tutorial (RNN)*. Datacamp.Com. <https://www.datacamp.com/tutorial/tutorial-for-recurrent-neural-network>
- ADS-B exchange. (2024, November 30). <https://globe.adsbexchange.com/>
- Automatic dependent surveillance-broadcast (ADS-B)*. (n.d.). Retrieved November 30, 2024, from https://www.faa.gov/air_traffic/technology/adsb
- Balakrishna, P., Ganesan, R., & Sherry, L. (2010). Accuracy of reinforcement learning algorithms for predicting aircraft taxi-out times: A case-study of Tampa Bay departures. *Transportation Research Part C: Emerging Technologies*, 18, 950–962. <https://doi.org/10.1016/j.trc.2010.03.003>
- Bishop, C. (2006). Pattern Recognition and Machine Learning. In *Journal of Electronic Imaging* (Vol. 16, pp. 140–155). <https://doi.org/10.1117/1.2819119>
- Bojarski, M., Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). *End to End Learning for Self-Driving Cars*. <https://doi.org/10.48550/arXiv.1604.07316>
- Bureau of Transportation Statistics (BTS) (2009) Airline on-time statistics and delay causes: on-time arrival performance*. (n.d.). Retrieved December 1, 2024, from http://www.transtats.bts.gov/OT_Delay/ot_delaycause1.asp?display=data&pn=1
- Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (n.d.). *Doctor AI: Predicting Clinical Events via Recurrent Neural Networks*.
- Chollet, F. (2017). *Deep Learning with Python*. <https://catalogue.library.cern/literature/wf1vv-31806>
- Cockburn, I. M., Henderson, R., Stern, S., Professor, H., Management, E., Business, H., & Morgan, S. (2018). *The Impact of Artificial Intelligence on Innovation*. <http://www.nber.org/papers/w24449>
- Coşkun, M., Yildirim, Ö., Uçar, A., & Demir, Y. (2017). AN OVERVIEW OF POPULAR DEEP LEARNING METHODS. In *European Journal of Technic EJT* (Vol. 7, Issue 2).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. <https://doi.org/10.48550/arXiv.1810.04805>
- Gatys, L., Ecker, A., & Bethge, M. (2015). A Neural Algorithm of Artistic Style. *ArXiv*. <https://doi.org/10.1167/16.12.326>
- Goodfellow, I., Bengio, Y., & Courville, A. (n.d.). *Deep Learning*.

- Gui, G., Liu, F., Sun, J., Yang, J., Zhou, Z., & Zhao, D. (2020). Flight delay prediction based on aviation big data and machine learning. *IEEE Transactions on Vehicular Technology*, 69(1), 140–150. <https://doi.org/10.1109/TVT.2019.2954094>
- Hafeez, A., & Sial, A. (2021). Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python [HEC Y Cat]. *International Journal of Advanced Trends in Computer Science and Engineering*, 10, 2281–2770. <https://doi.org/10.30534/ijatcse/2021/391012021>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Jordan, M., Kleinberg, J., & Schölkopf, B. (n.d.). *Pattern Recognition and Machine Learning*.
- Khaksar, H., & Sheikholeslami, A. (2019). Airline delay prediction by machine learning algorithms. *Scientia Iranica*, 26(5 A), 2689–2702. <https://doi.org/10.24200/sci.2017.20020>
- Kim, Y. J., Choi, S., Briceno, S., & Mavris, D. (2016). A deep learning approach to flight delay prediction. 1–6. <https://doi.org/10.1109/DASC.2016.7778092>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (n.d.). *ImageNet Classification with Deep Convolutional Neural Networks*. <http://code.google.com/p/cuda-convnet/>
- Laying down requirements for the performance and the interoperability of surveillance for the single european sky text with eea relevance*. (2022). <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02011R1207-20220125&from=EN>
- Lecun, Y., & Bengio, Y. (1995). Convolutional Networks for Images, Speech, and Time-Series. In *The Handbook of Brain Theory and Neural Networks*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, 521, 436–444. <https://doi.org/10.1038/nature14539>
- Li, Q., & Jing, R. (2022). Flight Delay Prediction from Spatial and Temporal Perspective. *Expert Systems with Applications*, 205, 117662. <https://doi.org/10.1016/j.eswa.2022.117662>
- Lindholm, A., Wahlström, N., Lindsten, F., & Schön, T. B. (2022). *MACHINE LEARNING A First Course for Engineers and Scientists*. <http://www.cambridge.org>.
- LM Po. (2024, July 6). *The Evolution of Artificial Neurons*. Medium.Com. <https://medium.com/@lmpo/the-evolution-of-artificial-neurons-90619f224f63>

- Ma, L., & Tian, S. (2020). A Hybrid CNN-LSTM Model for Aircraft 4D Trajectory Prediction. *IEEE Access*, 8, 134668–134680. <https://doi.org/10.1109/ACCESS.2020.3010963>
- Massaoudi, M., Abu-Rub, H., S. Refaat, S., Chihi, I., & Oueslati, F. (2021). Deep Learning in Smart Grid Technology: A Review of Recent Advancements and Future Prospects. *IEEE Access*, 9, 54558–54578. <https://doi.org/10.1109/ACCESS.2021.3071269>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133. <https://doi.org/10.1007/BF02478259>
- McCulloch, W. S., & Pitts, W. (1990). A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY* n. In *Bulletin of Mothemnticnl Biology* (Vol. 52, Issue 2).
- McKinney, W., & Safari, an O. M. C. (2017). *Python for Data Analysis, 2nd Edition* (1st edition). O'Reilly Media, Inc.
- Pirim, H. (n.d.). YAPAY ZEKA.
- Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2017). *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. <http://arxiv.org/abs/1711.05225>
- Ravish Raj. (n.d.). *Supervised, Unsupervised and Semi-supervised Learning with Real-life Usecase*. Retrieved December 7, 2024, from <https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning>
- Reynolds, D., Campbell, W., Richardson, F., Dagli, C., Sahin, C., Gadepally, V., Tran, A., Greenfield, K., Trepagnier, P., Hall, R., Zipkin, J., Roeser, C., Mohindra, S., Hennighausen, K., & Thornton, J. (2019). *Artificial Intelligence: Short History, Present Developments, and Future Outlook Final Report*.
- Rosenblatt, F. (n.d.). THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN 1. In *Psychological Review* (Vol. 65, Issue 6).
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533–536. <https://api.semanticscholar.org/CorpusID:205001834>
- Rumelhart, D., Hinton, G., & Williams, R. (2002). *Learning Representations by Back-Propagating Errors* (pp. 213–222). <https://doi.org/10.7551/mitpress/1888.003.0013>

- Saabith, A. S., Vinothraj, T., & Fareez, M. (n.d.). A Review on Python Libraries and IDEs for Data Science. In *International Journal of Research in Engineering and Science (IJRES)* ISSN. www.ijres.org
- Sheela, A., & Meena, Kp. (n.d.). *Combination of NumPy, SciPy and Matplotlib/PyLab-a good alternative methodology to MATLAB-A Comparative analysis.*
- Shi, Z., Xu, M., Pan, Q., Yan, B., & Zhang, H. (2018). *2018 International Joint Conference on Neural Networks (IJCNN) : 2018 proceedings.* IEEE.
- Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529, 484–489. <https://doi.org/10.1038/nature16961>
- Sutton, R., & Barto, A. G. (1999). Reinforcement learning. *Journal of Cognitive Neuroscience*, 11, 126–134.
- Tanja Grobrotek. (2019, July 11). *The European Airspace Challenge.* <https://Nats.Aero/Blog/2019/09/the-European-Airspace-Challenge/>.
- Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (n.d.). *Attention Is All You Need.*
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need.* <http://arxiv.org/abs/1706.03762>
- What Is Unsupervised Learning?* (2024, January 26). Eastgate Software. <https://eastgate-software.com/what-is-unsupervised-learning/>
- Yablonsky, G., Steckel, R., Constales, D., Farnan, J., Lercel, D., & Patankar, M. (2014). Flight delay performance at Hartsfield-Jackson Atlanta International Airport. *Journal of Airline and Airport Management*, 4(1). <https://doi.org/10.3926/jairm.22>