

**T.C.  
HASAN KALYONCU ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
ELEKTRONİK BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**



**FIRIN SENSÖR VERİLERİNİ KULLANARAK FOTOVOLTAİK PANEL  
ÜRETİM SÜRECİNDE ZAMAN SERİSİ VE LSTM YÖNTEMİ İLE ANOMALİ  
TESPİTİ**

**Yeşim GÖRÜR**

**YÜKSEK LİSANS TEZİ**

**GAZİANTEP- 2025**



**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**  
**YÜKSEK LİSANS TEZ KABUL VE ONAY FORMU**

Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Programı öğrencisi Yeşim GÖRÜR tarafından hazırlanan “**Fırın Sensör Verilerini Kullanarak Fotovoltaik Panel Üretim Sürecinde Zaman Serisi Analizi ve LSTM Yöntemi ile Anomali Tespiti**” başlıklı tez, **15/01/2025** tarihinde yapılan savunma sınavı sonucu başarılı bulunarak jürimiz tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

<b><u>Görevi</u></b>	<b><u>Unvanı, Adı ve Soyadı</u></b>	<b><u>Kurumu/Üniversitesi</u></b>	<b><u>İmzası:</u></b>
<b>Tez Danışmanı</b>	Dr. Öğr. Üyesi Ali Emre Öztürk	Elektrik Elektronik Mühendisliği/Hasan Kalyoncu Üniversitesi	
<b>Jüri Başkanı</b>	Doç. Dr. Bülent Haznedar	Bilgisayar Mühendisliği/Gaziantep Üniversitesi	
<b>Jüri Üyesi</b>	Dr. Öğr. Üyesi Erkan Kıymık	Elektrik Elektronik Mühendisliği/Hasan Kalyoncu Üniversitesi	
<b>Jüri Üyesi</b>	.....	.....	
<b>Jüri Üyesi</b>	.....	.....	

**Bu tez Enstitü Yönetim Kurulunca belirlenen yukarıdaki jüri üyeleri tarafından uygun görülmüş ve Enstitü Yönetim Kurulu kararı ile onaylanmıştır.**

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 material and results that are not original to this work.

Yeşim GÖRÜR

13.02.2025

**HASAN KALYONCU ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**  
**ELEKTRONİK BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**FIRIN SENSÖR VERİLERİNİ KULLANARAK FOTOVOLTAİK PANEL  
ÜRETİM SÜRECİNDE ZAMAN SERİSİ VE LSTM YÖNTEMİ İLE ANOMALİ  
TESPİTİ**

**Yeşim GÖRÜR**

**YÜKSEK LİSANS TEZİ**

**Danışman**  
**Dr. Öğr. Üyesi Ali Emre Öztürk**

**Eş Danışman**  
**Dr. Öğr. Üyesi Erkan Kıymık**

**ÖZET**

Enerji arzının yenilenebilir kaynaklardan sağlanması, güneş enerjisi sistemlerinin önemini arttırmış ve bu sistemlerin kullanımının hızla yaygınlaşmasına zemin hazırlamıştır. Bu bağlamda fotovoltaik sistemler, enerji sektöründe kritik bir rol üstlenmiştir. Fakat bu fotovoltaik sistemlerin üretim sürecindeki yüksek maliyetler ve operasyonel hassasiyetler üretim verimliliği ve kalite sürekliliği üzerinde doğrudan etkili olmaktadır. Endüstri 4.0 teknolojilerinin sağladığı dijitalleşme ve otomasyon, işletmelerdeki bileşenlerin haberleşmesine ve insanlarla gerçek zamanlı iletişim kurabilmesine olanak sağlamıştır. Bu durum, canlı süreç izleme yöntemlerinin de etkisiyle, yüksek boyutlu verilerin üretilmesine katkı sağlamaktadır. Tahminsel bakım kapsamında değerlendirilen, verilerin detaylı analizi ile arızaların gerçekleşmeden tahmin edilmesi, potansiyel arızaları önceden tespit etmeyi ve arızaya zamanında müdahaleyi hedefler. Böylece öngörülme arızalar önlenebilir, iş duruş süreleri minimize edilebilir, kaynak kullanımı ve ekipman varlık ömrü optimize edilebilir. Bu çalışma, Kalyon Fotovoltaik (PV) Güneş Teknolojileri Fabrikası'ndaki üretim hattındaki silisyum eritme ve kristal büyütme fırını üzerindeki sensörlerden elde edilen gerçek zamanlı verilerin bir araya getirilmesiyle, veri odaklı analiz ve tahmin süreçlerine odaklanmaktadır. Zaman serisi analizi, sıralı verilerdeki zamansal bağlantıları ve eğilimleri inceleyerek makinelerin dinamik davranışlarını anlamada önemli bir yöntem sunmaktadır. Büyük verilerdeki karmaşık desenleri tanımlama kapasitesine sahip derin öğrenme teknikleri, veri analizi ve tahmin süreçlerinde önemli rol oynamaktadır. Özellikle uzun kısa süreli bellek ağları (LSTM), zaman serisi verilerindeki karmaşık ilişkileri öğrenme yeteneği sayesinde, zamansal veriler üzerinde yüksek doğrulukla tahmin ve analiz yapılmasına olanak tanımıştır. Keşifsel veri analizi ve veri ön işleme süreçleriyle optimize edilen veri seti üzerinde sekiz deney gerçekleştirilmiş, modelin doğruluğunu artırmaya yönelik ek özellikler test edilmiştir. Sonuç olarak, modelin doğruluğu kademeli olarak iyileştirilmiş ve %98 doğruluk oranına ulaşılmıştır.

**Anahtar Kelimeler:** Kestirimci bakım, derin öğrenme, anomali tespiti, zaman serisi analizi, LSTM

**HASAN KALYONCU UNIVERSITY  
GRADUATE EDUCATION INSTITUTE  
DEPARTMENT OF ELECTRONICS COMPUTER ENGINEERING**

**FIRIN SENSÖR VERİLERİNİ KULLANARAK FOTOVOLTAİK PANEL  
ÜRETİM SÜRECİNDE ZAMAN SERİSİ VE LSTM YÖNTEMİ İLE ANOMALİ  
TESPİTİ**

**Yeşim GÖRÜR**

**MASTER THESIS**

**Supervisor**

**Asst. Prof. Dr. Ali Emre ÖZTÜRK**

**Co-Supervisor**

**Asst. Prof. Dr. Erkan KIYMIK**

**ABSTRACT**

The use of renewable energy sources has increased the importance of solar energy systems and led to their rapid expansion. In this context, photovoltaic systems play a key role in the energy sector. However, the high production costs and operational challenges of these systems directly affect production efficiency and quality. Industry 4.0 technologies, with their digitalization and automation capabilities, allow components in factories to communicate and interact with humans in real time. This, combined with live process monitoring, helps generate large amounts of valuable data. Predictive maintenance involves analyzing data to detect failures before they happen, allowing for early intervention. This helps prevent unexpected breakdowns, reduce downtime, optimize resource use, and extend equipment lifespan. This study focuses on analyzing and predicting data collected from sensors on the silicon melting and crystal growth furnace at the Kalyon Photovoltaic (PV) Solar Technologies Factory. Time series analysis is an important method for understanding machine behavior by identifying patterns and trends in sequential data. Deep learning techniques, which can recognize complex patterns in large datasets, play a key role in data analysis and prediction. In particular, Long Short-Term Memory (LSTM) networks can learn complex relationships in time series data, enabling highly accurate forecasts. After preparing and optimizing the dataset through data analysis and preprocessing, eight experiments were conducted to test improvements. As a result, the model's accuracy increased step by step, reaching 98% in the final experiment.

**Keywords:** Predictive Maintenance, Deep Learning, Anomaly Detection, Time Series Analysis, LSTM

## ÖNSÖZ

Bu çalışmanın her aşamasında bana rehberlik eden ve süreç boyunca değerli bilgi ve deneyimlerini paylaşarak yolumu aydınlatan Dr. Öğr. Üyesi Ali Emre Öztürk ve Dr. Öğr. Üyesi Erkan Kıymık'a teşekkürlerimi sunarım. Ayrıca, tezimi değerlendirme sürecinde gösterdikleri ilgi ve katkıları için kıymetli jüri başkanı Doç. Dr. Bülent Haznedar'a teşekkürlerimi iletmek isterim.

Bu süreçte güçlü vizyonu, derin kişiliği ve her daim hissettiğim desteğiyle bana ilham veren sevgili dostum Mihriban Türk Ay'a sonsuz teşekkürler. Ayrıca, bu çalışmada kullanılan veri setini paylaşarak çalışmanın gerçekleştirilmesine katkıda bulunan Kalyon Fotovoltaik (PV) Güneş Teknolojileri Fabrikası'na da teşekkür ederim.

Son olarak, bu çalışmayı her zaman yanımda olan, sevgileri ve destekleriyle bana güç veren, hayatımın en değerli parçaları olan aileme ithaf ediyorum.

Yeşim GÖRÜR

Gaziantep 2025

# İÇİNDEKİLER

ÖNSÖZ .....	vi
İÇİNDEKİLER .....	vii
TABLolar LİSTESİ .....	ix
ŞEKİLLER LİSTESİ .....	xi
KISALTMALAR LİSTESİ.....	xii
<b>1. GİRİŞ.....</b>	<b>1</b>
1.1. Araştırma Problemi ve Gerekçesi .....	1
1.2. Çalışmanın Amacı ve Hedefleri.....	2
1.3. Çalışmanın Önemi, Özgün Değeri ve Katkısı .....	3
1.4. Tez Özeti.....	3
<b>2. KAVRAMSAL ÇERÇEVE.....</b>	<b>4</b>
<b>3. MATERYAL VE YÖNTEM.....</b>	<b>8</b>
3.1. Çalışma Tasarımı .....	8
3.2. Materyal .....	8
3.2.1. Yapay zekâ (Artificial Intelligence).....	8
3.2.2. Makine öğrenimi.....	9
3.2.3. Derin öğrenme .....	10
3.2.3.1. Derin öğrenme ağları .....	10
3.2.3.1.1. Convolutional Neural Networks (CNN).....	10
3.2.3.1.2. Recurrent Neural Networks (RNN).....	11
3.2.3.1.3. Transformers.....	12
3.2.3.1.4. Generative Adversarial Networks (GAN) .....	12
3.2.3.1.5. Autoencoders (AE).....	12
3.2.3.1.6. Graph Neural Networks (GNN) .....	13
3.2.3.1.7. Long Short-Term Memory (LSTM).....	13
3.2.3.2. Derin öğrenme araçları .....	15
3.2.3.3. Platformlar .....	16
3.2.3.4. Donanım araçları.....	16
3.2. Yöntem.....	17
3.2.1. Derin Öğrenme ile anomali tespitine yönelik verilerin hazırlanması .....	18
3.2.1.1. Sensör verilerinin toplanması ve tanımı .....	18
3.2.1.2. Keşifsel Veri Analizi .....	21
3.2.1.3. Veri ön işleme .....	25
3.2.1.3.1. Zaman serisi verisinin normalizasyonu .....	25
3.2.1.4. Özellik çıkarımı .....	26
3.2.1.5. Zaman serisi verisinin eğitim /test verilerine ayrılması ve şekillendirilmesi.....	28
3.2.2. Model geliştirme ve değerlendirme .....	30

3.2.2.1. LSTM hiperparametreleri .....	30
3.3.2.1.1. Aktivasyon fonksiyonları (Activation Functions) .....	30
3.3.2.1.2. Optimizasyon algoritmaları (Optimization Algorithms) .....	31
3.3.2.1.3. Modelin diğ er hiperparametreleri .....	32
3.3.2.2. Model performansının iyileştirilmesi için hiperparametre ayarı Keras Tuner .....	35
3.3.2.3. Değ erlendirme metrikleri .....	35
<b>4. DENEYLER VE BULGULAR .....</b>	<b>36</b>
4.1. Deney 1 .....	36
4.2. Deney 2 .....	40
4.3. Deney 3 .....	43
4.4. Deney 4 .....	46
4.5. Deney 5 .....	49
4.6. Deney 6 .....	52
4.7. Deney 7 .....	54
4.8. Deney 8 .....	58
<b>5. SONUÇLAR VE ÖNERİLER .....</b>	<b>69</b>
5.1 Sonuç ve öneriler .....	69
<b>6. KAYNAKÇA .....</b>	<b>74</b>

## TABLULAR LİSTESİ

<b>Tablo 3. 1.</b> Sistem Özellikleri .....	17
<b>Tablo 3. 2.</b> Fırın sensörlerinden gelen verilerin özellikleri ve tanımları.....	19
<b>Tablo 3. 3.</b> Fırın sensör verilerinin özellikleri ve minimum, maksimum değerleri .....	20
<b>Tablo 3. 4.</b> A1 özelliğine ait numerik değerlerin temsil ettiği işlemler .....	20
<b>Tablo 3. 5.</b> Ham veri formatı.....	22
<b>Tablo 3. 6.</b> Verinin yeniden şekillendirilmesi.....	22
<b>Tablo 3. 7.</b> Zaman serisi veri setindeki tamamıyla “Null” değere sahip satırların örneği .....	23
<b>Tablo 3. 8.</b> Eksik değer tablosu.....	23
<b>Tablo 3. 9.</b> Veri setinin temel istatistik özeti .....	24
<b>Tablo 3. 10.</b> Verilerin Min- Max ölçeklendirme ile normalizasyonu .....	26
<b>Tablo 3. 11.</b> Veri setine eklenen zaman alanı özellikleri (Time Domain Features).....	27
<b>Tablo 3. 12.</b> Zaman alanı özellikleri formülleri (Ding & Ming, 2019).....	28
<b>Tablo 3. 13.</b> Eğitim ve Test Veri Kümesi Boyutları .....	29
<b>Tablo 4. 1.</b> Deney 1 giriş özellikleri.....	36
<b>Tablo 4. 2.</b> Deney 1 parametreleri.....	37
<b>Tablo 4. 3.</b> Veri setlerinin boyutları ve yapıları .....	38
<b>Tablo 4. 4.</b> Deney 1, model performans değerleri.....	38
<b>Tablo 4. 5.</b> Deney 2 giriş özellikleri.....	40
<b>Tablo 4. 6.</b> Deney 2 parametreleri.....	40
<b>Tablo 4. 7.</b> Veri setlerinin boyutları ve yapıları .....	41
<b>Tablo 4. 8.</b> Deney 2, model performans değerleri.....	41
<b>Tablo 4. 9.</b> Deney 3 giriş özellikleri.....	43
<b>Tablo 4. 10.</b> Deney 3 parametreleri.....	43
<b>Tablo 4. 11.</b> Veri setlerinin boyutları ve yapıları .....	44
<b>Tablo 4. 12.</b> Deney 3, model performans değerleri.....	44
<b>Tablo 4. 13.</b> Deney 4 giriş özellikleri.....	46
<b>Tablo 4. 14.</b> Deney 4 parametreleri.....	46
<b>Tablo 4. 15.</b> Veri setlerinin boyutları ve yapıları .....	47
<b>Tablo 4. 16.</b> Deney 4, model performans değerleri.....	47
<b>Tablo 4. 17.</b> Modelin öğrenme eğrisi (Loss Curve).....	48

<b>Tablo 4. 18.</b> Modelin ROC eğrisi ve sınıflandırma başarısı .....	48
<b>Tablo 4. 19.</b> Deney 5 giriş özellikleri.....	49
<b>Tablo 4. 20.</b> Deney 5 parametreleri.....	49
<b>Tablo 4. 21.</b> Veri setlerinin boyutları ve yapıları .....	50
<b>Tablo 4. 22.</b> Deney 5, model performans değerleri.....	50
<b>Tablo 4. 23.</b> İstatistiksel özelliklerin eklenmesiyle oluşan deney 6 giriş özellikleri.....	52
<b>Tablo 4. 24.</b> Deney 6 parametreleri.....	52
<b>Tablo 4. 25.</b> Veri setlerinin boyutları ve yapıları .....	53
<b>Tablo 4. 26.</b> Deney 6, model performans değerleri.....	54
<b>Tablo 4. 27.</b> İstatistiksel özelliklerin eklenmesiyle oluşan deney 7 giriş özellikleri.....	55
<b>Tablo 4. 28.</b> Deney 7 parametreleri.....	55
<b>Tablo 4. 29.</b> Veri setlerinin boyutları ve yapıları .....	56
<b>Tablo 4. 30.</b> Deney 7, model performans değerleri.....	56
<b>Tablo 4. 31.</b> İstatistiksel özelliklerin eklenmesiyle oluşan deney 8 giriş özellikleri.....	58
<b>Tablo 4. 32.</b> Deney 8 parametreleri.....	58
<b>Tablo 4. 33.</b> Veri setlerinin boyutları ve yapıları .....	59
<b>Tablo 4. 34.</b> Keras Tuner ile optimizasyonu yapılan hiperparametreler ve değer aralıkları .....	60
<b>Tablo 4. 35.</b> Keras Tuner ile elde edilen en iyi hiperparametre değerleri.....	61
<b>Tablo 4. 36.</b> Deney 8, model performans değerleri.....	61
<b>Tablo 5. 1.</b> Model performans değerleri: deney sonuçları .....	72

## ŞEKİLLER LİSTESİ

Şekil 3. 1. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki hiyerarşik ilişki .....	9
Şekil 3. 2. Recurrent Neural Networks (RNN) Yapısı .....	12
Şekil 3. 3. LSTM ağının makine öğrenmesi mimarisindeki yeri.....	14
Şekil 3. 4. Önerilen çalışma için LSTM ağının yapısı.....	15
Şekil 3. 5. Anomali tespiti için önerilen metodoloji diyagramı.....	18
Şekil 3. 6. Veri ön işleme altındaki farklı aşamalar/adımlar .....	18
Şekil 3. 7. Değişkenler Arasındaki İlişkilerin Korelasyon Matrisi.....	25
Şekil 3. 8. Zaman serisi verilerinin çerçevesi .....	25
Şekil 3. 9. Verilerin Min-Max ölçeklendirme ile normalizasyonu kodu .....	26
Şekil 3. 10. Kayan pencere algoritması .....	29
Şekil 3. 11. Veri setinin eğitim ve test verilerine ayrılması ve yeniden şekillendirilmesi .....	29
Şekil 4. 1. Kayan pencere algoritması .....	38
Şekil 4. 2. Modelin öğrenme eğrisi (Loss Curve).....	39
Şekil 4. 3. Modelin ROC eğrisi ve sınıflandırma başarısı .....	39
Şekil 4. 4. Modelin öğrenme eğrisi (Loss Curve).....	42
Şekil 4. 5. Modelin ROC eğrisi ve sınıflandırma başarısı .....	42
Şekil 4. 6. Modelin öğrenme eğrisi (Loss Curve).....	45
Şekil 4. 7. Modelin ROC eğrisi ve sınıflandırma başarısı .....	45
Şekil 4. 12. Modelin ROC eğrisi ve sınıflandırma başarısı .....	54
Şekil 4. 13. Modelin öğrenme eğrisi (Loss Curve).....	57
Şekil 4. 14. Modelin ROC eğrisi ve sınıflandırma başarısı .....	57
Şekil 5. 1. Önerilen modelin korelasyon matrisi .....	70
Şekil 5.2. Normalizasyondan önce ve sonra A39 özelliğinin değerleri.....	71

## KISALTMALAR LİSTESİ

<b>LSTM</b>	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
<b>IOT</b>	Internet of Things (Nesnelerin İnterneti)
<b>PV</b>	Photovoltaic Panel (Fotovoltaik Panel)
<b>ERP</b>	Enterprise Resource Planning (Kurumsal Kaynak Planlaması)
<b>AR-GE</b>	Research and Development (Araştırma ve Geliştirme)
<b>PCA</b>	Principal Component Analysis (Ana Bileşen Analizi)
<b>KNN</b>	K-Nearest Neighbors (K-En Yakın Komşu)
<b>VGG</b>	Visual Geometry Group
<b>RESNET</b>	Residual Networks
<b>AE</b>	Autoencoder (Oto Kodlayıcı)
<b>ML</b>	Machine Learning (Makine Öğrenimi)
<b>CNN</b>	Convolutional Neural Networks (Konvolüsyonel Sinir Ağları)
<b>RNN</b>	Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
<b>DNN</b>	Deep Neural Networks (Derin Sinir Ağları)
<b>GPU</b>	Graphics Processing Unit (Grafik İşlem Birimi)
<b>GAN</b>	Generative Adversarial Networks
<b>GNN</b>	Graph Neural Networks (Grafik Sinir Ağları)
<b>VM</b>	Virtual Machine (Sanal Makine)
<b>CPU</b>	Central Processing Unit (Merkezi İşlem Birimi)
<b>TPU</b>	Tensor Processing Unit (Tensor İşlem Birimi)
<b>MAE</b>	Mean Absolute Error (Ortalama Mutlak Hata)
<b>MSE</b>	Mean Squared Error (Ortalama Kare Hata)
<b>R<sup>2</sup></b>	R-squared (R-kare)
<b>EDA</b>	Exploratory Data Analysis (Keşifsel Veri Analizi)
<b>NULL</b>	Missing Data (Eksik Veri)
<b>RELU</b>	Rectified Linear Unit (Düzeltici Doğrusal Birim)
<b>ADAM</b>	Adaptive Moment Estimation (Uyarlanabilir Momentler)
<b>SGD</b>	Stochastic Gradient Descent (Stokastik Gradyan İnişi)
<b>MIN</b>	Minimum (Minimum)
<b>MAX</b>	Maximum (Maksimum)
<b>MEAN</b>	Average (Ortalama)
<b>VAR</b>	Variance (Varyans)

<b>STD</b>	Standard Deviation (Standart Sapma)
<b>RANGE</b>	Range (Aralık)
<b>SKEW</b>	Skewness (Çarpıklık)
<b>KURTOSIS</b>	Kurtosis (Basıklık)
<b>RMS</b>	Root Mean Square (Kök Ortalama Kare)



# 1. GİRİŞ

## 1.1. Araştırma Problemi ve Gerekçesi

Endüstri 4.0'ın en temel kapsamlarından biri gelişmiş teknolojilerin üretim süreçlerine entegrasyonudur. Nesnelerin interneti (IOT), Endüstri 4.0'ın en önemli kaynağıdır. IOT, birbirine bağlı cihazların, sensörlerin ve sistemlerin gerçek zamanlı veri üretmesini sağlar. Üretilen bu büyük ve karmaşık veri kümelerinden, gizli desenleri, korelasyonları ve insanın göremeyeceği diğer bilgileri öngörerek ham verilerden faydalı bilgi çıkarmak işletme açısından oldukça kritiktir. Büyük veri analitiği ve veri odaklı tekniklerin, üretimden enerjiye, ulaşımdan sağlığa pek çok alanda durum izlemek için kullanımı yaygınlaşmıştır. Durum izleme, ekipmanların ve sistemlerin performansını ve çalışmasını izleyerek, potansiyel sorunları ortaya çıkarmayı ve olası arızaları engellemeyi amaçlar. Durum izlemenin amacı, problemleri kritik hale gelmeden çözmek ve sistemin sürekliliğini sağlayarak verimliliği arttırmaktır. Ayrıca durum izleme, sorun ortaya çıktıktan sonra yapılan bakımı ifade eden geleneksel reaktif bakım ve belirli aralıklarla veya planlı olarak yapılan önleyici bakım türlerinden, sensörler ve veri analitiği kullanılarak arıza oluşmadan önce sorunların tahmin edilmesi ve bu doğrultuda müdahale edilmesini kapsayan modern kestirimci bakım yaklaşımına geçişi mümkün kılar (Bampoula vd., 2024).

Endüstride, üretim süreçlerinin izlenmesi, makinelerde yaşanan hataların teşhisi, kalite yönetimi, ERP sistemleri /ağ güvenliği ve altyapı izleme alanlarında derin öğrenme uygulamalarının kullanımı yaygınlaşmaktadır. Özellikle imalat sürecindeki anomali tespitinde kullanımında artış yaşanmaktadır (Yan vd., 2024).

Zaman serisi tahmini, finansal tahminlerde, trafik ve insan hareketi tahminlerinde geniş uygulama alanına sahip olan, istatistik ve makine öğrenimi alanında öne çıkan bir araştırma konusudur. Yinelenen sinir ağları (RNN), zamansal bağımlılıkları yakalama özelliği sayesinde ardışık verileri modellemede temel yöntem olarak kabul edilmiştir. RNN'lerin varyantı olan uzun kısa süreli bellek (LSTM) ağları ise klasik RNN'lerin karşılaştığı "vanishing gradient ve exploding gradient" problemine çözüm sunarak uzun boyutlu ardışık verileri işlemede daha iyi performans göstermiştir ve RNN türleri arasında popüler hale gelmiştir. Araştırmacılar zaman serisi analizinde bu LSTM ağlarını benimsemiş ve aynı zamanda etkinliğini kanıtlamışlardır (Kong vd., 2024).

Fotovoltaik panel (PV) üretim süreçlerinde silisyum eritme ve kristal büyütme fırını gibi kritik ve kompleks ekipmanlarda yaşanan arızaların büyük ekonomik kayıplara ve üretim sürecinde ciddi aksamalara sebep olabileceği, üretim süreçlerinde aktif görev yapan personelin saha deneyimleri ve gözlemleri sonucu ilettiği bir öngörüdür. Bu makinelerden üretilen veriler karmaşık, yoğun ve dinamik bir yapıya sahiptir. Ayrıca bu veriler uzun süreli bağımlılıkları olan etiketsiz zaman serisi verileridir.

Veriyi analiz etmek ve doğru yöntemlerle işleyip faydalı bilgiye dönüştürmek için gelişmiş analiz ve modelleme yeteneği gereklidir. Bu bağlamda RNN'lerin popüler türlerinden olan ve başarısı kanıtlanmış LSTM ağları karmaşık zaman serisi analizinde ve anomali tespitinde yüksek doğruluk oranları sağlamaktadır (Akkuzu Kaya & Yıldız, 2023).

## **1.2. Çalışmanın Amacı ve Hedefleri**

Çalışmanın amacı, derin öğrenme tabanlı LSTM ağlarıyla zaman serisi analizi yaparak, fırın sensör verilerinde anomali tespiti yapmaktır. Bu doğrultuda tahminsel bakım için bir dayanak oluşturulması ve olası arızaların önceden tespit edilerek, engellenmesi amaçlanmaktadır. İşletmenin öngörülemeyen arızalardan kaynaklı ekonomik kayıplarının minimize edilmesi ve kesintisiz çalışmanın sağlanması da önemli bir boyuttur.

Bu çalışmada Türkiye'de fotovoltaik panel fabrikası olarak üretim yapan ve ayrıca ileri teknoloji üretmek için Araştırma-Geliştirme (Ar-Ge) merkezini de içinde bulunduran Kalyon Fotovoltaik (PV) Güneş Teknolojileri Fabrikası'ndaki üretim hattında yer alan silisyum eritme ve kristal büyütme fırını üzerinde verilere dayalı tahmin yapılmıştır. Fırın üzerindeki sensörlerden toplanan verilere keşifsel veri analizi ve veri ön işleme işlemleri uygulanmıştır. Bu işlemlerden sonra veri seti zaman serisi verilerine dönüştürülmüştür. Böylece gerçek zamanlı veri seti LSTM modelinin eğitilmesi için hazır hale getirilmiştir.

LSTM modeli, ilgili veri seti ile eğitilmiş ve modelin doğruluğunun artırılması için hiper parametre ayarları yapılmıştır. Modelin doğruluğu, duyarlılığı ve performans metrikleri analiz edilmiştir. Modelin gerçek zamanlı verilerdeki başarısı test edilmiş olup

ayrıca potansiyel iyileştirme yöntemleri uygulanmıştır. Literatürde, zaman serisi analizi ile anomali tespiti, tahminsel bakım için derin öğrenme yöntemlerinin başarısı, endüstride tahminsel bakım için kullanılan yapay zekâ yöntemleri, üzerine yapılan benzer çalışmalar incelenmiştir. Bu tezde ise kompleks ve kritik bir ekipman olan ve ayrıca dinamik ve yoğun veri üreten silisyum eritme ve kristal büyütme fırınının sensör verilerine odaklanılmıştır.

### **1.3. Çalışmanın Önemi, Özgün Değeri ve Katkısı**

Çalışmamızda tasarladığımız LSTM tabanlı modelimiz, zaman serisi analizinde yüksek doğruluk oranı sunmaktadır. Bu yöntem sensörlerden gelen eksik, etiketsiz verilere ve verilerdeki heterojenlik gibi zorluklara çözüm sunmaktadır. Yapay zekâ tabanlı derin öğrenme yöntemlerinin bu kapsamda kullanılması, ileri teknolojinin endüstriyel süreçlere entegre edilmesini kapsamaktadır. Doğru bakım yaklaşımı oluşturmak için uygulanan bu sürecin doğru yönetilmesi, ayrıca iş gücü ve zaman yönetimini iyileştirmenin yanı sıra kaynakların verimli kullanılmasına da katkıda bulunur.

### **1.4. Tez Özeti**

Bu tez 6 bölümde ele alınmıştır. Giriş bölümünde, araştırma probleminden, problemin gerekçesinden, yapılan çalışmanın amacı ve öneminden bahsedilmiştir. Bir sonraki başlık olan kavramsal çerçevede literatür taraması yapılmış ve en genel kapsamdan başlanıp konuya özgü ayrıntılara doğru ilerleyerek bu doğrultudaki kavramsal çerçeve detaylandırılmıştır. Üçüncü bölüm olan materyal ve yöntem kısmında çalışma kapsamındaki tüm bilimsel başlıklar, yöntemler, araçlar, veri seti ve veri ön işleme aşamaları sistematik bir şekilde anlatılmıştır. Dördüncü bölüm deneyler ve bulgular kısmında çalışma boyunca yapılan deneylerden kritik öneme sahip olanlar seçilip sunulmuş ve bulguları değerlendirilmiştir. Beşinci bölümde, elde edilen bulgular doğrultusunda sonuçlar değerlendirilmiştir. Altıncı bölümde ise referanslar verilmiştir.

## 2. KAVRAMSAL ÇERÇEVE

Son yıllarda, güneş enerjisi teknolojilerinin yaygınlaşması, yenilenebilir enerji kaynaklarına küresel geçişi hızlandırmış, güneş çiftliklerini sürdürülebilir enerji çözümleri için kritik bir alt yapı haline getirmiştir (Qureshi vd., 2024). Fotovoltaik sistemlerin benimsenmesinin artışıyla, PV panel üretim süreçlerinin daha güvenilir hale gelmesi önem kazanmıştır. Üretim makinelerindeki hataların en aza indirilmesi ve işletme risklerinin azaltılması hem üretim verimliliği hem de kalite sürekliliği açısından büyük önem taşımaktadır (Güven & Bilhan, 2024).

Endüstri 4.0 olarak da bilinen teknoloji devrimi, işletmeleri geleneksel yöntemlerini değiştirmeye zorlayarak dijitalleşmeye ve otomasyona yönlendirmiştir. İşletmeleri, üretim, yönetim, hizmet ve bakım süreçlerinin her aşamasında son teknolojilerin kullanımına sevk etmiştir. Büyük veri, nesnelerin interneti, siber fiziksel sistemler gibi araçlar dijitalleşme sürecinde büyük gelişmelere olanak sunmaktadır. Bu gelişmeler işletmelerin kapsamlı ve akıllı işletmelere dönüşmesini sağlamaktadır. İşletmelerdeki bileşenler birbirleriyle ve insanlarla gerçek zamanlı iletişim kurabilirler ve böylece büyük miktarda verilerin üretilmesine de katkıda bulunurlar. Veriden tam anlamıyla faydalanabilmek için güçlü araçlarla doğru analiz etmek gereklidir (Arena vd., 2024).

Endüstri 4.0 kavramı döngüsel ekonomiyi destekleme potansiyeli nedeniyle de büyük ilgi görmektedir. Sürdürülebilir kalkınma için önemli bir kavramdır. Döngüsel ekonomi, insanın karar verme sorumluluğunu kaldırarak ve malzeme kaynağı verimliliği ve üretkenliği arttırarak sağlanabilir. Fakat potansiyel arızaların doğru tanımlanmaması, tahmini bakım stratejisi için hala zorluk oluşturmaktadır. Tahmine dayalı bakım, ekipmanların ne zaman değiştirileceğine karar vererek, plansız duruş sürelerini ve malzeme tüketimini azaltarak, malzeme kullanım oranını arttırmakta ve kalan faydalı ömrü uzatmaktadır. Böylece döngüsel ve sürdürülebilir ekonomi desteklenmektedir. Wall Street Journal ve Emerson (2018) tarafından yapılan araştırmaya göre işletmelerdeki yıllık plansız duruş süresinin mali yansımalarının 50 milyar ABD doları olduğu ileri sürülmektedir. Ekipman ve sistem arızası, bu plansız duruşların %42'sinin ana sebebinin oluşturmaktadır. Bu problemi çözmek için önleyici bakım önlemleri alınmıştır. Önleyici bakımda bileşenler zamanından önce değiştirilir. Bu da firma için gereksiz malzeme

israfına ve maliyete sebep olur. Bir şirket bileşenin durumunu iyi analiz ederek ve bakım stratejisini optimize ederek önleyici bakım maliyetini %50 azaltabilir (Li vd., 2024).

Tahminsel bakım, geleneksel bakım yöntemlerinin sınırlarını aşmak için stratejik bir çözüm olarak ortaya çıkmıştır. Bu yöntem tarihsel verilere ve makine sensör verilerinden gelen gerçek zamanlı bilgilere dayanır. Potansiyel arızaları önceden tespit etmeyi ve arızaya zamanında müdahaleyi hedefler. Böylece öngörülmeven arızaları önler, iş duruş sürelerini minimize eder, kaynak kullanımını ve ekipman varlık ömrünü optimize eder (Mohan Raparthy, 2023) (Qureshi vd., 2024). Tahminsel bakım makinenin arızalanma olasılığını tahmin etmek için sensör verilerini, veri analitiğini ve tahminsel modellemeyi kullanır (Arena vd., 2024).

Tahminsel bakım ayrıca Endüstri 4.0 kapsamında üreticiler için en önemli ve öncelikli konuların başında yer alır. Kritik ekipmanların arızası üretim süreçlerinde en önemli risklerden biri olarak görülür. Bu sebeple üreticiler bakım faaliyetlerini zorunluluk olarak değil, strateji gerektiren iş akış süresi olarak değerlendirmeye başlamıştır. Son zamanlarda “kırılana kadar çalıştır” şeklindeki maliyetli yaklaşımları veya “nasıl olursa olsun tamir et” şeklindeki verimsiz bakım stratejileri terk edilmektedir. Bunun yansira tahminsel bakım Endüstri 4.0’ın karışık ve değişken üretim sürecinde pratik olarak uygulanabilirliği bakımından çeşitli zorluklarla karşı karşıyadır (Bousdekis vd., 2020).

Endüstriyel süreçlerin otomatik izlenmesi, anormal durumların hızlı bir şekilde tespit edilmesini sağlar. Büyük verilerdeki karmaşık desenleri tanımlayabilen derin öğrenme bu süreçte etkin rol oynar. Standart derin öğrenme yöntemleri belirli bir veri tipine dayalı belirli görevleri çözmek için uygundur ve ayrıca eğitim esnasında büyük miktarlarda etiketli veri gerektirir. Fakat endüstriyel süreçlerin ve çevresel koşulların devamlı değişen yapısı, her olay için büyük ve etiketli veri elde etmeyi zor ve maliyetli hale getirmektedir (Yan vd., 2024).

Çalışmalar, tahminsel bakımda temel zorlukların verilerden etkili ve uygun özelliklerin nasıl çıkarılacağı, arızaların hatasız bir biçimde nasıl tanımlanıp sınıflandırılacağı ve potansiyel arızaların ve bozulmaların nasıl titiz bir şekilde tahmin edileceği olduğunu ifade etmektedir. Ayrıca uygun şekilde arıza tanımlama ve tahmin

sistemi oluşturmak için alanında uzman mühendisler ve kapsamlı bir alan bilgisi gereklidir. Veri analizinde son derece popüler olan derin öğrenme ağları, çoklu özellik destekleri ile uzmanların verileri manuel olarak yorumlamasının önüne geçerek otomatik veri yorumlamasını sağlar (Li vd., 2024).

Anomali veya diğer bir deyişle aykırı değer, literatüre göre diğer gözlemlerden belirli ölçüde farklılık gösteren ve muhtemelen farklı bir süreç sonucunda oluştuğu düşünülen bir gözlem verisidir. Bu anomaliler iki temel faktörle ortaya çıkabilir. Bunlar iç faktörler ve dış faktörlerdir. İç faktörler sensör arızası, kalibrasyon hatası, ısıtma elemanlarının aşırı ısınması ve arızalanması, sıcaklık ve hava akışı dengesizlikleri, yanlış veya tutarsız enerji beslemesi, sensör verilerinin yanlış işlenmesi, yazılım hataları şeklinde, dış faktörler ise çevresel koşullar, insan hataları, enerji kesintileri şeklinde sıralanabilir (Chalapathy & Chawla, 2019).

Zaman serisi verileri çoğunlukla finans, sağlık, üretim ve siber güvenlik endüstrilerinde kullanılan zamana bağlı verilerin kümesidir. Zaman serisi tahmini önceki gözlemlere dayanarak gelecekteki verileri ve olayları tahmin etmeye çalışır. Anomali tespiti ise verinin içerisinde beklenmedik sıra dışı, normalden farklı değerleri bulmaya odaklanır. Bu veriler sayısal, kategorik veya bir kombinasyon olabilir. Örneğin bir zaman serisi yaptığımız çalışmaya bağlı olarak saniyelerden yıllara kadar değişen sabit veya değişen bir gözlem dizisidir. Zaman serisi verileri trendleri, desenleri veya anormallikleri yakalamak için incelenir. Bu verilerin gelecekteki değerleri tahmin edilebilir veya veri dağılımındaki farklılıkları yakalanabilir. Zaman serisi verileri zamanla değişir ve ayrıca istatistiksel ve makine öğrenimi teknikleriyle analiz edilebilir (Iqbal & Amin, 2024).

Zaman serisi anomalilerini tespit etmek zordur. Bunun sebebi verilerin karmaşıklığı, endüstriyel süreçlerin otomasyon seviyesi ve tek değişkenli zaman serisinin herhangi bir endüstriyel süreci temsil etmedeki yetersizliğidir. Bu durum, tüm sürece hâkim olmak için daha fazla sensör kurulmasını ve çok değişkenli zaman serilerinden anomali tespiti yapmayı gerekli kılar. Endüstriyel süreçler sıcaklık, basınç ve nem gibi pek çok koşula bağlı olarak dinamik değişkenlik gösterir. Bu koşullar da verilerde dalgalanmalara sebep olabilir. Ayrıca anomali tespitini ve endüstriyel süreç üzerindeki kontrolü sürdürmeyi zorlaştırır (Yan vd., 2024).

Zaman serisi anomali tespiti, tarihsel verileri analiz etmekte ve yorumlamakta istatistiksel teknikleri kullanır. Anomali tespitinin iki temel yolu vardır. Bunlar denetimli anomali tespiti ve denetimsiz anomali tespitidir. Denetimli anomali tespitinde her zaman serisi için etiket oluşturmak gereklidir. İkili sınıflandırıcı şeklinde olan bu etiketler, normal/anormal şeklindedir. Denetimsiz anomali tespiti için ise sadece normal verilere bağlı anormallik puanı veya eşik değeri, anormal ve normal örnekleri ayırt etmek için kullanılabilir (Yan vd., 2024).

Verilerde etiketlenmiş anomaliler kısıtlıdır. Çünkü pratikte nadir olarak oluşurlar ve verilerin doğru etiketlenmesi için alan uzmanları gereklidir. Bu sebeple zaman serisi anomali tespiti çoğunlukla denetimsiz öğrenme modelleriyle formülize edilir ve uzun süreli bağımlılıklara dayanır (Choi vd., 2023).

## **2.1. Literatür Özeti**

Anomali tespiti, verinin yoğunluğunu modelleyerek bilinmeyen desenleri tespit edebilen ve aykırı değerleri ortaya çıkarabilen denetimsiz öğrenme yöntemidir. Anomali tespitinde 3 temel kategori vardır. Bunlar; mesafe tabanlı, kümeleme tabanlı ve istatistiksel metotlardır. Örneğin Ana Bileşen Analizi (PCA), önceden belirlenmiş güven eşiğine dayanan istatistiksel bir metottur. K-En Yakın Komşu (KNN) algoritması, her verinin anomali skorunu, en yakın komşusuna olan mesafe ortalamasını baz alarak hesaplayan mesafe tabanlı bir yöntemdir. Kümeleme tabanlı metotlar ise farklı veri kümelerini modeller ve önceden belirlenmiş aykırı değer skoru ile anomaliyi tespit eder. Fakat bu metotlar zamansal bağılıkları algılayamadıkları için çok değişkenli zaman serisi verilerinde yeterince başarılı olmayabilir. Bu problemin üstesinden gelmek için çok sayıda derin öğrenme tabanlı modeller oluşturulmuştur. İlk çalışmalarda çok yüksek boyutlu veri kümelerini analiz etmek için boyut indirgeme ve yoğunluk tahmini yöntemleri kullanılmıştır. Fakat bu yöntemlerde önemli bilgi kayıpları yaşanmıştır. Sonraki aşamalarda ise bilgi kaybını engellemek için önceden eğitilmiş modeller Visual Geometry Group (VGG), Residual Networks (ResNet) ve Autoencoder (AE) kullanılmıştır. Bu modeller, verinin yoğunluğunu modelleyerek daha başarılı anomali tespitleri yapmıştır. LSTM kullanılarak yapılan zaman serisi verilerinde anomali tespiti ise verilerdeki uzun süreli bağılıkları ve diziler içindeki bağlamsal bilgiyi algılama yetenekleri sayesinde öne çıkan yaklaşımlardan biridir. (Joung vd., 2024).

Bu tez kapsamında literatür taraması yapılırken Google Scholar, Scopus, Elsevier, IEEE, SpringerLink, Science Direct veritabanlarında anahtar kelimeler ile ilgili yayınlar filtrelenmiştir.

### **3. MATERYAL VE YÖNTEM**

#### **3.1. Çalışma Tasarımı**

Bu çalışmada PV panel üretim fabrikasındaki fırınlara yönelik özelleştirilmiş bir tahminsel bakım geliştirmek için, iki güçlü analitik yaklaşıma odaklanılmıştır. Zaman serisi analizi ve derin öğrenme. Zaman serisi analizi, sıralı verilerdeki zamansal bağlantıları ve eğilimleri kavrama yeteneğiyle makinelerin zaman içinde oluşan davranışını analiz etmekte oldukça başarılıdır. Derin öğrenme kapsamındaki LSTM ağları ise zaman serisi verilerindeki karmaşık ilişkileri öğrenme konusunda başarılıdır. Çalışmanın metodolojisi, veri toplama, keşifsel veri analizi, veri ön işleme, model geliştirme ve performans değerlendirmeyi içerir.

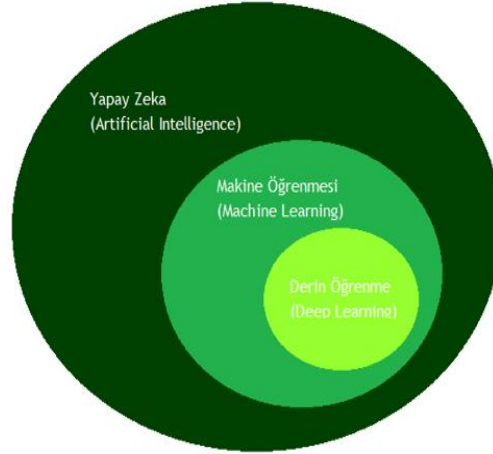
Zaman serisi anomali tespiti uygulamalarında model mimarisi seçimi son derece kritiktir. Eğitim sürecini kolaylaştırmak için zamansal bağımlılıkları algılayan ve değişik zaman ölçeklerindeki çözümleri tanıyabilen model mimarileri tercih edilir. CNN, LSTM ve Otomatik Kodlayıcılar bu mimarilerden popüler olanlarıdır. Bu modeller eğitildikleri veri ile alakalı belirli türdeki desenlere ve çözümlere öncelik verme eğilimindedirler. Model sonsuz sayıdaki olası çözümlerden birini öğrenmek zorundadır. Bu sebeple induktif ön yargılar, bu olası çözüm sayısı skalasını daraltır ve modelin daha etkin öğrenmesini sağlar. LSTM'ler kuvvetli induktif ön yargılara sahiptir ve daha az veriye ihtiyaç duyarlar. CNN ve LSTM ağları Otomatik Kodlayıcı olarak da kullanılabilirler. Böylece yüksek boyutlu zaman serilerini daha basite indirgeyerek anomali tespitini kolaylaştırabilirler (Yan vd., 2024).

#### **3.2. Materyal**

##### **3.2.1. Yapay zekâ (Artificial Intelligence)**

"Yapay zekâ, makineleri zeki hale getirme faaliyetidir; zekâ ise bir varlığın çevresinde uygun şekilde ve ileri görüşlülükle hareket etmesini sağlayan bir niteliktir."

Nils J. Nilsson tarafından yapılan bu tanım yapay zekânın tanımına faydalı bir bakış açısı oluşturmuştur (*ai100report10032016fnl\_singles*, t.y.) .



**Şekil 3. 1.** Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki hiyerarşik ilişki

### 3.2.2. Makine öğrenimi

Makine öğrenimi (ML), çevresel etkenlerden öğrenerek insan zekâsını taklit etmeyi hedefleyen hesaplama algoritmalarının evrimleşen bir dalıdır. Büyük veri çağı olarak adlandırılan bu çağda makine öğrenimi yöntemleri desen tanıma, bilgisayarlı görüde, uzay aracı mühendisliğinde, finans, eğlence, hesaplamalı biyoloji, biyomedikal ve tıbbi uygulamalar gibi pek çok alanda başarıyla uygulanmaktadır (El Naqa & Murphy, 2015).

Makine öğrenmesi, bilgisayar bilimlerinin bir dalı olup, bilgisayarların programlanmadan öğrenmesini sağlamayı hedefler. 1950’lerin yapay zekâ olgusunu temel alan bu alan özellikle tahmin ve optimizasyon araçlarına ve uygulamalarına odaklanır. Bilgisayarlar “tecrübe” yoluyla görevlerdeki performanslarını geliştirirler. Bu durum, verilere uyum sağlama anlamına gelir (Bi vd., 2019).

ML, açık komutların yerine, verilerdeki desenleri ve çıkarımları kullanarak istenilen görevleri yerine getirmek için istatistiksel modelleme yöntemlerini inceleyen bir alandır. Bilgisayarların belirli bir konuda öğrenmesini sağlar. ML, algoritmaları yapay zekânın bir alt kümesidir. Eğitim ve test verilerinden matematiksel bir model oluşturarak tahminlerde bulunmaya ve kararlar vermeye imkân sunar. Geniş algoritma seçeneği vardır ve kullanılan öğrenme şekline göre farklı kategorilere ayrılır (Shaala vd., t.y.).

En yaygın olarak kullanılan ML yöntemleri 3 temel gruba ayrılır. Bunlar; denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenmedir. Denetimli öğrenme, algoritmaları, verileri doğru şekilde sınıflandırmak ve sonuçları tanımlamak için etiketli veri kümelerini kullanarak eğitir. Denetimsiz öğrenme, etiketlenmemiş verileri inceleyerek aralarındaki deseni bulmayı amaçlar. Pekiştirmeli öğrenme ise, insanın öğrenme şekline benzerdir. Algoritma, çevresindekilerle etkileşime girerek gerçek zamanlı aldığı pozitif veya negatif bildirimler yoluyla öğrenir. Bu yöntem özellikle devamlı geri bildirim ve adaptasyonu gerekli kılan dinamik süreçlerde etkindir (Srisuwan & Innet, 2024).

### **3.2.3. Derin öğrenme**

Derin öğrenme, geleneksel yapay sinir ağlarından (Artificial Neural Networks) türetilen ve ML alanında üstün performanslar sergileyen algoritmalarından biridir. Dönüşüm ve grafik teknolojilerini eş zamanlı kullanarak çok katmanlı öğrenme modelleri oluşturur. Ses, konuşma işleme, görsel veri işleme, doğal dil işleme gibi alanlarda son zamanlarda dikkat çeken başarılar elde etmiştir. Derin öğrenme algoritmaları aracılığıyla özellik çıkarımı otomatik olarak gerçekleştirilir. Böylece araştırmacılar için en az insan çabası ve alan bilgisi ile ayırt edici özelliklerin çıkarılmasını kolaylaştırır. Derin öğrenme algoritmaları çok katmanlı bir veri temsil mimarisine sahiptir. Başlangıçtaki katmanlar düşük seviyeli özellikleri çıkarırken, son katmanlar yüksek seviyeli özellikleri çıkarır. Bu yapı, başlangıçta insan beynindeki temel duyu alanlarında gerçekleşen süreci modelleyen yapay zekâdan esinlenmiştir. İnsan beyni farklı görsellerden, olaylardan otomatik olarak veri çıkarabilir. Bu veriler, girdiyi temsil ederken, süreç sonunda nesnelere sınıflandırılması ise çıktıyı ifade eder. Kısacası derin öğrenme algoritmaları insan beyninin çalışmasını taklit etmeye çalışır. Bu durum derin öğrenmenin temel avantajlarından birini öne çıkarır (Alzubaidi vd., 2021).

#### **3.2.3.1. Derin öğrenme ağları**

##### **3.2.3.1.1. Convolutional Neural Networks (CNN)**

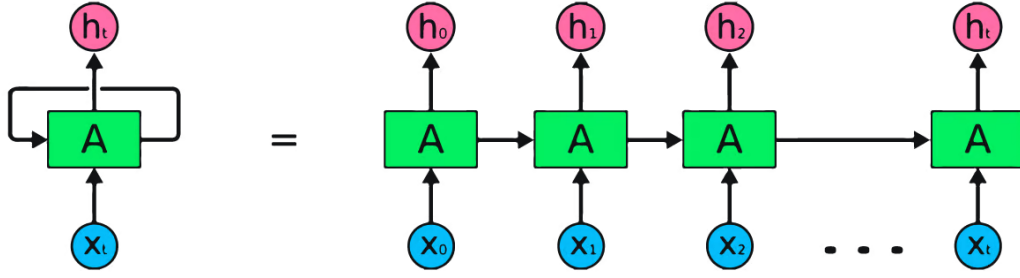
Derin Öğrenme ağlarından Convolutional Neural Networks (CNN) 'ler görüntü ve video analizi gibi alanlarda oldukça yaygın olarak kullanılan yöntemlerdir. CNN'ler insanın kolaylıkla yakalayamayacağı, karmaşık desenleri algılayabilirler. Ayrıca büyük

miktardaki verileri yönetebildikleri için sensör verilerinin yoğunlukla üretildiği endüstriyel uygulamalar için kullanımı uygundur. Fakat CNN'ler etiketli verilere ihtiyaç duyarlar ve veri homojen ve çok özellikli olduğu zaman, karmaşık verileri etkin şekilde analiz etmekte zorlanırlar. Bunun yanı sıra veri dizisinde bir adımdan sonraki adıma bilgi taşıma yeteneğine sahip olmadıkları için sıralı verileri işleme konusunda yetersizlerdir. Bu tür durumlar için Recurrent Neural Networks (RNNs) daha etkili bir yöntemdir (Bampoula vd., 2024).

### 3.2.3.1.2. Recurrent Neural Networks (RNN)

Sıralı veriler üzerinde çalışabilmek için optimize edilmiş bir derin öğrenme tekniğidir. Konuşma tanıma, doğal dil işleme ve zaman serisi tahmini gibi görevlerde yaygın olarak kullanılır. Geri bildirim döngüleri vasıtasıyla RNN'ler, zaman adımları boyunca bilgi akışını sağlayarak önceki birimlerin bilgisini hatırlayabilirler. Fakat uzun vadeli bağımlılıkları devam ettirmekte zorlanırlar. Giriş dizisinin boyutu arttıkça yöntemin doğruluğu zamanla azalabilir. Bu durum onları gerçek zamanlı tahminler için yetersiz kılar. Araştırmacılar bu problemlere çözüm bulmak için RNN'lerin varyantları olan LSTM ağlarını geliştirmişlerdir. LSTM ağları uzun boyutlu diziler için bilgiyi koruyarak ve bağımlılıkların daha etkin şekilde modellenmesine olanak sağlayarak RNN'lerin tahminsel bakımda daha verimli kullanılmasını mümkün kılmıştır (Bampoula vd., 2024).

RNN'ler, geleneksel sinir ağlarının geçmiş verilerle alakalı kısıtlı bellek problemini çözmek için geliştirilmiştir. Geleneksel sinir ağlarının hafızası, sadece eğitim esnasında öğrenilen bilgilerle sınırlıdır ve tahmin esnasında sadece mevcut giriş verileri baz alırlar. Bu durum daha önce görülen verilerden bilgi çıkarmayı zorlaştırır. Bu problemi çözmek için RNN'lerdeki nöronlar, ağın ileriki adımlarında kullanılacak gizli bir değişken üretirler. Bu değişken, bilginin ağ içerisinde kalmasına olanak tanır. Böylece RNN'ler, geleneksel sinir ağlarına göre zaman serisi verilerini tahmin etmede daha yetkin hale gelirler (*Contents*, t.y.).



Şekil 3. 2. Recurrent Neural Networks (RNN) Yapısı

### 3.2.3.1.3. Transformers

Transformer'lar kendi kendine dikkat (self-attention) mekanizmasını kullanarak ardışık verilerdeki bağılıkları yakalayan derin sinir ağları (DNN) türlerinden biridir. Sıralı verilerle yapılan görevlerde, giriş verileri arasındaki bağımlılıkları yönetme işleminde ve paralel işlemlerde oldukça başarılıdır. Kendi kendine dikkat mekanizmasını kullanarak verilerden özellikleri belirler ve doğal dil işleme alanında güçlü bir potansiyel sergiler. Geleneksel tekrarlama yöntemlerinin aksine, Transformer'lar dikkat mekanizmasını kullanarak sıralamanın tamamından öğrenirler. Bağlamın gerçek anlamını yakalayabilirler. Paralel çalışabildikleri için hızlıdırlar. Grafik işlem birimi(GPU) kullanarak yüksek boyutlu veri kümeleri üzerinde hızlı hesaplamalar yapabilirler (Islam vd., 2024) .

### 3.2.3.1.4. Generative Adversial Networks (GAN)

Generative Adversial Networks (GAN)'lar, oyun teorisinden esinlenerek oluşturulmuş bir yapay zekâ modelidir. Üretici(generator) ve ayırt edici(discriminator) olmak üzere birbiriyle rekabet halinde olan iki ağdan oluşur. Bu ağların amacı Nash denklemini sağlayabilmektir. Üretici ağın ana prensibi, gerçek verilerin potansiyel dağılımına uyacak şekilde yapay veriler üretmektir. Ayırt edici ağ ise bu yapay verileri, gerçek verilerden ayırt etmeye çalışır. Eğer ayırt edici ağ, gelen verilerin gerçek veri kümesinden mi yoksa sahte verilerden mi geldiğini ayırt edemezse üretici ağ başarılı bir modele dönüşmüş demektir. Üretici ağ gerçek verilerin dağılımını öğrenmiştir ve en iyi durum sağlanmış olur (Pan vd., 2019).

### 3.2.3.1.5. Autoencoders (AE)

Autoencoder birbirine bağlı iki ağdan oluşan bir sinir ağıdır. Bu ağlardan biri kodlayıcı(encoder) diğeri ise çözücü(decoder) olarak çalışır. Kodlayıcı ağı girdi verisini alarak daha küçük ve yoğun bir örneğe çevirir, çözücü ise bu örneği alır ve ilk girdiyeye

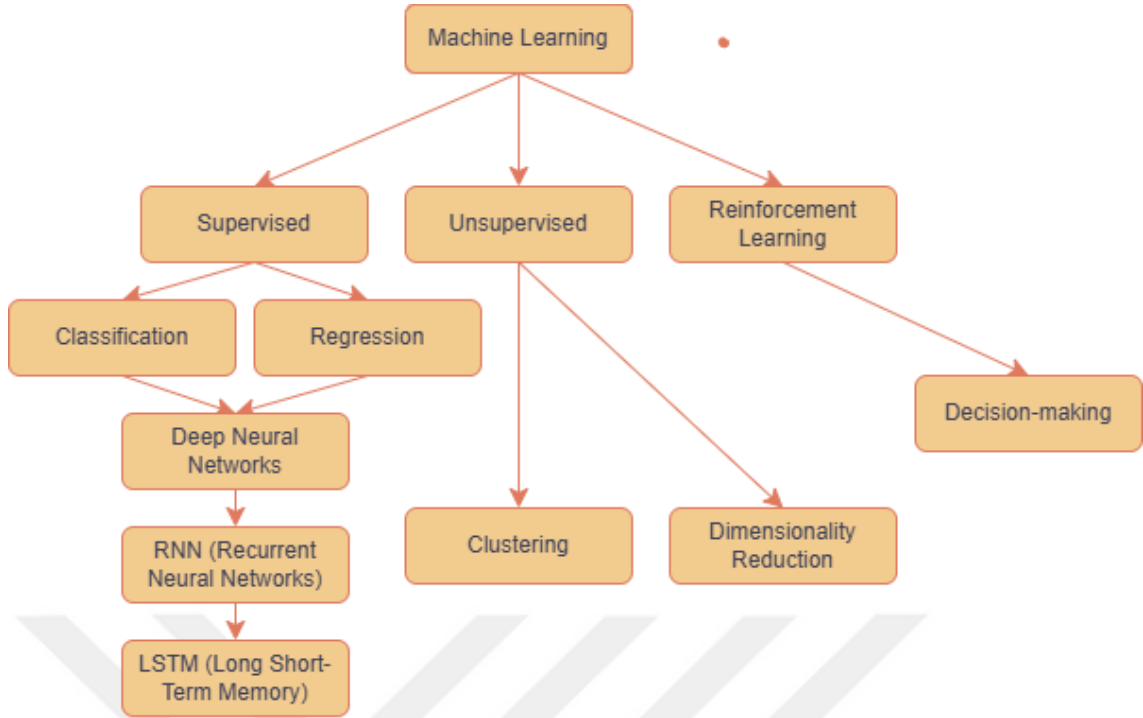
dönüştürmek için kullanır. Ana prensip olarak Autoencoder 'lar ağıın giriş ve çıkışlarını en doğru şekilde temsil etmeyi hedeflerler. Bu sürece veriyi sıkıştırma ve sonrasında çözme süreci de denilebilir. Autoencoder, girdi verilerini soyut olarak nasıl küçülteceğini ve sonrasında gerçek halleriyle nasıl geri alacağını öğrenir. En nihayetinde Autoencoder orijinal problemin boyutunu sıkıştırmayı ve gürültülü verileri temizlemeyi öğrenir. Mimari olarak ise Autoencoder'lar ileri beslemeli ve tekrarlanmayan sinir ağılarıdır. Giriş katmanı, çıkış katmanı ve çok sayıda gizli katmandan oluşur. Çıkış katmanının girdiyi en iyi şekilde yeniden oluşturabilmesi için, giriş katmanı ile aynı sayıda düğümü içermesi gerekir. Eğitim için etiketli verileri gerektirmeyen, denetimsiz bir derin öğrenme modelidir. Autoencoderlar, her daim encoder ve decoder şeklinde iki ağıdan oluşurlar (Hu & Dai, 2022).

#### **3.2.3.1.6. Graph Neural Networks (GNN)**

Graph Neural Networks (GNN)'ler, grafik verileri üzerine çalışan bir çeşit derin öğrenme modelidir. Grafikler üzerindeki verileri başarılı bir şekilde işleme yeteneklerine sahiptirler. Son zamanlarda CNN'ler, öklidyen veri dediğimiz görsel ve metin verileri üzerinde başarılarını kanıtladıkları için GNN'lerin gelişimine ilham kaynağı olmuşlardır. CNN'ler düzenli verileri işlemede güçlü yeteneğe sahiptirler fakat doğrudan öklidyen olmayan, grafik verileri gibi verilere uygulanamazlar. Grafik verilerindeki yerel filtreler ve havuzlama işlemleri CNN'lerin öklidyen olmayan verilere uygulanmasında zorluk oluştururlar. Bu problemi çözmek için grafiklerde, derin öğrenme modellerini genelleştirmeyi hedefleyen "Geometrik Derin Öğrenme" ortaya çıkmıştır. GNN'ler grafik verilerinde derin öğrenme modellerini geliştirmenin bir temsilidir (Zhou vd., 2020).

#### **3.2.3.1.7. Long Short-Term Memory (LSTM)**

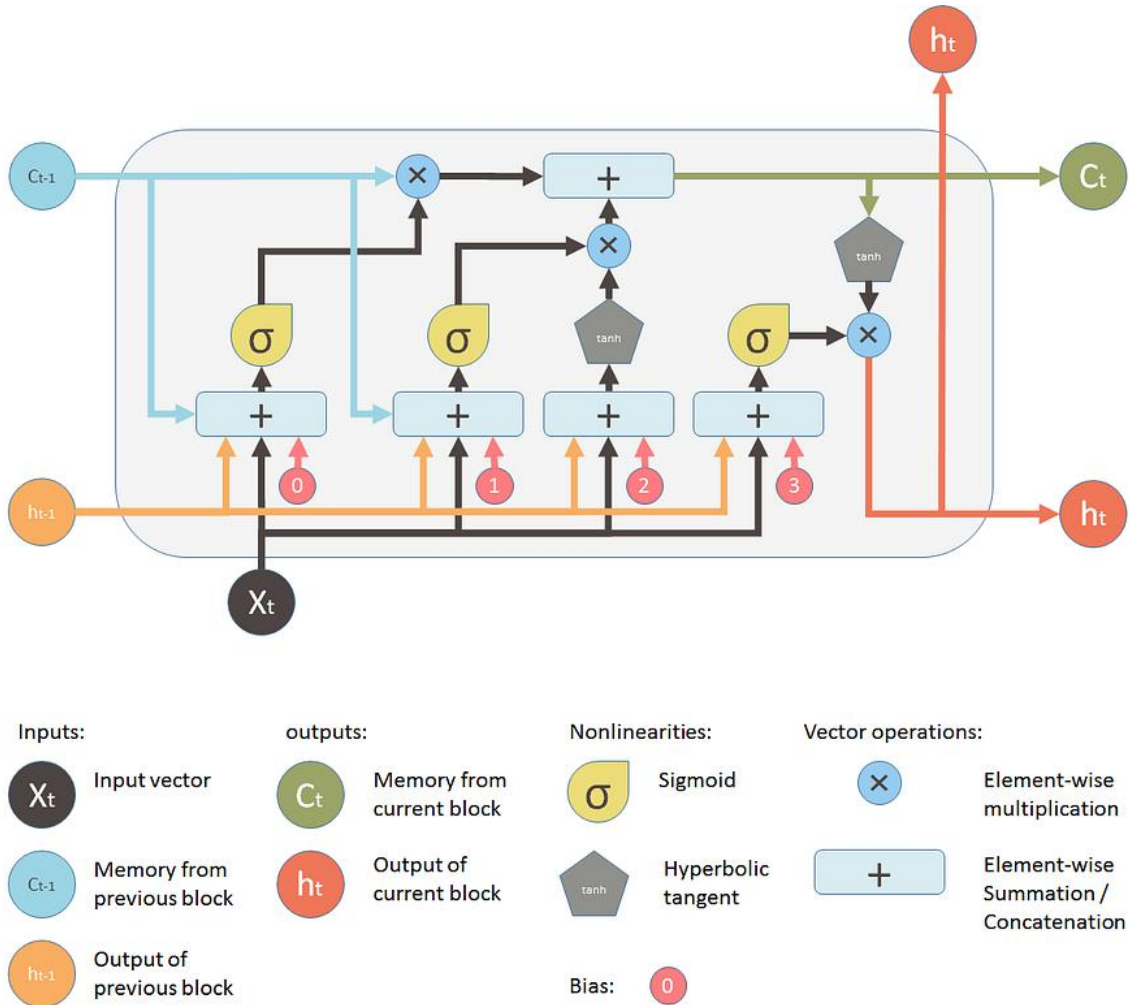
Uzun kısa süreli bellek ağıları, RNN mimarisi baz alınarak tasarlanmıştır. RNN tabanlı olan bu yapı uzun kısa süreli hafıza hücrelerine sahiptir. LSTM hücresi birden fazla dahili ağırlık parametresi içerir ve geçmiş çıktılardan gelen kayda değer bilgileri algılayıp tutmak için spesifik olarak geliştirilmiştir (*Contents*, t.y.).



**Şekil 3. 3.** LSTM ağının makine öğrenmesi mimarisindeki yeri

LSTM hücresi, geleneksel RNN’lerde görülen ve uzun süreli bağımlılıkları kavramayı zorlaştıran gradyan kaybolma (vanishing-gradient) sorununu ortadan kaldırmak için (Hochreiter ve Schmidhuber, 1997) tarafından tasarlanmıştır. Gradyan kaybolma sorunu, geleneksel RNN hücrelerinin öğrenme aşamasında değişmeyi bırakması olayını kapsar. Bu durum güncel bilgilerin öncelikli olarak ele alınmasına ve eski bilgilerin göz ardı edilmesine sebep olabilir. Bu sebeple uzun süreli bağımlılıklar başarılı bir şekilde öğrenilemez. LSTM, nöronlar arasındaki bütün bilgi akışını yönetmek amacıyla geliştirilmiştir. Bu işlem için, hücre durumundan (cell state) adım adım iletilen bilginin eklenmesi ve silinmesi işlemlerini yöneten bir kapı mekanizmasına sahiptir. Bu mekanizma, unutma sürecini etkin şekilde kontrol edebilir ve hem kısa süreli hem de uzun süreli bağımlılıkların daha başarılı modellenmesini sağlar. LSTM’in RNN’den farkı, LSTM’de çıktı sadece önceki çıktılar veya sadece girişlerden türetilmez. Bunun yerine hücre durumuna dayanır. Hücre durumu (cell state), unutma kapısı (forget gate) ve ekleme kapısı (add gate) tarafından adım adım güncellenir. İnsan hafızasına benzemeye çalışan bu yapı, insan hafızasının sınırlı kapasiteye sahip olduğunu temel alır. Bu doğrultuda süreç boyunca her yinelemede bilgi eklenir ve silinir. Çıkış kapısı, güncellenmiş hücre durumunu (cell state) baz alarak istenilen çıktıyı üretir. Böylece geçmişteki önemli olayların ağırlığı korunurken, düşük bilgi yoğunluğu olan güncel girdilerin göz ardı

edilmesi de mümkün olur. LSTM hücresi pek çok farklı sinir ağı mimarisiyle birleştirilebilir (Lindemann vd., 2021).



Şekil 3. 4. Önerilen çalışma için LSTM ağının yapısı

### 3.2.3.2. Derin öğrenme araçları

Derin öğrenme, makine öğrenmesinin bir alt başlığı olarak, yüksek boyutlu verileri işleme ve bu verilerdeki karmaşık desenleri öğrenme konusunda başarılıdır. Bilgisayar görüşü, doğal dil işleme ve sağlık alanlarında yapılan uygulamalarla başarısını kanıtlamıştır. Derin öğrenme alanındaki başarının temeli, sinir ağlarının kolaylıkla oluşturulmasını, eğitilmesini ve dağıtımını kolaylaştıran güçlü ve yenilikçi çerçevelerin oluşturulmasıdır. Keras ve TensorFlow sıklıkla kullanılan derin öğrenme kütüphanelerindedir. Keras başlangıçta sinir ağlarının modellenmesi için tasarlanmış, basitliği ve kolaylığı ile bilinen yüksek seviyeli bir API'dir. Araştırmacılar ve geliştiriciler, Keras ile en az kod ile hızlı prototipler oluşturabilir ve karmaşık modeller inşa edebilirler.

Keras, TensorFlow, Theano ve Microsoft Cognitive Toolkit (CNTK) gibi yapıları desteklemektedir. TensorFlow ise Google Brain ekibi tarafından geliştirilmiş olup, makine öğrenimi modelleri oluşturmak ve dağıtmak için gelişmiş araçlar sunan güçlü ve esnek bir çerçevedir. TensorFlow hem yüksek seviyeli hem de düşük seviyeli API'ler sağlayarak geniş bir uygulama alanını kapsar. TensorFlow ekosistemi, TensorFlow Extended (TFX), TensorFlow Lite ve TensorFlow Serving gibi araçlardan oluşur ve modelin eğitimi, optimizasyonu ve dağıtımında önemli destek sunar. (Medavarapu, 2024).

### **3.2.3.3. Platformlar**

. Colaboratory, toplulukların ML eğitimi ve araştırmalarını yaygınlaştırmayı amaçlayan bir yapıdır. Colaboratory, Jupyter Notebooks tabanlı bir yapıya sahiptir ve Google Docs gibi paylaşılabilir bir formda çalışır. Kullanıcılar aynı not defteri üzerinde eş zamanlı düzenlemeler yapabilir. Colaboratory Python 2 ve 3 için çalışma ortamları sunar. Bu ortamlar popüler makine öğrenimi ve yapay zekâ kütüphaneleri olan TensorFlow, Matplotlib ve Keras'ı içinde bulundurur. Çalışma zamanı altında bulunan sanal makine (VM), belirli bir süre sonra devre dışı bırakılır. Bu durum kullanıcının verilerinin ve yapılandırmalarının kaybolmasına sebep olur fakat not defteri korunur. Bunun yanısıra sanal makinenin sabit diskinden kullanıcının Google Drive hesabına dosya aktarımı yapabilir. Ayrıca Colaboratory, tamamen yapılandırılmış bir GPU hızlandırılmış çalışma zamanı da sağlayarak, kullanıcılara yüksek performanslı işlem gücü sunar. Google Colaboratory altyapısı, Google Cloud platformu üzerinde tutulmaktadır(Carneiro vd., 2018).

### **3.2.3.4. Donanım araçları**

Bulutta sıklıkla kullanılan donanım ortamları arasında CPU, TPU ve GPU bulunmaktadır. Derin öğrenme modelleri, yüksek miktarda işlem gücü gerektirir ve böyle yoğun hesaplama uygulamalarını hızlandırmak amacıyla özel amaçlı hızlandırıcılar kullanılır. Tensor İşlem Birimleri (TPU'lar) ve Grafik İşlem Birimleri (GPU'lar), derin öğrenme uygulamalarında sıklıkla kullanılan hızlandırıcılardır. Bu hızlandırıcılar paralel işlem yapabilme yetenekleri ve yüksek bellek bant genişlikleri sayesinde, CPU'lara göre çok daha iyi performans gösterebilirler. Colab'ta ek bir yapılandırmaya gerek duymadan TPU'lar ve GPU'lar ücretsiz kullanılabilir (Kimm vd., 2021).

### 3.2. Yöntem

Derin öğrenme, çok katmanlı sinir ağlarını eğiten bir makine öğrenme türü olarak zaman serisi tahminlerinde ve anomali tespitinde kullanılan güçlü bir araçtır. Derin öğrenme modelleri verilerdeki karmaşık ve doğrusal olmayan desenleri öğrenmedeki başarısı sayesinde doğru tahminler yaparak üstün performans gösterirler. İstatistik ve makine öğrenme yöntemleri zaman serisi verilerini incelemede oldukça yaygındır (Iqbal & Amin, 2024).

Bu çalışmada zaman serisi tahmini ve anomali tespiti için derin öğrenme modellerinden LSTM kullanılmıştır. Bu tezdeki çalışmalar **Hata! Başvuru kaynağı bulunamadı.**'de gösterilen yazılım ve donanım sistemi üzerinde gerçekleştirilmiştir. Uygulama Python programlama dili ile yazılmış, kullanılan kütüphaneler Python ortamında çalıştırılmıştır. Kodların implementasyonu, Google Colaboratory (Colab) platformu üzerinde yapılmıştır.

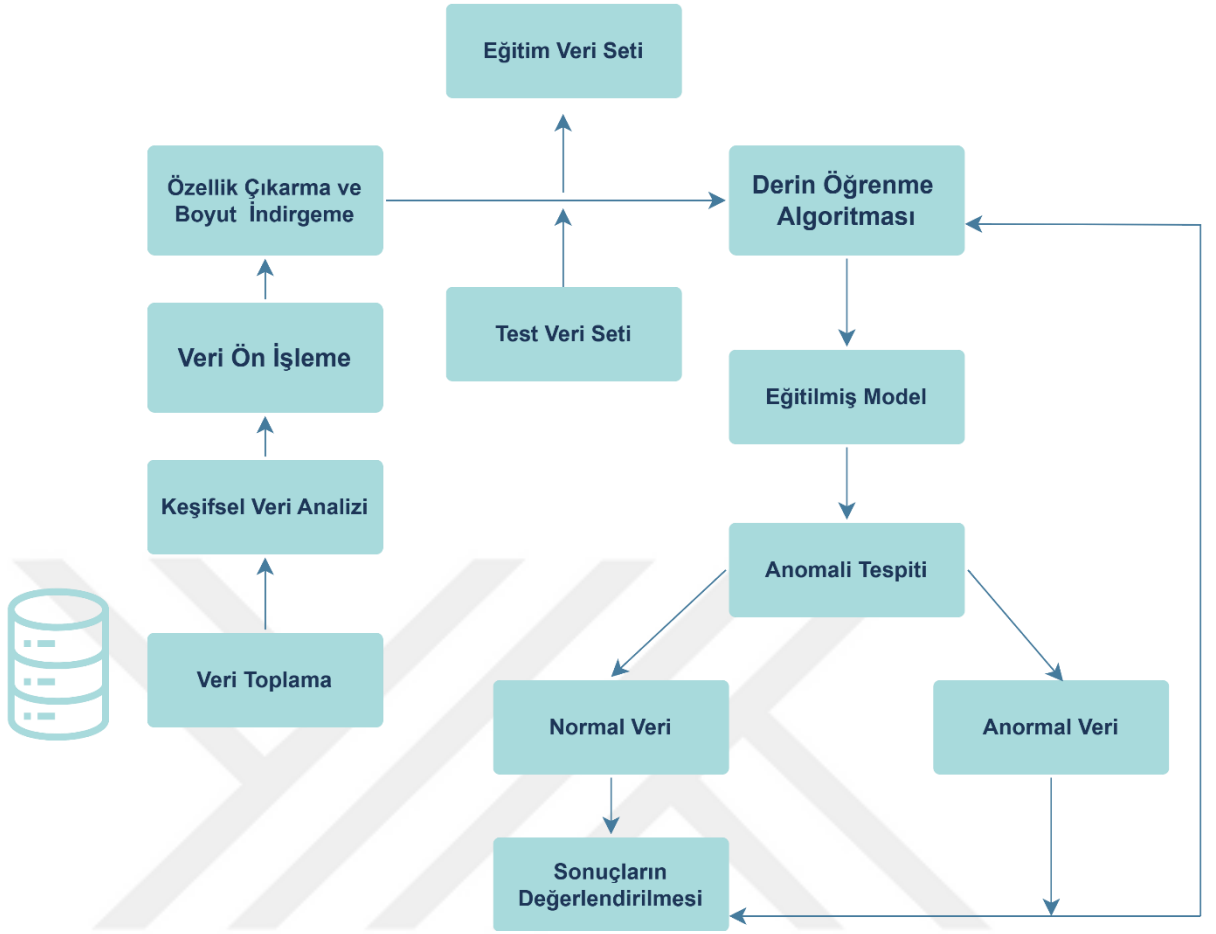
**Tablo 3. 1.** Sistem Özellikleri

Adı	Açıklaması
İşlemci	Intel(R) Core (TM) i5-12450H CPU
Ekran Kartı	Nvidia Tesla K80
İşletim Sistemi	Windows 11
RAM	16 GB
Kullanılan Paketler	Python, Keras, TensorFlow

Çalışmanın temelini oluşturan veri ön işleme, model geliştirme ve değerlendirme işlemleri için Python kütüphaneleri ve araçları kullanılmıştır. Veri ön işleme adımında Scikit-Learn kütüphanesinden MinMaxScaler aracı kullanılarak veri normalizasyonu yapılmıştır.

Keras kütüphanesi, LSTM modelinin oluşturulması ve eğitilmesi için yüksek seviyeli bir API olarak kullanılmıştır. Keras, derin öğrenme mimarilerinin prototiplerini oluşturmak için TensorFlow üzerinde çalışarak büyük modellerin eğitilmesini sağlar. Bu sebeple LSTM katmanının kolaylıkla yapılandırılması ve sıralı verilerin modellenmesi için tercih edilmiştir.

Eğitilen modelin performansı Mean Absolute Error (MAE), Mean Squared Error (MSE) ve R-squared ( $R^2$ ) metrikleri ile değerlendirilmiştir.



Şekil 3. 5. Anomali tespiti için önerilen metodoloji diyagramı

### 3.2.1. Derin Öğrenme ile anomali tespitine yönelik verilerin hazırlanması



Şekil 3. 6. Veri ön işleme altındaki farklı aşamalar/adımlar

#### 3.2.1.1. Sensör verilerinin toplanması ve tanımı

Teknolojik gelişmeler doğrultusunda veri odaklı karar verme, endüstride makinaların bakımının nasıl yapılacağını, performansının nasıl optimize edileceğini, devrim niteliğinde değiştiren önemli bir unsur haline gelmiştir. Ayrıca tahminsel bakım alanında da kritik öneme sahip olmuştur. Veri odaklı karar verme, arızaların ne zaman

yaşanacağını tahmin etmek için gelişmiş analitiklerin, sensör teknolojisinin ve makine öğreniminin gücünden faydalanarak, zamanında ve maliyet olarak etkin bakım müdahalelerine olanak tanır (Kharazian vd., 2024).

Bununla birlikte tahminsel bakımdaki en önemli zorluklardan biri, herkese açık gerçek veri setlerinin eksikliğidir. Bunun sebebi şirketlerin gizlilik politikalarıdır. Gerçek veri setleri güçlü modeller oluşturabilmek için oldukça önemlidir. Gerçek dünya veri setlerinin eksikliği, araştırmacıları simüle veri setlerini kullanmaya mecbur bırakır. Bu durum sentetik verilerin oluşumunu gerekli kılar. Fakat sentetik veriler, gerçek verilerin karmaşıklığından uzaktır. Dengesiz etiketler, sansürlü okumalar, gürültü ve sinyaller arasındaki karmaşık ilişkiler gibi bir çok bağlantıyı gözden geçirir (Kharazian vd., 2024).

Çalışmada kullanılan veriler, Kalyon Fotovoltaik (PV) Güneş Teknolojileri Fabrikası'ndaki üretim hattında yer alan silisyum eritme ve kristal büyütme fırını üzerindeki sensörlerden toplanan gerçek verilerdir. Bu veriler 1 Mayıs 2023 saat 00:08:56 ile 30 Eylül 2023 saat 11:59 PM arasındaki 153 günü kapsamaktadır. 9 dk aralıklarla tutulan veriler mayıs, haziran, temmuz, ağustos ve eylül ayını kapsam üzere 5 farklı Excel dosyasında depolanmıştır. Bu veriler kaydedildiği şekliyle, işlenmemiş, düzenlenmemiş ve filtrelenmemiş olan ham verilerdir. 10 ayrı sensörden toplanan verilerin her bir sensör ismi ve sensörün işlevini ifade eden açıklaması Tablo 3. 2'de verilmiştir.

**Tablo 3. 2.** Fırın sensörlerinden gelen verilerin özellikleri ve tanımları

Tag İsmi	Açıklama
A1	Süreç modeli
A11	Tohum kaldırma gerçek zamanlı değeri
A15	Potanın dönüş hızı gerçek zamanlı değeri
A2	Toplam çalışma süresi (saat)
A21	Ana vakum gerçek zamanlı değeri
A23	Kelebek tampon gerçek zamanlı değeri
A26	Hava akışı gerçek zamanlı değeri
A35	Kristal ağırlığı gerçek zamanlı değeri
A36	Kristal uzunluğu gerçek zamanlı değeri
A39	Erimiş sıcaklık gerçek zamanlı değeri

Bu özelliklerden bir kısmının alabileceği minimum değer ve maksimum değer bilgileri verilmiştir. (Tablo 3. 3)

**Tablo 3. 3.** Fırın sensör verilerinin özellikleri ve minimum, maksimum değerleri

Adı	Veri Tanımlama	Min Değer	Max Değer
A15	Potanın dönüş hızı gerçek zamanlı değeri	0	11
A21	Ana vakum gerçek zamanlı değeri	0	20
A23	Kelebek tampon gerçek zamanlı değeri	0	100
A26	Hava akışı gerçek zamanlı değeri	0	250
A35	Kristal ağırlığı gerçek zamanlı değeri	0	400
A36	Kristal uzunluğu gerçek zamanlı değeri	0	3900
A39	Erimiş sıcaklık gerçek zamanlı değeri	1400	1480

Ignot furnace data limit A1 özelliğine ait numerik değerlerin temsil ettiği işlemler Tablo 3. 4'te gösterilmiştir.

**Tablo 3. 4.** A1 özelliğine ait numerik değerlerin temsil ettiği işlemler

A1	Process model	0	Manual
		1	Pump Down
		2	Leakback
		3	Pressurization
		4	Melting
		5	Dip
		6	STAB TEMP
		7	Neck Growth
		8	Manual Neck
		9	Crown Growth
		10	Should Growth
		11	Body Growth
		12	Tail Growth
		13	Shutdown
		14	Isolation
		15	Purge
		16	Bakeout
17	Oxidation		

		18	Stabilization II
		19	Cooling
		20	Ingot Out
		21	Seed Down
		22	Crystallization
		23	Shoulder Out
		24	Remove Slag
		25	NAC Out
		26	Recharge
		28	Sticky Slag
		30	P/C Right

### 3.2.1.2. Keşifsel Veri Analizi

Veri görselleştirme araçları ile desteklenen keşifsel veri analizi (EDA) veri kümelerini keşfetmek ve analiz etmek için bilim insanları tarafından sıklıkla kullanılır. Modeli oluşturmadan önce EDA'nın sunacağı bilgileri titizlikle analiz etmek büyük önem taşır. Çünkü tabular veriyi analiz ederken önemli veri özelliklerini ayırt etmek zordur. İstatistiksel yaklaşımlar zaman alıcı, sıkıcı veya korkutucu olabilir. Bu sebeplerle EDA önemlidir (Iqbal & Amin, 2024).

Keşifsel veri analizinin temel amacı verinin modelleme için uygun olup olmadığını değerlendirmektir. Bunun için verideki desenleri ve ilişkileri gözlemlemeyi, potansiyel problemleri belirlemeyi kapsar. Matplotlib, Seaborn, Pandas, NumPy kütüphaneleri bu doğrultuda kullanılan Python kütüphaneleridir.

Veri setindeki ham veriler Tablo 3. 5'te gösterilen yapıdaydı. Bu veriler analiz için daha uygun olan Tablo 3. 6'te gösterilen tabular veri formatına dönüştürülmüştür. Tablo 3. 5'te tüm sensörlerin aynı zaman dilimine ait değerleri sırayla gösterilmektedir. Tablo 3. 6, ise her bir zaman dilimindeki tüm sensör değerlerinin tek bir satırda yer aldığı ve analiz için daha uygun olan yapıyı ifade etmektedir. Bu işlem tüm aylara uygulanmıştır.

**Tablo 3. 5.** Ham veri formatı

TagName	Description	DateTime	Value
FurnacePullers55.A1	Process model	5.1.23 12:08 AM	11
FurnacePullers55.A11	Seed lift real-time value	5.1.23 12:08 AM	114,9000015
FurnacePullers55.A15	Crucible rotation real-time value	5.1.23 12:08 AM	7,199999809
FurnacePullers55.A2	Total running time (hour)	5.1.23 12:08 AM	103
FurnacePullers55.A21	Main vacuum real-time value	5.1.23 12:08 AM	12,89999962
FurnacePullers55.A23	Butterfly bumper real-time value	5.1.23 12:08 AM	27,29999924
FurnacePullers55.A26	Air flow real-time value	5.1.23 12:08 AM	70,5
FurnacePullers55.A35	Crystal weight real-time value	5.1.23 12:08 AM	318,6560059
FurnacePullers55.A36	Crystal length real-time value	5.1.23 12:08 AM	3271
FurnacePullers55.A39	Melt temp real-time value	5.1.23 12:08 AM	1408

**Tablo 3. 6.** Verinin yeniden şekillendirilmesi

Datetime	A1	A11	A15	A2	A21	A23	A26	A35	A36	A39
2023-05-01 00:08:56	11	114,9	7,2	103	12,9	27,3	70,5	318,656	3271	1408
2023-05-01 00:17:52	11	93,4	7,2	103	12,9	26,4	70,5	320,315	3287,7	1409
2023-05-01 00:26:47	11	117,7	7,2	103	13	26,3	70,4	321,829	3303,7	1410
2023-05-01 00:35:43	11	117,7	7,1	103	12,9	27,1	70,5	323,433	3320,2	1410
2023-05-01 00:44:39	11	113,2	7,1	103	12,9	26,6	70,5	324,939	3336,5	1410

Veri seti incelendiğinde bazı günlere ait hiç kayıt tutulmadığı tespit edilmiştir. Tablo 3. 7’de küçük bir kesiti gösterilmiştir. Eksik veri (Null) durumları, yaşanan kesintilerden, sensör arızalarından veya veri aktarımındaki problemlerden kaynaklı olabilir. Eksik satırların toplamı tüm veri satırlarıyla kıyaslandığında sayısının nispeten az olması sebebiyle ilgili satırlar tamamıyla silinmiştir. Her bir aya ait veri setine bu işlem uygulanmıştır. Böylece tüm verilerden tamamıyla Null olan satırlar silinmiştir.

**Tablo 3. 7.** Zaman serisi veri setindeki tamamıyla “Null” değere sahip satırların örneği

Datetime	A1	A11	A15	A2	A21	A23	A26	A35	A36	A39
2023-05-12 16:23:43	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null
2023-05-12 16:32:38	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null
2023-05-12 16:41:34	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null
2023-05-12 16:50:30	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null
2023-05-12 16:59:26	Null	Null	Null	Null	Null	Null	Null	Null	Null	Null

Null değerli satırlar kaldırıldıktan sonra bazı sütunlarda eksik değerler tespit edilmiştir. Eksik değerlerin bulunduğu sütunlar ve her birindeki eksik değer sayısı Tablo 3. 8’de verilmiştir.

Buradaki "null" değerler, veri bütünlüğünü sağlamak amacıyla yapılan üç farklı yöntemde sırasıyla "0" ile doldurulmuş, "dropna" yöntemiyle silinmiş ve ortalama (mean) ile doldurulmuştur.

**Tablo 3. 8.** Eksik değer tablosu

A11	12
A15	1
A21	2
A23	14
A26	11
A35	19
A36	16
A39	1

Verinin temel istatistik özeti çıkarıldığında her bir özelliğe ait çeşitli merkezi eğilim ve yayılım ölçüleri elde edilmiştir. Her bir sütuna ait ortalama, standart sapma, minimum ve maksimum değerler bulunmuştur (Tablo 3. 9). Böylece veri setindeki uç değerler gözlemlenmiş ve anomali tespiti için dayanak noktası oluşturulmuştur.

**Tablo 3. 9.** Veri setinin temel istatistik özeti

Sütun	A1	A11	A15	A2	A21	A23	A26	A35	A36	A39
count	22640	22640	22640	22640	22640	22640	22640	22640	22640	22640
mean	9.93	81.65	6.02	158.41	19.42	41.53	65.87	91.32	1419.74	-307.61
std	5.94	82.84	3.09	94.39	30.56	31.98	18.45	99.08	24.19	852.60
min	0	0	0	0	0	0	-10.7	-2.23	0	-4408.1
25%	7	0	2	78	12.9	23.8	70.5	0.11	1404	0
50%	11	92.6	8	158	13	27.6	70.5	55.54	1410	7
75%	11	116.6	8	233	13	33.2	70.6	149.78	1444	8
max	30	500	10	342	150.5	105.6	203.7	350.02	1559	8

Veri setindeki değişkenler arasındaki doğrusal ilişkiyi anlayıp analiz edebilmek için korelasyon matrisi çıkarılmıştır. Lstm modellerinde zaman serisi verilerinde birbirine yüksek derecede korele olan değişkenler birbirinin yerine geçebilir. Bu durum modelin doğruluğunu olumsuz etkileyebilir. Görsel şekilde sunulan Şekil 3. 7’deki matris hangi değişkenlerin modeli oluştururken kullanılacağını hangilerinin kullanılmayacağını belirlemede faydalı olacaktır.



### Şekil 3. 7. Değişkenler Arasındaki İlişkilerin Korelasyon Matrisi

Zaman serisi verilerini modellemenin amacı geçmişteki gözlemlere dayanarak gelecekteki değerlerin tahmin edilmesini sağlamaktır. Bu doğrultuda veri setinin geçmiş  $n\_minute$  zaman dilimindeki değerleri ve tahmin edilecek hedef değişkenleri içerecek şekilde çerçevesi gereklidir (Şekil 3. 8). Yeni veri çerçevesi(reframed\_df) zaman serisi tahmin modelleri için uygun yapıda veri girişi sunar.

```
reframed_df = pd.DataFrame(  
    series_to_supervised(scaled_veri.to_numpy(), n_minute,  
    len([target]))  
    reframed_df.index = veri.index[n_minute:]
```

Şekil 3. 8. Zaman serisi verilerinin çerçevesi

#### 3.2.1.3. Veri ön işleme

Toplanan veriler modeli geliştirmeden önce tutarlılık, doğruluk ve uygunluk sağlamak amacıyla detaylı bir ön işlemeden geçirilmiştir. Veri temizleme teknikleri sensör verilerindeki eksik veri (kayıtlı tutulmamış) problemini, aykırı değerleri ve tutarsızlığı ortadan kaldırmak için uygulanmaktadır. Zaman serisi verileri, tahminsel bakım modellerinin zamansal ayrıntılarıyla hizalanması için titiz bir şekilde düzenlenmiştir (Qureshi vd., 2024).

##### 3.2.1.3.1. Zaman serisi verisinin normalizasyonu

Normalizasyon ve özellik ölçeklendirme gibi ön işleme adımlarının hem zaman serisi analizinde hem de anormallik tespitinde önemli yeri vardır (Iqbal & Amin, 2024).

Çalışmada verileri daha iyi gözlemleyip, analiz edebilmek için(Chen vd., 2024) veriler, [0, 1] arasında min-max normalizasyon yöntemi ile normalize edilmiştir. Bu işlemle farklı aralıklardaki ham veriler 0 ile 1 arasında olacak şekilde ölçeklenmiştir. Minimum değer 0'a maksimum değer 1'e endekslenmiştir.

```

scaler = MinMaxScaler(feature_range=(0, 1))
scaled_veri =
pd.DataFrame(scaler.fit_transform(veri[cols_to_analyze]),
              index=veri.index,
              columns=cols_to_analyze)

```

**Şekil 3. 9.** Verilerin Min-Max ölçeklendirme ile normalizasyonu kodu

Normalizasyon sonrası oluşan yeni değerler Tablo 3. 10'da verilmiştir.

**Tablo 3. 10.** Verilerin Min-Max ölçeklendirme ile normalizasyonu

	A1	A11	A15	A2	A21	A23	A26	A35	A36	A39
0	0.3667	0.2298	0.72	0.3012	0.0857	0.2585	0.3787	0.9107	0.7818	0.9049
1	0.3667	0.1868	0.72	0.3012	0.0857	0.2500	0.3787	0.9155	0.7835	0.9055
2	0.3667	0.2354	0.72	0.3012	0.0864	0.2491	0.3783	0.9198	0.7851	0.9062
3	0.3667	0.2354	0.71	0.3012	0.0857	0.2566	0.3787	0.9243	0.7868	0.9062
4	0.3667	0.2264	0.71	0.3012	0.0857	0.2519	0.3787	0.9286	0.7885	0.9062

### 3.2.1.4. Özellik çıkarımı

Özellik çıkarımı ve boyut azaltma orijinal veri kümesinden yeni özellikler çıkarmayı kapsar. Önemli özellikler kaybolmadan çok büyük veri setlerinde yüksek boyutları azaltır. Dönüşüm yöntemi cebirsel dönüşüm yöntemiyle ve bazı optimizasyon kriterlerine göre yapılır. Ayrıca yüksek boyutlu problemlerle uğraşırken önemli bilgileri de gözden kaçırmaz. Boyut azaltma yöntemleri, fazla bilgi kaybını engelleyerek orijinal veri yapısını korur. Aşırı uyuma daha az maruz kalır ve ayrıca sınıflandırma doğruluğu için özellik seçimi yöntemlerine göre daha iyi performans gösterir (Zebari vd., 2020). Özellik çıkarımının amacı var olan verilerden yeni özellikler geliştirerek Makine Öğrenme algoritmalarının tahmin etme yeteneğini yükseltmektir (Shaala vd., t.y.).

Bu çalışmada veri seti üzerinde zaman alanı özellik çıkarma işlemi uygulanmıştır. Eklenen bu 10 yeni özellik, zaman alanındaki verilerin temel istatistiksel dağılımını ve yayılımını ifade etmektedir. Böylece modelin daha iyi genelleştirme yeteneği kazanması hedeflenmiştir.

Çalışmamızda özellik çıkarma yöntemi ile veri setine Time Domain Features'lar eklenmiştir.

**Tablo 3. 11.** Veri setine eklenen zaman alanı özellikleri (Time Domain Features)

Zaman Özellikleri	Alanı	Açıklama
Mean (Ortalama)		Verilerin genel dağılımını temsil eder. Tüm değerlerin toplamının veri sayısına bölünmesiyle elde edilir.
Min (Minimum değer)		Veri setindeki en küçük değeri temsil eder ve alt sınırı oluşturur.
Max (Maksimum değer)		Veri setindeki en büyük değeri gösterir ve üst sınırı oluşturur.
Var (Varyans)		Verilerin ortalamadan ne kadar uzaklaştığının değeridir. Varyans değerinin büyüklüğü verilerin daha geniş bir alanda dağılım gösterdiğini ifade eder.
Std (Standart Sapma)		Verilerin yayılımını ve değişkenliğini ölçer.
Range (Değer Aralığı)		Veri kümesindeki en büyük değer ile en küçük değer arasındaki farkı ifade eder.
Skewness (Çarpıklık)		Verilerin simetrikliğini ifade eder. Dağılımın hangi yöne eğimli olduğunu ifade eder.
Kurtosis (Basıklık)		Verilerin uç değerlerindeki yoğunluğu ölçer..
RMS (Kök Ortalama Karesi)		Verilerin enerjisini ve genliğini temsil eder.

Zaman alanı özellikleri eklenirken Tablo 3. 12'de verilen formüller ile ilgili özelliğin değerleri hesaplanmıştır.

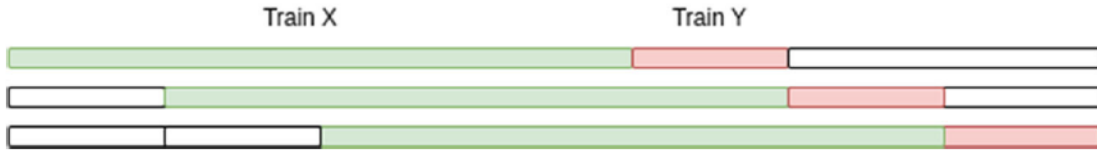
**Tablo 3. 12.** Zaman alanı özellikleri formülleri (Ding & Ming, 2019)

$\text{Range} = \max(a_i) - \min(a_i), \quad i \in \{1, 2, \dots, n\}$ $\text{Max} = \max(a_i), \quad i \in \{1, 2, \dots, n\}$ $\text{Min} = \min(a_i), \quad i \in \{1, 2, \dots, n\}$ $\text{Mean} = \frac{1}{n} \sum_{i=1}^n a_i$ $\text{Var} = \frac{1}{n} \sum_{i=1}^n (a_i - \text{Mean})^2$ $\text{Std} = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i - \text{Mean})^2}$ $\text{Skewness} = \frac{1}{n} \sum_{i=1}^n \left( \frac{a_i - \text{Mean}}{\text{Var}} \right)^3$ $\text{Kurtosis} = \frac{1}{n} \sum_{i=1}^n \left( \frac{a_i - \text{Mean}}{\text{Var}} \right)^4$ $\text{RMS} = \sqrt{\frac{1}{n} (a_1^2 + a_2^2 + \dots + a_n^2)}$
--

### 3.2.1.5. Zaman serisi verisinin eğitim /test verilerine ayrılması ve şekillendirilmesi

Veri seti modelin etkinliğini değerlendirebilmek adına %80'i eğitim ve %20 si test olmak üzere ikiye ayrılmıştır. Veri bölme işlemi esnasında zaman serilerindeki ardışıklığın korunması için veriler karıştırılmamıştır. Shuffle=False parametresi ile gerçekleştirilen bu durum zaman serisinin sıralı yapısının bozulmamasını ve modelin doğru şekilde eğitilmesini sağlar.

LSTM'in yapısı gereği giriş verileri, zaman bilgisini içeren üç boyutlu bir tensöre dönüştürülmelidir (Jin & Kang, 2024). Bu sebeple kaydırmalı pencere tekniği uygulanmıştır. Bu teknik ile 2 boyutlu sensör verileri 3 boyutlu tensör verilerine dönüştürülür. Bu tensör yapısı (örnek sayısı, zaman adımı, özellik sayısı) formatındadır. Modelin, eski verilere dayanarak gelecekteki verileri tahmin etmesinin temelini oluşturur.



**Şekil 3. 10.** Kayan pencere algoritması

```
X_train_df, X_test_df, y_train_df, y_test_df = train_test_split(
    X, y, train_size=train_size, shuffle=False)
X_train = X_train_df.to_numpy().reshape(
    (X_train_df.shape[0], n_minute, len(cols_to_analyze)))
X_test = X_test_df.to_numpy().reshape(
    (X_test_df.shape[0], n_minute, len(cols_to_analyze)))
```

**Şekil 3. 11.** Veri setinin eğitim ve test verilerine ayrılması ve yeniden şekillendirilmesi

**Tablo 3. 13.** Eğitim ve Test Veri Kümesi Boyutları

Veri Kümesi	Boyutlar (Örnek Sayısı, Zaman Adımı, Özellik Sayısı)
Eğitim Verisi (X_train)	(18108, 4, 19)
Test Verisi (X_test)	(4528, 4, 19)
Eğitim Hedefi (y_train_df)	(18108, 10)
Test Hedefi (y_test_df)	(4528, 10)

## 3.2.2. Model geliştirme ve değerlendirme

### 3.2.2.1. LSTM hiperparametreleri

Derin sinir ağı modelleri, belirli bir veri kümesinde en iyi performansa ulaşabilmek için yapılandırılabilen pek çok parametreye sahiptir.(Liao vd., 2022) Derin öğrenme modellerinin başarısı, hiperparametre adı verilen bu parametrelerin en uygun şekilde optimize edilmesine bağlıdır. Hiperparametreler model eğitilmeden önce belirlenir (Yadav vd., 2020). Hiperparametreler, modelin öğrenme algoritmasını veya altındaki istatistiksel modelin yapısını kontrol eder. (19. *Hyperparameter Optimization — Dive into Deep Learning 1.0.3 documentation*, t.y.). LSTM derin öğrenme algoritmasına özgü hiperparametreler şunlardır.

#### 3.3.2.1.1. Aktivasyon fonksiyonları (Activation Functions)

İyi bir performans ve daha doğru sonuçlar için, ağıdaki gizli katman sayısı, eğitim metodları, hiperparametre optimizasyonu gibi pek çok yapı titizlikle oluşturulmalıdır. Aktivasyon fonksiyonu buradaki en önemli parametrelerden biridir. Aktivasyon fonksiyonu seçimi yapılmak istenen göreve bağlıdır. Seçim için tek bir doğru yol yoktur (Sharma vd., 2020).

Aktivasyon fonksiyonları, derin sinir ağlarında bilginin bir sonraki nörona aktarılıp aktarılmamasını belirler. Öğrenme aşamasının başarılı şekilde ilerlemesi, overfitting'in engellenmesi, doğruluk oranının artırılması ve hesaplama maliyetinin düşürülmesi gibi durumlar temel alınarak aktivasyon fonksiyonları geliştirilmiştir. Bu fonksiyonlar gerçek dünya problemlerindeki karmaşık desenleri ortaya çıkararak derin öğrenme modelinin daha başarılı çalışmasına ve öğrenmesine olanak sunar. Aktivasyon fonksiyonu belirlenmeden eğitilen derin öğrenme modelleri, tam bir öğrenme gerçekleştirmediği için yeterli başarıya ulaşamaz (Kılıçarslan & Adem, 2021).

#### Sigmoid fonksiyonu

Değerleri 0 ile 1 arasında dönüştüren, doğrusal olmayan, yaygın olarak kullanılan aktivasyonlarından biridir. Sigmoid fonksiyonu 0 etrafında simetrik değildir. Yani

nöronların tüm çıkış değerlerinin işaretleri birbirinin aynısıdır. Bu durum, sigmoid fonksiyonun ölçeklendirilmesiyle düzeltilebilir(Sharma vd., 2020).

### **Tanh fonksiyonu**

Hiperbolik Tanjant fonksiyonudur. Tanh fonksiyonu, sigmoid fonksiyonuna benzer şekildedir, fakat orijinal etrafında simetriktir. Bu durum önceki katmanlardan çıkan farklı işaretlerdeki çıkışların, sonraki katmana giriş olarak verilmesine olanak sunar. Tanh fonksiyonunda değerler -1 ile 1 arasındadır. Bu fonksiyon süreklidir ve türevi alınabilir. Tanh fonksiyonu, gradyanlarını belirli yönde sınırlandırmadığı ve sıfır merkezli olduğu için sigmoid fonksiyonuna tercih edilir (Sharma vd., 2020).

### **Relu fonksiyonu**

Relu fonksiyonu, sinir ağlarında sıklıkla kullanılan, doğrusal olmayan bir aktivasyon fonksiyonudur. Rectified Linear Unit(Düzeltilici doğrusal birim) ifadesinin kısaltmasıdır. Bu fonksiyonun artı yönü tüm nöronların aynı zamanda aktive olmamasıdır. Bu durum bir nöronun sadece doğrusal dönüşümünün çıktısı 0 olduğunda devre dışı bırakılacağı anlamına gelir. Bu bağlamda tüm nöronları aynı zamanda aktive olmadığı için, belli sayıda nöron aktive olduğu için daha verimli çalışır (Sharma vd., 2020).

#### **3.3.2.1.2. Optimizasyon algoritmaları (Optimization Algorithms)**

Optimizasyon algoritmaları, bir makinenin deneyim ile öğrenebilmesinin alt yapısını oluşturur. Bu algoritmalar, gradyanları hesaplayarak kayıp fonksiyonunu en aza indirmeye çalışır. Çeşitli optimizasyon algoritmalarıyla öğrenme şekli farklılaştırılır. Makine öğrenimi ve derin öğrenme, giriş verilerinin parametrelerini öğrenebilmek için optimizasyon algoritmalarını kullanırlar. Bu sebeple optimizasyon algoritmaları gerçek dünya problemlerinde başarılı sonuçlar üretilebilmek için kritik önem taşırlar (Zaheer & Shaziya, 2019).

### **Adam**

Bu algoritma adını "adaptive moments"(uyarlanabilir momentler) ifadesinden almıştır. Adam, RMSProp ve momentum tekniklerinin bir araya getirilmesiyle oluşmuştur. Güncelleme aşamasında sadece gradyanın düzgünleştirilmiş hali baz alınır. Ayrıca önyargı düzeltme mekanizmasını da kapsar (Zaheer & Shaziya, 2019).

## **Adagrad**

Adagrad, gradyan iniş optimizasyon algoritmasıdır. Bu yöntem ile öğrenme oranı tüm yakınsama süreci boyunca uyarlanabilir hale gelebilir. Bu sebeple Adagrad algoritması özellikle seyrek veri setlerinde başarılı bir performans sergiler. Fakat eğitim esnasında öğrenme oranının zamanla yavaşlaması algoritmanın dezavantajıdır. Bu durum, çoklu epoch'larda öğrenme performansını düşürür (Zhang vd., 2018).

## **RMSProp**

RMSProp algoritması, Adagrad algoritmasını eğimin birikme şekli açısından değiştirir. Eğimler üstel ağırlıklı bir ortalama biriktirilir. RMSProp, geçmiş verilere bağlı kalmak yerine sadece en güncel eğitim bilgilerini saklar. RMSProp ve onun türevlerini ele alır (Zaheer & Shaziya, 2019).

## **Stochastic gradient descent(SDG)**

SDG algoritması son zamanlarda en sık kullanılan gradient descent optimizasyon algoritmalarının bir çeşidi olmasına rağmen, birçok dezavantajı da vardır. Örneğin SDG'nin yakınsama süreci istikrarlı değildir. Bunun yanısıra aşırı öğrenme(overfitting) sıklıkla görülmektedir. Bu durum, çoklu epochlarda yetersiz performansa sebep olmaktadır(Zhang vd., 2018).

### **3.3.2.1.3. Modelin diğer hiperparametreleri**

#### **Birim sayısı (Number of Units)**

Modelin LSTM katmanındaki bellek hücrelerinin sayısını ifade eder. Birim sayısı artırılarak modeldeki daha karmaşık desenlerin öğrenilmesi sağlanır. Fakat birim sayısını gereğinden fazla arttırmak aşırı öğrenme(overfitting) yol açabilir. Bu parametre, modelin öğrenme başarısının temel etkenidir.

#### **Katman sayısı (Number of Layers)**

LSTM modelinin derinliğini ifade eder. Katman sayısı arttıkça karmaşık desenlerin öğrenilmesi kolaylaşır. Fakat bu durum modelin hesaplama maliyetinde artışa sebep olabilir ve aşırı öğrenme(overfitting) sorununa sebep olabilir.

### **Dizi uzunluđu (Sequence Length)**

Modelin eski verilerden ne kadar bilgi tutacađını belirler. Uzun boyutlu diziler arasında daha fazla bağlam bilgisi vardır. Dizi uzunluđu, modelin hesaplama maliyetini arttırır. Zaman serisi problemlerinde dizi uzunluđu oldukça önemlidir.

### **Batch boyutu (Batch Size)**

LSTM modelinin her bir adımda işlediđi veri sayısını ifade eder. Küçük batch boyutları eğitim süresini kısaltır fakat sonuçlar daha gürültülü olabilir. Batch boyutunun büyük olması ise daha yavaş fakat daha dengeli bir öğrenme sağlar.

### **Dropout oranı (Dropout Rate)**

Modelin eğitilmesi esnasında aşırı öğrenme(overfitting) durumunu engellemek için belirli nöronlar rastgele devre dışı bırakılır. Böylece ađın daha güçlü özellikler öğrenmesi sağlanır. Bu rastgelelik, nöronların birbirlerinin aktivasyonlarına aşırı bađımlı olmasını engeller. Bu nöronların miktarı genellikle %20-%50 lik bir oranı kapsar. Bu durum ađ organizasyonunun ihtiyacı ile yeterli bilgi akışının korunması arasında bir denge sağlar(Baldi & Sadowski, 2013).

Derin sinir ađları doğrusal olmayan çok sayıda gizli katmana sahiptir. Bu sayede girdiler ve çıktılar arasındaki son derece kompleks ilişkileri öğrenebilen güçlü modeller oluşturabilirler. Fakat eğitim verisinin kısıtlı olduđu durumlarda bu kompleks ilişkilerin pek çođu sadece örnekleme gürültüsünden kaynaklanır. Eğitim verisinde mevcut olan bu ilişkiler, aynı dağılımdan gelse bile gerçek test verisinde bulunmaz. Bu durum aşırı öğrenmeye(overfitting) sebep olur (Srivastava vd., t.y.).

Dropout, hem aşırı öğrenmeyi(overfitting) engelleyen hem de üstel sayıda farklı sinir ađı mimarisini yaklaşık olarak ve verimli bir şekilde birleştirmeyi mümkün kılan bir tekniktir. "Dropout" terimi sinir ađındaki birimlerin (gizli ve görünür) devre dışı bırakılmasını temsil eder. Rastgele bir birimin ađdan geçici olarak çıkarılmasını ve onun tüm giriş ve çıkış bağlantılarının kaldırılmasını kapsar (Srivastava vd., t.y.).

### **Öğrenme hızı (Learning Rate)**

Modelin, ağırlıklarının yenileme hızını ifade eder. Düşük öğrenme hızı, yakınsama sürecini yavaşlatabilir. Yüksek öğrenme hızı ise kararsız öğrenme sürecine sebep olur. En iyi öğrenme hızı deneme yanılma yöntemiyle bulunur.

### **Gradyan sınırlandırma (Gradient Clipping)**

Eğitim esnasında gradyan büyüklüğünü kısıtlayarak modelin dengesini sağlar. Gradyan patlaması sorununu minimuma indirmek için kullanılan bir yöntemdir.

### **Epoch sayısı (Epochs)**

LSTM modelinin eğitim veri kümesini kaç kere ele alacağını sayısını ifade eder. Model öğrenme performansını arttırabilmek için yeterli sayıda epoch değeri girilmelidir. Fakat fazla epoch sayısı aşırı öğrenme (overfitting) sorununa yol açabilir.

### **Gizli durum boyutu (Hidden State Size)**

Modelin her bir iterasyonda tuttuğu bilginin boyutunu ifade eder. Modelin performansını etkileyen, modelin belleği olarak nitelendirilen bir hiperparametredir.

### **Dizi döndürme (Return Sequences)**

Modelin her bir zaman adımında çıktı verip vermeyeceğini kontrol eden parametrelerdir. True olarak ayarlanırsa tüm zaman adımları için çıktı verir. False olarak ayarlanırsa sadece son adımda çıktı üretir.

### **Çift yönlü LSTM (Bidirectional LSTM)**

Veri setini hem ileri hem geri yönde işleyerek, veriler arasındaki bağlamı daha iyi öğrenmeyi amaçlar. Doğal dil işleme uygulamalarında oldukça faydalıdır.

### **Kayıp fonksiyonu (Loss Function)**

Modelin tahmin ettiği değer ile gerçek veri seti arasındaki farkı belirler. Problemin türüne göre kayıp fonksiyonu tercih edilmelidir.

### 3.3.2.2. Model performansının iyileştirilmesi için hiperparametre ayarı Keras Tuner

Hiperparametre ayarlama modelin performansını optimize etmek ve tahminin doğruluk oranını arttırmak için yapılır (Qureshi vd., 2024).

Keras Tuner, Keras altında bulunan kütüphanelerden biridir ve hiperparametre optimizasyonu için kullanılır. Konvolüsyon katmanların sayısı, giriş nöronların sayısı ve optimal epoch sayısı gibi hiperparametrelerin ayarlanmasını sağlar. En iyi model için en iyi hiperparametre setini oluşturmaya yardımcı olur (Ng, t.y.).

### 3.3.2.3. Değerlendirme metrikleri

RMSE,  $R^2$  ve MAE, makine öğrenimi modellerinin performansını değerlendirebilmek için sıklıkla kullanılan makine öğrenimi metriklerdir. Yorumlanabilirliği, matematiksel temelleri ve pratik faydası bakımından değerli bir performans istatistikleridir (Srisuwan & Innet, 2024). Bu çalışmada modelin tahmini için test verisi kullanılarak Kök Ortalama Kare Hata (RMSE), R-kare ( $R^2$ ) ve Ortalama Mutlak Hata (MAE) hesaplanmıştır. Daha düşük RMSE değeri, modelin bir durum için en iyi tahmin performansını ifade eder.  $R^2$  değerinin yüksek olması modelin daha yüksek doğrulukla çalıştığını gösterir ve MAE değerinin küçük olması tahmin edilen değer ile gerçek değer arasındaki farkın az olduğunu gösterir.

RMSE, tahmin hatasının standart sapmasını ölçmek için sıklıkla kullanılan metriktir (Srisuwan & Innet, 2024).

$R^2$ , modelin bağımlı değişkendeki toplam değişkenliği azaltabildiği miktarı ifade eder. Yüksek  $R^2$  değeri modelin daha iyi uyum sağladığını gösterir.

- $R^2 = 1$ : Modelin veri setine tam uyumlu olduğunu ifade eder.
- $R^2 = 0$ : Modelin hiçbir şekilde hedef değişkeni açıklayabilecek uyumlulukta olmadığını ifade eder.
- $R^2 < 0$ : Modelin tahmin değeri zayıftır ve ortalama değeri temel alan bir tahminden bile daha düşük sonuçlar vermiştir.

MAE, test verilerindeki gerçek değerler ile modelin tahmin ettiği değerler arasındaki ortalama farkı ölçer.

Modelin eğitildikten sonra, tahmin ettiği değerler ile veri setindeki gerçek değerlerin yakınlığı karşılaştırılmak üzere  $R^2$ , RMSE ve MAE performans metrikleri hesaplanmıştır.

#### 4. DENEYLER VE BULGULAR

Çalışma kapsamında 16 farklı deney yapılmıştır. Bunlar arasından ayırt edici özellikler içeren ve ekstra uygulamalar eklenen, diğer bir ifadeyle temel dönüm noktalarına sahip 8 deney belirlenerek sunulmuştur. Bu deneylerdeki parametreler ve bulgular sırasıyla paylaşılmıştır. Deneyler yapılırken parametreler ve özellikler kademe kademe değiştirilmiş ve her eklenen yeni özelliğin veya uygulamanın başarı oranına katkısı değerlendirme metrikleri ile ölçülmüştür. Tüm deneylerde kullanılan veri setine veri ön işleme adımında şu işlemler uygulanmıştır. Tamamı “Null” değer içeren satırlar silinmiştir. Modelin eğitiminde ve testinde kullanılan veri seti bu şekilde oluşturulmuştur.

##### 4.1. Deney 1

Deney 1’de kullanılan veri seti 4 aylık veriyi kapsayan 17810 satır ve 11 sütundan oluşan zaman serisi veri kümesidir. Veri setindeki özellikler (feature) doğrudan sensörlerden gelen ölçümleri temsil etmektedir (Tablo 4. 1). Tamamı eksik (Null) olan satırlar, bilgi içermediği ve analize katkı sağlamadığı için veri setinden çıkarılmıştır. Deney 1’de kısmi olarak ‘Null’ değere sahip hücreler ‘0’ ile doldurulmuştur. Deney 1’de, modelin  $R^2$  değeri 0.7909 olarak hesaplanmış, RMSE 0.0719 ve MAE 0.0265 olarak ölçülmüştür.

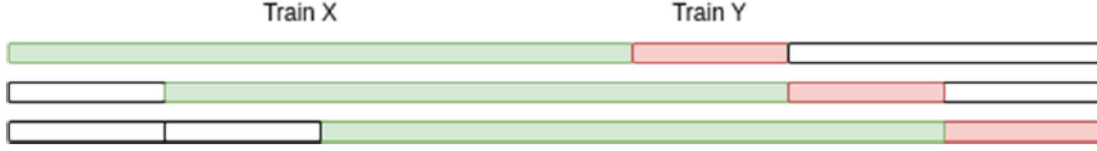
**Tablo 4. 1.** Deney 1 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39

**Tablo 4. 2.** Deney 1 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE
Optimizer	Adam (learning_rate=0.0001)
Return Sequence	False
Target	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_minute	8
Train Size	0.8
Data Shape	(17810, 11)
Time Period	4 aylık veri
Columns to Analyze	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_in (Input Window Size)	8
n_out (Output Window Size)	1
Step (Overlap)	2

Modelin eğitim aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme (windowing) stratejisindeki değerler Tablo 4. 3'te verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri (overlap) ise 2 olarak belirlenmiştir. Bu durum şu şekilde ifade edilebilir. Giriş için kullanılan 8 örnekten son iki tanesi, bir sonraki giriş için tekrar kullanılmıştır. Şekil 4. 1. Kayan pencere algoritması'te gösterilen çalışma prensibi baz alınmıştır.



**Şekil 4. 1.** Kayan pencere algoritması

Giriş verileri (Columns to Analyze) tahmin için kullanılacak 10 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir. Kısacası tüm özellikler, giriş verisinde tahmin için kullanılır ve çıkışta da aynı özelliklerin tahmin edilmesi istenir.

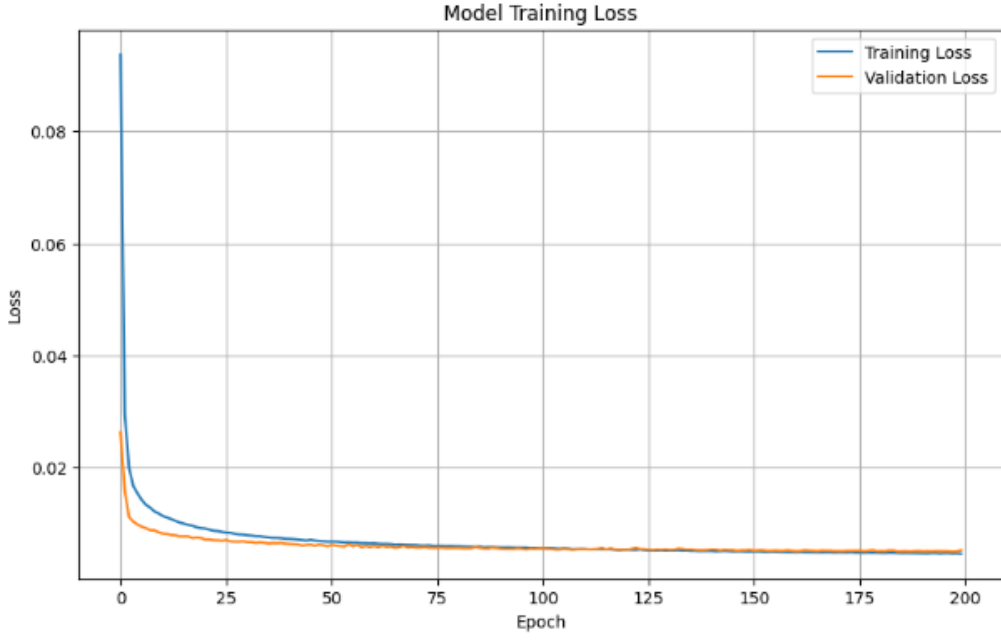
**Tablo 4. 3.** Veri setlerinin boyutları ve yapıları

Veri Seti	Satır Sayısı (n)	Sütun Sayısı (Zaman Dilimi x Özellik)	Boyutlar
X_train	14241	8 x 10	(14241, 8, 10)
X_test	3561	8 x 10	(3561, 8, 10)
y_train_df	14241	10	(14241, 10)
y_test_df	3561	10	(3561, 10)

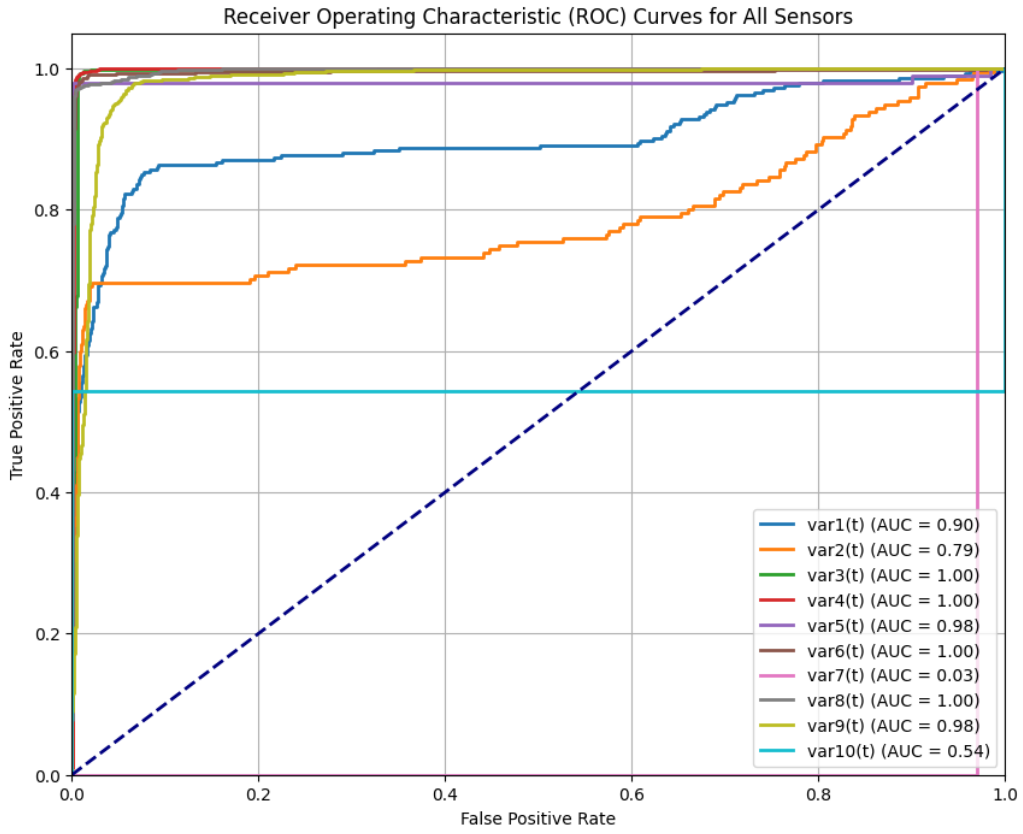
Tablo 4. 4'te LSTM modelinin tahmin performansını değerlendiren üç temel metrik gösterilmektedir.

**Tablo 4. 4.** Deney 1, model performans değerleri

Model	R <sup>2</sup>	RMSE	MAE
LSTM	0.7909	0.0719	0.0265



Şekil 4. 2. Modelin öğrenme eğrisi (Loss Curve)



Şekil 4. 3. Modelin ROC eğrisi ve sınıflandırma başarısı

## 4.2. Deney 2

Deney 2’de, tamamen eksik (Null) olan satırlar veri setinden çıkarılmış, kısmi eksik değerlere sahip satırlar ise eksik veri işleme tekniklerinden biri olan "dropna" yöntemiyle silinmiştir. Deney 1 ile tüm uygulamalar aynıdır; tek fark, kısmi eksik verilerin işleme yöntemidir. Bu işlem,  $R^2$  değerinde önemli bir artış sağlamış ve 0.79'dan 0.86'ya yükselmesine neden olmuştur. Bunun yanısıra RMSE değeri 0.0674 ve MAE değeri 0.0258 olarak ölçülmüştür.

**Tablo 4. 5.** Deney 2 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39

**Tablo 4. 6.** Deney 2 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE
Optimizer	Adam (learning_rate=0.0001)
Return Sequence	False
Target	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_minute	8
Train Size	0.8
Data Shape	(17810, 11)
Time Period	4 aylık veri
Columns to Analyze	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_in (Input Window Size)	8
n_out (Output Window Size)	1
Step (Overlap)	2

Modelin eğitim aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme(windowing) stratejisindeki değerler Tablo 4. 6’da verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri(overlap) ise 2 olarak belirlenmiştir.

Giriş verileri (Columns to Analyze) tahmin için kullanılacak 10 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

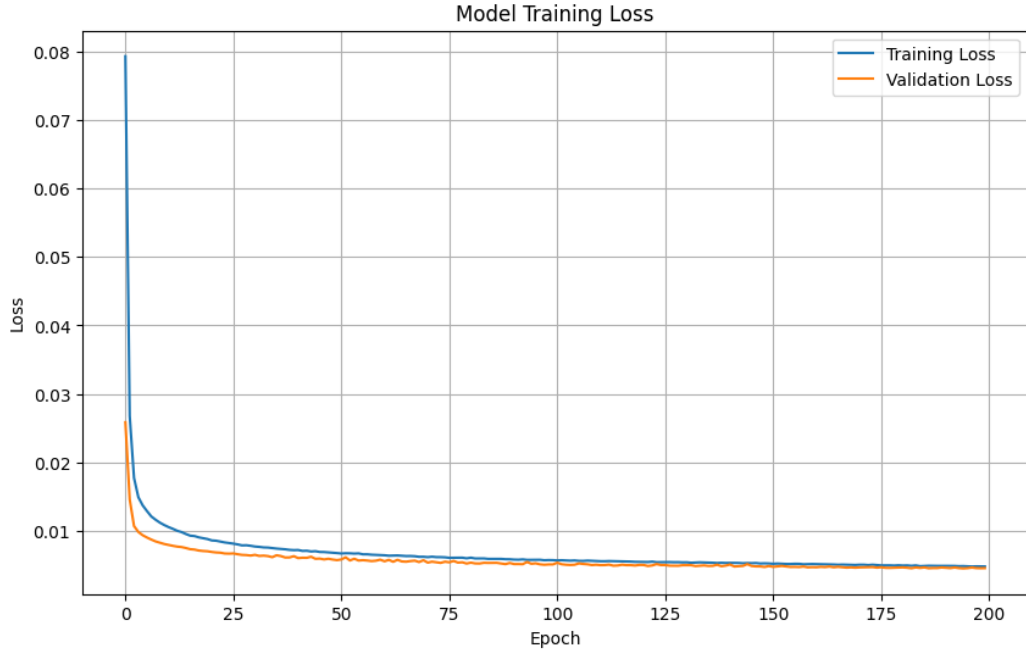
**Tablo 4. 7.** Veri setlerinin boyutları ve yapıları

Veri Seti	Satır Sayısı (n)	Sütun Sayısı (Zaman Dilimi x Özellik)	Boyutlar
X_train	14225	8 x 10	(14225, 8, 10)
X_test	3557	8 x 10	(3557, 8, 10)
y_train_df	14225	10	(14225, 10)
y_test_df	3557	10	(3557, 10)

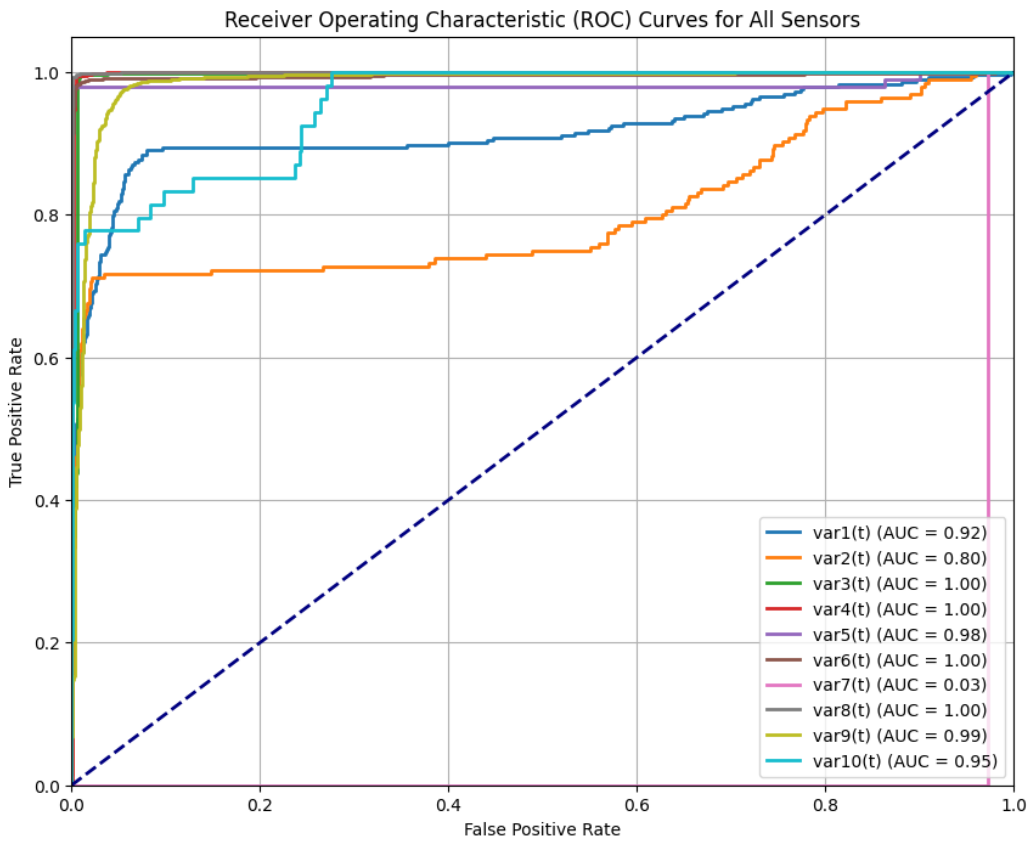
Tablo 4. 8’de LSTM modelinin tahmin performansını değerlendiren üç temel metrik gösterilmektedir.

**Tablo 4. 8.** Deney 2, model performans değerleri

Model	R2	RMSE	MAE
LSTM	0.8676	0.0674	0.0258



Şekil 4. 4. Modelin öğrenme eğrisi (Loss Curve)



Şekil 4. 5. Modelin ROC eğrisi ve sınıflandırma başarısı

### 4.3. Deney 3

Deney 3'te, tamamıyla eksik (Null) olan satırlar veri setinden çıkarılmış, kısmi eksik değerlere sahip satırlar ise ilgili sütunların ortalama (mean) değerleri ile doldurulmuştur. Bu işlem, deney 2 ile aynı adımları izlerken, tek fark kısmi eksik verilerin işleme yönteminde bulunmaktadır.  $R^2$  değeri 0.8646, RMSE değeri 0.0685 ve MAE değeri ise 0.0265 olarak ölçülmüştür. Deney 3, deney 2 ile kıyaslandığında her iki deney de benzer düzeyde doğruluk sunmaktadır. Sonuç olarak, her iki deney de benzer bir performans seviyesine sahip olup, Deney 3'te elde edilen küçük farklar modelin doğruluğunda önemli bir iyileşme sağlamamaktadır. Ancak, kısmi eksik verilerin ortalama ile doldurulmasının genel model performansını biraz daha iyileştirdiği söylenebilir.

**Tablo 4. 9.** Deney 3 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39

**Tablo 4. 10.** Deney 3 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE
Optimizer	Adam (learning_rate=0.0001)
Return Sequence	False
Target	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_minute	8
Train Size	0.8
Data Shape	(17810, 11)
Time Period	4 months
Columns to Analyze	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']

<b>n_in (Input Window Size)</b>	8
<b>n_out (Output Window Size)</b>	1
<b>Step (Overlap)</b>	2

Modelin eğitim aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme(windowing) stratejisindeki değerler Tablo 4. 10'da verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri(overlap) ise 2 olarak belirlenmiştir.

Giriş verileri (Columns to Analyze) tahmin için kullanılacak 10 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

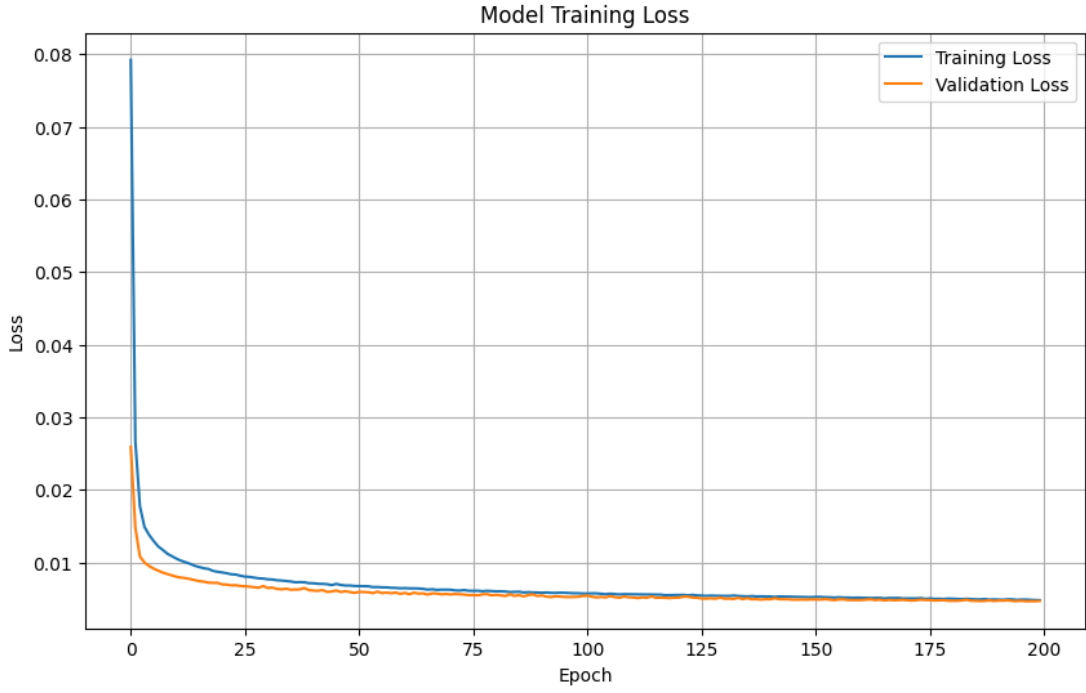
**Tablo 4. 11.** Veri setlerinin boyutları ve yapıları

<b>Veri Seti</b>	<b>Satır Sayısı (n)</b>	<b>Sütun Sayısı (Zaman Dilimi x Özellik)</b>	<b>Boyutlar</b>
X_train	14241	8 x 10	(14241, 8, 10)
X_test	3561	8 x 10	(3561, 8, 10)
y_train_df	14241	10	(14241, 10)
y_test_df	3561	10	(3561, 10)

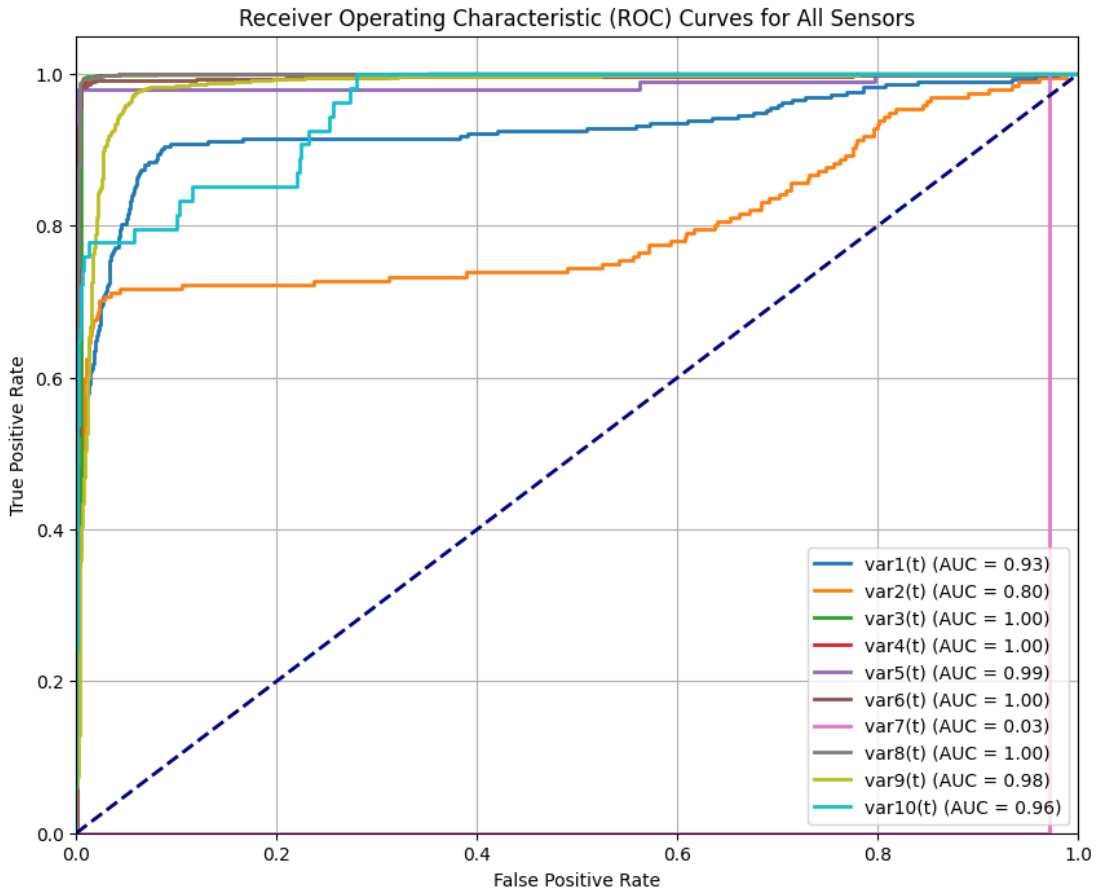
Tablo 4. 12'de LSTM modelinin tahmin performansını değerlendiren üç temel metrik gösterilmektedir.

**Tablo 4. 12.** Deney 3, model performans değerleri

<b>Model</b>	<b>R2</b>	<b>RMSE</b>	<b>MAE</b>
LSTM	0.8646	0.0685	0.0265



Şekil 4. 6. Modelin öğrenme eğrisi (Loss Curve)



Şekil 4. 7. Modelin ROC eğrisi ve sınıflandırma başarısı

#### 4.4. Deney 4

Deney 4'te, deney 2'de elde edilen olumlu sonuçlar göz önünde bulundurularak aynı uygulamalar tekrarlanmıştır. Tamamıyla eksik (Null) olan satırlar veri setinden çıkarılmıştır. Kısmi eksik değerlere sahip satırlar ise, eksik veri işleme tekniklerinden biri olan "dropna" yöntemiyle silinmiştir. Bu deneyde test edilmek istenen ana faktör, overlap miktarının etkisidir. Deney 2'deki overlap miktarı 2 dir. Deney 3'te ise deney 2'deki tüm parametreler ve uygulamalara ek olarak, overlap miktarı 4 olarak belirlenmiş ve sonuçlar bu yeni parametreyle değerlendirilmiştir.  $R^2$  değeri 0.8676, RMSE değeri 0.0674 ve MAE değeri 0.0258 olarak elde edilmiştir. Deney 4, deney 3 ile karşılaştırıldığında çok küçük bir iyileşme göstermektedir, özellikle  $R^2$  değeri üzerinden. Ancak RMSE ve MAE değerleri açısından, model performansında belirgin bir fark gözlemlenmemektedir. Bu, overlap miktarının 4 olarak ayarlanmasının modelin doğruluğunu biraz artırdığı ancak hata seviyelerinde önemli bir değişim sağlamadığı sonucunu ortaya koymaktadır.

**Tablo 4. 13.** Deney 4 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39

**Tablo 4. 14.** Deney 4 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE
Optimizer	Adam (learning_rate=0.0001)
Return Sequence	False
Target	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_minute	8
Train Size	0.8

<b>Data Shape</b>	(17810, 11)
<b>Time Period</b>	4 months
<b>Columns to Analyze</b>	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
<b>n_in (Input Window Size)</b>	8
<b>n_out (Output Window Size)</b>	1
<b>Step (Overlap)</b>	4

Modelin eğitilmesi aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme(windowsing) stratejisindeki değerler Tablo 4. 14’te verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri (overlap) ise 4 olarak belirlenmiştir.

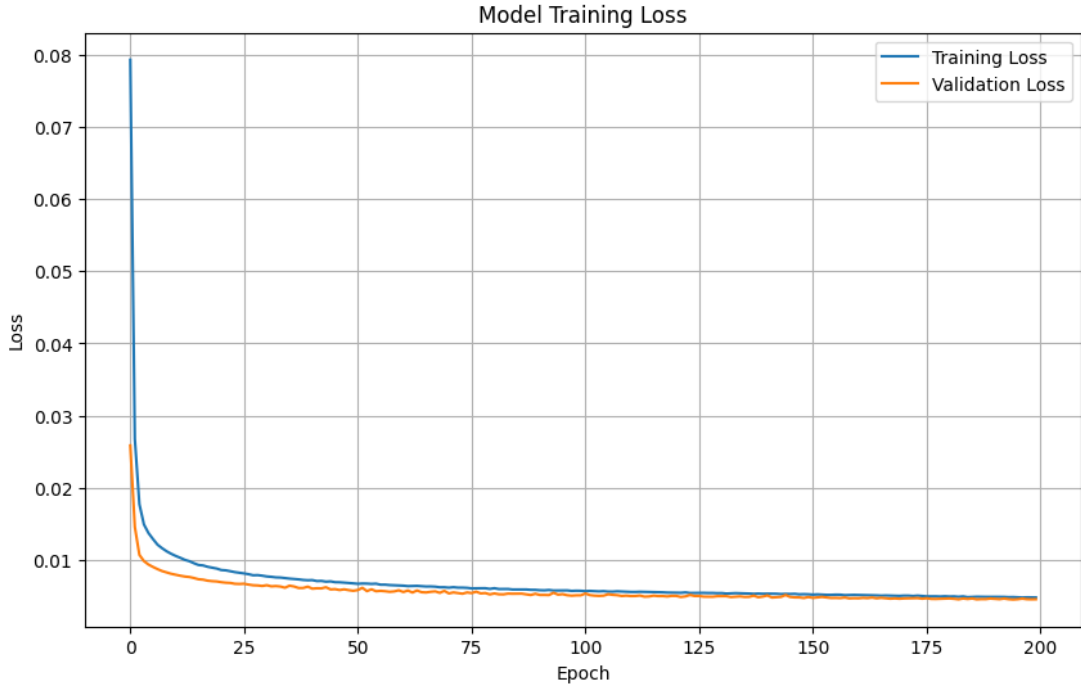
Giriş verileri (Columns to Analyze) tahmin için kullanılacak 10 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

**Tablo 4. 15.** Veri setlerinin boyutları ve yapıları

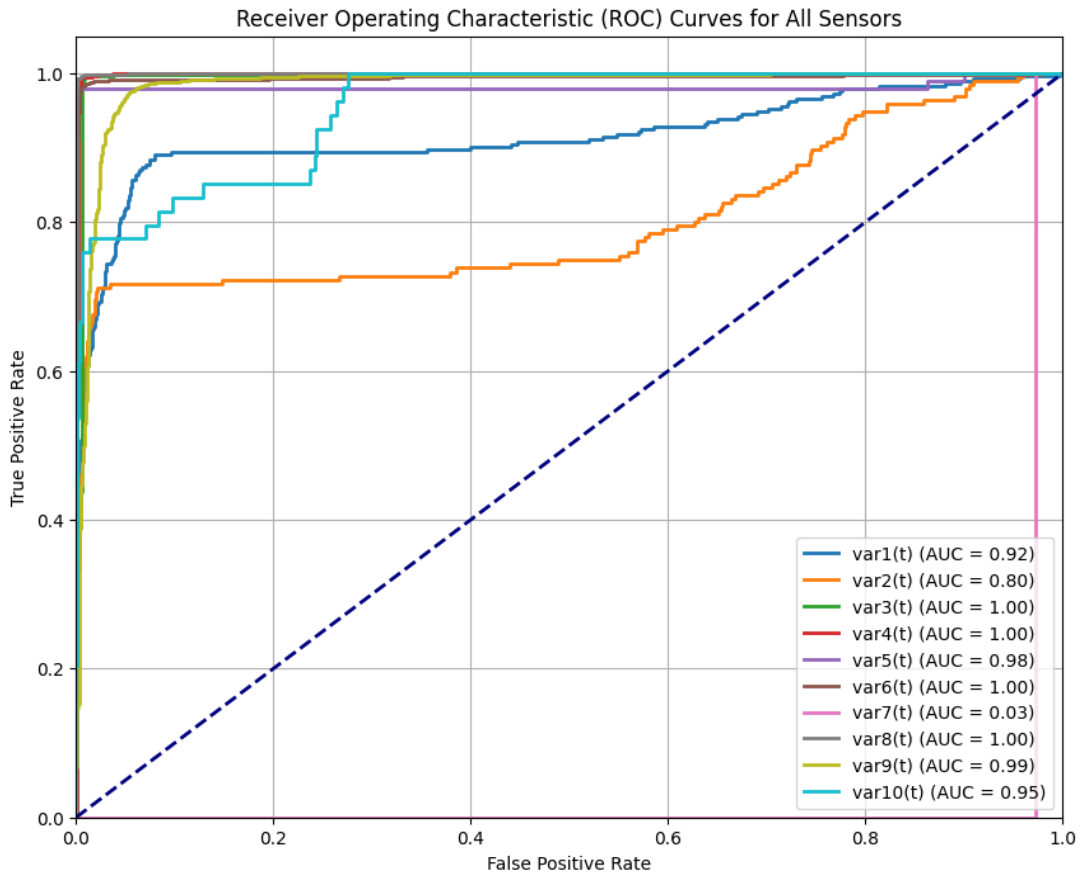
<b>Veri Seti</b>	<b>Satır Sayısı (n)</b>	<b>Sütun Sayısı (Zaman Dilimi x Özellik)</b>	<b>Boyutlar</b>
X_train	14225	8 x 10	(14225, 8, 10)
X_test	3557	8 x 10	(3557, 8, 10)
y_train_df	14225	10	(14225, 10)
y_test_df	3557	10	(3557, 10)

**Tablo 4. 16.** Deney 4 model performans değerleri

<b>Model</b>	<b>R2</b>	<b>RMSE</b>	<b>MAE</b>
LSTM	0.8676	0.0674	0.0258



**Tablo 4. 8.** Modelin öğrenme eğrisi (Loss Curve)



**Tablo 4. 9.** Modelin ROC eğrisi ve sınıflandırma başarısı

#### 4.5. Deney 5

Deney 5'te, deney 4'te uygulanan tüm uygulamalar aynen takip edilmiştir. Tamamıyla eksik (Null) olan satırlar veri setinden çıkarılmış, kısmi eksik verilere sahip satırlar ise eksik veri işleme tekniklerinden biri olan "dropna" yöntemiyle silinmiştir. Bu deneyde test edilmek istenen faktör, giriş penceresi boyutunun etkisidir. Deney 4'te giriş penceresi boyutu 8 dir. Deney 5'te ise deney 4'teki tüm parametreler ve uygulamalara ek olarak, giriş penceresi boyutu 4 olarak belirlenmiş ve elde edilen sonuçlar değerlendirilmiştir.  $R^2$  değeri 0.8674, RMSE değeri 0.0675 ve MAE değeri ise 0.0262 olarak ölçülmüştür. Deney 4 ile sonuçlar yakındır.

**Tablo 4. 17.** Deney 5 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39

**Tablo 4. 18.** Deney 5 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE
Optimizer	Adam (learning_rate=0.0001)
Return Sequence	False
Target	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_minute	8
Train Size	0.8
Data Shape	(17810, 11)
Time Period	4 months
Columns to Analyze	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_in (Input Window Size)	4

<b>n_out (Output Window Size)</b>	1
<b>Step (Overlap)</b>	2

Modelin eğitim aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme(windowing) stratejisindeki değerler Tablo 4. 18'de verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 4 tür. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri(overlap) ise 2 olarak belirlenmiştir.

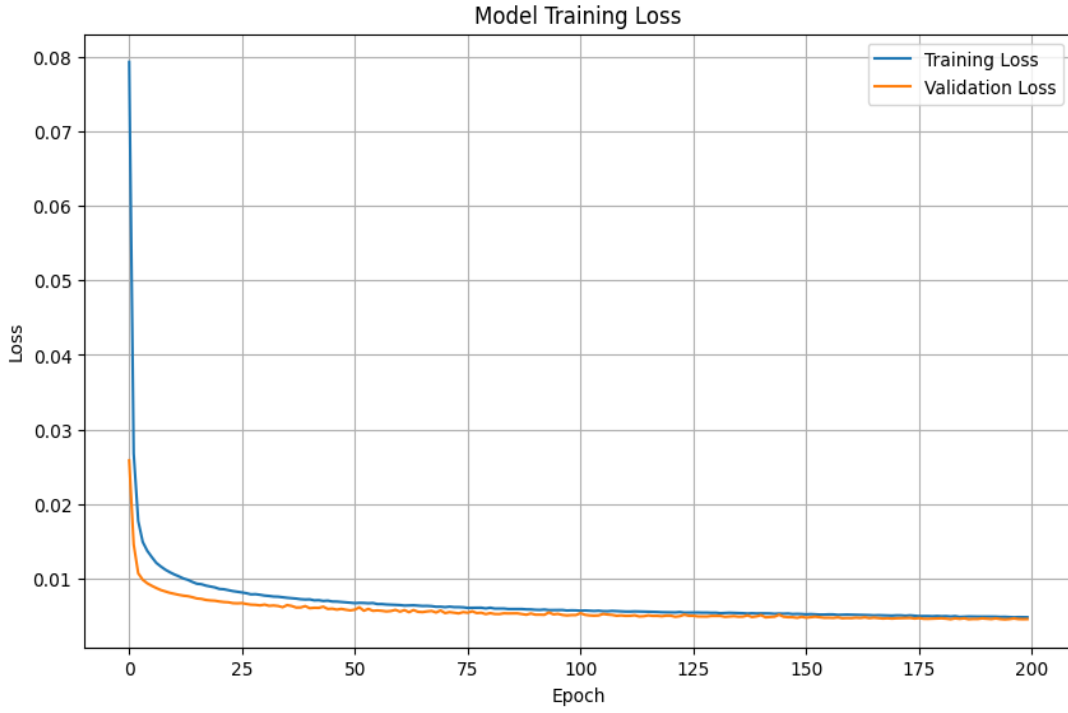
Giriş verileri (Columns to Analyze) tahmin için kullanılacak 10 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

**Tablo 4. 19.** Veri setlerinin boyutları ve yapıları

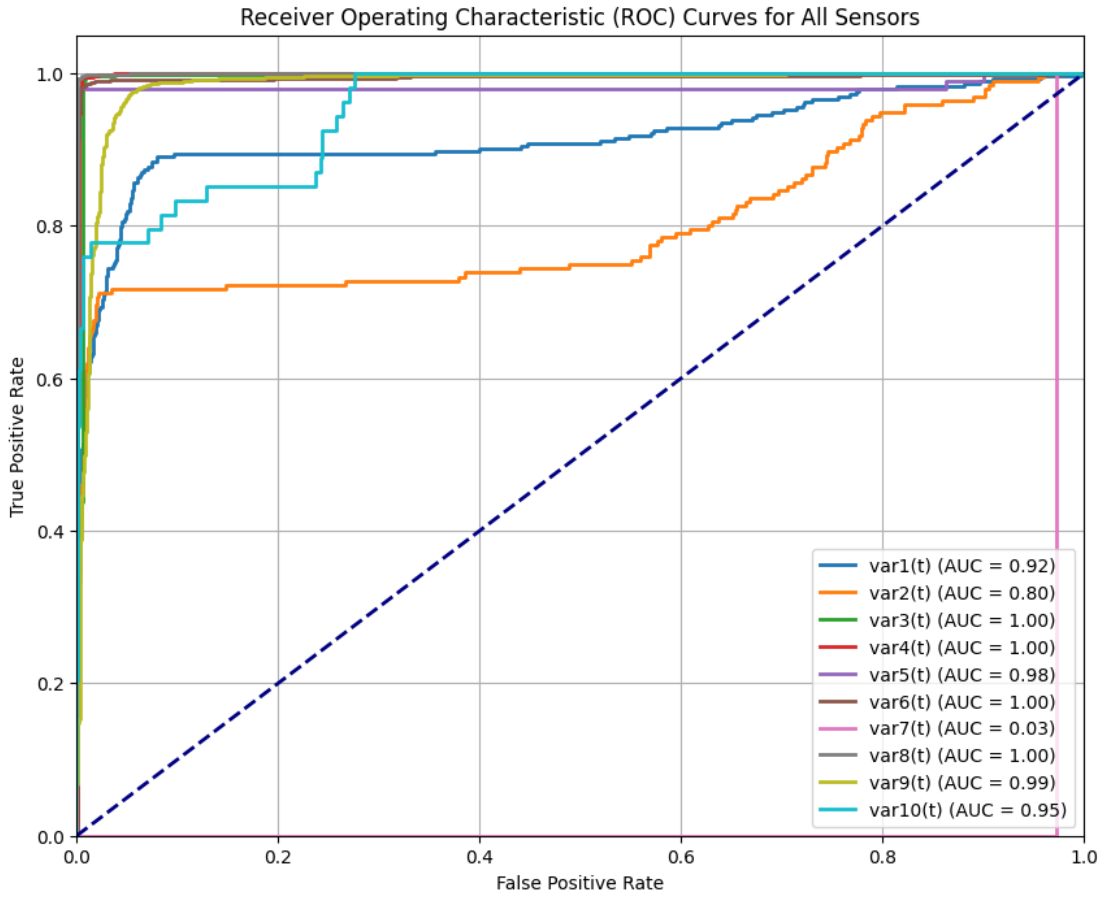
<b>Veri Seti</b>	<b>Satır Sayısı (n)</b>	<b>Sütun Sayısı (Zaman Dilimi x Özellik)</b>	<b>Boyutlar</b>
X_train	14228	4 x 10	(14228, 4, 10)
X_test	3558	4 x 10	(3558, 4, 10)
y_train_df	14228	10	(14228, 10)
y_test_df	3558	10	(3558, 10)

**Tablo 4. 20.** Deney 5, model performans değerleri

<b>Model</b>	<b>R2</b>	<b>RMSE</b>	<b>MAE</b>
LSTM	0.8674	0.0675	0.0262



Şekil 4. 10. Modelin öğrenme eğrisi (Loss Curve)



Şekil 4. 11. Modelin ROC eğrisi ve sınıflandırma başarısı

#### 4.6. Deney 6

Deney 6'da, Deney 5'teki uygulamalar aynen takip edilmiştir. Tamamıyla eksik (null) olan satırlar veri setinden çıkarılmış, kısmi eksik verilere sahip satırlar ise eksik veri işleme tekniklerinden biri olan "dropna" yöntemiyle silinmiştir. Bu deneyde test edilmek istenen ana faktör, giriş verilerine ek olarak yeni özelliklerin hesaplanarak veri setine dahil edilmesidir. Deney 4'teki tüm parametreler ve uygulamalara ek olarak, sensörlerden gelen ham verilere dayalı olarak Min (Minimum), Max (Maksimum), Mean (Ortalama), Var (Varyans), Std (Standart Sapma), Range (Aralık), Skew (Çarpıklık), Kurtosis (Basıklık) ve Rms (Kök Ortalama Kare) gibi istatistiksel özellikler hesaplanmış ve veri setine eklenmiştir. Bu işlem sonucunda her bir zaman dilimi için giriş özelliklerinin sayısı 19'a çıkmıştır. Ancak, burada önemli bir nokta, hesaplanan bu 19 giriş özelliğinin hedef değişken olmadığıdır. Tahmin edilmek istenen özellikler yalnızca gerçek sensör verileridir. Giriş özellikleri 19 iken, hedef özelliklerinin sayısı 10'dur.  $R^2$  değeri 0.9890, RMSE değeri 0.0674 ve MAE değeri ise 0.0097 olarak ölçülmüştür.

Deney 6' da, giriş verilerine ek olarak hesaplanan istatistiksel özelliklerin (Min, Max, Mean, Var, Std, Range, Skew, Kurtosis, Rms) eklenmesi modelin performansını önemli ölçüde artırdığı gözlemlenmiştir.

**Tablo 4. 21.** İstatistiksel özelliklerin eklenmesiyle oluşan deney 6 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39
	Min	Max	Mean	Var	Std	Range	Skew	Kurtosis	Rms	

**Tablo 4. 22.** Deney 6 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE

<b>Optimizer</b>	Adam (learning_rate=0.0001)
<b>Return Sequence</b>	False
<b>Target</b>	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
<b>n_minute</b>	8
<b>Train Size</b>	0.8
<b>Data Shape</b>	(17810, 20)
<b>Time Period</b>	4 months
<b>Columns to Analyze</b>	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39', 'Min', 'Max', 'Mean', 'Var', 'Std', 'Range', 'Skew', 'Kurtosis', 'Rms']
<b>n_in (Input Window Size)</b>	8
<b>n_out (Output Window Size)</b>	1
<b>Step (Overlap)</b>	4

Modelin eğitilmesi aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme(windowing) stratejisindeki değerler Tablo 4. 22’te verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri(overlap) ise 4 olarak belirlenmiştir.

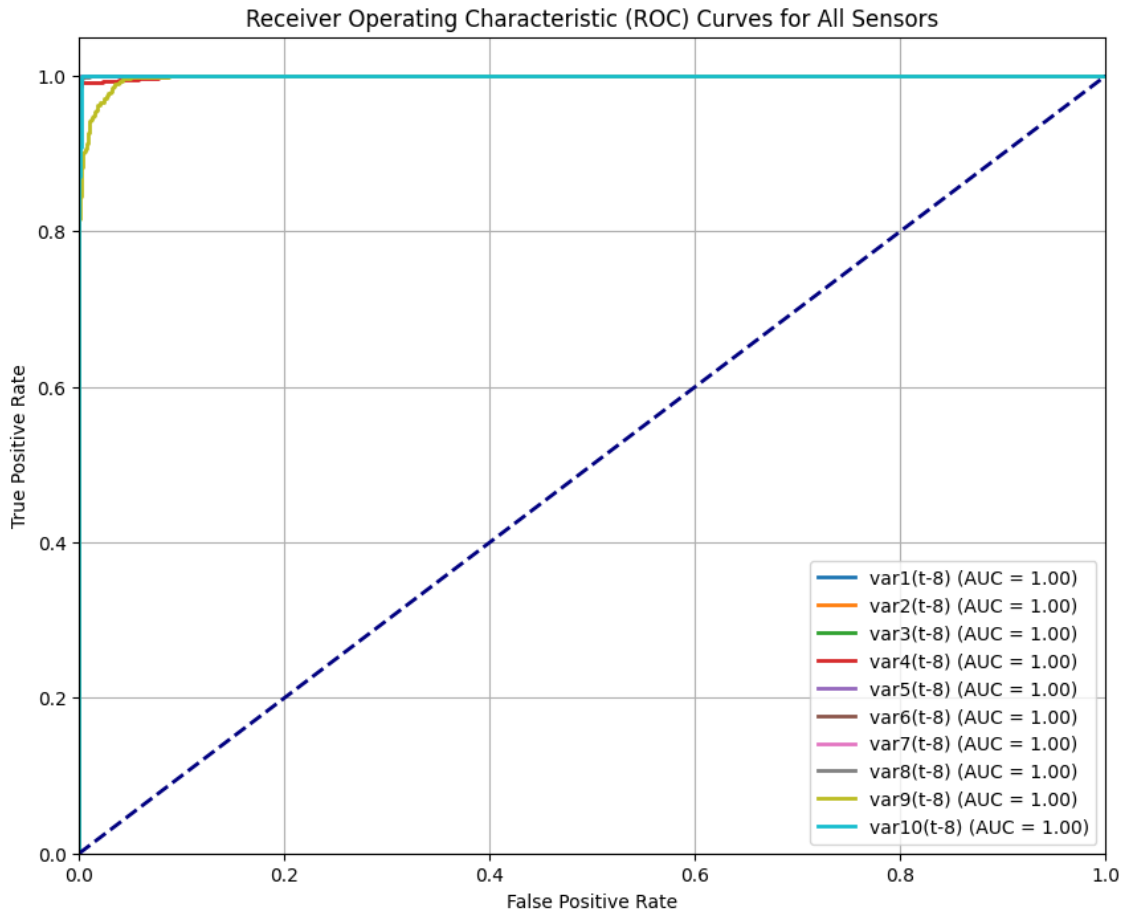
Giriş verileri(Columns to Analyze), tahmin için kullanılan 19 özelliği kapsamaktadır. Çıkış verileri(Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

**Tablo 4. 23.** Veri setlerinin boyutları ve yapıları

<b>Veri Seti</b>	<b>Satır Sayısı (n)</b>	<b>Sütun Sayısı (Zaman Dilimi x Özellik)</b>	<b>Boyutlar</b>
X_train	14225	8 x 19	(14225, 8, 19)
X_test	3557	8 x 19	(3557, 8, 19)
y_train_df	14225	10	(14225, 10)
y_test_df	3557	10	(3557, 10)

**Tablo 4. 24.** Deney 6, model performans deęerleri

Model	R2	RMSE	MAE
LSTM	0.9890	0.0674	0.0097



**Şekil 4. 8.** Modelin ROC eğrisi ve sınıflandırma başarısı

#### 4.7. Deney 7

Deney 7'de, deney 6'daki uygulamalar aynen takip edilmiştir. Tamamı eksik (Null) olan satırlar veri setinden çıkarılmış, kısmi eksik verilere sahip satırlar ise eksik veri işleme tekniklerinden biri olan "dropna" yöntemiyle silinmiştir. Bu deneyde test edilmek istenen durum, veri miktarının model sonuçları üzerindeki etkisidir. Deney 6'da 4 aylık veri (17810, 20) kullanılırken, deney 7'de 5 aylık veri (22640, 20) kullanılmıştır.

Giriş verisine ek olarak, yeni özellikler istatistiksel hesaplamalarla elde edilerek veri setine dahil edilmiştir. Eklenen yeni özellikler; Min (Minimum), Max (Maksimum), Mean (Ortalama), Var (Varyans), Std (Standart Sapma), Range (Aralık), Skew (Çarpıklık), Kurtosis (Basıklık), ve Rms (Kök Ortalama Kare) olup, her bir zaman dilimi

için 19 giriş özelliği elde edilmiştir. Burada önemli olan nokta, hesaplanan bu 19 giriş özelliğinin hedef değişken olmadığıdır. Tahmin edilmek istenen özellikler yalnızca gerçek sensör verileridir. Giriş özellikleri 19 iken, hedef özelliklerinin sayısı 10'dur. Modelin  $R^2$  değeri 0.9874, RMSE değeri 0.0193 ve MAE değeri ise 0.0129 olarak ölçülmüştür.

**Tablo 4. 25.** İstatistiksel özelliklerin eklenmesiyle oluşan deney 7 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39
	Min	Max	Mean	Var	Std	Range	Skew	Kurtosis	Rms	

**Tablo 4. 26.** Deney 7 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32
Loss Function	MSE
Optimizer	Adam (learning_rate=0.0001)
Return Sequence	False
Target	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
n_minute	8
Train Size	0.8
Data Shape	(22640, 20)
Time Period	5 months
Columns to Analyze	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39', 'Min', 'Max', 'Mean', 'Var', 'Std', 'Range', 'Skew', 'Kurtosis', 'Rms']
n_in (Input Window Size)	8
n_out (Output Window Size)	1
Step (Overlap)	4

Modelin eğitilmesi aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme (windowing) stratejisindeki değerler Tablo 4. 26’te verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri (overlap) ise 4 olarak belirlenmiştir.

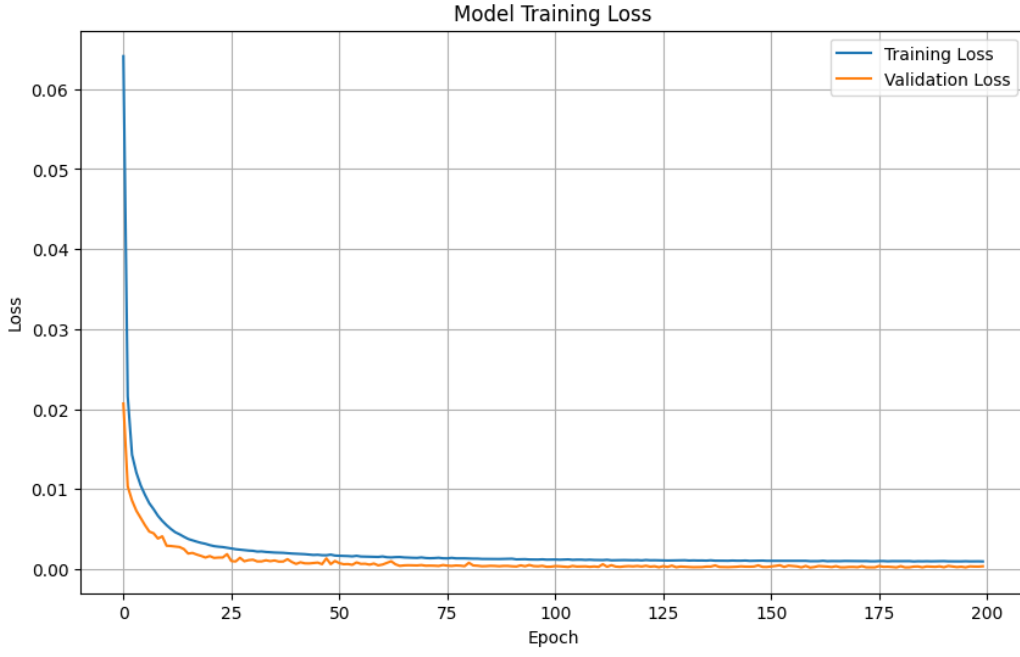
Giriş verileri (Columns to Analyze), tahmin için kullanılan 19 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

**Tablo 4. 27.** Veri setlerinin boyutları ve yapıları

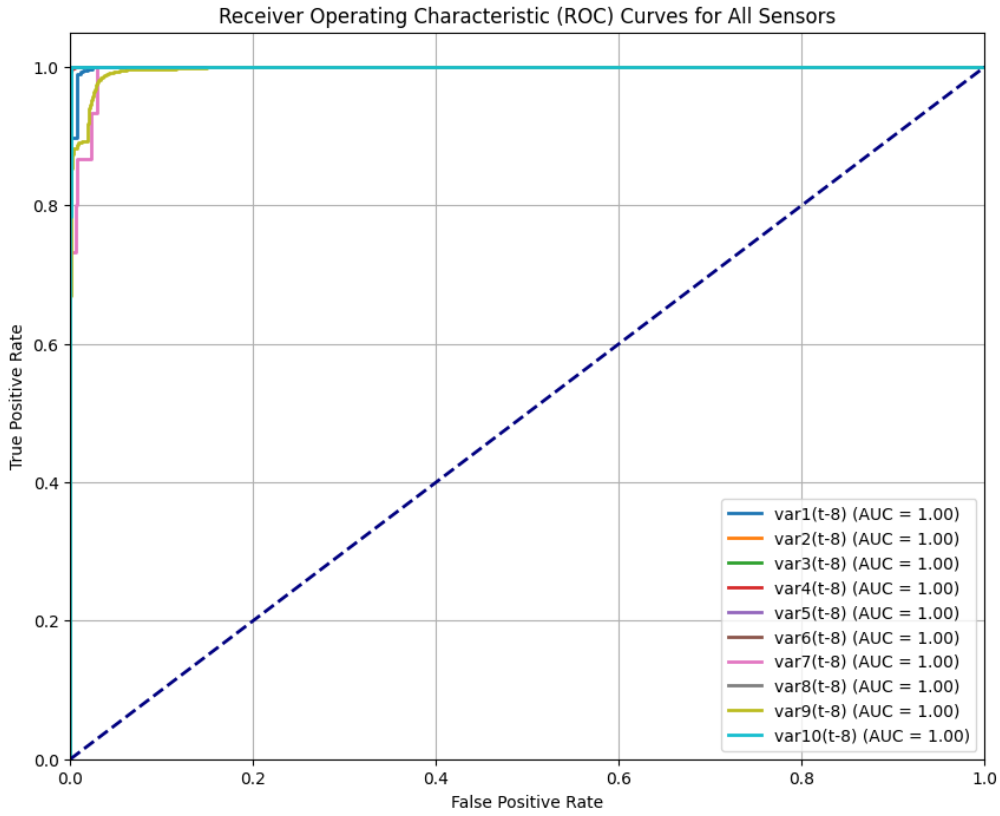
Veri Seti	Satır Sayısı (n)	Sütun Sayısı (Zaman Dilimi x Özellik)	Boyutlar
X_train	18088	8 x 19	(18088, 8, 19)
X_test	4523	8 x 19	(4523, 8, 19)
y_train_df	18088	10	(18088, 10)
y_test_df	4523	10	(4523, 10)

**Tablo 4. 28.** Deney 7, model performans değerleri

Model	R2	RMSE	MAE
LSTM	0.9874	0.0193	0.0129



Şekil 4. 9. Modelin öğrenme eğrisi (Loss Curve)



Şekil 4. 10. Modelin ROC eğrisi ve sınıflandırma başarısı

#### 4.8. Deney 8

Deney 8’de deney 7’deki uygulamalar üzerinden gidilmiştir. Tamamı eksik (Null) olan satırlar, veri setinden çıkarılmıştır. Kısmi olarak “null” değerlere sahip satırlar, eksik veri işleme tekniklerinden biri olan “dropna” yöntemiyle silinmiştir. 5 aylık veri (22640, 20) kullanılmıştır. Giriş verisine ek olarak yeni özellikler istatistiksel hesaplamalar sonucu hesaplanarak eklenmiştir. Eklenen yeni özellikler; Min (Minimum), Max (Maksimum), Mean (Ortalama), Var (Varyans), Std (Standart Sapma), Range (Aralık), Skew (Çarpıklık), Kurtosis (Basıklık), Rms (Kök Ortalama Kare) dir. Böylece her bir zaman dilimi giriş için 19 özelliğe sahiptir. Burada dikkat edilmesi gereken nokta hesaplamalar sonucu elde edilen 19 giriş özelliğinin hedef değişken olmamasıdır. Tahmin edilmek istenen özellikler sadece gerçek sensör özellikleridir. Giriş özellikleri 19 iken hedef özelliklerinin sayısı 10 dur.

Deney 8 kapsamında, modelin performansını artırmak amacıyla Keras Tuner kütüphanesi ve Hyperband yöntemi kullanılarak en iyi hiperparametre kombinasyonları belirlenmiştir. Hiperparametrelerin otomatik olarak seçilmesi için 10 farklı model denemesi yapılmıştır.

R2 değeri 0.9866, RMSE değeri 0.0174 ve MAE değeri 0.0114 olarak elde edilmiştir.

**Tablo 4. 29.** İstatistiksel özelliklerin eklenmesiyle oluşan deney 8 giriş özellikleri

Özellikler										
Datetime	A1	A23	A11	A26	A15	A35	A2	A36	A21	A39
	Min	Max	Mean	Var	Std	Range	Skew	Kurtosis	Rms	

**Tablo 4. 30.** Deney 8 parametreleri

Parameter	Value
Layer Size	2
Units	80
SEED	123
Learning Rate	0.0001
Layer	LSTM
Activation Function	Relu
Dropout Rate	0.1
Epochs	200
Batch Size	32

<b>Loss Function</b>	MSE
<b>Optimizer</b>	Adam (learning_rate=0.0001)
<b>Return Sequence</b>	False
<b>Target</b>	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39']
<b>n_minute</b>	8
<b>Train Size</b>	0.8
<b>Data Shape</b>	(22640, 20)
<b>Time Period</b>	5 aylık veri
<b>Columns to Analyze</b>	['A1', 'A11', 'A15', 'A2', 'A21', 'A23', 'A26', 'A35', 'A36', 'A39', 'Min', 'Max', 'Mean', 'Var', 'Std', 'Range', 'Skew', 'Kurtosis', 'Rms']
<b>n_in (Input Window Size)</b>	8
<b>n_out (Output Window Size)</b>	1
<b>Step (Overlap)</b>	4

Modelin eğitim aşamasında kullanılan parametreler, giriş değişkenleri, hedef değişkenleri, pencereleme(windowing) stratejisindeki değerler Tablo 4. 30'de verilmiştir. Burada modelin her tahmin için gözlemleyeceği geçmiş zaman dilimi sayısı 8 dir. Modelin tahmin edeceği sonraki zaman dilimi sayısı 1 dir. Pencerenin her adımda ne kadar kaydırılacağı değeri(overlap) ise 4 olarak belirlenmiştir.

Giriş verileri (Columns to Analyze) tahmin için kullanılacak 19 özelliği kapsamaktadır. Çıkış verileri (Target) ise modelin tahmin edeceği 10 hedef değişkeni temsil etmektedir.

**Tablo 4. 31.** Veri setlerinin boyutları ve yapıları

<b>Veri Seti</b>	<b>Satır Sayısı (n)</b>	<b>Sütun Sayısı (Zaman Dilimi x Özellik)</b>	<b>Boyutlar</b>
X_train	18088	8 x 19	(18088, 8, 19)
X_test	4523	8 x 19	(4523, 8, 19)
y_train_df	18088	10	(18088, 10)
y_test_df	4523	10	(4523, 10)

Deney 8 kapsamında, modelin performansını artırmak amacıyla Keras Tuner ile hiperparametreler optimize edilmiştir. Optimizasyon için kullanılan hiperparametreler ve değer aralıkları şu şekilde belirlenmiştir:

- **units:** 32, 64, 96 ve 128 aralıklarında optimizasyon yapılmıştır.
- **activation:** 'relu', 'tanh' ve 'sigmoid' aktivasyon fonksiyonları arasında seçim yapılmıştır.
- **dropout\_rate:** 0.1 ile 0.5 arasında, adım olarak 0.1 değerleriyle optimize edilmiştir.
- **units\_2:** 32, 64, 96 ve 128 aralıklarında optimize edilmiştir.
- **activation\_2:** 'relu', 'tanh' ve 'sigmoid' aktivasyon fonksiyonları arasında seçim yapılmıştır.
- **dropout\_rate\_2:** 0.1 ile 0.5 arasında, adım olarak 0.1 değerleriyle optimize edilmiştir.
- **optimizer:** 'adam', 'sgd' ve 'adagrad' optimizasyon algoritmaları arasında seçim yapılmıştır.
- **learning\_rate:** 0.01, 0.001 ve 0.0001 değerleriyle optimize edilmiştir.

**Tablo 4. 32.** Keras Tuner ile optimizasyonu yapılan hiperparametreler ve değer aralıkları

Hiperparametre	Optimize Edilen Değer Aralığı
<b>units</b>	32, 64, 96, 128
<b>activation</b>	'relu', 'tanh', 'sigmoid'
<b>dropout_rate</b>	0.1 ile 0.5 (adım: 0.1)
<b>units_2</b>	32, 64, 96, 128
<b>activation_2</b>	'relu', 'tanh', 'sigmoid'
<b>dropout_rate_2</b>	0.1 ile 0.5 (adım: 0.1)
<b>optimizer</b>	'adam', 'sgd', 'adagrad'
<b>learning_rate</b>	0.01, 0.001, 0.0001

En iyi hiperparametrelerin (Tablo 4. 33) belirlenmesinin ardından, bu parametrelerle model yeniden eğitilerek, optimize edilen yapı üzerinden daha yüksek doğruluk ve güvenilirlik elde edilmesi hedeflenmiştir. Bu süreç, modelin performansını artırmak için gerekli parametrelerin doğru bir şekilde yapılandırılmasını sağlamaktadır.

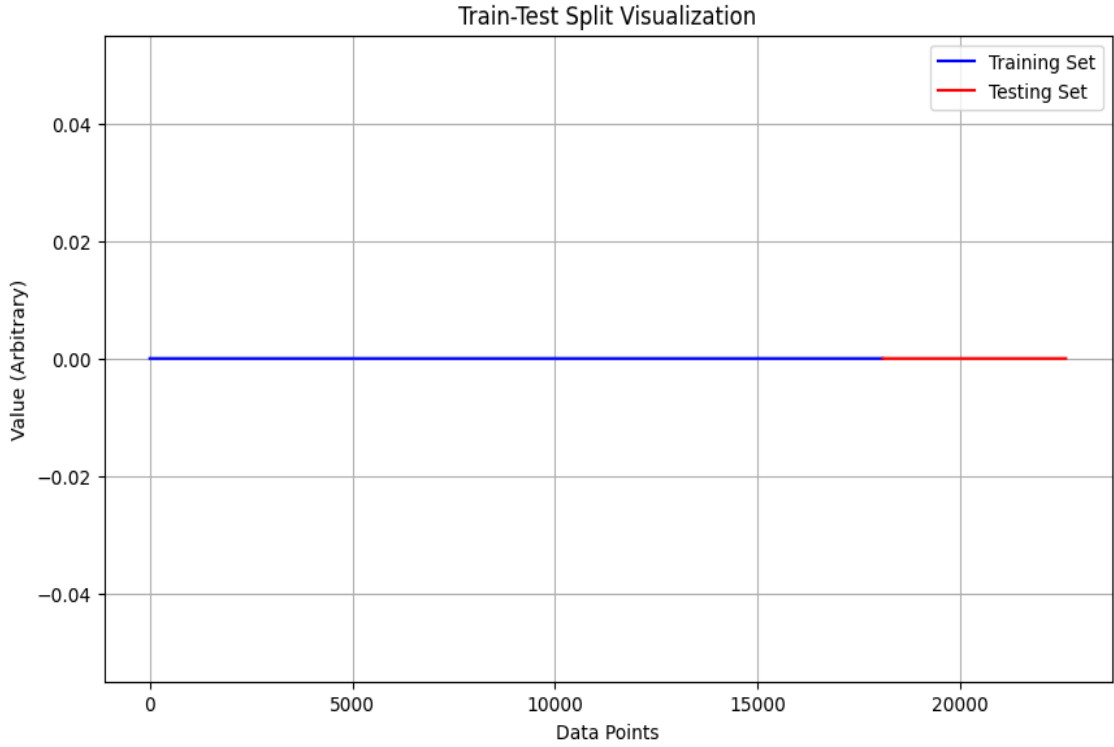
**Tablo 4. 33.** Keras Tuner ile elde edilen en iyi hiperparametre deęerleri

Parameter	Value
Units	32
Learning Rate	0.001
Dropout Rate	0.2
Activation	tanh
Units_2	128
Activation_2	tanh
Dropout Rate_2	0.4
Optimizer	Adam

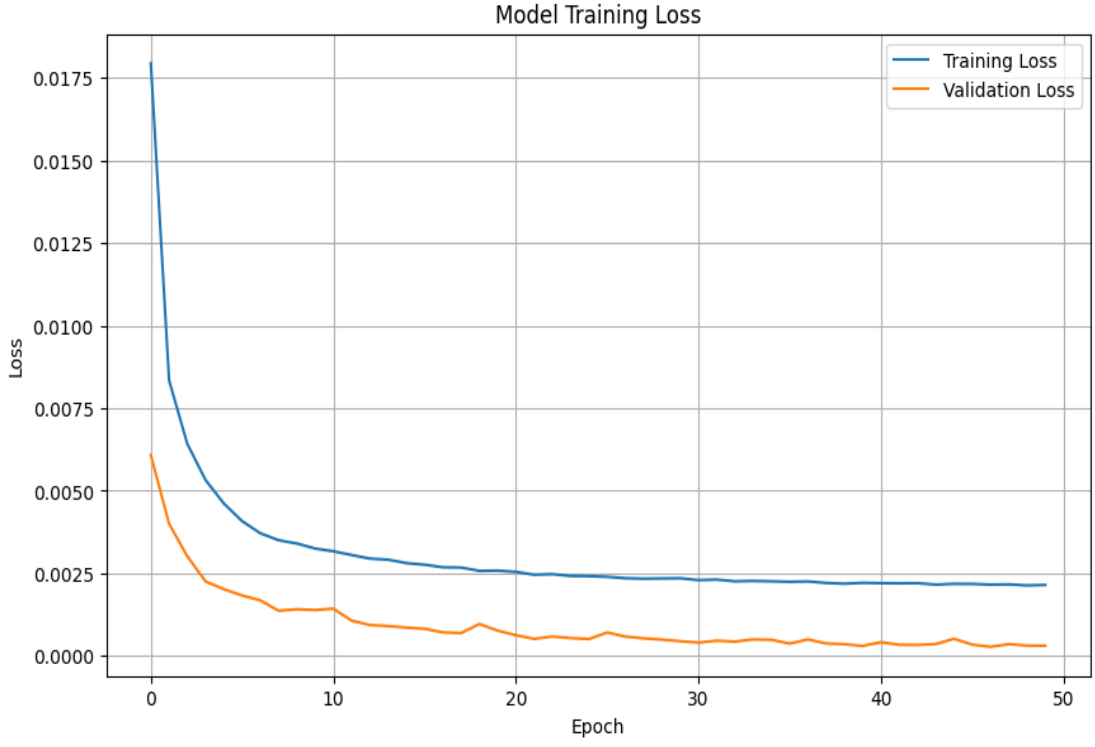
**Tablo 4. 34.** Deney 8, model performans deęerleri

Model	R2	RMSE	MAE
LSTM	0.9866	0.0174	0.0114

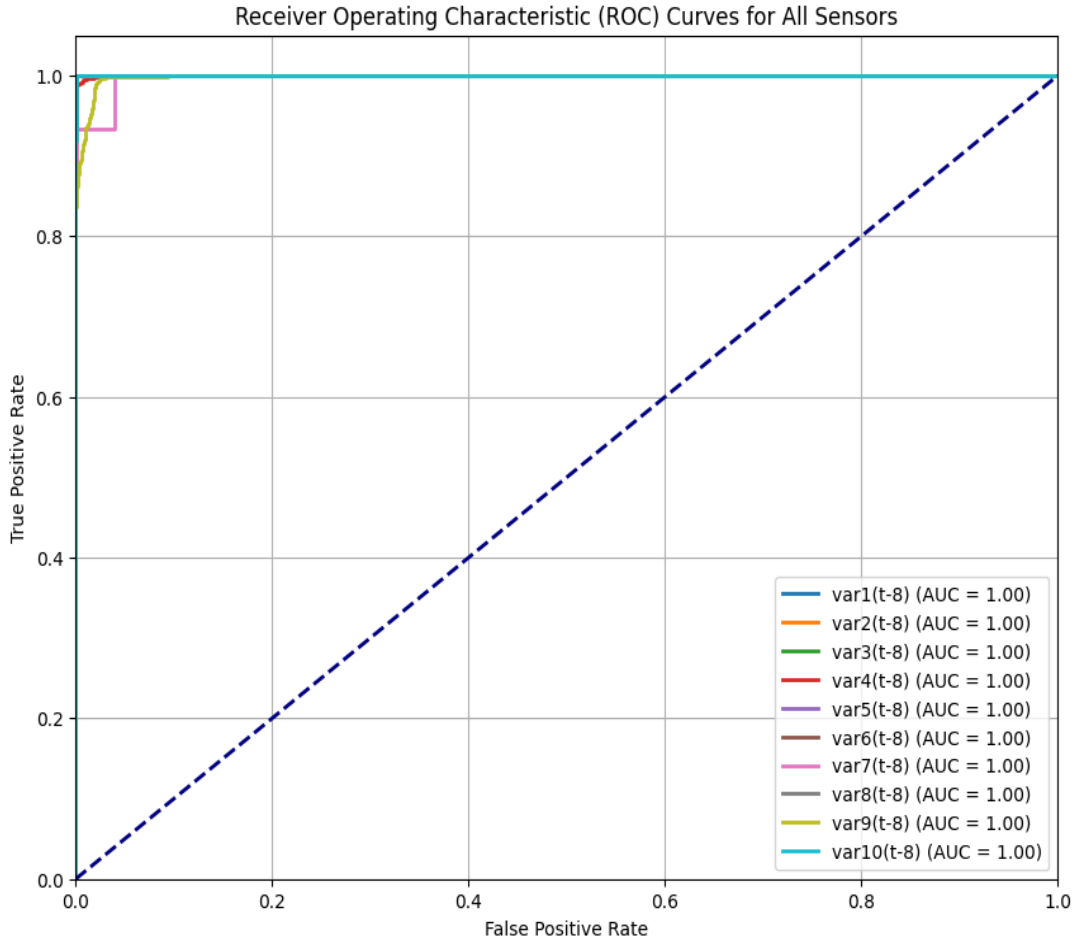
Veri seti %80 eęitim ve %20 test verisi olarak 2 ye ayrılmıřtır (řekil 4. ).



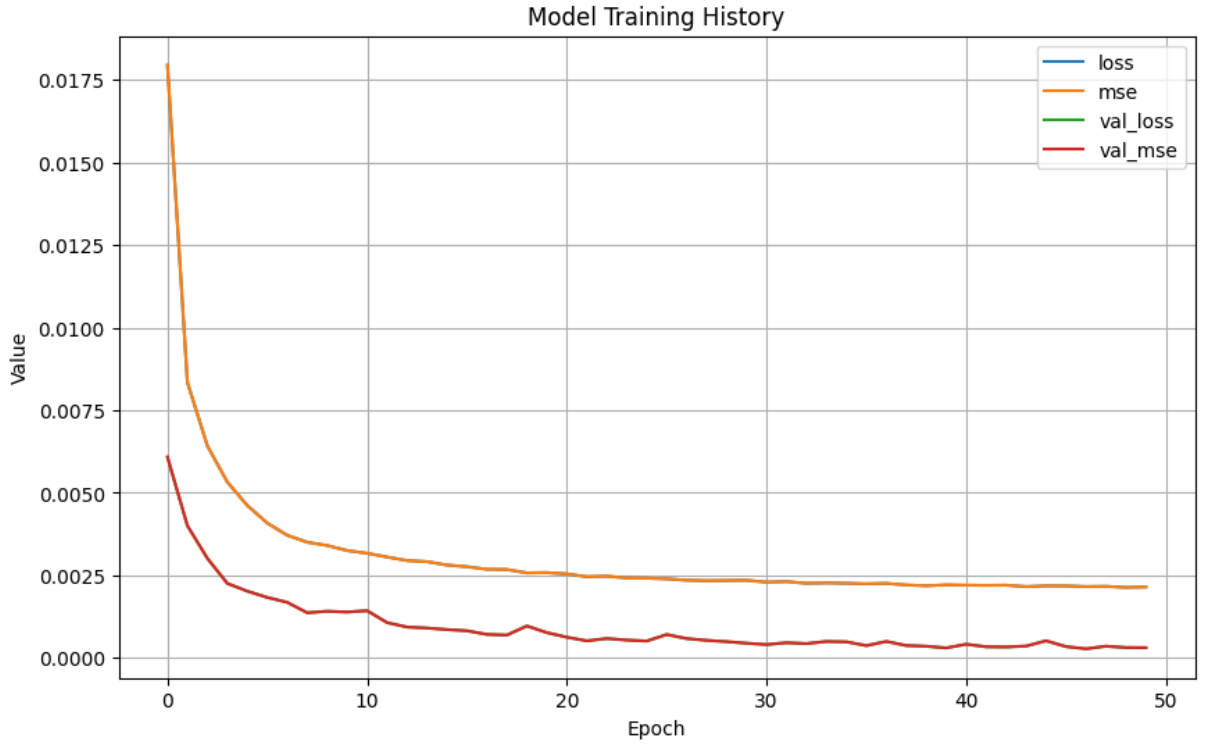
**řekil 4. 15.** Veri setinin eęitim ve test verisi olarak ayrılması



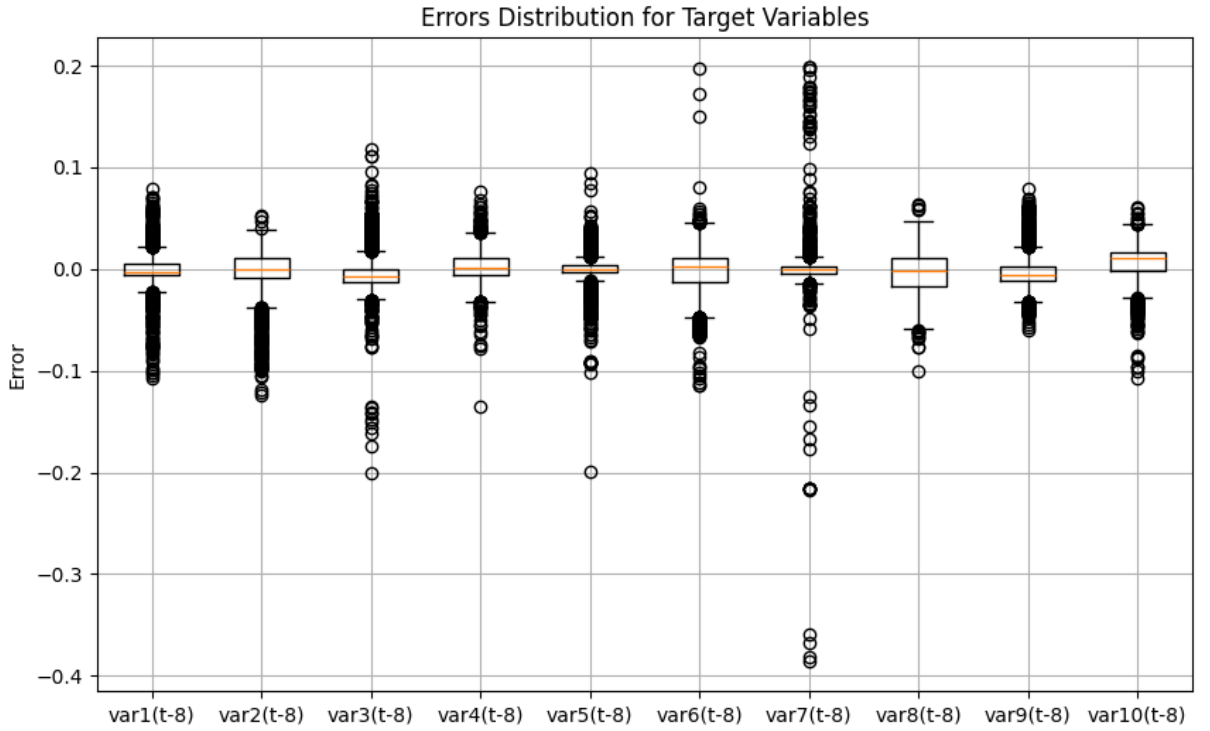
Şekil 4. 16. Önerilen modelin öğrenme eğrisi (Loss Curve)



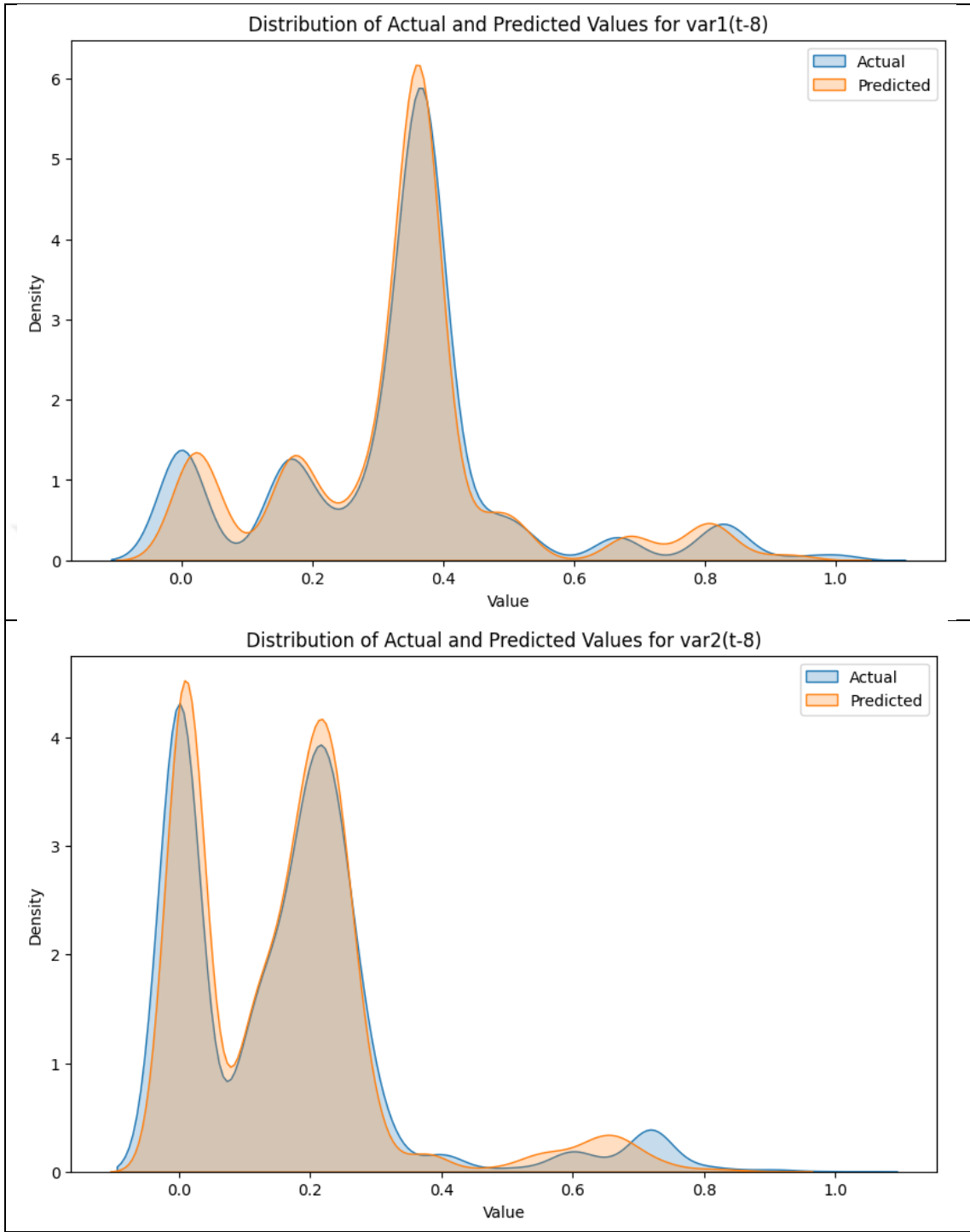
Şekil 4. 17. Önerilen modelin ROC eğrisi ve sınıflandırma başarısı

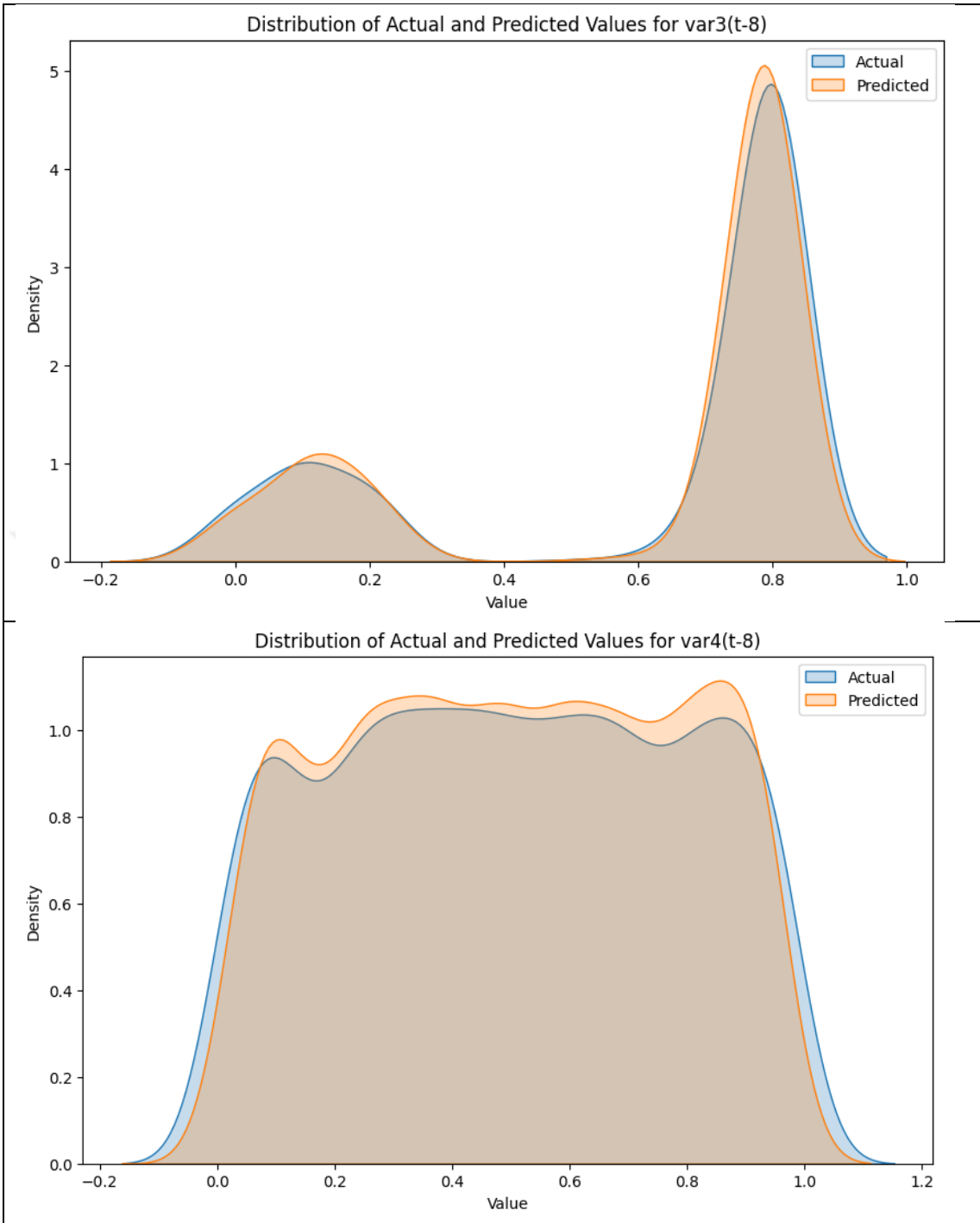


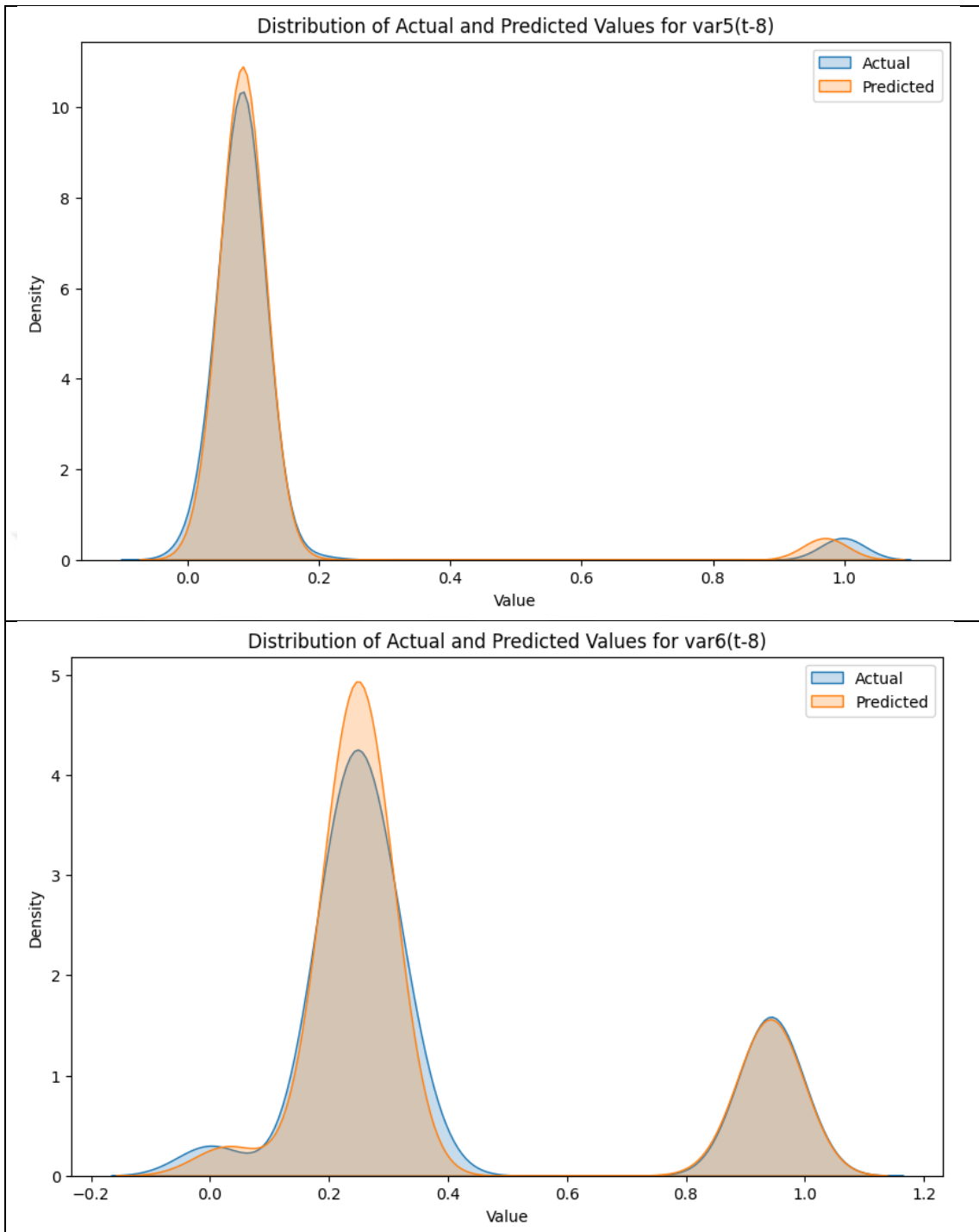
Şekil 4. 18. Model eğitim geçmişi

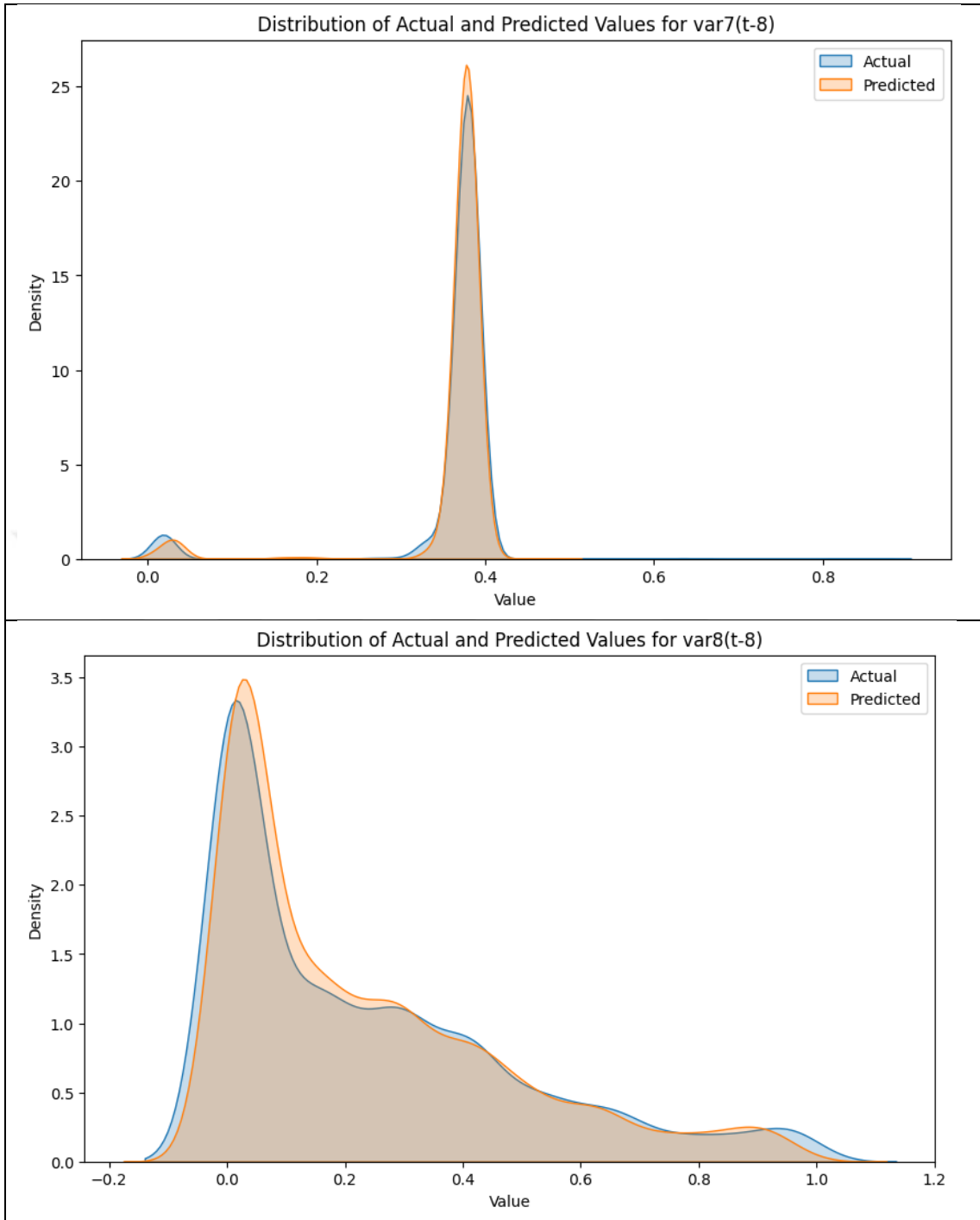


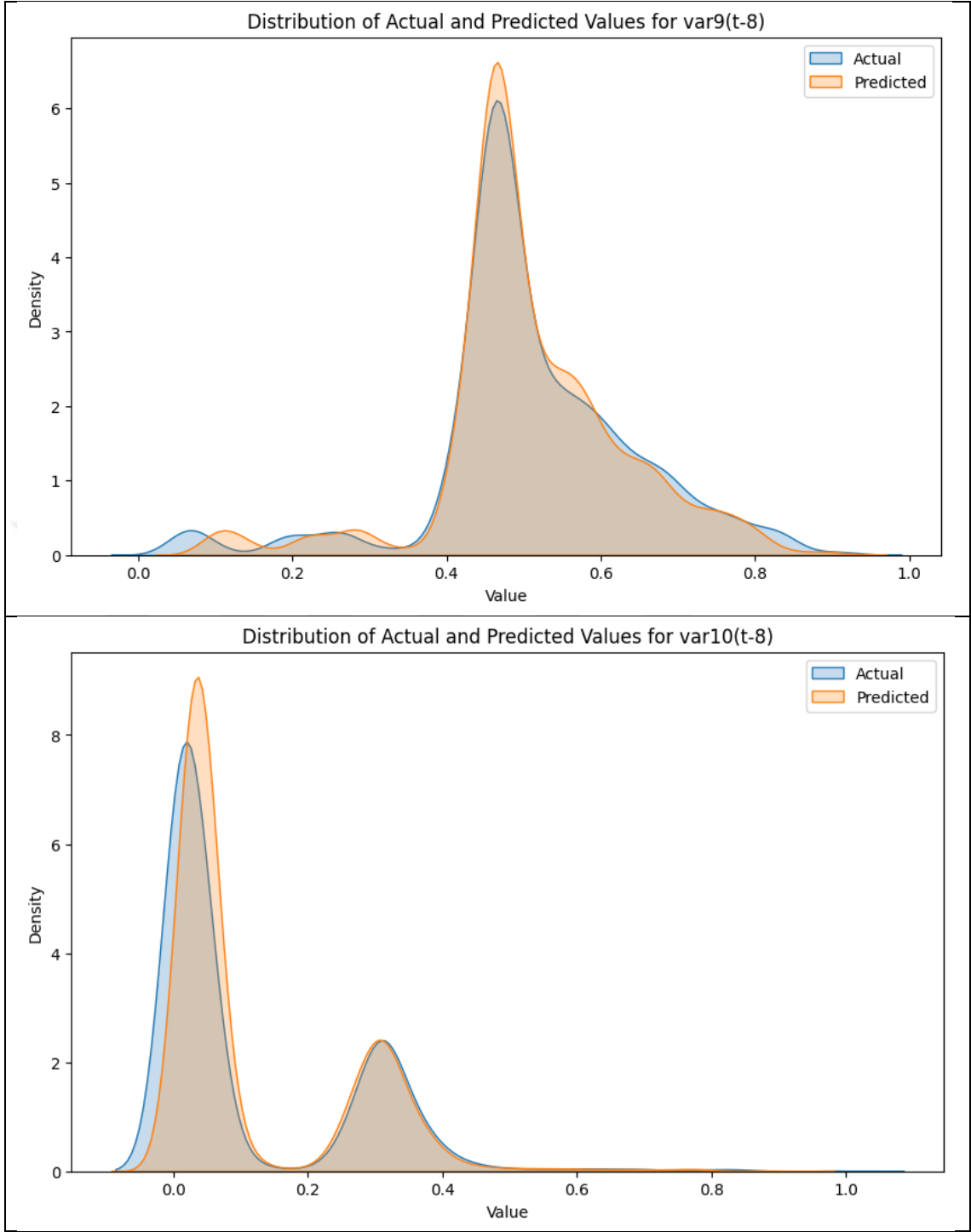
Şekil 4. 19. Hedef değişkenler için hata dağılımı











Şekil 4. 20. Model tahminleri ve değişkenlerin gerçek değerlerinin yoğunluk dağılımı

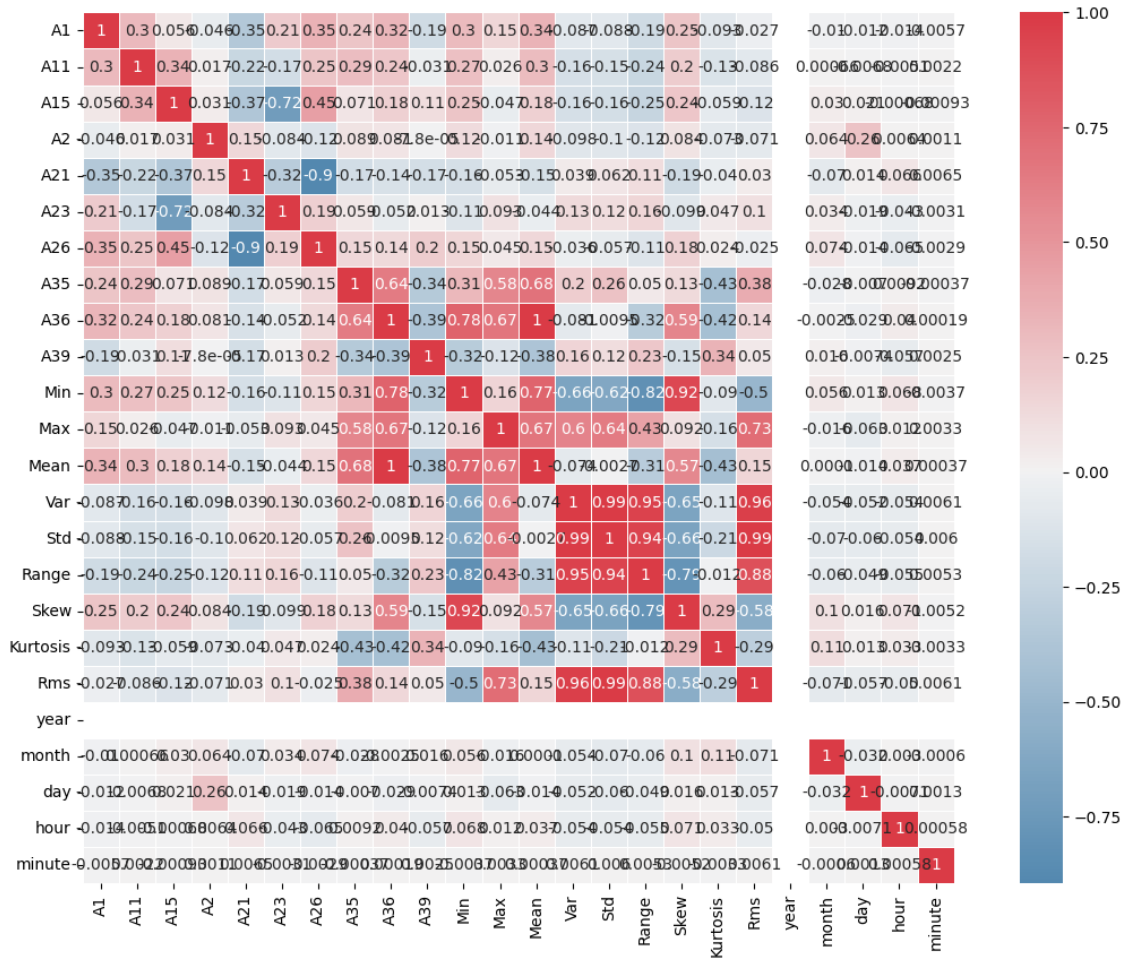
## 5. SONUÇLAR VE ÖNERİLER

### 5.1 Sonuç ve öneriler

Çalışmadaki en kritik süreçlerden biri, veri toplama ve hazırlama işlemidir. Makineden gerçek zamanlı sensör verilerini toplamak için uzun zaman aralığı gereklidir. Çalışmada, Mayıs, Haziran, Temmuz, Ağustos ve Eylül ayları boyunca, her 9 dakikalık aralıklarla toplanan toplam 153 günlük veri seti incelenmiştir. Veriler, başlangıçta kaydedildiği şekliyle, işlenmemiş, düzenlenmemiş ve filtrelenmemiş olan ham verilerdir. 5 ayrı Excel dosyasında tutulan ve 10 ayrı sensörden toplanan bu ham verilere, verinin modelleme için uygun olup olmadığını değerlendirmek adına keşifsel veri analizi uygulanmıştır. Bu kapsamda veriler tabular veri formatına dönüştürülmüştür. Ayrı ayrı tutulan aylar birleştirilmiştir. Bazı günlere ait hiç kayıt tutulmadığı için 'Null' değer içeren eksik kayıtların tamamı silinmiştir.

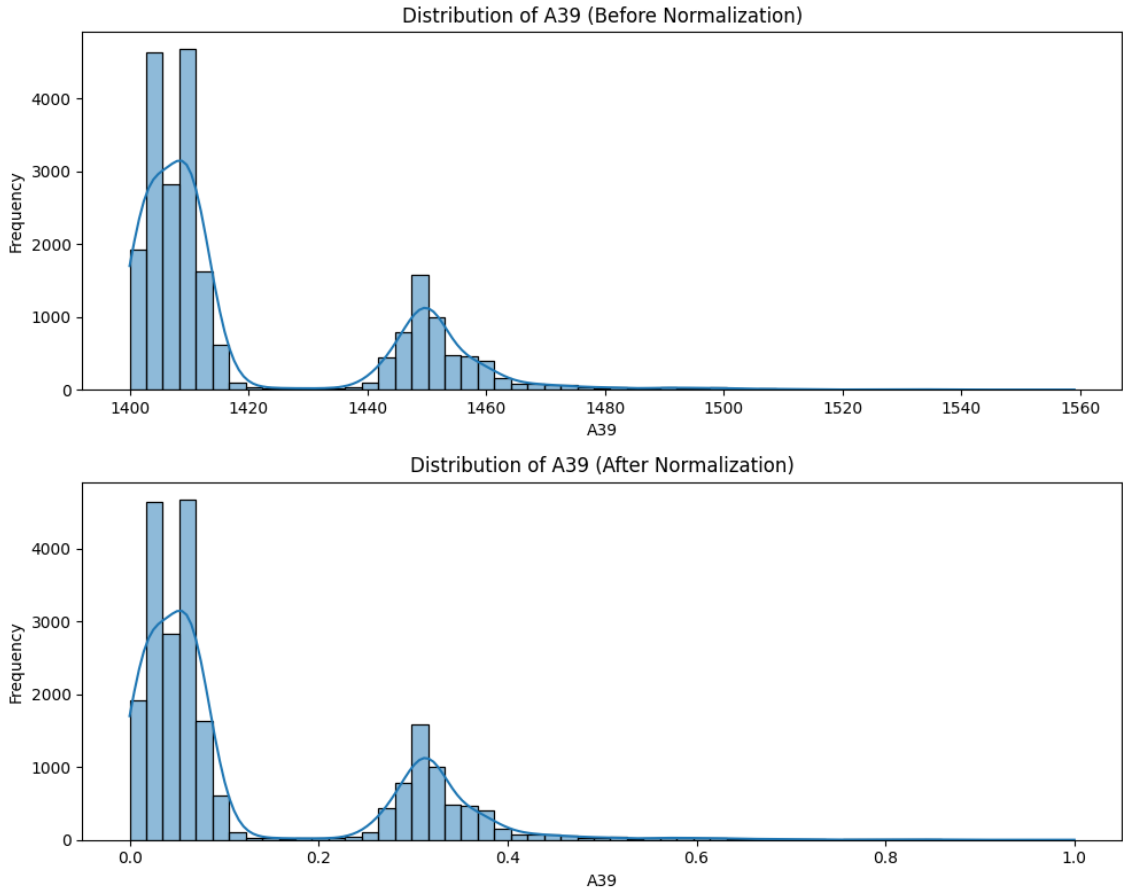
Verinin temel istatistik özeti çıkarılmış ve veri setindeki uç değerler gözlemlenmiş ve anomali tespiti için dayanak noktası oluşturulmuştur.

Veri setindeki değişkenler arasındaki doğrusal ilişkiyi anlayıp analiz edebilmek için korelasyon matrisi çıkarılmıştır.



Şekil 5. 1. Önerilen modelin korelasyon matrisi

Veri setinin geçmiş “n\_minute” zaman dilimindeki değerleri ve tahmin edilecek hedef değişkenleri içerecek şekilde çerçevesi sağlanmıştır. Ayrıca keşifsel veri analizine ek olarak veri ön işleme kapsamında zaman serisi verisinin normalizasyonu yapılmıştır. Örneğin sensör verilerinden A39 verileri için normalizasyon yapılmadan önce ve sonraki değerleri Şekil 5.2’te sunulmuştur.



**Şekil 5.2.** Normalizasyondan önce ve sonra A39 özelliğinin değerleri

Veri seti modelin etkinliğini değerlendirebilmek adına %80’i eğitim ve %20 si test olmak üzere ikiye ayrılmıştır. Shuffle=False parametresi ile zaman serilerindeki ardışıklığın korunması için veriler karıştırılmamıştır.

Bu noktaya kadar yapılan işlemler tüm deneyler için ortaktır. Sonrasında yapılan deneylerin sonuçları şu şekildedir.

### **Deney sonuçları**

Deney sonuçlarına göre, eksik verilerin “0” ile doldurulduğu deney 1’de modelin doğruluk seviyesi düşük kalmıştır. Eksik verilerin “dropna” yöntemi ile tamamen silindiği deney 2’de model performansı belirgin şekilde iyileşmiştir. Benzer şekilde, eksik verilerin sütun ortalamaları ile doldurulduğu deney 3’te de modelin doğruluğu yüksektir. Deney 2 ve deney 3 benzer performans göstermiştir.

Overlap miktarının değiştirilmesinin hata metrikleri üzerindeki etkisinin oldukça küçük olduğu gözlemlenmiştir. Aynı şekilde giriş penceresi boyutunun değiştirilmesi de sonucu belirgin şekilde değiştirmemiştir.

İstatistiksel özelliklerin giriş verilerine eklenmesi, modelin genel performansını önemli ölçüde iyileştirmiştir. Modelin tahmin başarısı artmıştır. Tüm değerlendirme metriklerinin değerleri incelendiğinde, hata oranlarında kayda değer azalma gözlemlenmiştir. Bu durum, zaman serisi tabanlı analizlerde veri ön işleme aşamasında istatistiksel özelliklerin dahil edilmesinin, model performansı üzerinde kritik bir etkiye sahip olduğunu ortaya koymaktadır.

Burada dikkat edilmesi gereken nokta şudur: Eklenen yeni özellikler modelin eğitimi esnasında kullanılırken, hedef değişkenler arasında olmamalıdır. Tahmin değişkenleri sadece sensörlerden gelen özellikler olmalıdır.

Keras Tuner kütüphanesi kullanılarak Hyperband yöntemi ile hiperparametre optimizasyonu gerçekleştirilmiştir. Keras Tuner, geniş bir hiperparametre aralığında en uygun kombinasyonları belirlemek için kullanılmıştır. Böylece manuel denemeler yerine otomatik optimizasyon sağlayarak model geliştirme daha otomatik hale getirilmiştir. Hyperband yöntemi ise, rastgele seçilen hiperparametre kombinasyonlarını adım adım değerlendirmiştir ve düşük performans gösterenleri erken aşamalarda dışarıda bırakarak, hesaplama kaynaklarını daha verimli bir şekilde kullanılmasını sağlamıştır. Tablo 5. 1’de tüm deneylerin performans değerleri verilmiştir.

**Tablo 5. 1.** Model performans değerleri: deney sonuçları

	<b>R2</b>	<b>RMSE</b>	<b>MAE</b>
<b>Deney 1</b>	0.7909	0.0719	0.0265
<b>Deney 2</b>	0.8676	0.0674	0.0258
<b>Deney 3</b>	0.8646	0.0685	0.0265
<b>Deney 4</b>	0.8676	0.0674	0.0258
<b>Deney 5</b>	0.8674	0.0675	0.0262
<b>Deney 6</b>	0.9890	0.0674	0.0097
<b>Deney 7</b>	0.9874	0.0193	0.0129
<b>Deney 8</b>	0.9866	0.0174	0.0114

Önleyici bakım gibi geleneksel bakım yöntemleri ekipmanların mevcut durumunu göz önünde bulundurmadıkları için her zaman verimli olmayabilirler. Bu çalışmada silisyum eritme ve kristal büyütme fırınındaki sensör verilerinin analiziyle prediktif bakım yaklaşımı incelenmiştir. Bakım stratejileri oluşturmak için makine öğrenmesi

tabanlı anomali tespit modelinin potansiyel faydası vurgulanmıştır. Bu modeller önemli bir arıza yaşanmadan önce, yaklaşan arızayı tespit etmek için öneri sunabilir. Böylece ekipmanın güvenilirliğini artırma, plansız duruş sürelerini azaltma ve ekonomik maliyetleri azaltma noktasında büyük faydalar sağlar. Geliştirilen LSTM tabanlı anomali tespit modeli, PV panel üretim fabrikasının bel kemiği niteliğindeki fırının olası arızasını tespit etmek için kullanılabilir. Model zaman verilerinin tahmininde %98'lik doğruluk oranı sunmaktadır.

Bu çalışmada LSTM modeli üzerine çalışılmıştır. Süreç boyunca yapılan araştırmalar sonucu LSTM Autoencoder yapısının da zaman serisi verilerinde, verilerin normal modellerini öğrenme ve bu modelden sapmaları tespit etmek için etkili olduğu bilgisine varılmıştır. Gelecekteki çalışmada LSTM Autoencoder yapısı ile modelleme yaparak anomali tespiti planlanmaktadır. Sonraki aşamalarda geliştirilen modelin gerçek zamanlı olarak üretim sürecine entegresinin uygunluğu değerlendirilecektir.

## 6. KAYNAKÇA

19. *Hyperparameter Optimization—Dive into Deep Learning 1.0.3 documentation.* (t.y.).

Geliş tarihi 11 Aralık 2024, gönderen [https://d2l.ai/chapter\\_hyperparameter-optimization/index.html](https://d2l.ai/chapter_hyperparameter-optimization/index.html) *Ai100report10032016fnl\_singles.* (t.y.).

Akkuzu Kaya, G., & Yıldız, M. (2023). Time Series Anomaly Detection Embedded Systems By Using LSTM. *International Journal of Multidisciplinary Studies and Innovative Technologies*, 7(2), 90. <https://doi.org/10.36287/ijmsit.7.2.90>

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), 53. <https://doi.org/10.1186/s40537-021-00444-8>

Arena, S., Florian, E., Sgarbossa, F., Sølvsberg, E., & Zennaro, I. (2024). A conceptual framework for machine learning algorithm selection for predictive maintenance. *Engineering Applications of Artificial Intelligence*, 133, 108340. <https://doi.org/10.1016/j.engappai.2024.108340>

Baldi, P., & Sadowski, P. J. (2013). Understanding Dropout. *Advances in Neural Information Processing Systems*, 26. [https://papers.nips.cc/paper\\_files/paper/2013/hash/71f6278d140af599e06ad9bf1ba03cb0-Abstract.html](https://papers.nips.cc/paper_files/paper/2013/hash/71f6278d140af599e06ad9bf1ba03cb0-Abstract.html)

Bampoula, X., Nikolakis, N., & Alexopoulos, K. (2024). Condition Monitoring and Predictive Maintenance of Assets in Manufacturing Using LSTM-Autoencoders and Transformer Encoders. *Sensors*, 24(10), 3215. <https://doi.org/10.3390/s24103215>

- Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is Machine Learning? A Primer for the Epidemiologist. *American Journal of Epidemiology*, kwz189. <https://doi.org/10.1093/aje/kwz189>
- Bousdekis, A., Apostolou, D., & Mentzas, G. (2020). Predictive Maintenance in the 4th Industrial Revolution: Benefits, Business Opportunities, and Managerial Implications. *IEEE Engineering Management Review*, 48(1), 57-62. <https://doi.org/10.1109/EMR.2019.2958037>
- Carneiro, T., Medeiros Da Nóbrega, R. V., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, 6, 61677-61685. <https://doi.org/10.1109/ACCESS.2018.2874767>
- Chalapathy, R., & Chawla, S. (2019). *Deep Learning for Anomaly Detection: A Survey* (Versiyon 2). arXiv. <https://doi.org/10.48550/ARXIV.1901.03407>
- Chen, H., Hsu, H.-Y., Hsieh, J.-Y., & Hung, H.-E. (2024). A differential privacy-preserving federated learning scheme with predictive maintenance of wind turbines based on deep learning for feature compression and anomaly detection with state assessment. *Journal of Mechanical Science and Technology*, 38(7), 3413-3429. <https://doi.org/10.1007/s12206-024-0616-9>
- Choi, H., Kim, S., & Kang, P. (2023). Recurrent auto-encoder with multi-resolution ensemble and predictive coding for multivariate time-series anomaly detection. *Applied Intelligence*, 53(21), 25330-25342. <https://doi.org/10.1007/s10489-023-04764-5>
- Contents. (t.y.). Ar5iv. Geliş tarihi 06 Aralık 2024, gönderen <https://ar5iv.labs.arxiv.org/html/2105.06756>

- El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? İçinde I. El Naqa, R. Li, & M. J. Murphy (Ed.), *Machine Learning in Radiation Oncology: Theory and Applications* (ss. 3-11). Springer International Publishing. [https://doi.org/10.1007/978-3-319-18305-3\\_1](https://doi.org/10.1007/978-3-319-18305-3_1)
- Güven, G., & Bilhan, A. (2024). Güneş Panellerinin Verimliliğini Üretim ve Kurulum Sürecinde Etkileyen Faktörlerin İncelenmesi. *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*. <https://doi.org/10.28948/ngumuh.1535918>
- Hu, L., & Dai, G. (2022). Estimate remaining useful life for predictive railways maintenance based on LSTM autoencoder. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-021-06051-1>
- Iqbal, A., & Amin, R. (2024). Time series forecasting and anomaly detection using deep learning. *Computers & Chemical Engineering*, 182, 108560. <https://doi.org/10.1016/j.compchemeng.2023.108560>
- Islam, S., Elmekki, H., Elsebai, A., Bentahar, J., Drawel, N., Rjoub, G., & Pedrycz, W. (2024). A comprehensive survey on applications of transformers for deep learning tasks. *Expert Systems with Applications*, 241, 122666. <https://doi.org/10.1016/j.eswa.2023.122666>
- Jin, M.-J., & Kang, C.-G. (2024). Anomaly Detection With Transformer for a Railway Vehicle Air Compressor. *International Journal of Control, Automation and Systems*, 22(8), 2644-2657. <https://doi.org/10.1007/s12555-023-0721-z>
- Joung, B. G., Nath, C., Li, Z., & Sutherland, J. W. (2024). Bearing anomaly detection in an air compressor using an LSTM and RNN-based machine learning model. *The International Journal of Advanced Manufacturing Technology*. <https://doi.org/10.1007/s00170-024-14322-z>

- Kharazian, Z., Lindgren, T., Magnússon, S., Steinert, O., & Reyna, O. A. (2024). *SCANIA Component X Dataset: A Real-World Multivariate Time Series Dataset for Predictive Maintenance* (No. arXiv:2401.15199). arXiv. <https://doi.org/10.48550/arXiv.2401.15199>
- Kılıçarslan, S., & Adem, K. (2021). *An overview of the activation functions used in deep learning algorithms.*
- Kimm, H., Paik, I., & Kimm, H. (2021). Performance Comparison of TPU, GPU, CPU on Google Colaboratory Over Distributed Deep Learning. *2021 IEEE 14th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*, 312-319. <https://doi.org/10.1109/MCSoc51149.2021.00053>
- Kong, Y., Wang, Z., Nie, Y., Zhou, T., Zohren, S., Liang, Y., Sun, P., & Wen, Q. (2024). *Unlocking the Power of LSTM for Long Term Time Series Forecasting* (Versiyon 1). arXiv. <https://doi.org/10.48550/ARXIV.2408.10006>
- Li, Z., He, Q., & Li, J. (2024). A survey of deep learning-driven architecture for predictive maintenance. *Engineering Applications of Artificial Intelligence*, 133, 108285. <https://doi.org/10.1016/j.engappai.2024.108285>
- Liao, L., Li, H., Shang, W., & Ma, L. (2022). An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks. *ACM Transactions on Software Engineering and Methodology*, 31(3), 1-40. <https://doi.org/10.1145/3506695>
- Lindemann, B., Maschler, B., Sahlab, N., & Weyrich, M. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, 131, 103498. <https://doi.org/10.1016/j.compind.2021.103498>
- Medavarapu, S. S. (2024). *Advancements in Deep Learning: A Review of Keras and TensorFlow Frameworks.*

- Mohan Raparthy, E. Al. (2023). Predictive Maintenance in IoT Devices using Time Series Analysis and Deep Learning. *Dandaao Xuebao/Journal of Ballistics*, 35(3), 01-10.  
<https://doi.org/10.52783/dxjb.v35.113>
- Ng, S. K. E. (t.y.). *Fine-tuning Neural Networks using Keras Tuner and Fuzzy Algorithm*.
- Pan, Z., Yu, W., Yi, X., Khan, A., Yuan, F., & Zheng, Y. (2019). Recent Progress on Generative Adversarial Networks (GANs): A Survey. *IEEE Access*, 7, 36322-36333. <https://doi.org/10.1109/ACCESS.2019.2905015>
- Qureshi, M. S., Umar, S., & Nawaz, M. U. (2024). *Machine Learning for Predictive Maintenance in Solar Farms*. 01(03).
- Shaala, A., Baglee, D., & Dixon, D. (t.y.). *Machine learning model for predictive maintenance of modern manufacturing assets*.
- Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, 04(12), 310-316.  
<https://doi.org/10.33564/IJEAST.2020.v04i12.054>
- Srisuwan, S., & Innet, S. (2024). An Investigation of Machine Learning Algorithms for Predictive Maintenance in High Pressure Processing Systems. *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 94-98. <https://doi.org/10.1109/JCSSE61278.2024.10613634>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (t.y.). *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*.
- Yadav, A., Jha, C. K., & Sharan, A. (2020). Optimizing LSTM for time series prediction in Indian stock market. *Procedia Computer Science*, 167, 2091-2100.  
<https://doi.org/10.1016/j.procs.2020.03.257>

- Yan, P., Abdulkadir, A., Luley, P.-P., Rosenthal, M., Schatte, G. A., Grewe, B. F., & Stadelmann, T. (2024). A Comprehensive Survey of Deep Transfer Learning for Anomaly Detection in Industrial Time Series: Methods, Applications, and Directions. *IEEE Access*, *12*, 3768-3789. <https://doi.org/10.1109/ACCESS.2023.3349132>
- Zaheer, R., & Shaziya, H. (2019). A Study of the Optimization Algorithms in Deep Learning. *2019 Third International Conference on Inventive Systems and Control (ICISC)*, 536-539. <https://doi.org/10.1109/ICISC44355.2019.9036442>
- Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, *1*(1), Article 1. <https://doi.org/10.38094/jastt1224>
- Zhang, N., Lei, D., & Zhao, J. F. (2018). An Improved Adagrad Gradient Descent Optimization Algorithm. *2018 Chinese Automation Congress (CAC)*, 2359-2362. <https://doi.org/10.1109/CAC.2018.8623271>
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57-81. <https://doi.org/10.1016/j.aiopen.2021.01.001>

