

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**İoT AĞLARI İÇİN YENİ BİR SALDIRI
TESPİT SİSTEMİ TASARIMI**

YÜKSEK LİSANS TEZİ

Tuğba ULUSOY

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

TEMMUZ 2025

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**IoT AĞLARI İÇİN YENİ BİR SALDIRI
TESPİT SİSTEMİ TASARIMI**

YÜKSEK LİSANS TEZİ

Tuğba ULUSOY

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

Tez Danışmanı: Prof. Dr. Ünal ÇAVUŞOĞLU

TEMMUZ 2025

Tuğba ULUSOY tarafından hazırlanan “IoT Ağları İçin Yeni Bir Saldırı Tespit Sistemi Tasarımı” adlı tez çalışması 08.07.2025 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim **Dalı Bilgisayar Mühendisliği Bilim** Dalı’nda **Yüksek Lisans tezi**olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :

Jüri Üyesi :

Jüri Üyesi :



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “**IoT AĞLARI İÇİN YENİ BİR SALDIRI TESPİT SİSTEMİ TASARIMI**” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(...../...../20.....).

(imza)

Tuğba Ulusoy





Değerli anneme, babama ve kardeşlerime



TEŞEKKÜR

Bu tez çalışmamın yürütülmesi sırasında kıymetli bilgilerini ve desteğini esirgemeyen tez danışmanım Prof. Dr. Ünal Çavuşoğlu'na eğitim katkılarından dolayı çok teşekkür ederim.

Bütün eğitim hayatım boyunca her zaman yanımda olan ve bana yol gösteren en büyük destekçim değerli aileme çok teşekkür ederim.

Tuğba Ulusoy



İÇİNDEKİLER

Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	v
TEŞEKKÜR	ix
İÇİNDEKİLER	xi
KISALTMALAR	xv
SİMGELER	xvii
TABLO LİSTESİ	xix
ŞEKİL LİSTESİ	xxi
ÖZET	xxiii
SUMMARY	xxv
1. GİRİŞ	1
1.1. Tezin Kapsamı	2
1.2. Tezin Amacı	3
1.3. Arka Plan ve Problemin Tanımı	4
1.4. Araştırmanın Önemi	5
1.5. Literatür Araştırması	6
1.5.1. IoT güvenlik sorunları ve saldırı türleri	6
1.5.3. CIC-IoT 2023 veri seti incelemesi	8
2. YÖNTEMLER	11
2.1. LSTM (Long Short-Term Memory)	11
2.2. CNN (Convolutional Neural Networks)	13
2.2.1. CNN'in temel yapısı ve öğrenme yeteneği	13
2.2.2. CNN ile uzamsal özelliklerin çıkarımı	15
2.2.3. CNN tabanlı anomali tespitinde uzamsal öğrenmenin rolü	16
2.2.4. CNN mimarilerinde derinlik, aktivasyon ve optimizasyon teknikleri	17
2.2.5. CNN-LSTM hibrit mimarilerinin saldırı tespitindeki etkinliği	19
2.3. BiLSTM (Bidirectional LSTM)	20
2.3.1. BiLSTM mimarisinin temel yapısı	20
2.3.2. Attention katmanları ile güçlendirilmiş BiLSTM yapısı	21
2.3.3. Modelin aşamaları ve katman yapısı	22
2.3.4. Literatürde BiLSTM performansı ve başarı oranları	22
2.3.5. BiLSTM kullanımının avantajları ve sınırlılıkları	22
2.4. Transformer	23
2.4.1. Transformer mimarisi ve temelleri	23
2.4.2. Zaman serisi verileri üzerindeki uygulaması	25
2.4.3. IoT ağlarında attention tabanlı IDS kullanımı	26
2.4.4. Transformer mimarilerinin avantaj ve sınırlılıkları	27
2.5. Mevcut Çalışmaların Başarı Durumları	29
3. VERİ SETİ VE ÖN İŞLEME	33
3.1. CIC-IoT 2023 Veri Seti Nitelikleri	33
3.1.1. Sayısal özelliklerin istatistiksel özeti	33
3.1.2. Özellikler arası korelasyon analizi	34

3.2. Etiket Yapısı ve Sınıf Dengesi	35
3.2.1. Etiketlenen sınıflar ve örnek sayıları.....	35
3.2.2. Etiket dengesizliği ve etkileri.....	36
3.2.3. Veri türü yapısı.....	36
3.2.4. Başlangıçtaki örnekleme stratejisi.....	36
3.2.5. Güncellenen örnekleme stratejisi	38
3.3. Eksik / Sonsuz Değer Temizliği	38
3.3.1. Eksik (NaN) değer temizliği	39
3.3.2. Sonsuz (inf) değerlerin temizlenmesi.....	39
3.3.3. Zamanla ilgili sütunların çıkarılması.....	39
3.4. Özellik Seçimi Yöntemleri	39
3.4.1. Chi-Square (χ^2) testi ile özellik seçimi	40
3.4.2. ExtraTrees classifier ile özellik önemi	41
3.5. PCA Yöntemi Kullanılarak Boyut İndirgeme	41
4. KULLANILAN YÖNTEM.....	43
4.1. Veri Setini Hazırlama	43
4.1.1. Sınıf dağılımı ve yapay dengeleme	43
4.1.2. Eksik ve anlamsız verilerin temizlenmesi.....	43
4.1.3. Özellik ölçekleme.....	44
4.1.4. Etiket dönüşümü ve sınıf kodlama	44
4.1.5. Özellik seçimi.....	44
4.1.6. Zaman serisi formatı.....	45
4.2. Model Mimarisinin Tanımı (LSTM).....	45
4.3. Model Mimarisinin Tanımı (BiLSTM)	46
4.3.1. DDoS, mirai ve benign multilabel classification için model mimarisi	46
4.3.2. Benign ve mirai binary classification için model mimarisi.....	47
4.3.3. DDoS ve benign binary classification için model mimarisi.....	47
4.4. Model Mimarisinin Tanımı (CNN-LSTM + Attention).....	49
4.5. Eğitim Parametreleri.....	49
4.6. Eğitimde Kullanılan Teknikler	50
4.6.1. Dropout katmanları	50
4.6.2. Batch normalization	51
4.6.3. Class weight	51
4.6.4. SMOTE (synthetic minority oversampling technique)	51
4.6.5. Erken durdurma ve öğrenme oranı azaltma	51
5. DEĞERLENDİRME VE SONUÇLAR.....	53
5.1. Accuracy, Precision, Recall, F1	53
5.2. PR Ve ROC Eğrileri	54
5.2.1. LSTM modeli ROC eğrileri	54
5.2.2. BiLSTM modeli ROC eğrileri	55
5.2.3. CNN-LSTM + attention modeli PR eğrileri.....	55
5.3. Confusion Matrix Analizi.....	56
5.3.1. LSTM modeli confusion matrix değerlendirmesi	56
5.3.2. BiLSTM modeli confusion matrix değerlendirmesi	57
5.3.3. CNN-LSTM + attention modeli confusion matrix değerlendirmesi	58
5.4. Deney Sonuçları	59
5.4.1. LSTM modeli genel değerlendirme	59
5.4.2. BiLSTM modeli genel değerlendirme.....	60
5.4.3. CNN-LSTM + attention modeli genel değerlendirme	61
5.5. Model Gecikme Süreleri (Latency)	62

5.5.1. LSTM modeli gecikme süresi	62
5.5.2. BiLSTM modeli gecikme süresi	63
5.5.3. CNN-LSTM + attention modeli gecikme süresi	63
5.6. Benign + Mirai + DDoS Eğitimi	65
5.6.1. LSTM modeli sonuçları	66
5.6.2. BiLSTM modeli sonuçları	69
5.6.3. CNN-BiLSTM + attention modeli sonuçları	72
5.7. Benign + Mirai Binary Model Eğitimi	77
5.7.1. LSTM modeli sonuçları	77
5.7.2. BiLSTM modeli sonuçları	79
5.8. Benign + DDoS Binary Model Eğitimi	82
5.8.1. LSTM modeli sonuçları	82
6. TARTIŞMA	85
6.1. Literatür ile Karşılaştırmalar	85
6.1.1. LSTM	85
6.1.2. BiLSTM	87
6.1.3. CNN-LSTM + attention	89
6.2. Modellerin Güçlü ve Zayıf Yönleri	91
6.2.1. LSTM	91
6.2.2. BiLSTM	94
6.2.3. CNN-LSTM + attention	96
6.3. Gerçek Hayatta Uygulanabilirlik	98
7. SONUÇ VE ÖNERİLER.....	101
7.1. Sonuç	101
7.2. Gelecek Çalışmalar İçin Öneriler	102
7.3. Genel Değerlendirme ve Sonuçların Önemi	103
KAYNAKLAR	105
ÖZGEÇMİŞ.....	109



KISALTMALAR

AUC	: Area Under the Curve
BiLSTM	: Bidirectional Long Short-Term Memory
CNN	: Convolutional Neural Network
FN	: False Negative
FP	: False Positive
LSTM	: Long Short-Term Memory
PR	: Precision-Recall
ROC	: Receiver Operating Characteristic
TN	: True Negative
TP	: True Positive



SİMGELER

$A_{i,j}$	: Aktivasyon çıktısı (örneğin ReLU son)
α	: Öğrenme oranı (learning rate)
b	: Bias terimi
c_t	: LSTM hücre durumu (cell state)
ϵ	: Kayıp fonksiyonu toleransı
f_t	: Unutma kapısı (forget gate) çıktısı
h_t	: LSTM hücre durumu (cell state)
$ht \Rightarrow ht \oplus \leftarrow ht$	: BiLSTM toplam çıktısı (concatenation)
$\rightarrow ht$	: BiLSTM ileri yön gizli durum
$\leftarrow ht$	: BiLSTM geri yön gizli durum
i_t	: Giriş kapısı (input gate) çıktısı
λ	: L2 regularizasyon katsayısı
o_t	: Çıkış kapısı (output gate) çıktısı
ρ	: Derinlik-genişlik oranı ($\rho = W/D \backslash rho = W/D \rho = W/D$)
σ	: Sigmoid aktivasyon fonksiyonu
t	: Zaman
$\tanh \tanh$	: Hiperbolik tanjant fonksiyonu
U	: Önceki gizli durumdan gelen ağırlık matrisi
W	: Girişten gelen ağırlık matrisi
x_t	: t anındaki giriş vektörü
X^2	: Ki-kare (Chi-square) testi
$Z_{i,j}$	: CNN evrişim sonucu (pre-activation değer)
$\odot$	: Eleman bazlı çarpma işlemi



TABLO LİSTESİ

Sayfa

Tablo 2.1. Derin öğrenme mimarilerinin karşılaştırması: güçlü yönler, zayıf yönler ve tipik kullanım alanları.	29
Tablo 2.2. Literatürdeki çeşitli derin öğrenme modellerinin saldırı tespiti performansları.	30
Tablo 2.3. Literatürdeki çeşitli derin öğrenme modellerinin saldırı tespiti performansları detaylı karşılaştırılması.	31
Tablo 3.1. Bazı sayısal sütunların istatistiksel özet değerleri.	33
Tablo 3.2. CIC-IoT 2023 veri seti içerisindeki sınıflar ve örnek sayıları.	35
Tablo 3.3. Başlangıçta kullanılan dengeli sınıf dağılımı (1milyon örnek/sınıf).	37
Tablo 3.4. Güncellenmiş örnekleme stratejisi (mirai sınıfları artırılmıştır).	38
Tablo 5.1. Farklı derin öğrenme modellerinin performans ölçütleri.	53
Tablo 5.2. IDS modellerinin eğitim süresi, veri seti ve mimari özelliklerine göre karşılaştırılması.	64
Tablo 5.3. Ddos + benign + mirai eğitiminde farklı derin öğrenme modelleri sonuçları performans karşılaştırması.	77
Tablo 6.1. LSTM tabanlı modellerin literatürle başarımların karşılaştırması.	86
Tablo 6.2. BiLSTM tabanlı modellerin literatürle başarımların karşılaştırması.	88
Tablo 6.3. CNN-LSTM + attention ve benzeri literatür modellerinin performans karşılaştırması.	90
Tablo 6.4. LSTM modelinin güçlü yönlerinin uygulama bazlı değerlendirmesi.	92
Tablo 6.5. LSTM modelinin zayıf yönlerinin uygulama bazlı değerlendirmesi.	93
Tablo 6.6. BiLSTM modelinin güçlü yönlerinin uygulama bazlı değerlendirmesi. ..	94
Tablo 6.7. BiLSTM modelinin zayıf yönlerinin uygulama bazlı değerlendirmesi. ..	95
Tablo 6.8. CNN-LSTM + attention modelinin güçlü yönlerinin uygulama bazlı değerlendirilmesi.	96
Tablo 6.9. CNN-LSTM + attention modelinin zayıf yönlerinin uygulama bazlı değerlendirilmesi.	97



ŞEKİL LİSTESİ

Sayfa

Şekil 2.1. LSTM mimarisinin temel hücre yapısı ve kapı mekanizmaları (Sak et al., 2014).	11
Şekil 2.2. Klasik LSTM ve derin, projeksiyonlu varyantlarının yapısal gösterimi (Sak et al., 2014).	12
Şekil 2.3. Sigmoid, ReLU ve leaky ReLU aktivasyon fonksiyonları (LeCun et al., 1998).	14
Şekil 2.4. CNN'in tipik mimarisi ve veri akışı (Elsayed et al., 2021).	15
Şekil 2.5. BiLSTM + attention tabanlı model mimarisi (Zrira et al., 2024).	21
Şekil 2.6. Transformer (attention mekanizması) kullanarak saldırı tespiti süreci.	24
Şekil 3.1. Seçilen özellikler arasında pearson korelasyon katsayısı ile oluşturulan ısı haritası.	34
Şekil 3.2. Modelin saldırı tespit performansı.	36
Şekil 3.3. Eğitim süreci boyunca model performansı.	37
Şekil 3.4. SHAP analizi ile model çıktılarına en çok etki eden değişkenlerin görselleştirilmesi.	40
Şekil 3.5. PCA ile 2 boyutlu sınıf dağılımı görselleştirmesi.	42
Şekil 5.1. LSTM modeli için ROC AUC eğrileri (tüm sınıflar için $AUC \geq 0.99$).	54
Şekil 5.2. BiLSTM modeli için ROC AUC eğrileri (özellikle mirai alt türlerinde sınırlı kayıplar gözlenmiştir).	55
Şekil 5.3. CNN-LSTM + attention modeline ait precision-recall eğrileri (mirai alt türlerinde daha düşük AP değerleri).	56
Şekil 5.4. LSTM modeline ait confusion matrix (mirai alt türlerinde belirgin sapmalar gözlenmiştir).	57
Şekil 5.5. BiLSTM modeline ait confusion matrix (mirai sınıflarında hata oranı LSTM'ye göre biraz daha düşüktür).	58
Şekil 5.6. CNN-LSTM + attention modeline ait confusion matrix (özellikle mirai sınıflarında ayırım gücünde düşüş gözlenmiştir).	59
Şekil 5.7. LSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek, ancak Mirai alt türlerinde küçük sapmalar mevcuttur).	60
Şekil 5.8. BiLSTM modeline ait sınıflandırma raporu (özellikle mirai sınıflarında LSTM'den daha düşük hata oranları gözlenmiştir).	60
Şekil 5.9. CNN-LSTM + attention modeline ait sınıflandırma raporu.	61
Şekil 5.10. LSTM modelinin epoch başına eğitim süresi (ortalama: ~311 sn/epoch).	62
Şekil 5.11. BiLSTM modelinin epoch başına eğitim süresi (ortalama: ~89 sn/epoch).	63
Şekil 5.12. CNN-LSTM + attention modeline ait kısmi eğitim süreci çıktısı (ortalama süre belirtilmemiştir ancak işlem yükü yüksektir).	64
Şekil 5.13. LSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).	66
Şekil 5.14. LSTM modeline ait confusion matrix.	66

Şekil 5.15. LSTM modeline ait test loss grafiği.	67
Şekil 5.16. LSTM modeline ait test accuracy grafiği.	68
Şekil 5.17. LSTM modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).	68
Şekil 5.18. BiLSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).	69
Şekil 5.19. BiLSTM modeline ait confusion matrix.	69
Şekil 5.20. BiLSTM modeline ait test loss grafiği.	70
Şekil 5.21. BiLSTM modeline ait test accuracy grafiği.	71
Şekil 5.22. BiLSTM modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).	72
Şekil 5.23. CNN-BiLSTM + attention modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).	73
Şekil 5.24. CNN-BiLSTM + attention modeline ait confusion matrix.	73
Şekil 5.25. CNN-BiLSTM + attention modeline ait test loss grafiği.	74
Şekil 5.26. CNN-BiLSTM + attention modeline ait test accuracy (doğruluk) grafiği.	75
Şekil 5.27. CNN-BiLSTM + attention modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).	76
Şekil 5.28. LSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).	78
Şekil 5.29. LSTM modeline ait confusion matrix.	78
Şekil 5.30. BiLSTM modeline ait tüm katlama (foldlar) için epoch lara göre kayıp değerleri.	79
Şekil 5.31. BiLSTM modeline ait en iyi katlama (fold) kaybı.	80
Şekil 5.32. BiLSTM modeline ait tüm katlamalar (foldlar) için epoch'lara göre accuracy (doğruluk) değerleri.	81
Şekil 5.33. BiLSTM modeline ait en iyi fold doğruluğu (accuracy).	81
Şekil 5.34. LSTM modeline ait confusion matrix.	82
Şekil 5.35. LSTM modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).	83
Şekil 5.36. LSTM modeline ait test loss grafiği.	83
Şekil 5.37. LSTM modeline ait test accuracy (doğruluk) grafiği.	84

İoT AĞLARINDA YENİ BİR SALDIRI TESPİT SİSTEMİ

ÖZET

Nesnelerin İnterneti (IoT) teknolojisinin yaygınlaşması, beraberinde önemli güvenlik açıklarını da gündeme getirmiştir. Düşük donanım gücüne sahip, çeşitli protokollerle çalışan ve genellikle güvenlik açısından yetersiz olan bu cihazlar, Mirai gibi kötü amaçlı yazılımların hedefi hâline gelmekte ve büyük ölçekli DDoS saldırılarına zemin hazırlamaktadır. Kullanılan veri seti, güncel ve gerçekçi yapısıyla bilinen CIC-IoT2023'tür. 105 farklı cihaz ve 33 saldırı türü içeren bu veri setinden özellikle Mirai varyantları ve DDoS kategorileri seçilmiş, bu sayede sistemin gerçek dünya koşullarına daha yakın senaryolarda test edilmesi sağlanmıştır. Model, TensorFlow tabanlı olarak Google Colab ortamında eğitilmiş, veri dengesizliği SMOTE ve class-weight gibi stratejilerle giderilmiştir. Özellik seçimi için chi-square ve LightGBM gibi yöntemler kullanılmıştır.

CIC-IoT2023 veri setinin bu çalışmada kullanılan versiyonu, çeşitli DDoS ve Mirai tabanlı saldırı türlerinin yanı sıra normal (Benign) trafiği de içerecek şekilde yapılandırılmıştır. LSTM, BiLSTM ve CNN-BiLSTM+Attention modelleri ile çok sınıflı (multi-class) çalışılmıştır. Veri setindeki eksik ve anlamsız verilerin temizlenerek model eğitimdeki sapmalar minimize edilmiştir. Özellikle modelde ezberlemeye neden olan sütunların veri setinden çıkarılması, overfitting riskini önemli ölçüde azaltmıştır.

Çalışmada, özelliklerin değer aralıklarını normalize etmek amacıyla RobustScaler ve MinMaxScaler birlikte kullanılmıştır. Yüksek boyutlu veriyle çalışıldığı için SelectKBest yöntemiyle χ^2 testi uygulanmış ve en iyi 15 özellik seçilmiştir. LSTM ve BiLSTM modelleri %98'lik başarımla değerleri ile birbirine oldukça yakın sonuçlar verirken, CNN-BiLSTM + Attention mimarisi, özellikle Mirai-Greeths ve Mirai-Greips sınıflarında daha düşük performans sergilemiştir. Model performansının sınıf bazlı detaylı analizini yapabilmek adına, ROC (Receiver Operating Characteristic) ve PR (Precision-Recall) eğrileri değerlendirilmiştir. LSTM modeline ait ROC eğrisi, modelin genel anlamda oldukça başarılı bir ayırt etme kapasitesine sahip olduğunu ortaya koymaktadır.

Benign, DDoS-SYN, DDoS-TCPs, DDoS-UDP ve Mirai-udpplains gibi sınıflar için AUC = 1.00 seviyesinde değerler elde edilmiş, Mirai-Greeths ve Mirai-Greips gibi örüntüleri benzeşen alt sınıflarda ise AUC = 0.99 değerleri gözlemlenmiştir. BiLSTM modeli, çift yönlü öğrenme kapasitesi sayesinde özellikle zaman bağımlı örüntülerde daha derin bağlamları kavrayabilmektedir. BiLSTM modeli için ROC eğrilerinde model, Benign ve DDoS türlerinde AUC = 1.00 düzeyinde bir başarıya ulaşmıştır. Ancak Mirai-Greeths sınıfı için AUC = 0.97, Greips için ise AUC = 0.98 değerleri gözlemlenmiştir. CNN-BiLSTM + Attention modeli Precision-Recall (PR) eğrileri, özellikle dengesiz veri yapılarında sınıf bazlı doğruluğu ölçmek açısından kritiktir. Bu modelde Benign ve DDoS sınıflarında AP (Average Precision) değeri 1.00 olarak ölçülmüş olsa da, Mirai-Greeths sınıfı için AP = 0.84 ve Mirai-Greips için AP = 0.87 değeri elde edilmiştir. Üç modelin confusion matrix sonuçları birlikte

değerlendirildiğinde Benign, DDoS-SYN, DDoS-TCPs ve DDoS-UDP gibi temel sınıflarda modellerin yüksek doğrulukta tahmin yaptığı anlaşılmaktadır. Ancak Mirai alt sınıflarında, benzer davranış desenleri sebebiyle modeller için daha zorlayıcı olmuştur. Bu durum, model mimarisi seçiminde sınıf yapısının ayrıştırılabilirliğinin göz önünde bulundurulması gerektiğini ortaya koymaktadır.

Elde edilen sonuçlar, literatür ile karşılaştırıldığında veri setinde tercih edilen yöntem mimarilerinin hem performans hem de işlem süresi açısından dengeli bir çözüm sunduğu gözlemlenmiştir. Bu tez, IoT ortamlarında saldırı tespitine yönelik yapay zekâ destekli çözümlerin etkinliğini ortaya koymakta ve sektörel uygulamalar için öncü nitelikte bir sistem sunmaktadır.



A NEW ATTACK DETECTION SYSTEM IN IoT NETWORKS

SUMMARY

The widespread adoption of Internet of Things (IoT) technology has brought significant security vulnerabilities to the forefront. These devices, which have low hardware power, operate with various protocols, and are generally inadequate in terms of security, have become targets for malicious software such as Mirai, paving the way for large-scale DDoS attacks. The data set used is CIC-IoT2023, which is known for its up-to-date and realistic structure. From this data set, which includes 105 different devices and 33 attack types, Mirai variants and DDoS categories were specifically selected to ensure that the system could be tested in scenarios closer to real-world conditions. The model was trained in a Google Colab environment based on TensorFlow, and data imbalance was addressed using strategies such as SMOTE and class-weight. Methods such as chi-square and LightGBM were used for feature selection. The version of the CIC-IoT2023 dataset used in this study is configured to include various types of DDoS and Mirai-based attacks as well as normal (benign) traffic. Multi-class studies were conducted using LSTM, BiLSTM, and CNN-BiLSTM+Attention models. Missing and meaningless data in the dataset were cleaned to minimise deviations in model training. In particular, removing columns that caused memorisation in the model significantly reduced the risk of overfitting.

In this study, RobustScaler and MinMaxScaler were used together to normalise the value ranges of the features. Since high-dimensional data was used, the χ^2 test was applied with the SelectKBest method, and the best 15 features were selected. While the LSTM and BiLSTM models yielded very similar results with 98% accuracy, the CNN-BiLSTM + Attention architecture performed worse, especially in the Mirai-Greeths and Mirai-Greips classes. ROC (Receiver Operating Characteristic) and PR (Precision-Recall) curves were evaluated to perform a detailed class-based analysis of model performance. The ROC curve for the LSTM model shows that the model has a highly successful discrimination capacity in general. AUC values of 1.00 were obtained for classes such as Benign, DDoS-SYN, DDoS-TCPs, DDoS-UDP, and Mirai-udpplains, while AUC values of 0.99 were observed for subclasses with similar patterns such as Mirai-Greeths and Mirai-Greips. The BiLSTM model can understand deeper contexts, especially in time-dependent patterns, thanks to its bidirectional learning capability. In the ROC curves for the BiLSTM model, the model achieved an AUC of 1.00 for Benign and DDoS types. However, AUC values of 0.97 were observed for the Mirai-Greeths class and 0.98 for Greips. The Precision-Recall (PR) curves of the CNN-BiLSTM + Attention model are critical for measuring class-based accuracy, especially in imbalanced data structures. In this model, while the AP (Average Precision) value was measured as 1.00 for the Benign and DDoS classes, AP = 0.84 was obtained for the Mirai-Greeths class and AP = 0.87 for the Mirai-Greips class. When the confusion matrix results of the three models are evaluated together, it is understood that the models make high-accuracy predictions in basic classes such as Benign, DDoS-SYN, DDoS-TCPs, and DDoS-UDP. However, in the Mirai

subclasses, it has been more challenging for the models due to similar behaviour patterns. This situation reveals that the separability of the class structure should be considered in the model architecture selection. When compared with the literature, the results obtained show that the preferred method architectures in the data set offer a balanced solution in terms of both performance and processing time. This thesis demonstrates the effectiveness of artificial intelligence-supported solutions for attack detection in IoT environments and presents a pioneering system for sectoral applications.

The results of the LSTM model in the Benign + Mirai + DDoS training, particularly the precision, recall, and f1-score metrics hovering around 0.999 levels across all classes, clearly demonstrate the model's success on time series data. This indicates that the model can detect true positives at a high rate and, in general, offers balanced classification performance. The fact that the validation loss is close to the training loss indicates that the model has not overfitted and has high generalisability. The loss value of 0.0015 calculated on the test data also supports this. Both training and validation accuracies remained at high levels, exceeding 99.9% especially after the 10th epoch. The absence of fluctuations in accuracy values as the epoch count increases demonstrates that the model has a stable learning process. Consequently, the 0.9995 accuracy rate obtained on the test data confirms the model's high performance. ROC curves are an important metric used to evaluate the classification performance of LSTM models. The fact that the AUC (Area Under Curve) value is calculated as 1.00 for all classes indicates that the model's ability to distinguish between positive and negative classes is close to perfect. The classification performance of the BiLSTM model has been summarised.

The precision, recall, and F1-score values being at 0.999 levels for all classes indicate that the model produces highly balanced and accurate results. Compared to LSTM, the BiLSTM model has the advantage of being able to evaluate past and future contexts simultaneously in time series data. Although the overall success is quite high, a certain degree of confusion is observed, particularly between the Mirai and DDoS classes. The fact that 101 examples belonging to Mirai were predicted as DDoS indicates that these two attack types may exhibit similar behaviour patterns in some time windows. This situation is noteworthy in that it shows that structural similarities in models with bidirectional data flow may be perceived as similar by the model. However, the low overall error distribution proves that the model has strong learning capacity. The achieved test accuracy of 0.9995 confirms that the model can perform inter-class discrimination with high precision. Despite this success, the slight fluctuations observed in the validation curve indicate that the model can maintain flexible decision boundaries when encountering complex structures.

It presents the comparative results of the LSTM, Bi-LSTM, and CNN-BiLSTM + Attention models. While all three models achieved high success rates, the CNN-BiLSTM + Attention model stood out in all metrics. This model not only made more accurate predictions with an accuracy rate of 0.9996 and a test loss of 0.0011, but also achieved a more stable and low-loss learning process. The 0.9997–0.9996 levels achieved in critical performance metrics such as precision, recall, and f1-score demonstrate that this model is more reliable in sensitive areas such as attack detection. The fact that the ROC-AUC value is 1.00, as with other models, indicates that top-level performance is maintained in terms of discriminative power. Therefore, choosing the CNN-BiLSTM + Attention model is a reasonable and recommendable approach, both in terms of metric superiority and operational reliability. When applying the

LSTM model for benign and Mirai binary classification, the K-fold cross-validation method was preferred. This method ensures that the model is evaluated more reliably by testing it on different data subsets rather than just a single training-test split. The dataset is divided into a specific number of equal parts and the model is tested on each part while being trained on the remaining parts. This ensures that the model's overall performance is reflected more fairly across the entire dataset. During the K-Fold application, the training and validation loss and accuracy values of the model in each fold were monitored separately. It was observed that the loss values decreased steadily across all folds and the accuracy rates rose rapidly. In the best fold the test accuracy was close to 100% and the test loss was measured at only 0.0006. All precision, recall, and f1-score values reached levels of 0.9999, demonstrating that the model was successful not only in terms of accuracy but also in maintaining balance between classes. In this context, it can be said that the K-Fold method helped the model produce a more generalised representation without overfitting. In particular, evaluating the data distribution across all layers, independent of random splits, has increased the model's adaptability to different scenarios. Training loss initially dropped rapidly and then stabilised at low levels. Similarly, validation loss also decreased steadily, demonstrating the model's strong generalisation performance.

BiLSTM can operate with minimal loss within a specific data slice, and the model parameters contribute optimally to both the learning process and validation performance. It has been observed that training and validation accuracies start at 0.99998, rapidly exceed 0.99999, and maintain this value throughout the process. When applying the LSTM model for benign and DDoS binary classification, the K-fold cross-validation method was preferred. This method utilised the K-Fold cross-validation technique. This application ensures that the model is tested more consistently and evenly across the entire dataset, without being confined to a specific subset of data. The model consistently delivered high accuracy rates during both training and validation across different folds. The AUC value for both classes was observed to be 1.00 on the ROC curve. According to the confusion matrix, the LSTM model classified the Benign and DDoS classes with very high accuracy. Classification errors occurred in only nine examples, clearly demonstrating the model's high classification capability. When examining the ROC curves, the AUC (Area Under Curve) values of 1.00 for the Benign and DDoS classes indicate that the model distinguishes between the classes without error and possesses very high discriminative power. In the loss graph obtained from the test data, it is observed that the training and validation loss values decrease over time and stabilise. Model's test accuracy, it is seen that the accuracy rates rise rapidly as the epoch progresses and stabilise at a high level. Furthermore, thanks to the K-Fold cross-validation method, it is observed that the model adapts to wider variations by being tested on different data subsets and that its generalisability increases. This study demonstrates that different deep learning architectures can detect attacks occurring in IoT networks with high accuracy rates.

Specifically, the BiLSTM architecture provided the most balanced result in terms of both processing time and accuracy rate. The BiLSTM model was able to distinguish between DDoS and Mirai-type attacks with up to 98% accuracy, keeping inter-class confusion to a minimum. The inclusion of the CNN component contributed to spatial pattern extraction, but this contribution increased the training time in some cases. Although the attention architecture theoretically has the potential to learn more advanced patterns thanks to its attention mechanism, the results of this study show that in practical applications, a high number of parameters can lead to errors. Complex

models struggled to distinguish between classes, particularly when there were significant similarities between subclasses, such as Mirai-Greeths and Mirai-Greips.

The studies conducted revealed that model performance is not solely dependent on architectural choices; data balance, feature selection, and hyperparameter tuning are equally decisive factors. In experiments using features selected specifically by the χ^2 method, the model showed less tendency towards overfitting. ROC curves, accuracy-F1 balance, and confusion matrix analyses also corroborate these findings. It can be stated that the system proposed in this thesis is particularly applicable in edge-based IoT security solutions and offers satisfactory results in terms of speed and accuracy. When the results obtained are compared with the literature, it is observed that the proposed solution is competitive and demonstrates superior performance in some respects.



1. GİRİŞ

Günümüzde teknolojinin hayatın pek çok alanına girmesiyle birlikte, internete bağlı cihaz sayısı da oldukça artmış durumda. Artık sadece bilgisayarlar değil, evdeki akıllı prizden, şehirdeki trafik lambalarına kadar birçok cihaz internet üzerinden iletişim kurabiliyor. Bu durum, “Nesnelerin İnterneti (IoT)” olarak bilinen yapının oluşmasına yol açmıştır (Nazir et al., 2024). İlk başta bu teknoloji hayatı kolaylaştırmak için tasarlanmış olsa da, zamanla bazı güvenlik risklerini de beraberinde getirmiştir. Çünkü bu cihazlar genelde düşük donanımlı, ucuz üretimli ve çoğu zaman yazılımsal olarak da eksik güvenlik önlemleriyle piyasaya sürülüyor. Durum böyle olunca saldırganlar için kolay hedef haline geliyorlar. Özellikle de Mirai gibi kötü amaçlı yazılımların bu cihazları ele geçirip botnet ağına katması sonucunda büyük çaplı DDoS saldırıları meydana gelebiliyor (Ge et al., 2019). Sadece bireysel kullanıcılar değil, hastane sistemleri ya da enerji altyapısı gibi kritik alanlar da bu tür saldırıların hedefi olabiliyor (Sinha et al., 2025). Mevcut güvenlik sistemleri IoT için yeterli değil. Çünkü klasik güvenlik çözümleri, genellikle sabit yapıya sahip sistemlerde çalışmak üzere tasarlanmış. IoT ise çok daha dinamik, cihaz çeşitliliği fazla ve her biri farklı protokollerle haberleşebiliyor. Bu yüzden yeni yöntemlere ihtiyaç duyuluyor. Son zamanlarda özellikle yapay zeka ve derin öğrenme yöntemleri bu alanda öne çıkmaya başladı (Alsoufi et al., 2021). Bu yöntemler, özellikle bilinmeyen saldırı türlerini tespit etme konusunda daha başarılı sonuçlar veriyor. Derin öğrenme yöntemlerinden CNN ve LSTM, ağ trafiği üzerinde mekânsal ve zamansal özellikleri öğrenebilen güçlü yapılar. Bu iki model birleştirildiğinde hem zaman serisi analizi hem de paket düzeyinde örüntü yakalama açısından avantaj sağlanabiliyor (Halbouni et al., 2022). Ayrıca son katmanlara eklenen Attention mekanizmaları, modele daha fazla odaklanma yetisi kazandırıyor. Bu da modelin hangi bilgiyi ne zaman dikkate alacağı konusunda daha isabetli kararlar vermesini sağlıyor (Gueriani et al., 2024). Literatürde yapılan bazı çalışmalarda CNN-LSTM mimarilerinin klasik makine öğrenmesi tekniklerine kıyasla çok daha iyi sonuçlar elde ettiği görülmüştür. Örneğin, Sinha ve arkadaşları tarafından yapılan bir çalışmada, önerilen CNN-LSTM modelinin %99.87

gibi oldukça yüksek bir doğruluk oranına ulaştığı ve false-positive oranının da oldukça düşük olduğu belirtilmiştir (Sinha et al., 2025). Yine benzer şekilde Nazir ve ekibi de üç farklı veri seti üzerinde %99 seviyesine kadar çıkan başarı oranları rapor etmiştir (Nazir et al., 2024). Bu çalışmada ise CIC-IoT2023 veri seti kullanılarak, IoT ağlarında görülen DDoS ve Mirai türündeki saldırıların tespiti için LSTM, BiLSTM ve CNN-LSTM tabanlı Attention katmanı içeren bir hibrit model önerilmiştir. Hedef, hem doğruluk oranını yüksek tutmak hem de bu tespiti gerçek zamana yakın bir hızda gerçekleştirebilmek. Çünkü akademik olarak iyi sonuç almak bir yana, gerçek dünya koşullarında çalışabilecek bir sistem ortaya koymak asıl önceliktir. CIC-IoT2023 veri setinin 105 farklı cihazı kapsamı ve 33 farklı saldırı tipini içermesi, bu çalışmanın daha pratik ve gerçekçi olmasını sağlamaktadır (Jony & Arnob, 2024).

1.1. Tezin Kapsamı

Bu tez çalışması, özellikle IoT ağlarında yaygın şekilde görülen DDoS ve Mirai saldırılarını ele alır. Buradaki amaç, bu iki saldırı türünün tespitine yönelik yüksek doğruluklu ve pratik bir derin öğrenme modeli oluşturmaktır. Yani çalışmada bütün saldırı türleri ele alınmamıştır, daha çok bu iki kategoriye odaklanılmıştır. Bunun sebebi ise, literatürde bu saldırıların IoT altyapısında en çok karşılaşılan ve en fazla zarar veren türler arasında olmasıdır (Sinha et al., 2025; Jony & Arnob, 2024). Veri seti olarak CIC-IoT2023 kullanılmıştır. Bu veri seti, 105 farklı cihazdan toplanan ağ trafiğini içerdiği için, hem geniş kapsamlı hem de gerçek dünyaya yakın bir ortam sunmaktadır. 33 saldırı tipini barındıran veri seti içinden DDoS ve Mirai kategorileri seçilerek çalışmaya dâhil edilmiştir. Böylece modelin farklı Mirai alt varyantları (örneğin Mirai-UDP, Mirai-Greeths, Mirai-Greips gibi) üzerinde nasıl performans gösterdiği de gözlemlenebilmiştir (Nazir et al., 2024). Tezin kapsamı teknik olarak da belirli bir çerçevede tutulmuştur. Örneğin sadece derin öğrenme teknikleri kullanılmış; klasik makine öğrenmesi yöntemlerine (Random Forest, SVM vs.) yer verilmemiştir. Çünkü odak, zaman-serisi yapısına uygun olan LSTM, BiLSTM, CNN ve Attention gibi katmanları içeren mimarilerde yoğunlaşmaktadır (Halbouni et al., 2022). Modelin girişine yönelik öznetelik seçimi işlemleri ise daha çok χ^2 testi ve LightGBM skorlaması gibi istatistiksel ön adımlarla yapılmıştır, yani burada da kapsam sınırlandırılmıştır. Bu çalışma bir saldırının gerçekleşmesini önlemeyi değil, saldırının ağ trafiği üzerinden zamanında tespit edilmesini amaçlamaktadır. Yani bu tezde bir güvenlik duvarı ya da engelleme algoritması geliştirilmemiştir. Tespit modeli pasif

olarak ağdaki davranışları inceleyip, neyin şüpheli olduğunu sınıflandırmakla ilgilenir. Son olarak, modelin eğitim ve test süreçleri belirli donanım sınırlamaları altında gerçekleştirilmiştir. Eğitim işlemleri Google Colab platformu üzerinde GPU destekli olarak yapılmış, ancak sistemin gerçek zamanlı çalışabilirliği sadece simülasyon ortamlarında gözlemlenmiştir. Yani modelin saha ortamındaki davranışı bu çalışmanın doğrudan kapsamı içerisinde değildir. Bu da çalışmanın gelecekte genişletilebileceği alanlardan biri olabilir.

1.2. Tezin Amacı

Bu çalışmanın temel amacı, IoT tabanlı ağlarda meydana gelen Dağıtık Hizmet Reddi (DDoS) ve Mirai kaynaklı saldırıların yüksek doğrulukla ve düşük gecikmeyle tespit edilmesini sağlayacak bir saldırı tespit sistemi (IDS) geliştirmektir. Özellikle CIC-IoT2023 gibi geniş kapsamlı ve güncel bir veri seti üzerinde LSTM mimarisi, BiLSTM mimarisi ve CNN-LSTM+Attention hibrit derin öğrenme mimarisi ile çalışarak klasik yöntemlerin sınırlı kaldığı noktaların aşılması hedeflenmektedir. Mevcut literatür incelendiğinde, IoT ortamlarına özel önerilen IDS çalışmalarında genel olarak iki yönlü bir eğilim olduğu göze çarpar: ya doğruluk oranı yüksek ama gerçek zamanlılıktan uzak mimariler tercih edilmekte ya da hızlı çalışan ama başarı oranı düşük modeller kullanılmaktadır. Halbuki saldırıların doğası gereği, her iki yönün de birlikte ele alınması gerekir (Gueriani et al., 2024). Örneğin, Halbouni ve arkadaşlarının yaptığı çalışmada CNN ve LSTM'nin birleşimiyle oluşturulan hibrit modelin yüksek doğruluk oranlarına ulaştığı, ancak dikkat çekici noktanın modelin performansına rağmen gerçek zamanlı uygulamalarda henüz test edilmediği belirtilmiştir (Halbouni et al., 2022). Diğer yandan, Nazir et al. (2024), farklı veri setlerinde test edilen CNN-LSTM modelinin özellikle 14'ten fazla saldırı tipine karşı %99'a varan doğruluk sergilediğini ancak bu başarıların sınıfların dengeli olduğu test ortamlarında sağlandığını ifade etmiştir. Araştırmamızda hedeflenen şey, literatürdeki bu sınırlamaları aşarak hem doğruluk hem de uygulama düzeyinde güçlü bir sistem ortaya koymaktır. Bu kapsamda, CIC-IoT2023 veri setinin sağladığı gerçekçi cihaz çeşitliliği ve saldırı yoğunluğu (105 IoT cihazı, 33 saldırı tipi) kullanılarak modelin hem genişletilebilirliği hem de saldırı tipi fark etmeksizin başarıyı test edilmiştir (Jony & Arnob, 2024). Öte yandan, çalışmanın bir diğer önemli amacı da geleneksel makine öğrenmesi algoritmalarının (Random Forest, SVM, Decision Tree vb.) ötesine geçerek, derin öğrenmenin sunduğu avantajları IoT ortamında ölçülebilir hâle

getirmektir. Çünkü klasik algoritmalar çoğunlukla elle seçilmiş özniteliklere bağımlıdır ve bu durum modelin genelleme kapasitesini kısıtlamaktadır (Alsoufi et al., 2021). Derin öğrenme ise bu bağımlılığı ortadan kaldırarak, sistemin kendi içinde örüntü öğrenmesini ve aykırı davranışları otomatik şekilde tespit etmesini sağlar (Ge et al., 2019). Bu araştırmada sadece akademik bir model sunmakla kalmayıp, aynı zamanda IoT ortamlarında uygulanabilir bir IDS yaklaşımı geliştirmeyi, saldırı tespiti konusunda literatürdeki boşlukları doldurmayı ve gerçek dünya koşullarında test edilebilir bir yapı kurmayı amaçlamaktadır.

1.3. Arka Plan ve Problemin Tanımı

İnternetin gelişmesiyle birlikte, birbirine bağlı cihazların sayısı milyonları çoktan geçmiş durumda. Bu cihazların bir kısmı evlerdeki akıllı ampuller ya da termostatlar gibi basit görünen sistemler olsa da, bazıları üretim tesislerinde ya da sağlık sektöründe kritik işlevlere sahip. Bu yapı, Nesnelerin İnterneti (IoT) kavramını doğurmuştur. IoT sistemleri, sürekli veri üreten ve genellikle otomatik çalışan cihazlardan oluşur. Ancak bu faydalı yapı, saldırganlar için de cazip bir hedef oluşturmaktadır (Nazir et al., 2024). IoT cihazlarının genel yapısı itibarıyla güvenlik önlemleri zayıf kalabilmektedir. Üreticilerin çoğu, bu cihazları düşük maliyetli ve hızlı üretim amacıyla piyasaya sunduklarından, güvenlik ikinci planda kalmaktadır. Güncellenmeyen firmware'ler, varsayılan şifrelerin değiştirilmemesi, zayıf protokol kullanımları gibi nedenlerle bu cihazlar savunmasız hâle gelir (Ge et al., 2019). Özellikle Mirai gibi kötü amaçlı yazılımlar, bu zayıf noktaları kullanarak cihazları botnet ağına katarak DDoS saldırılarında kullanmaktadır. Problem burada başlıyor. Çünkü geleneksel saldırı tespit sistemleri (IDS) ya bu cihazları görmezden geliyor ya da onları koruyacak kapasiteye sahip değil. Ayrıca bu sistemler, çoğunlukla bilinen saldırı imzalarını yakalayabilen yapılara sahiptir. Ancak IoT ağlarında saldırı desenleri oldukça değişken ve karmaşık olabilir. Bu da anomali tespiti yapabilen, daha esnek ve öğrenebilen sistemlere olan ihtiyacı doğurur (Alsoufi et al., 2021). Son yıllarda bu problemle başa çıkmak için yapay zekâ ve özellikle derin öğrenme temelli modeller öne çıkmıştır. CNN ve LSTM gibi mimariler, ağ trafiğindeki örüntüleri analiz ederek, olağandışı durumları tanımlamada oldukça etkili sonuçlar vermiştir (Halbouni et al., 2022). Ancak burada da yeni bir zorluk ortaya çıkmaktadır: Modelin doğruluk oranı yüksek olsa da gerçek zamanlı çalışabilmesi, düşük gecikmeyle karar verebilmesi ve farklı saldırı türlerini ayırt edebilmesi gerekir. Yani sadece doğru tahmin yetmez, hızlı ve pratik de olmalı.

Bu çalışmanın ortaya çıkış noktası da tam olarak budur. Mevcut sistemlerin gerek saldırı tespit başarımı, gerekse operasyonel verimliliği konusunda ciddi eksiklikler bulunmaktadır. Özellikle Mirai ve DDoS gibi yaygın saldırı türlerinin, gerçek IoT trafiği üzerinde zamanında tespit edilememesi, hem bireysel hem de kurumsal ağlar açısından büyük risk oluşturmaktadır (Sinha et al., 2025).

1.4. Araştırmanın Önemi

IoT sistemlerinin bu kadar yaygınlaşması, beraberinde siber güvenliği de daha karmaşık hâle getirdi. Çünkü bu cihazlar sıradan sistemlerden farklı. Mesela bir dizüstü bilgisayar gibi düşünülüp koruma altına alınamıyorlar. Her biri farklı üreticiden, farklı protokolde ve farklı güvenlik seviyesiyle çalışıyor. Bu durum, klasik güvenlik araçlarının çoğu zaman yetersiz kalmasına yol açıyor (Ge et al., 2019). Yani artık daha esnek ve öğrenebilen sistemlere ihtiyaç duyuluyor. Araştırmamız burada devreye giriyor. Çünkü literatürde yapılan çoğu çalışmada modeller ya yalnızca belirli saldırı tipleri üzerinde eğitilmiş ya da veri setleri dengesiz olduğu için bazı saldırılar çok daha iyi tespit edilmiş, bazıları ise gözden kaçmış. Hatta bazen sadece “benign vs attack” olarak ikili sınıflandırma yapılmış ve gerçek dünyada karşılaşılabilecek çoklu saldırı türleri hesaba katılmamış (Alsoufi et al., 2021). Bizim çalışmamızda ise CIC-IoT 2023 gibi çok daha çeşitli bir veri seti kullanılarak, sistemin çoklu saldırı tiplerine karşı genel başarısı değerlendirilmiştir (Jony & Arnob, 2024). Bir diğer önemli taraf da şu: Literatürde bazı modeller yüksek doğruluk veriyor ama çalışma ortamları çok temiz. Gerçek dünya trafiğinde, örneğin gecikmeler, benzer oturumlar, sistem gürültüsü gibi durumlar varken performans düşüyor. Bu nedenle geliştirdiğimiz modelin amacı, sadece teorik olarak yüksek doğruluk sağlamak değil, aynı zamanda bu koşullar altında da tutarlı kalabilmesi. Özellikle Attention katmanı, modelin önemli sinyallere odaklanmasına yardımcı olduğu için, bu tür gerçekçi durumlarda avantaj sağlayabilir (Gueriani et al., 2024). Ayrıca bu çalışmanın önemli olmasının bir başka nedeni de, IoT sistemlerine saldırı gerçekleştiğinde sonuçların çok daha yaygın etkiler oluşturması. Yani bu cihazlara yapılan bir saldırı, sadece bir ağın değil, bağlı olduğu tüm sistemlerin çalışmasını etkileyebiliyor. Örneğin bir akıllı enerji sistemine yapılan DDoS saldırısı, sadece cihazı değil, şehri etkileyebilir. Bu yüzden saldırı tespiti erken olmalı, hızlı olmalı, doğru olmalı. Bizim modelimiz de bu üçüne birden odaklanıyor (Sinha et al., 2025). Sonuç olarak bu tez, pratikte uygulanabilirliği yüksek, güncel saldırı tiplerini tanıyabilen ve bunu kısa sürede yapabilen bir sistem önerdiği için

sadece akademik deęil, aynı zamanda sektörel anlamda da önem taşımaktadır. Özellikle IoT ekosisteminin güvenliğine katkı sağlayacak çalışmalara olan gereksinim, günden güne artmaktadır (Nazir et al., 2024).

1.5. Literatür Araştırması

1.5.1. IoT güvenlik sorunları ve saldırı türleri

Son yıllarda IoT cihazlarının her alanda yaygınlaşmasıyla birlikte güvenlik sorunları da artmaya başladı. Bu cihazlar çoęu zaman düşük donanım kapasitesine sahip olduęu için gelişmiş güvenlik yazılımlarını çalıştıramıyor. Birçok üretici, ürünleri hızlıca pazara sunmak için yazılım güvenliğini ikinci plana atabiliyor. Saldırganların özellikle Mirai gibi botnetler aracılığıyla kolayca bu cihazlara sızmasına neden oluyor (Sharma et al., 2023). IoT cihazları çok farklı alanlarda kullanılıyor: akıllı şehirlerde trafik yönetimi, ev otomasyonu, sanayi üretimi gibi. Ama bu çeşitlilik aynı zamanda güvenlik açıklarının da çeşitlenmesine yol açıyor. Literatürde belirtildięi gibi, IoT cihazlarında karşılaşılan başlıca tehditler arasında kimlik sahtecilięi (spoofing), veri sızdırma, hizmet aksatma (DoS ve DDoS), botnet saldırıları ve yetkisiz erişim girişimleri yer alıyor (Varalakshmi et al., 2021). Özellikle DDoS saldırıları, IoT ortamları için önemli bir tehdit durumuna gelmiştir. Çünkü saldırganlar, zayıf korunan IoT cihazlarını ele geçirip bir botnet ağına dahil ediyor. Bu ağlar sonrasında hedef sunuculara büyük hacimli trafik göndererek hizmet kesintisine neden olabiliyor. Mirai botnet örneğinde olduęu gibi, bu saldırılar saatler sürebiliyor ve milyarlarca paketle sistemleri çökertiyor (Varalakshmi et al., 2021). Bu tip saldırılar, genellikle UDP flooding, TCP SYN flooding, ACK attacks, Scan gibi farklı alt türlere ayrılmakta. Mirai zararlısı örneğın, farklı IoT cihazlara bulaşarak bu saldırı türlerinin çoęunu gerçekleştirebiliyor (Sharma et al., 2023). Bu saldırılar sadece aęa zarar vermekle kalmıyor, baęlı olan tüm sistemlerin işleyişini de sekteye uğratabiliyor. Özellikle akıllı üretim hatlarında ya da saęlık sistemlerinde bu durum hayati sonuçlara neden olabilir. IoT ağlarının karakteristik özellikleri – düşük bant genişlięi, enerji sınırlamaları, çeşitlilik – bu tehditlerle mücadeleyi daha da zorlaştırmaktadır. Bu nedenle IDS sistemleri, klasik sistemlerin aksine daha özel çözümlerle tasarlanmalı, cihaz davranışlarını daha hassas şekilde izleyebilmelidir (Chen et al., 2022). IoT ağlarında karşılaşılan güvenlik sorunları hem donanımsal sınırlamalardan hem de yazılımsal zayıflıklardan kaynaklanmaktadır. Saldırganlar bu zayıflıkları kötüye kullanarak, sistemleri hem hedefli hem de yaygın saldırılarla etkisiz hale getirebilmektedir. Bu

yüzden IoT güvenliği teknik bir sorun olmanın yanı sıra, stratejik bir öncelik olarak da değerlendirilmelidir (Xu et al., 2022).

1.5.2. Saldırı tespit sistemleri (IDS/IPS)

IoT ağlarının karmaşık ve dinamik yapısı, geleneksel güvenlik çözümlerini çoğu zaman yetersiz bırakıyor. Güvenlik duvarları ya da statik erişim denetimleri gibi klasik yöntemler, IoT cihazlarının heterojen ve kaynak kısıtlı doğası karşısında etkisiz kalabiliyor. Bu nedenle özellikle saldırı önleme sistemleri (IPS) ve saldırı tespit sistemleri (IDS) IoT ortamları için daha uygun bir çözüm olarak öne çıkmaktadır (Sharma et al., 2023). IDS sistemlerinin temel amacı, ağ trafiğini sürekli izleyerek anormal ya da saldırgan davranışları tespit etmektir. Bu sistemler genellikle iki temel yaklaşıma dayanır; anomali tabanlı (anomaly-based) ve imza tabanlı (signature-based) yöntemler. İmza tabanlı sistemler daha çok bilinen saldırıları tespit etmede başarılıyken, yeni ve bilinmeyen saldırılar karşısında etkisiz kalabilir. Anomali tabanlı sistemler ise, normal ağ davranışlarını öğrenip bu davranışların dışına çıkan durumları potansiyel tehdit olarak işaretleyebilir (Dai et al., 2024). IoT ortamlarında ise bu sistemlerin bazı sınırlamaları vardır. Kaynak kısıtları (CPU, RAM, enerji vs.) IDS sistemlerinin IoT cihazlarında doğrudan çalıştırılmasını zorlaştırabilir. Bu yüzden çoğu zaman merkezi bir sunucuya ya da edge cihazlara yönlendirme yapılır. Bununla birlikte cihazlar arası iletişim protokollerinin farklılığı (MQTT, CoAP gibi) da IDS mimarilerini daha karmaşık hale getirmektedir (Chen et al., 2022). Geleneksel IDS sistemleri, daha çok klasik ağ altyapıları (bilgisayarlar, sunucular) için tasarlandığından, IoT'ye özgü zayıflıkları tanımakta zorlanabilir. Bu nedenle literatürde, özellikle IoT odaklı özelleştirilmiş IDS sistemlerinin geliştirilmesi önerilmektedir. Örneğin CNN, LSTM ve Attention tabanlı yapay zekâ mimarileriyle zenginleştirilmiş sistemler, IoT ağına özel saldırı örüntülerini daha iyi öğrenebilmekte ve tespit oranlarını artırabilmektedir (Tuli et al., 2022; Xu et al., 2022). Bazı yeni nesil IDS sistemleri yalnızca saldırıyı tespit etmekle kalmayıp, ona karşı gerçek zamanlı tepki de verebilmektedir. Bu tür sistemler IDS ile IPS'i birleştiren hibrit çözümler sunar. Özellikle edge computing ya da federated learning ile entegre edilen yapılar sayesinde IDS modelleri daha hızlı ve cihazlara daha yakın konumda çalıştırılabilir hale gelmiştir (Chen et al., 2022). Bazı yapılan çalışmalarda CNN-BiLSTM-Attention gibi yapılar kullanılarak hem doğruluk hem de düşük false-positive oranı sağlanmış, ayrıca azınlık sınıfı saldırıların da daha iyi tespit edilebildiği gösterilmiştir (Dai et al.,

2024)Network_Intrusion_Detec.... Bu da IDS sistemlerinin artık daha fazla derin öğrenme ve hem içeride destekli şekilde gelişmekte olduğunu ortaya koymaktadır. IoT ortamlarında etkili bir saldırı tespiti sağlamak için geleneksel IDS yaklaşımlarının ötesine geçilmesi gerekmektedir. Gerek veri yoğunluğu, gerek cihaz kısıtlamaları gerekse yeni tehdit çeşitliliği nedeniyle, yapay zekâ temelli ve özelleştirilebilir IDS çözümleri IoT güvenliğinde temel bir gereksinim hâline gelmiştir.

1.5.3. CIC-IoT 2023 veri seti incelemesi

Saldırı tespit sistemlerinde doğru ve etkili modeller geliştirebilmek için, kullanılan veri setinin içerik ve yapısal olarak gerçekçi ve güvenilir olması oldukça önemlidir. Literatürde NSL-KDD, CICIDS2017, N-BaIoT, Bot-IoT, ToN-IoT gibi pek çok veri seti önerilmiş olsa da, özellikle IoT temelli ağlarda yeni tehditleri yansıtan, güncel ve çok cihazlı veri kümeleri oldukça sınırlıdır (Nazir et al., 2024; Sharma et al., 2023). CIC-IoT2023 veri seti, bu açıdan literatürde öne çıkan çalışmalardan biridir. Canadian Institute for Cybersecurity (CIC) ve ITU Copenhagen iş birliğiyle geliştirilen bu veri setinin, hem çeşitlilik açısından hem de hacim yönünden diğer veri setlerine kıyasla önemli avantajları vardır. Bu veri setinde 105 farklı IoT cihazı yer almakta ve 33 saldırı türü 7 ana kategoride (DDoS, DoS, Recon, Web tabanlı, Brute-force, Spoofing, Mirai) toplanmıştır (Jony & Arnob, 2024). Veri toplama süreci de dikkat çekici şekilde detaylıdır. Gerçek IoT cihazları bir ev ortamını taklit edecek şekilde yerleştirilmiş, trafiğin bir kısmı Wireshark ve TCPdump gibi araçlarla doğrudan toplanmış, saldırılar Snort ve Suricata gibi IDS sistemleri aracılığıyla etiketlenmiştir. Bu da veri setinin sentetik olmaktan uzak, gerçek dünyaya daha yakın bir yapıya sahip olmasını sağlamıştır (IJITCS, 2024). CIC-IoT2023'ü literatürdeki diğer veri setleriyle karşılaştırmak gerekirse:

- **N-BaIoT** veri seti, çoğunlukla Mirai ve Bashlite saldırılarını kapsar. 9 IoT cihazı üzerinden elde edilmiştir. Ancak saldırı tipi çeşitliliği kısıtlıdır ve sadece botnet odaklıdır (Nazir et al., 2024).
- **Bot-IoT** veri seti simülasyon tabanlıdır. Gerçek cihazlar yerine sanal senaryolarla oluşturulmuştur ve bu yüzden gerçek trafik örüntülerinden sapmalar görülebilir (IJITCS, 2024).
- **CICIDS2017** veri seti genel ağ saldırılarını kapsar fakat IoT cihazları içermez, bu yüzden doğrudan IoT ortamları için yeterli temsil gücü taşımaz (Nazir et al., 2024).

CIC-IoT2023'ün avantajlarından biri de her cihazın hem saldıran hem de kurban olabileceği senaryolara sahip olmasıdır. Bu durum, saldırı tespiti yapacak modellerin sadece saldırıyı değil aynı zamanda ağ içi rol değişimlerini de öğrenmesine olanak tanır. Ayrıca veri setinde yer alan saldırıların çoğu farklı ağ katmanlarına yöneliktir. Örneğin, TCP SYN Flood ağ seviyesinde, HTTP Slow POST uygulama katmanındadır; bu da çok katmanlı analiz yapılabilmesini sağlar (IJITCS, 2024). Veri seti 115 özelliğe sahiptir ve bazı çalışmalarda bu yüksek boyutluluk bir problem olarak ele alınmıştır. Bu yüzden literatürde sıkça bu veri setiyle çalışılırken mutual information, ANOVA F-test, χ^2 testi gibi tekniklerle özellik seçimi yapılmakta; ardından PCA veya autoencoder gibi yöntemlerle boyut indirgenmektedir (Ogobuchi et al., 2024). CIC-IoT2023 hem cihaz sayısı hem saldırı çeşitliliği hem de veri toplama yöntemi bakımından güncel IoT saldırılarını modellemek için en uygun veri kümelerinden biri hâline gelmiştir. Literatürde CNN, LSTM, Attention gibi modellerin büyük çoğunluğu bu veri setiyle test edilmekte ve doğruluk oranı %90'ların üzerine çıkmaktadır (Jony & Arnob, 2024).

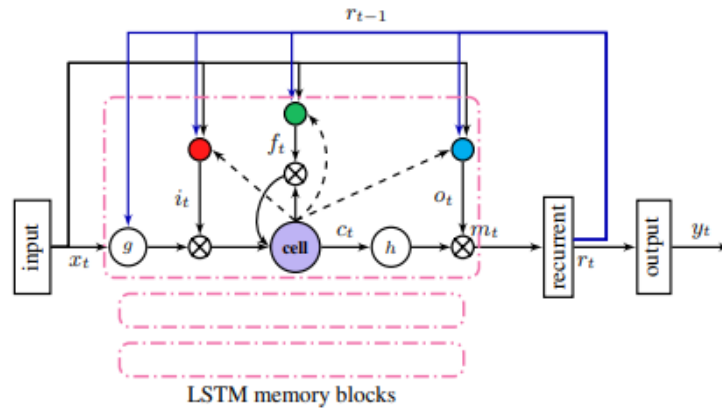


2. YÖNTEMLER

Saldırıları tespit edebilmek için IoT ağlarında kullanılan geleneksel makine öğrenmesi yöntemleri (Random Forest, Decision Tree, SVM gibi) son yıllarda yerini daha çok derin öğrenme modellerine bırakmaya başladı. Bunun en temel sebebi, derin öğrenmenin yüksek boyutlu, karmaşık ve zamana bağlı veriler üzerinde daha etkin sonuçlar vermesi. Özellikle zaman serisi verisiyle çalışan IoT ortamlarında LSTM, CNN, BiLSTM ve Attention mimarileri dikkat çekici başarılarla imza atmıştır (Sharma et al., 2023)

2.1. LSTM (Long Short-Term Memory)

Klasik RNN'lerin zaman içindeki bilgi taşıma kapasitesinin zayıflığına karşı LSTM (Long Short-Term Memory) mimarisi geliştirilmiştir. Bu yapı, özellikle uzun dönemli bağımlılıkların önemli olduğu zaman serisi verilerde etkili sonuçlar üretmektedir. LSTM'nin en temel özelliği, veri akışını kontrol eden giriş kapısı (input gate), unutma kapısı (forget gate) ve çıkış kapısı (output gate) olmak üzere üç ana kapı mekanizmasına sahip olmasıdır. Bu kapılar aracılığıyla hücre durumu (cell state) güncellenir ve geçmişten gelen bilginin ne kadarının korunup ne kadarının silineceği öğrenilir. Şekil 2.1'de klasik bir LSTM hücresinin iç yapısı ve kapılar arası veri akışı şematik olarak sunulmuştur (Sak et al., 2014).



Şekil 2.1. LSTM mimarisinin temel hücre yapısı ve kapı mekanizmaları (Sak et al., 2014).

$$\text{Giriş kapısı: } i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.1)$$

$$\text{Unutma kapısı: } f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (2.2)$$

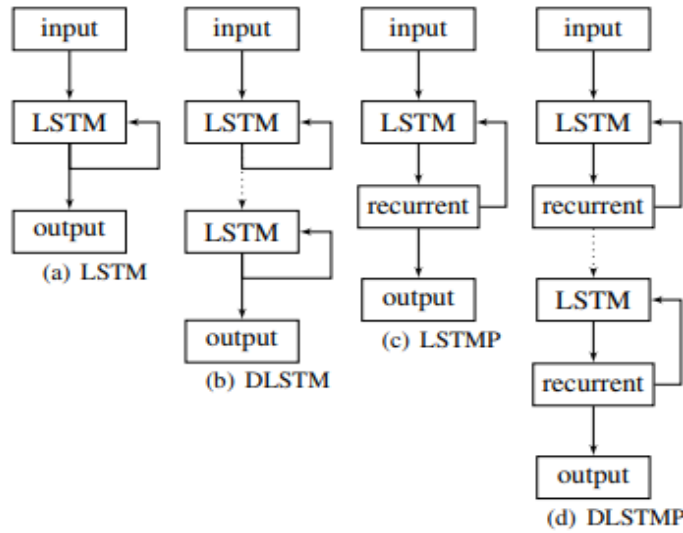
$$\text{Yeni hücre adayı: } \tilde{c}_t = \tanh \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (2.3)$$

$$\text{Hücre durumu: } c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (2.4)$$

$$\text{Çıkış kapısı: } o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.5)$$

$$\text{Gizli durum: } h_t = o_t \odot \tanh \tanh(c_t) \quad (2.6)$$

Matematiksel olarak bu yapının denklemleri 2.1, 2.2, 2.3, 2.4, 2.5 ve 2.6'daki gibi olup x_t giriş vektörü, h_{t-1} bir önceki gizli durum, c_t hücre durumu, σ sigmoid fonksiyonu, $\odot$ ise eleman bazlı çarpma işlemini temsil etmektedir. Bu yapı, zaman bağımlı verilerde uzun süreli bağımlılıkların öğrenilmesini mümkün kılmak için özel olarak tasarlanmıştır (Sak et al., 2014, s. 3).



Şekil 2.2. Klasik LSTM ve derin, projeksiyonlu varyantlarının yapısal gösterimi (Sak et al., 2014).

Bu yapı sayesinde ağ, hangi bilginin tutulması gerektiğini öğrenerek uzun vadeli bağımlılıkları kaybetmeden işleyebilir. LSTM yapıları farklı şekillerde genişletilebilir.

Örneğin derin LSTM (DLSTM), birden fazla LSTM katmanının üst üste yerleştirilmesiyle oluşturulur. LSTM mimarisinde ise projeksiyon katmanı aracılığıyla parametre sayısı azaltılırken modelin öğrenme gücü korunur. DLSTM ise hem derinlik hem de projeksiyon mekaniğini birleştirerek daha kompakt ama güçlü mimariler sunar. Şekil 2.2’de bu mimarilerin temel farkları yapısal olarak gösterilmiştir (Sak et al., 2014).

2.2. CNN (Convolutional Neural Networks)

Bu bölümde (Convolutional Neural Networks) CNN’den bahsedilecektir.

2.2.1. CNN’in temel yapısı ve öğrenme yeteneği

Convolutional Neural Network (CNN) modelleri, ilk olarak görsel örüntü tanıma gibi alanlarda geliştirilmiş olsa da, son yıllarda IoT tabanlı ağ trafiği analizinde de etkili şekilde kullanılmaya başlanılmıştır. Bu yapı, ağ verilerinde özellikle bayrak dizilimleri, port geçişleri ve veri yoğunluk değişimleri gibi mekânsal örüntüleri öğrenerek saldırı tespitinde önemli katkılar sağlamaktadır (Elsayed et al., 2021). CNN mimarisi üç temel bileşenden oluşur. Bunlar ; evrişim katmanı (convolution layer), havuzlama katmanı (pooling layer) ve tam bağlantılı katman (fully connected layer) dır. Bu yapı, sırasıyla özellik çıkarımı ve sınıflandırma işlemlerini üstlenir. Şekil 2.3’de bu mimarinin temel akışı gösterilmektedir.

İlk olarak giriş verisi, üzerinde kaydırılan filtreler aracılığıyla parçalar hâlinde işlenir ve buna “özellik haritası” (feature map) adı verilir. Bu işlem şu şekilde ifade edilir:

$$Z_{i,j} = (X \cdot W)_{i,j} + b \quad (2.7)$$

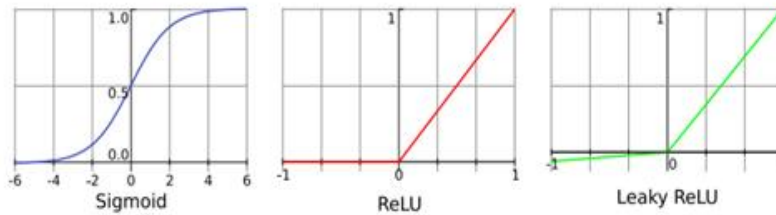
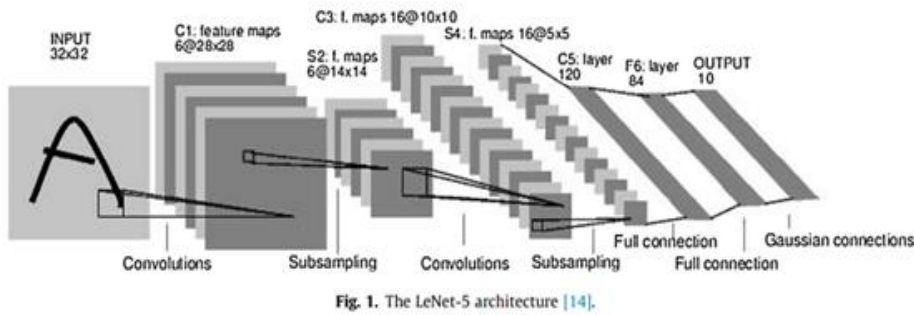
$$A_{i,j} = (0, Z_{i,j}) \quad (2.8)$$

Denklem 2.7’de X, giriş matrisi; W, öğrenilen filtre (kernel); b ise bias terimidir. Bu işlem sonucunda elde edilen değerler, doğrusal olmayan bir aktivasyon fonksiyonundan geçirilir. Genellikle kullanılan aktivasyon fonksiyonu ReLU olup, bu işlem denklem 2.8’deki gibi ifade edilir:

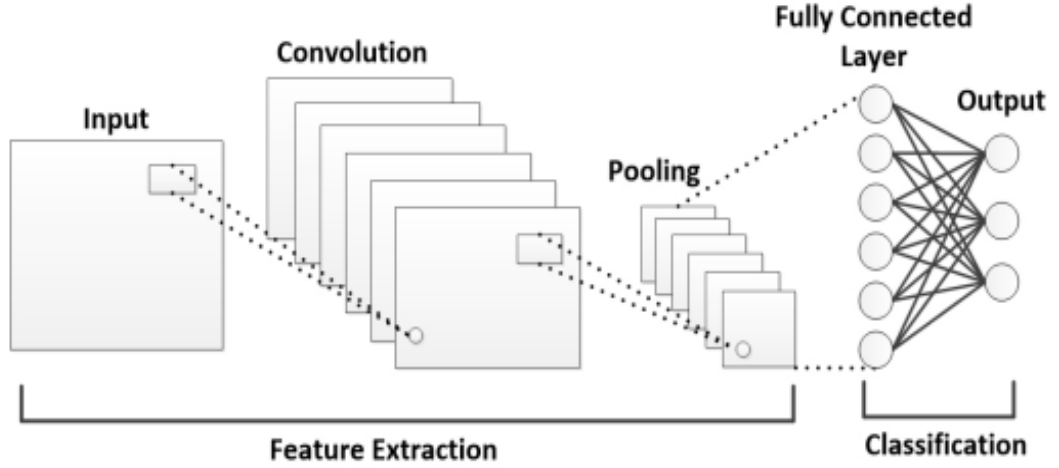
Yukarıda verilen formül 2.7 ve formül 2.8, Indolia ve arkadaşlarının çalışmasında tanımlanan CNN yapılarına dayalı olarak bu tez için uyarlanmıştır (Indolia et al., 2018, s. 681).

Aktivasyon işlemi olarak ReLU'ya ek olarak sigmoid ve leaky ReLU gibi fonksiyonlar da kullanılabilir. ReLU'nun en önemli avantajı, negatif değerleri sıfıra eşitleyerek modelin yalnızca pozitif örüntülere odaklanmasını sağlamasıdır (LeCun et al., 1998). Konvolüsyon katmanlarının ardından gelen havuzlama (genellikle max-pooling) işlemiyle, boyut azaltımı sağlanır. Bu işlemde girişin küçük bölgelerindeki en büyük değer seçilerek, modelin konumsal değişkenliklere karşı daha dayanıklı hâle gelmesi amaçlanır. Son katmanda yer alan fully connected yapı ise, çıkarılan tüm özellikler bir vektöre indirger ve sınıflandırma adımını gerçekleştirir.

IoT ağlarından gelen veriler çoğu zaman doğrudan görsel veri (görüntü) olmadığından, klasik CNN yapıları bu verilere doğrudan uygulanamaz. Bu sebeple, bazı çalışmalarda (örneğin, Elsayed et al., 2021) 1D verilerin 2D forma çevrilerek 8x6 boyutunda matris haline getirildiği ve CNN'e uygun giriş formatına dönüştürüldüğü görülmektedir. Bu dönüşüm, hem örüntü öğrenme kabiliyetini hem de parametre paylaşımını optimize eder. ReLU'nun yanında sigmoid ve leaky ReLU gibi fonksiyonlar da CNN'lerde tercih edilebilmektedir. Şekil 2.4'de bu üç aktivasyon tipi grafiksel olarak sunulmuştur. ReLU, negatif değerleri sıfıra eşitleyerek modelin sadece pozitif örüntülere odaklanmasını sağlar (LeCun et al., 1998). Evrişim katmanlarını takip eden havuzlama (genelde max-pooling) işlemi, boyut küçültme sağlar. Bu katman, girişin her küçük bölgesinden en büyük değeri alarak örüntülerin konumsal bağımsızlıkla öğrenilmesini sağlar.



Şekil 2.3. Sigmoid, ReLU ve leaky ReLU aktivasyon fonksiyonları (LeCun et al., 1998).



Şekil 2.4. CNN'in tipik mimarisi ve veri akışı (Elsayed et al., 2021).

Son olarak, fully connected katman, tüm çıkarılan özellikleri tek vektöre indirger ve sınıflandırır. IoT veri kümelerinde bu yapı doğrudan uygulanamayabilir çünkü ağ trafiği görüntü formatında değildir. Bu yüzden bazı çalışmalarda, örneğin Elsayed ve arkadaşlarının çalışmasında, 1D veri 2D'ye çevrilerek CNN'e uygun hâle getirilmiştir. 48 özellikten oluşan giriş verisi 8x6 matris olarak CNN'e aktarılmış ve böylece hem parametre paylaşımı hem de örüntü algısı optimize edilmiştir (Elsayed et al., 2021).

CNN mimarisinin en önemli avantajı, parametre paylaşımı sayesinde büyük veri setlerinde bile aşırı öğrenme (overfitting) ihtimalini azaltması ve aynı zamanda yüksek boyutlu verilerde hesaplama yükünü ciddi oranda düşürmesidir. Literatürde CNN'in özellikle saldırı öncesi oluşan düşük seviye anomalileri tanımada başarılı olduğu rapor edilmiştir (Elsayed et al., 2021).

2.2.2. CNN ile uzamsal özelliklerin çıkarımı

IoT ağlarında oluşan trafik, her ne kadar zaman serisine dayalı gibi görünse de, aslında içerisinde birçok uzamsal örüntü barındırır. Bu örüntüler, paket başlıklarındaki bayrak (flag) kombinasyonları, port kullanımları, TTL (Time-to-Live) değerleri, trafik yoğunluğu, akış yönü gibi statik fakat örüntüsel bilgiler üzerinden modellenebilir. Bu durumda CNN mimarileri, bu tür uzamsal ilişkileri öğrenme ve tanıma konusunda etkili bir yöntem sunar (Elsayed et al., 2021). Örneğin bir DDoS saldırısında sürekli tekrar eden SYN bayrağı kombinasyonları veya Mirai gibi botnet saldırılarında tipik UDP flood davranışları, CNN filtreleri tarafından tespit edilebilir hale gelir. Çünkü CNN'in evrişim (convolution) katmanları, giriş vektörlerinde birbirine yakın yerlerde

tekrar eden örüntüleri tespit etmeye programlanmıştır. Özellikle 1D veya 2D yapıdaki verilerde bu filtreler, belirli paternleri algılayarak hangi saldırı türünün gerçekleştiğine dair sinyaller üretir. Bazı çalışmalar bu tür uzamsal özelliklerin daha iyi yakalanabilmesi için, ağ trafiğini doğrudan görsel forma çevirmiştir. Örneğin Elsayed ve arkadaşlarının yaptığı çalışmada, 48 öznitelik içeren ağ trafiği özellikleri 8x6 matris hâline getirilmiş ve CNN'e bu yapıyla verilmiştir. Bu sayede klasik zaman serisi temsili yerine, örüntü ağırlıklı bir yapıyla çalışılmıştır ve bu dönüşüm modelin doğruluk oranını doğrudan artırmıştır (Elsayed et al., 2021). CNN filtreleri, ağ trafiğindeki örüntüleri çeşitli boyutlarda algılayabilir. İlk katmandaki filtreler daha temel yapıları (örneğin port tekrarı, SYN flag yoğunluğu gibi) yakalarken, derin katmanlar daha karmaşık saldırı örüntülerini tanıyabilir. Şekil 1'de görülen klasik CNN yapısında bu durum açıkça gösterilmektedir. İlk katmanlarda “kenar gibi” düşük seviye sinyaller, ilerleyen katmanlarda “nesne gibi” bütünsel yapılar hâline gelir (LeCun et al., 1998). Aktivasyon fonksiyonları da burada önemli rol oynar. ReLU ya da Leaky ReLU gibi doğrusal olmayan dönüşümler, örüntülerin daha belirgin hâle gelmesini sağlar. Çünkü bu dönüşümler, sinyalin pozitif kısımlarını korurken, negatif ya da önemsiz olanları bastırarak öğrenme sürecinde modelin odak noktasını optimize eder. Şekil 2'de bu fonksiyonların davranışları grafiksel olarak sunulmuştur. CNN, IoT ağ trafiğindeki örüntülerin yalnızca var olup olmadığını değil, nerede, ne şekilde ve hangi tekrar sıklığıyla ortaya çıktığını öğrenebilir. Bu özellik, saldırıların sadece varlığını değil, hangi saldırı türünün gerçekleştiğini de modelin anlamasına yardımcı olur. Böylece sadece “anomaly” değil, aynı zamanda saldırı tipi de öğrenilebilecek hâle gelir.

2.2.3. CNN tabanlı anomali tespitinde uzamsal öğrenmenin rolü

IoT tabanlı ağ trafiği verileri çoğunlukla zaman serisi şeklinde incelense de, bu veriler aynı zamanda yoğun miktarda uzamsal (spatial) bilgi barındırır. CNN mimarileri, özellikle bu tür uzamsal örüntülerin tanınmasında başarılı yapılar arasında yer alır. CNN'in temel yeteneği, giriş verisindeki bölgesel ilişki ve benzerlikleri küçük filtreler (kernel) ile tarayarak öğrenmesidir. Bu özelliği sayesinde, örneğin ağ paketlerindeki TCP/UDP bayrak kombinasyonları ya da SYN ataklarındaki tekrar eden yapı gibi düşük seviyeli sinyaller, CNN tarafından yakalanabilir hâle gelir (Elsayed et al., 2021). Elsayed ve arkadaşlarının yaptığı çalışmada, CNN'in doğrudan ağ trafiği üzerine uygulanamayacağı ancak özellik mühendisliği ile uygun bir forma getirilebileceği

gösterilmiştir. 48 boyutlu ağ trafiği özellikleri, 8x6'lık matris yapısına dönüştürülerek 2D-CNN mimarisine beslenmiştir. Bu yöntem sayesinde CNN, klasik görüntü işleme mantığında olduğu gibi, ağ trafiğindeki yerel örüntüleri bölgesel olarak analiz edebilmiştir (Elsayed et al., 2021). CNN filtrelerinin ağırlıkları (weights), evrişim işlemi ile giriş üzerinde kaydırılarak öğrenilir. Bu işlem sonucu elde edilen feature map'ler, belirli saldırı türlerine özgü davranışları modellemekte kullanılmaktadır. Şekil 1'de gösterildiği üzere, ilk katmandaki filtreler genellikle temel özellikleri (örneğin edge veya ani sıçrama) çıkarırken, daha derin katmanlar daha karmaşık saldırı örüntülerini yakalayabilir (LeCun et al., 1998). Özellikle DDoS ve Mirai türü saldırılar gibi tekrar eden davranışlarda CNN'in başarı oranının yüksek olduğu görülmüştür. Shen ve arkadaşları tarafından geliştirilen DeepMAD adlı matematiksel mimaride, CNN'in derinliği ve genişliği arasında optimal bir oran belirlenmiş ve bu oranın $\rho=DW$ aralığında [0.1,2.0] arasında tutulmasıyla örüntü tanıma kapasitesinin en üst düzeye çıkarıldığı rapor edilmiştir (Shen et al., 2023). Bu da CNN'in sadece giriş özellikleriyle değil, aynı zamanda mimari tasarımıyla da uzamsal öğrenme kapasitesinin artırılabilceğini göstermektedir. CNN'in aktivasyon fonksiyonları da uzamsal öğrenmenin kalitesini doğrudan etkiler. LeNet mimarisi temel alınarak yapılan karşılaştırmalarda, sigmoid fonksiyonunun doygunluk problemi nedeniyle bilgi kaybına yol açabileceği, buna karşın ReLU ve Leaky ReLU fonksiyonlarının negatif değerleri bastırarak öğrenmeyi daha etkili hâle getirdiği rapor edilmiştir (LeCun et al., 1998). Şekil 2'de gösterilen farklı üç aktivasyon fonksiyonunun çıkış davranışları karşılaştırmalı şekilde gösterilmiştir. CNN mimarileri, IoT veya SDN ağlarında saldırı tespitinde sadece yüksek doğruluk sunmakla kalmaz, aynı zamanda düşük seviye ve bölgesel örüntüleri ayırt etmede klasik tekniklere kıyasla fazlasıyla esnek çözüm sunmaktadır. Özellikle saldırıların belirli bir protokol ya da trafik akışı üzerinde yoğunlaştığı durumlarda, CNN filtrelerinin bu örüntüyü yakalayarak modelin daha sağlam sınıflandırma yapmasını sağladığı görülmektedir.

2.2.4. CNN mimarilerinde derinlik, aktivasyon ve optimizasyon teknikleri

Convolutional Neural Network (CNN) mimarisinde başarıyı etkileyen en kritik faktörlerden biri, modelin derinliği ve bu derinliğin doğru yapılandırılmasıdır. Özellikle IoT gibi kompleks, yüksek boyutlu ve heterojen ağ trafiği verileri üzerinde çalışan CNN modelleri için aktivasyon fonksiyonları, katman sayısı, düzenleme (regularization) stratejileri ve filtre boyutları dikkatle planlanmalıdır. Doğru mimari

kurulmadığında model, ya aşırı öğrenme (overfitting) ya da öğrenememe (underfitting) sorunları ile karşı karşıya kalır (Elsayed et al., 2021). CNN mimarilerinde derinlik, katman sayısı ile ölçülür. Çok katmanlı CNN'ler (deep CNN) daha karmaşık örüntüleri yakalayabilir; ancak bu derinlik beraberinde eğitim süresini uzatır ve parametre sayısını arttırır. Shen ve arkadaşlarının (2023) geliştirdiği DeepMAD yaklaşımında, mimarinin derinliği (D) ile genişliği (W) arasında optimal bir oran olan $\rho=WD$ üzerinden mimari verimlilik analizi yapılmıştır. Bu oran $0.1 \leq \rho \leq 2.00.1$ ve $2.00.1 \leq \rho \leq 2.0$ aralığında tutulduğunda CNN'in daha dengeli ve verimli çalıştığı görülmüştür (Shen et al., 2023). CNN yapısında kullanılan aktivasyon fonksiyonları da öğrenme başarımını doğrudan etkiler. Klasik sigmoid fonksiyonu düşük gradyan üretimi sebebiyle derin ağlarda “vanishing gradient” sorununa yol açabilmektedir. Bu nedenle modern CNN mimarilerinde daha çok Leaky ReLU ve ReLU (Rectified Linear Unit) tercih edilmektedir. ReLU fonksiyonu negatif değerleri sıfıra indirerek öğrenme sürecini hızlandırırken, Leaky ReLU ise negatif değerleri tamamen sıfırlamak yerine küçük bir eğimle geçiş sağlayarak bilgi kaybını azaltır (LeCun et al., 1998).

CNN mimarilerinin aşırı öğrenme (overfitting) eğilimi, özellikle de veri setlerinde sınıf dengesizliği varsa daha belirgindir. Bu durumu azaltmak için iki ana optimizasyon tekniği kullanılır:

1. Dropout: Eğitim esnasında nöronlardan bazıları rastgele devre dışı bırakılmasıyla öğrenme sürecinde modelin aşırı ezber yapması engellenir. Elsayed ve arkadaşları, 0.25 ve 0.5 oranlarında uygulanan dropout'un CNN modelinin genellebilirliğini artırdığını ve test doğruluğunu %90.79'dan %93.83'e çıkardığını göstermiştir (Elsayed et al., 2021).
2. L2 Regularizasyonu: Ağırlıkların büyüklüğüne ceza ekleyerek modelin çok büyük ağırlıklara sapmasını önler. Bu da hem overfitting'i baskılar hem de modelin farklı veri setlerine uyumunu artırır. Aynı çalışmada L2 regularizasyonu, CNN modelinde false-positive oranlarını düşürmüştür.

Mimari tasarımda 1D yerine 2D CNN kullanımı, veri girişinin matris biçimine çevrilerek parametre paylaşımının artırılması açısından etkili bir yöntemdir. Elsayed'in çalışmasında 48 ağ özelliği 8x6 matris yapısına çevrilmiş ve 2D-CNN ile öğrenilmiştir. Bu yöntem, filtrelerin daha etkili çalışmasına olanak sağlamış ve özellik çıkarımında daha etkili sonuçlar elde edilmiştir (Elsayed et al., 2021). CNN

mimarisinde katman sayısı, aktivasyon tipi ve düzenleme tekniklerinin doğru kombinasyonu, modelin öğrenme gücünü doğrudan etkilemektedir. Aksi takdirde, eğitim süreci fazla uzun sürer veya model yeni verilerde başarısız sonuçlar üretir. Literatürde bu konulara odaklanan çalışmalar, özellikle IoT ağ trafiği gibi karmaşık ortamlarda CNN mimarilerinin başarıya ulaşmasında bu yapıların önemini vurgulamaktadır.

2.2.5. CNN-LSTM hibrit mimarilerinin saldırı tespitindeki etkinliği

Geleneksel derin öğrenme mimarilerinde tekil modellerin belirli sınırlılıkları bulunmakta; örneğin CNN mimarileri yalnızca uzamsal (spatial) örüntüleri öğrenirken, LSTM yapıları ise zaman içinde oluşan ilişkileri yakalayabilmektedir. Bu iki yapının eksik yönlerini kapatmak ve avantajlarını bir araya getirmek amacıyla hibrit mimariler geliştirilmiş; özellikle CNN-LSTM kombinasyonu, ağ trafiğindeki karmaşık saldırı davranışlarını tanımada literatürde öne çıkmıştır (Elsayed et al., 2021). CNN-LSTM yapısı, genellikle iki aşamalı bir öğrenme süreci ile çalışır. İlk aşamada CNN katmanları, girdi verisi üzerinde lokal örüntüleri çıkarır. Bu örüntüler, ardından LSTM katmanlarına aktarılır ve burada zaman içinde oluşan ilişkiler modellenir. Özellikle IoT ya da SDN trafiği gibi hem kısa süreli hem de uzun süreli saldırı davranışlarının görüldüğü ortamlarda bu yapı oldukça etkilidir. Örneğin DDoS saldırıları kısa vadeli patlamalarla ortaya çıkarken, botnet iletişimi gibi davranışlar uzun süreli ve düşük yoğunluklu olabilir. CNN-LSTM bu iki farklı örüntüyü aynı anda işleyebilir (Elsayed et al., 2021).

Yapılan bazı karşılaştırmalı deneylerde, CNN-LSTM modelinin tek başına kullanılan CNN veya LSTM yapılarına göre daha yüksek başarı sağladığı rapor edilmiştir. Elsayed ve arkadaşları tarafından yapılan deneylerde, standart CNN modeli %90.79 doğruluk sağlarken, LSTM modeli %93.85, dropout ve L2 regularizasyon uygulanmış CNN modeli %93.83 doğruluk elde etmiştir. Buna karşın CNN-LSTM hibrit mimarisi, %96.32 doğruluk oranı ve %97.42 F1 skoru ile tüm diğer yöntemleri geride bırakmıştır (Elsayed et al., 2021). Aynı çalışmada yer alan confusion matrix sonuçları da CNN-LSTM modelinin özellikle saldırı sınıfını tanımada diğer yöntemlerden daha başarılı olduğunu ortaya koymuştur. CNN-LSTM ile yapılan sınıflandırmada, saldırı sınıfında %97'lik doğru tanılama başarısı elde edilirken, normal trafik için ise bu oran %94 civarındadır. Bu fark, özellikle eğitim verisinde normal trafik örneklerinin daha az olmasıyla ilişkilendirilmiştir. Bunun yanı sıra ROC (Receiver Operating

Characteristic) eğrisinin altında kalan alan AUC (Area Under Curve) değeri de CNN-LSTM modelinde en yüksek seviyede rapor edilmiştir. AUC değeri CNN-LSTM için 0.956 iken, LSTM modeli 0.945, CNN (L2 Reg.) 0.947 ve standart CNN modeli ise 0.928 AUC değerine ulaşmıştır. Bu da, hibrit mimarinin sınıflar arasındaki ayrımı daha net ve kararlı şekilde yaptığı anlamına gelir. Bu performans artışının temel sebeplerinden biri, CNN tarafından çıkarılan yüksek boyutlu fakat anlamlı uzamsal temsillerin, LSTM tarafından zaman içinde işlenerek karar verici katmana iletilmesidir. Böylece verideki küçük ama önemli paternler, sadece zamansal ya da sadece uzamsal değil, her iki bağlamda da işlenmiş olur. Ancak CNN-LSTM yapılarının da bazı zayıf yönleri mevcuttur.

Modelin derinliği arttıkça parametre sayısı da artmakta, bu da özellikle sınırlı donanım kaynaklarında (örneğin edge cihazlar) kullanımı kısıtlamaktadır. Ayrıca sınıf dengesizliği olan veri setlerinde LSTM katmanları, azınlık sınıfların örüntülerini öğrenmekte zorlanabilir. Bu nedenle bazı çalışmalarda class weighting, focal loss veya SMOTE tabanlı örnekleme gibi destekleyici stratejiler önerilmektedir (Shen et al., 2023). CNN-LSTM hibrit mimarileri, IoT ve SDN gibi karmaşık yapılı, çok değişkenli sistemlerde hem doğruluk oranı hem de genelleme kabiliyeti açısından önemli avantajlar sunmaktadır. Ayrıca günümüzde, IDS modelleri arasında en çok tercih edilenlerdendir.

2.3. BiLSTM (Bidirectional LSTM)

BiLSTM(Bidirectional Long Short-Term Memory), veriyi ileri ve geri yönde okuyarak zaman içindeki ilişkiyi çift taraflı olarak öğrenir. Bu sayede özellikle normal/saldırı arasındaki geçiş davranışları daha başarılı şekilde yakalanabilir. Literatürde CNN-BiLSTM mimarisinin, klasik CNN-LSTM modeline göre daha düşük false-positive oranlarına ulaştığı görülmüştür (Nazir et al., 2024) Ancak bu mimarinin hesaplama yükü biraz daha fazladır, bu yüzden edge veya düşük donanımlı sistemlerde zorluk çıkartabilir.

2.3.1. BiLSTM mimarisinin temel yapısı

BiLSTM ağları, klasik LSTM yapısına çift yönlü öğrenme yetisi ekleyerek, zaman serisi verilerde geçmiş ve gelecek bağlamların eş zamanlı olarak öğrenilmesini sağlar. LSTM yapısındaki giriş kapısı (input gate), unutma kapısı (forget gate) ve çıkış kapısı (output gate); bilgi kaybını minimize ederken uzun dönemli bağımlılıkların

korunmasına olanak tanır. BiLSTM ise, bu yapıyı hem ileri hem geri yönde çalıştırarak daha zengin bir temsil öğrenimi sağlar (Zhao et al., 2023). Her bir zaman adımı için hem geçmişe hem geleceğe dönük bilgilerin elde edilmesi, özellikle ağ trafiği gibi doğrusal olmayan ve zamana duyarlı verilerde kritik önemdedir.

BiLSTM'nin çıktısı, denklem 2.9' da görüldüğü üzere her zaman adımı için ileri yönlü gizli durum vektörü ($\rightarrow h_t$) ile geri yönlü gizli durum vektörünün ($\leftarrow h_t$) birleşiminden oluşur:

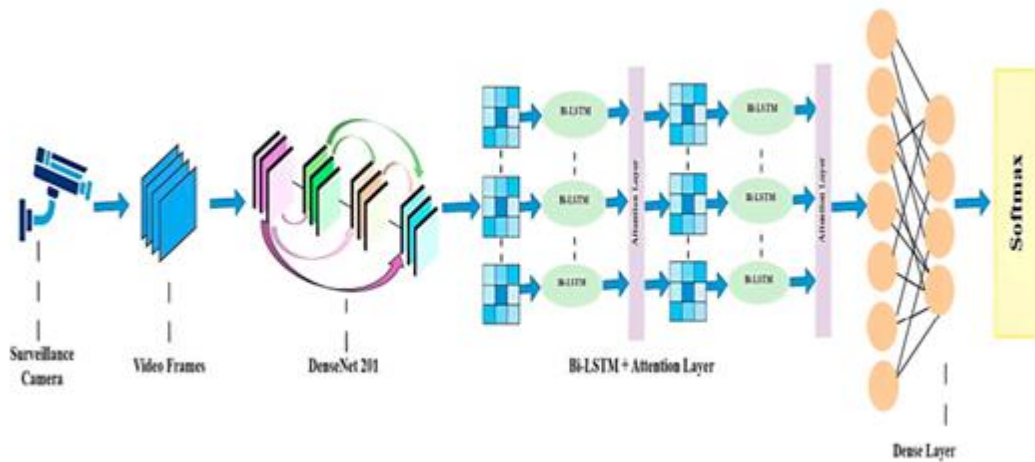
$$h_t = \rightarrow h_t \oplus \leftarrow h_t \quad (2.9)$$

Bu çift yönlü yapı, özellikle DDoS ve Mirai gibi saldırı türlerinin tespitinde sıklıkla tercih edilmektedir. Çünkü bu saldırılar genellikle zamansal paternler içerir ve tek yönlü modeller bu paterni tam olarak yakalayamaz.

2.3.2. Attention katmanları ile güçlendirilmiş BiLSTM yapısı

BiLSTM katmanlarının dikkat (attention) mekanizmasıyla birleştirilmesi, modelin öğrenilen zamansal bilgileri ağırlıklandırmasını ve kritik bilgileri daha öne çıkarmasını sağlar. Bu yaklaşım, özellikle ağ trafiği verilerinde yüksek boyutlu ve redundant bilgi yapılarıyla başa çıkabilmek için oldukça etkilidir.

Attention mekanizması, her bir zaman adımına ağırlık vererek hangi bilgilerin daha anlamlı olduğunu öğrenir (Zhao et al., 2023). Böylece model sadece verinin zaman içindeki sıralı yapısını değil, aynı zamanda bağlamsal önem derecesini de kavrayabilir.



Şekil 2.5. BiLSTM + attention tabanlı model mimarisi (Zrira et al., 2024).

Şekil 2.5'te yer alan mimaride de görüldüğü üzere, DenseNet ile çıkarılan öznetelikler sırasıyla BiLSTM katmanlarına verilir ve ardından her katmandan sonra attention uygulanarak anlamlı örüntülerin öne çıkarılması sağlanır (Saba et al., 2022).

2.3.3. Modelin aşamaları ve katman yapısı

Şekil 2.5'te sunulan mimari, güvenlik kamerasından alınan video verilerinin DenseNet201 üzerinden çıkarılarak zaman serisi formuna dönüştürülmesini ve ardından üçlü BiLSTM katmanı ile işlenmesini içermektedir. Bu BiLSTM katmanları, dikkat (attention) katmanları ile desteklenmiş ve her bir alt dizilim için önemli örüntüler vurgulanmıştır. Nihayetinde, yoğun katmanlar (dense layers) aracılığıyla sınıflandırma yapılmış ve Softmax fonksiyonundan yararlanılarak çok sınıflı bir çıktı elde edilmiştir. Bu yapı, özellikle yüksek gürültülü ve çoklu saldırı içeren veri kümelerinde avantaj sağlamaktadır. Katmanlar arasında bilgi kaybını önlemek amacıyla dropout gibi düzenleme teknikleri de kullanılmıştır (Zrira et al., 2024). Bu yapı saldırı tespiti gibi hassas görevlerde güçlü doğruluk sağlamaktadır.

2.3.4. Literatürde BiLSTM performansı ve başarı oranları

Literatürde BiLSTM tabanlı modellerin özellikle CIC-IDS2017 ve CIC-DDoS2019 gibi veri setlerinde yüksek doğruluk ve F1 skorları sağladığı görülmektedir. Örneğin Zhao ve arkadaşları tarafından geliştirilen CNN-BiLSTM-Attention modelinde %95.67 doğruluk, %95.90 doğruluk, ve %95.86 F1 skoru elde edilmiştir (Zhao et al., 2023). Bu sonuçlar, klasik LSTM ve CNN tabanlı modellere kıyasla istatistiksel olarak önemli derecede daha başarılıdır. Ayrıca, Sensors dergisinde yayımlanan bir başka çalışmada, yalnızca BiLSTM yerine attention destekli BiLSTM kullanıldığında modelin ortalama hata oranında (RMSE) %10 civarında iyileşme görülmüştür (Zrira et al., 2024). Bu bağlamda, BiLSTM modelleri, hem doğruluk hem de düşük yanlış alarm oranı ile IoT güvenlik uygulamalarında öne çıkmaktadır.

2.3.5. BiLSTM kullanımının avantajları ve sınırlılıkları

BiLSTM modelleri çift yönlü bağlam öğrenimi ile özellikle zaman serisi odaklı veri kümelerinde oldukça güçlüdür. Ancak modelin karmaşık yapısı daha fazla hesaplama gücü ve belleğe ihtiyaç duyduğu için, özellikle IoT gibi sınırlı kaynak ortamlarında optimize edilmeden uygulanması zor olabilir. Ayrıca, zaman serisinin uzunluğu arttıkça modelin dikkat mekanizması performansı da düşebilir. Bu nedenle, genellikle attention mekanizmasıyla desteklenerek önem derecesi düşük veriler elenmelidir.

Diğer yandan, uygun hiperparametre seçimi yapılmadığında BiLSTM modelleri overfitting eğilimi gösterebilir. Bu gibi durumlarda erken durdurma (early stopping) ve dropout yöntemleri önerilmektedir (Saba et al., 2022).

2.4. Transformer

Transformer mimarileri son yıllarda NLP'de gösterdiği başarıyı zaman serisi ve siber güvenlik alanlarına da taşımaya başlamıştır. Transformer, self-attention mekanizması sayesinde verinin zaman içindeki uzak ilişkilerini çok daha iyi modelleyebilir. Ayrıca paralel hesaplama yapısına olanak sağlaması nedeniyle LSTM'e kıyasla daha hızlı çalışabilir (Xu et al., 2022). Bazı çalışmalarda, özellikle GTA ve Anomaly Attention gibi yapılarla hem grafik tabanlı topoloji bilgisi hem de zamansal ilişki bir arada işlenmiştir (Chen et al., 2022). Bu modellerde örneğin Gumbel-Softmax sampling ile sensörler arası bağlantı yapısı öğrenilmiş, attention katmanlarıyla örüntü propagasyonu yapılmıştır. Sonuçlar, Attention tabanlı modellerin LSTM ve CNN modellerine göre daha iyi genellenebilirlik sunduğunu göstermektedir (Tuli et al., 2022).

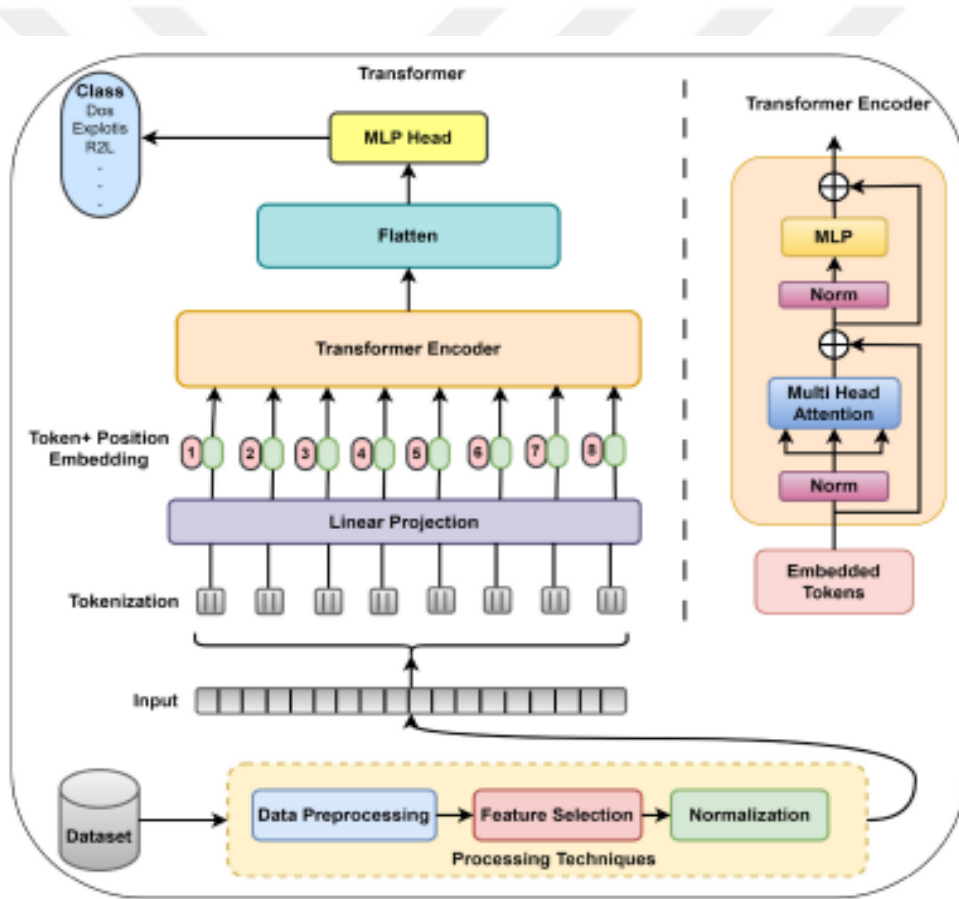
2.4.1. Transformer mimarisi ve temelleri

İlk kez doğal dil işleme alanında tanıtılmış olan Transformer mimarisi, sonrasında zaman serisi verileri gibi sıralı veri türlerine uygulanmaya başlanmıştır. Bu mimarinin temel yapı taşı “self-attention” yani kendi üzerine dikkat mekanizmasıdır. Bu mekanizma sayesinde, her girdi ögesinin diğer öğelerle aralarındaki ilişkiyi hesaplayarak, hangi girdinin daha önemli olduğunu kendi başına öğrenir. Geleneksel RNN ya da LSTM yapılarından farklı olarak, Attention paralel hesaplama yapabildiği için daha hızlıdır ve uzun bağımlılıkları daha başarılı şekilde modelleyebilir (Akuthota & Bhargava, 2025). Matematiksel olarak dikkat mekanizması, üç temel vektör üzerinden çalışır: Value (V), Query (Q) ve Key (K). Her bir giriş vektörü bu üç vektöre dönüştürülür ve ardından aşağıdaki formülle dikkat ağırlıkları hesaplanır:

$$Attention(Q, K, V) = Soft \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.10)$$

Bu ifade, Vaswani ve arkadaşlarının (2017, s. 4) önerdiği Scaled Dot-Product Attention yapısına dayalıdır. Burada Key vektörlerinin boyutunu ifade eder ve paydanın kareköküyle ölçeklendirme işlemi, iç çarpım değerlerinin çok büyük olmasını engelleyerek eğitim sürecinde sayısal kararlılığı artırır (Vaswani et al., 2017;

Akuthota & Bhargava, 2025). Attention yapısı tipik olarak Encoder ve Decoder' den oluşan iki temel bloktan oluşsa da, saldırı tespiti gibi sınıflandırma problemlerinde genellikle yalnızca Encoder kısmı kullanılır. Her bir Encoder bloğu içinde katman normalizasyonu (LayerNorm), sıralı tam bağlantılı katmanlar (MLP), çok başlı dikkat (multi-head attention) ve artık bağlantılar (residual connections) bulunmaktadır. Çok başlı dikkat mekanizması, veriyi farklı temsillerde paralel olarak işleyerek modelin çeşitli ilişki örüntülerini aynı anda öğrenmesini sağlar (Zeng et al., 2023). Bu mimaride pozisyon bilgisi eksik olduğu için giriş vektörlerine pozisyon gömme (positional encoding) eklenerek sıradaki öğelerin konumu da modele öğretilir. Böylece zaman serisindeki örneklerin sıralı yapısı korunur. Bu yapı, özellikle sıralı ve zamana duyarlı ağ verilerinin işlendiği durumlarda etkili performans sergiler.



Şekil 2.6. Transformer (attention mekanizması) kullanarak saldırı tespiti süreci.

Şekil 2.6, Attention tabanlı bir saldırı tespit sisteminde kullanılan tipik işlem hattını görselleştirmektedir. Aşamalar, veri ön işleme adımlarından başlayarak tokenlaştırma, pozisyon gömme, çok başlı dikkat mekanizması içeren Attention encoder katmanları ve son sınıflandırma adımlarına kadar detaylı biçimde sunulmuştur. Modelin girişine

alınan zaman serisi ağ verileri, ardışık şekilde gömme katmanları ve encoder modüllerinden geçirilir. Bu süreçte özellikle pozisyonel bilgilendirme ve dikkat mekanizması sayesinde sıralı ilişkiler başarıyla modellenir.

Transformer mimarisi sayesinde uzun sıralı bağımlılıkları yakalamak mümkün hale gelirken, aynı zamanda paralel işlem yetisi sayesinde eğitim süreleri de ciddi oranda azalır. Bu da onu IoT tabanlı saldırı tespit sistemleri gibi büyük veri içeren alanlarda ideal bir seçenek haline getirmiştir (Xie et al., 2021; Akuthota & Bhargava, 2025).

2.4.2. Zaman serisi verileri üzerindeki uygulaması

Transformer mimarisinin zaman serisi veriler üzerindeki uygulaması, özellikle ağ trafiği gibi çok boyutlu ve sıralı yapıdaki veri tiplerinde giderek önem kazanmaktadır. IoT sistemlerinden elde edilen veriler genellikle ardışık sensör çıktılarından oluştuğu için, bu tarz verilerin işlenmesinde zaman bağımlılıklarını modellemek hayati öneme sahiptir. Transformer mimarisi ise self-attention yapısı sayesinde bu ardışık örüntüleri, klasik RNN veya LSTM mimarilerinde olduğu gibi sırayla değil, doğrudan paralel olarak işlemektedir. Bu da zaman serisi verilerde hem kısa hem uzun dönemli ilişkilerin daha verimli şekilde öğrenilmesini mümkün kılar (Zeng et al., 2023). Attention yapısında giriş verileri belirli uzunlukta kaydırmalı pencere (sliding window) yaklaşımı ile bölünür. Her pencere, sabit uzunlukta bir zaman dizisi dilimi (örneğin $x_t-w+1:t$) olarak ele alınır. Bu pencere, giriş embedding katmanına iletilmeden önce normalize edilir ve gerekirse boyutsal dönüşüm uygulanır. Ardından pozisyon gömme (positional encoding) eklenerek zaman eksenindeki sıralama bilgisi korunur (Akuthota & Bhargava, 2025). Zaman serisi verilerde Attention uygulamaları genellikle iki yaklaşım altında gruplanır: tahmin temelli (prediction-based) ve yeniden inşa temelli (reconstruction-based). Birinci yaklaşımda model, verilen zaman penceresinden sonraki değeri tahmin etmeye çalışırken, ikinci yaklaşıma göre model orijinal pencereyi tekrar oluşturmaya çalışır ve orijinal ile yeniden inşa edilen veri arasındaki farktan anomali çıkarımı yapılır (Xie et al., 2021). Özellikle bu ikinci yaklaşım, saldırı tespiti gibi anomali bazlı uygulamalarda yaygın şekilde tercih edilmektedir.

Attention tabanlı yapılarda, dikkat (attention) mekanizması girişteki tüm zaman adımlarını eşzamanlı olarak işler ve her bir zaman adımının diğeriyle olan ilişkisini değerlendirir. Bu işlem, klasik RNN tabanlı yapılarla kıyaslandığında zaman içinde dağıtılmış ilişkileri daha isabetli bir şekilde modelleyebilmeyi mümkün kılar. Zeng ve arkadaşları tarafından geliştirilen Adformer modeli, iki aşamalı dikkat blokları ve ön

bilgi (prior knowledge) stratejisi ile, zaman serisinde kısa dönemli örüntüleri daha iyi yakalayarak küçük boyutlu anomalileri bile tespit edebilmektedir (Zeng et al., 2023). Zaman serisi analizinde dikkatli olunması gereken diğer konu, sıralama maskesi (sequence masking) kullanımıdır. Eğer veride gelecek bilgiye erişim istenmiyorsa, üçgen maske (causal mask) uygulanarak yalnızca geçmiş bilgiye odaklanılması sağlanır. Ancak saldırı tespiti gibi uygulamalarda genellikle tüm zaman penceresi birlikte işlendiği için bu tür maskelere ihtiyaç duyulmamaktadır. Transformer mimarisi zaman serisi verilerin işlenmesinde hem yüksek doğruluk hem de hesaplama verimliliği sağlayan güçlü bir alternatiftir. Özellikle IoT tabanlı ağ ortamlarında karşılaşılan yüksek boyutlu ve çok sensörlü zaman dizilerini işlemede Transformer'ın paralel işlem kapasitesi büyük avantaj sağlamaktadır (Akuthota & Bhargava, 2025; Zeng et al., 2023).

2.4.3. IoT ağlarında attention tabanlı IDS kullanımı

Nesnelerin İnterneti (IoT) tabanlı ağlar, birbirine bağlı milyonlarca cihazdan oluştuğu için hem geniş hacimli hem de düzensiz özellik gösteren ağ trafiği üretir. Bu trafiğin güvenli şekilde izlenebilmesi için saldırı tespit sistemlerinin (IDS) hem yüksek doğrulukta hem de düşük gecikmeyle çalışması gerekmektedir. Transformer mimarisi, özellikle self-attention mekanizması sayesinde, bu ihtiyaçlara yanıt veren kuvvetli bir çözüm olmaktadır (Akuthota & Bhargava, 2025). IoT ağlarında trafik örüntüleri genellikle zamansal korelasyon içerdiğinden, geleneksel makine öğrenmesi yöntemleri bu örüntüleri sınırlı ölçüde öğrenebilir. LSTM gibi sırayla çalışan yapılar da uzun dizilerde bilgi kaybı yaşayabilir. Attention yapıları ise her zaman adımındaki örneği, dizideki tüm diğer örneklerle aynı anda değerlendirme avantajına sahiptir. Bu özellik, özellikle düşük hacimli ancak yüksek etkili saldırıların tespitinde önemli bir katkı sağlar (Zeng et al., 2023). Attention tabanlı IDS yapılarında genellikle işlem şu şekilde ilerler: IoT ağından alınan çok değişkenli zaman serisi verileri önce normalize edilir ve kaydırmalı pencere yaklaşımıyla bölünür. Her pencere, belirli bir zaman dilimine ait tüm sensör verilerini içerir. Bu pencere vektörü, Attentionencoder'a beslenir. Encoder içerisinde yer alan çok başlı dikkat katmanları, her zaman noktasının diğerleriyle olan ilişkisini değerlendirerek anomali içeren paternleri ortaya çıkarır (Xie et al., 2021). Örneğin, Akuthota ve Bhargava (2025) tarafından önerilen Attention temelli saldırı tespit modeli, UNSW-NB15 ve NSL-KDD veri kümelerinde %99.5'in üzerinde doğruluk sağlamıştır. Modelin başarısı, özellikle "tokenizasyon", pozisyon

gömme ve çok başlı dikkat katmanlarının etkili kullanımı ile ilişkilendirilmektedir. Model, ön işlem olarak özellik seçimi (correlation based selection) uygulamakta ve ardından dönüştürülen verileri encoder katmanları üzerinden geçirerek nihai sınıflandırmayı gerçekleştirmektedir.

IoT sistemlerinin sınırlı kaynaklarla çalıştığı dikkate alındığında, Attention yapısının paralel hesaplama avantajı oldukça önemlidir. Geleneksel tekrarlayan ağlarda olduğu gibi zamansal sırayı tek tek işlemeden, tüm pencere paralel biçimde değerlendirildiğinden sistem üzerindeki işlem yükü azaltılabilmektedir (Akuthota & Bhargava, 2025). Transformer mimarisi, IoT ağlarında saldırı tespiti için yalnızca yüksek doğrulukta kararlar vermekle kalmayıp, aynı zamanda küçük anomali paternlerini de tespit edebilme, yüksek hızda işlem yapabilme ve çok değişkenli verileri birlikte değerlendirme becerisi sayesinde diğer yöntemlerin önüne geçmektedir.

2.4.4. Transformer mimarilerinin avantaj ve sınırlılıkları

Transformer mimarisi, özellikle saldırı tespit sistemlerinde sağladığı yüksek doğruluk, esnek yapı ve paralel işlem kapasitesi gibi avantajlarla literatürde öne çıkmaktadır. IoT gibi zaman serisine dayalı veri kümelerinde klasik yöntemlerin yetersiz kaldığı noktalarda Attention yapıları güçlü bir alternatif sunmaktadır. Bunun başlıca nedeni, modelin tüm zaman adımlarını eş zamanlı işleyerek uzun vadeli ilişkileri yakalayabilme yeteneğidir (Xie et al., 2021). Transformer mimarisinin en büyük avantajlarından biri, işlem süresini büyük ölçüde azaltmasıdır.

LSTM ve RNN gibi yapıların sıralı bilgi akışına ihtiyaç duyması, hesaplamaları ardışık hale getirirken, Transformer mimarisi bu işlemleri tamamen paralel yaparak eğitim süresini ciddi ölçüde kısaltır (Akuthota & Bhargava, 2025). Bununla birlikte, çok başlı dikkat (multi-head attention) sayesinde model, verideki farklı ilişki türlerini aynı anda öğrenebilir; bu da doğruluğu ve genelleme kapasitesini artırır. Attention yapısının modülerliği, farklı veri türleri veya görevler için kolayca uyarlanabilir olmasını sağlar. Örneğin CNN- Transformer veya BiLSTM- Transformer hibritleri geliştirerek hem uzamsal hem zamansal bağıntılar yakalanabilir (Zeng et al., 2023). Transformer mimarisinin bazı sınırlılıkları da vardır. Özellikle dikkat mekanizmasının hesaplama karmaşıklığı, girdi uzunluğu arttıkça karesel şekilde büyür:

$$O(n^2 \cdot d) \quad (2.11)$$

Bu ifade, tüm pozisyonlar arası dikkat skorlarının ayrı ayrı hesaplanması gerektiğini gösterir. Transformer mimarisiyle ilgili bu detay, Vaswani ve arkadaşlarının çalışmasında da açıkça belirtilmiştir (Vaswani et al., 2017, s. 6). Bu nedenle uzun dizilerle çalışırken, özellikle bellek kullanımı ve işlem süresi açısından self-attention yapısının verimliliği önemli bir faktördür. Burada n giriş dizisinin uzunluğu, d ise gömme boyutudur. Bu durum, düşük donanım kapasitesine sahip IoT cihazlarında modelin doğrudan çalıştırılmasını zorlaştırabilir (Akuthota & Bhargava, 2025). Bir diğer sınırlılık, eğitim için ihtiyaç duyduğu veri hacmidir. Transformer modelleri çoğunlukla büyük miktarda veri hacmine gereksinim duyar ve aşırı öğrenme (overfitting) eğilimi gösterebilir. Bu nedenle, dikkatli regularizasyon ve dikkat katmanlarının sınırlı derinlikte kullanımı önerilir. Ayrıca, pozisyonel bilgi doğrudan modele öğretilmediğinden, zaman duyarlılığı için pozisyon gömme stratejisinin dikkatli tasarlanması gerekir (Xie et al., 2021). Transformer mimarisi ağ güvenliğinde özellikle çok boyutlu ve çok sınıflı veri setlerinde yüksek verim sağlarken; bellek kullanımı, zaman karmaşıklığı ve veri ihtiyacı gibi birtakım kısıtlamaları da vardır. Bu bağlamda, hibrit ve optimize edilmiş Attention yapılarının geliştirilmesi üzerine yoğunlaşan çalışmalar dikkat çekmektedir. Attention modelleri çok boyutlu ve sıralı veriler üzerinde sağlam sonuçlar verirken, yapısal karmaşıklıkları ve yüksek veri ihtiyacı sebebiyle özellikle kaynak kısıtlı sistemlerde dikkatli uygulanmalıdır.

Derin öğrenme tabanlı farklı mimarilerin güçlü ve zayıf yönlerinin karşılaştırmalı olarak değerlendirildiği Tablo 2.1’de literatürde sıkça kullanılan yaklaşımların hangi koşullarda avantaj sağladığını, hangi modelin hangi durumlarda tercih edildiğini özetlemektedir. Tablo 2.1’deki karşılaştırma gösteriyor ki; IoT gibi dinamik ve çok kaynaklı veri ortamlarında Transformer mimarileri paralel işlem ve uzun bağımlılık öğrenme açısından ciddi avantajlar sunmaktadır. Fakat modelin büyük miktarda veri ihtiyacı ve karmaşık yapısı, bu yöntemi yalnızca güçlü donanım kaynaklarının bulunduğu sistemler için kullanışlı hale getirmektedir. Daha düşük kaynak tüketen ve zaman sınırlı uygulamalarda ise LSTM veya CNN temelli yaklaşımlar tercih edilebilmektedir. Dolayısıyla, veri yapısı, saldırı tipi ve sistem kaynaklarına bağlı olarak doğru mimari seçimi, saldırı tespit sisteminin performansına doğrudan tesir eden ana unsurlardandır (Akuthota & Bhargava, 2025; Xie et al., 2021).

Tablo 2.1. Derin öğrenme mimarilerinin karşılaştırması: güçlü yönler, zayıf yönler ve tipik kullanım alanları.

Mimari	Güçlü Yön	Zayıf Yön	Tipik Kullanım
LSTM	Zaman serisi takibi	Eğitim yavaş, overfitting riski	SYN/UDP flood
CNN	Özellik filtreleme	Zaman bağımlılığı zayıf	TCP/UDP özellik sınıflama
BiLSTM	Geçiş örüntüleri	Hesaplama yükü yüksek	Karmaşık saldırı zincirleri
Transformer	Uzun bağımlılık, paralel çalışma	Büyük veri ihtiyacı	Çok cihazlı IoT ağları

2.5. Mevcut Çalışmaların Başarı Durumları

Günümüzde IoT ağlarında saldırıları saptamak amacıyla derin öğrenme tabanlı geliştirilen sistemler, geleneksel yöntemlere kıyasla oldukça yüksek başarı oranlarıyla öne çıkmaktadır. Son yıllarda özellikle hibrit yapılar, hem zamansal hem uzamsal bilgiyi öğrenme artıları sayesinde sıkça tercih edilmektedir (Nazir et al., 2024). CNN-LSTM mimarisi, örneğin Sun et al. (2020) tarafından önerilen DL-IDS sisteminde olduğu gibi, çok sınıflı saldırı tespitinde %98.67 doğruluk elde ederek dikkat çekmiştir. Bu model, hem CICIDS2017 hem de güncel veri kümelerinde yüksek performans sergileyebilmiştir (Sun et al., 2020). Benzer şekilde, Gueriani ve arkadaşları (2024) tarafından önerilen CNN-BiLSTM mimarisi, CIC-IoT2023 verisi üzerinde %98.42 doğruluk ve %98.57 F1 skoru ile literatürdeki birçok çalışmadan üstün performans göstermiştir (Gueriani et al., 2024). Attention tabanlı modeller, özellikle uzun süreli bağımlılıkları öğrenme kabiliyetleri dolayısıyla modern sistemlerde tercih edilmektedir. Tuli et al. (2022) tarafından geliştirilen Anomaly Attention modeli, CICIDS2017 veri seti üzerinde %94'ün üzerinde F1 skoru elde etmiş ve eğitim süresini LSTM tabanlı modellere göre %60 oranında kısaltmıştır (Tuli et al., 2022). Bu avantajlar, büyük IoT ağları için gerçek zamanlı sistemlere entegrasyon konusunda Attentionmimarilerine güçlü bir temel sunmaktadır (Xu et al., 2022). LSTM ve BiLSTM gibi tek başına sıralı model mimarileri de halen geçerliliğini korumaktadır. Halbouni et al. (2022) tarafından geliştirilen CNN-LSTM modeli, ikili

ve çoklu sınıflandırmada yüksek başarı elde etmiş, özellikle batch normalization ve dropout katmanlarıyla modelin genellenebilirliği artırılmıştır (Halbouni et al., 2022). Jony ve Arnob (2024), CIC-IoT2023 veri seti üzerinde farklı modelleri karşılaştırdıkları çalışmalarında CNN-BiLSTM mimarisinin %98.8 F1 skoru ve %99.1 doğruluk neticesinde en başarılı sistem olduğunu göstermişlerdir. Özellikle 105 farklı cihaz ve 33 saldırı türünü içeren bu veri setinde bu denli yüksek başarı oranlarına ulaşılması, modelin genel geçer yeteneğini ortaya koymaktadır (Jony & Arnob, 2024).

Tablo 2.2. Literatürdeki çeşitli derin öğrenme modellerinin saldırı tespiti performansları.

Çalışma	Mimari	Veri Seti	Doğruluk (%)	F1 Skoru (%)	False Positive (%)
Sun et al. (2020)	CNN-LSTM	CICIDS2017	98.67	>99.50 (her sınıf)	-
Gueriani et al. (2024)	CNN-BiLSTM	CIC-IoT2023	98.42	98.57	9.17
Halbouni et al. (2022)	CNN-LSTM	CICIDS2017, UNSW-NB15	Yüksek	Yüksek	Düşük
Jony & Arnob (2024)	CNN-BiLSTM	CIC-IoT2023	99.10	98.80	-
Tuli et al. (2022)	Attention (TranAD)	CICIDS2017	-	94.00	-
Xu et al. (2022)	Anomaly Transformer	Çoklu	-	~94.00	-
Nazir et al. (2024)	CNN-LSTM (Gelişmiş)	CICIDS2017, IoT-23	99.00	99.00	-

Literatürde CNN-LSTM, BiLSTM-Attention ve Attention tabanlı sistemlerin yüksek başarı oranlarına ulaştığı görülmektedir. Ancak özellikle sınıf dengesizliği, overfitting ve gerçek zamanlı sistemlerdeki entegrasyon gibi sorunlar halen çözülmesi gereken

başlıca zorluklardandır (Ogobuchi et al., 2024). Tablo 2.2’de farklı çalışmalarda kullanılan derin öğrenme mimarilerinin IoT ağlarında saldırı tespiti üzerine elde ettiği başarı ölçütlerini göstermektedir. CNN-LSTM ve BiLSTM tabanlı hibrit yapılar F1 skorları ve yüksek doğruluk oranlarıyla ileri çıkarken, Attention tabanlı mimariler ise özellikle eğitim süresi ve paralel işleme kabiliyeti ile avantaj sağlamaktadır. Tablo 2.3’te literatürdeki çeşitli derin öğrenme modellerinin saldırı tespiti performansları detaylı karşılaştırılması yer almaktadır.

Tablo 2.3. Literatürdeki çeşitli derin öğrenme modellerinin saldırı tespiti performansları detaylı karşılaştırılması.

Kaynak / Çalışma	Model Türü	Geliştirme Yöntemi	Veri Seti	Sınıflandırma Tipi	Accurac y (%)	Precisio n (%)	Recall (%)	F1-score (%)
Sun et al. (2020)	CNN-LSTM	Derin Öğrenme	CICIDS 2017	Multi Class DDos, Brute F.	98.67	>99.50	>99.50	>99.50
Nazir et al. (2024)	CNN-LSTM	Derin Öğrenme	IoT-23	Multi Class	94.89	94.89	94.32	93.99
Nazir et al. (2024)	CNN-LSTM	Derin Öğrenme	CICIDS 2017	Multi Class	DDoS - 94.32 Min. - 82.34	DDoS - 99.32 Min. - 82.91	DDoS - 97.88 Min. - 76.53	DDoS - 98.60 Min. - 82.34
Mahmoud Elsayed et al. (2021)	SD-Reg ve CNN	Derin Öğrenme	InSDN (SDN ağlarında)	Binary ve Multi Class	96.32	-	-	-
Gueriani et al. (2024)	CNN-LSTM	Derin Öğrenme	CIC-IoT2023	Binary	98.42	-	-	98.57
Akuthota & Bhargava (2025)	Transformer	Geleneksel Makine Öğrenmesi	UNSW-NB15 / NSL-KDD	Binary ve Multi Class	>99.00	>99.00	>99.00	>99.00

Tablo 2.3.(Devamı) Literatürdeki çeşitli derin öğrenme modellerinin saldırı tespiti performansları detaylı karşılaştırılması.

Kaynak / Çalışma	Model Türü	Geliştirme Yöntemi	Veri Seti	Sınıflan- dırma Tipi	Accura cy (%)	Precisi on (%)	Recall (%)	F1- score (%)
Jony & Arnob (2024)	LST M	Derin Öğrenme	CIC-Io T2023	Binary (Tüm atak türleri vs. Benign)	98.75	98.66	98.75	98.59
Gueriani et al. (2024)	CNN- LST M	Derin Öğrenme	CIC- IoT202 3	Binary	98.42	98.85	98.43	98.57
Priyansh u Sinha et al. (2025)	LST M - CNN	Derin Öğrenme	BoT - IoT	Multi Class	99.87	99.89	99.85	99.87
Jony & Arnob (2024)	LST M	Makine Öğrenmesi	CIC-Io T2023	Binary - DDoS	LR: 0.8275 KNN: 0.9380 DT: 0.9919 RF: 0.9916	LR: 0.8473 KNN: 0.9366 DT: 0.9920 RF: 0.9913	LR: 0.8275 KNN: 0.9380 DT: 0.9919 RF: 0.9916	LR: 0.8034 KNN: 0.9364 DT: 0.9919 RF: 0.9909

Priyanshu Sinha et al. (2025) çalışmasında accuracy >%99 seviyelerine ulaşmış olsa da genel saldırılarda %90 doğruluk elde etmiştir.

Bu tezde, literatürdeki mevcut diğer çalışmalarla kıyaslandığında hem doğruluk oranı hem de dengeli metrik başarımları açısından güçlü bir alternatif sunan LSTM ve BiLSTM mimarileri önerilmiştir.

Elde edilen sonuçlar, literatürdeki diğer benzer mimari sonuçları ile kıyaslandığında rekabetçi bir seviyede yer almaktadır. Özellikle CIC-IoT2023 gibi karmaşık veri kümeleri üzerinde uygulanabilirliğini göstermektedir.

3. VERİ SETİ VE ÖN İŞLEME

Bu bölümde kullanılan veri seti ve uygulanan ön işleme metotları anlatılmaktadır.

3.1. CIC-IoT 2023 Veri Seti Nitelikleri

Bu çalışmada, University of New Brunswick tarafından yayımlanan CIC-IoT2023 veri seti kullanılmıştır. Veri seti, IoT ortamlarında gerçekleşen ağ trafiğini hem normal hem de çeşitli saldırı türlerini içerecek gerçekçi bir biçimde modellemektedir. Bu çalışmada kullanılan alt küme, `balanced_mixed_ddos_mirai.csv` adlı dosyadır ve veri kümesi toplam 27.103.989 adet örnek ve 40 sütun içermektedir.

3.1.1. Sayısal özelliklerin istatistiksel özeti

Veri setindeki sayısal değişkenler üzerinden yapılan istatistiksel özetleme sonucunda, bazı temel gözlemler yapılmıştır.

Tablo 3.1. Bazı sayısal sütunların istatistiksel özet değerleri.

Özellik	Ortalama	Std Sapma	Min	Maks
Header_Length	12.36	8.52	0	60
Protocol Type	18.01	15.46	6	47
Time_To_Live	66.64	43.83	0	255
syn_flag_number	0.15	0.35	0	1
ack_flag_number	0.39	0.17	0	1

Örneğin, Header_Length özelliği ortalama 12.37 değeri ile paket başlık boyutlarını temsil ederken, Protocol Type ortalaması 18.01 olarak gözlemlenmiştir. Rate değişkeni, yüksek varyansa ve bazı durumlarda sonsuz (inf) değerlere sahiptir. Bu sebeple dikkat çekmektedir. Ayrıca TCP/IP protokolü bayrakları (syn_flag_number, ack_flag_number vb.) çoğunlukla 0 değerindedir, bu da veri setinde saldırı olmayan

normal örneklerin çok sayıda bulunduğuna işaret etmektedir. Bazı sayısal sütunların istatistiksel özet değerleri Tablo 3.1’de gösterilmiştir.

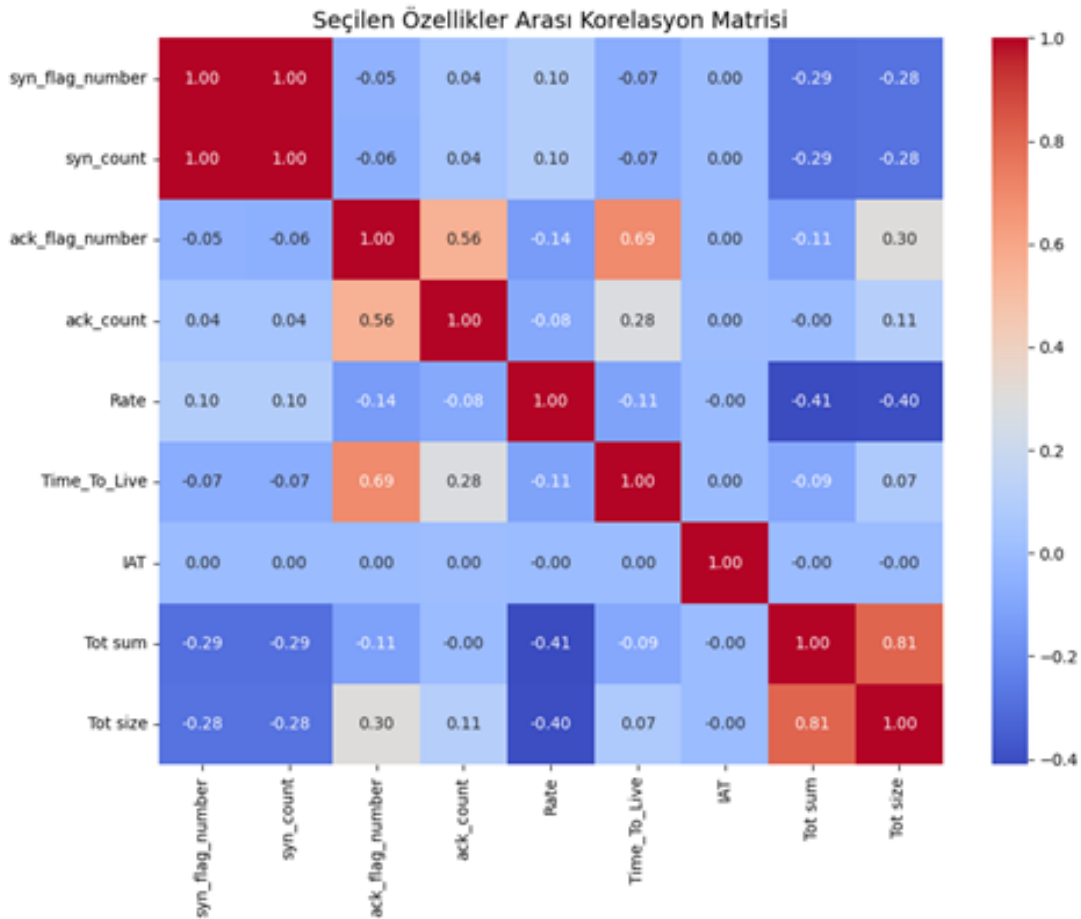
3.1.2. Özellikler arası korelasyon analizi

Seçilen sayısal özellikler arasında yapılan korelasyon analizi sayesinde, belirli değişken çiftleri arasında mânalı bağlantılar olduğu görülmüştür.

Örnek vermek gerekirse, Şekil 3.1’de görüldüğü üzere syn_flag_number ve syn_count özellikleri arasında mükemmel bir pozitif korelasyon ($r = 1.00$) gözlemlenmiştir.

Benzer şekilde, ack_flag_number ile ack_count arasında da ciddi seviyede pozitif ilişki bulunmaktadır ($r = 0.56$).

Öte yandan Rate değişkeni ile Tot size ve Tot sum arasında ise orta seviyede negatif korelasyonla karşılaşmıştır (yaklaşık $r = -0.40$).



Şekil 3.1. Seçilen özellikler arasında pearson korelasyon katsayısı ile oluşturulan ısı haritası.

3.2. Etiket Yapısı ve Sınıf Dengesi

Bu çalışmada kullanılan CIC-IoT 2023 veri seti, farklı DDoS ve Mirai tabanlı botnet saldırılarının yanı sıra normal (Benign) ağ trafiğini de içerecek şekilde yapılandırılmıştır. Kullanılan sürüm olan `balanced_mixed_ddos_mirai.csv`, veri sınıfları arasında denge sağlanarak oluşturulmuştur. Her bir örnek, ağ trafiğine ait belirli bir gözlemi temsil etmekte olup, label sütunu aracılığıyla ilgili saldırı türü ya da normal trafik olarak etiketlenmiştir. Bu yapı sayesinde modelin hem saldırı hem de normal trafiği ayırt edebilme kabiliyeti dengeli biçimde test edilebilmiştir.

3.2.1. Etiketlenen sınıflar ve örnek sayıları

Veri setindeki toplam 27.103.989 örnek, 8 farklı sınıfa dağılmıştır. Sınıfların örnek sayıları ve genel dağılımı Tablo 3.2’de sunulmuştur.

Tablo 3.2. CIC-IoT 2023 veri seti içerisindeki sınıflar ve örnek sayıları.

Sınıf Adı	Örnek Sayısı
DDoS-TCPs	7.456.505
DDoS-UDP	5.412.231
DDoS-SYN	4.059.179
Mirai-udpplains	1.781.148
Mirai-Greips	1.571.454
Benign (Normal Trafik)	1.098.191
Mirai-Greeths	1.012.751
DDoS-HTTP_Flood-.pcap	28.790

Bu tablo analiz edildiğinde, özellikle DDoS-TCPs, DDoS-UDP ve DDoS-SYN gibi saldırı türlerinin veri setinde büyük yer kapladığı görülmektedir. Benign sınıfının sayıca daha az olması, veri setinde saldırı davranışlarının daha yoğun temsil edildiğini göstermektedir. Bununla birlikte, her sınıf için milyonlar seviyesinde örnek bulunması, derin öğrenme modelleri için yeterli veri hacmini sağlamaktadır.

3.2.2. Etiket dengesizliği ve etkileri

Veri seti genel olarak dengelenmiş bir yapı sunsa da, Benign trafiğe ait örnek sayısının diğer bazı saldırı türlerine kıyasla daha az olması yönünden ilgi çekmektedir. Özellikle DDoS-TCPs sınıfı, toplam örneklerin yaklaşık %27'sini oluşturarak baskın duruma gelmiştir. Bundan dolayı, model eğitimi sırasında sınıf dengesizliği kaynaklı bias oluşma riskini doğurabilir. Bu nedenle veri işleme aşamasında sınıf dengesi göz önünde bulundurulmalı, gerekiyorsa SMOTE, class weighting veya resampling gibi yöntemler kullanılmalıdır.

3.2.3. Veri türü yapısı

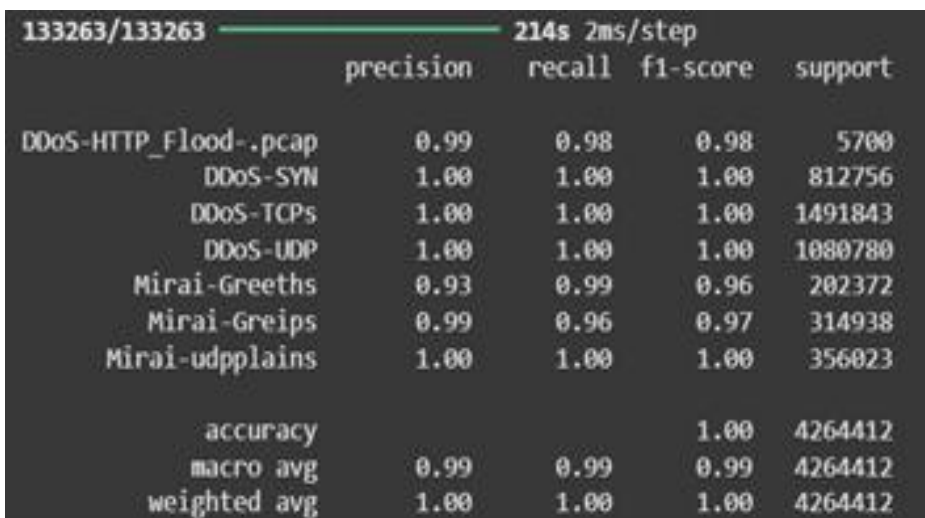
Veri setinde yer alan 40 sütunun veri türleri aşağıdaki gibidir:

- float64: 30 sütun (ör. Rate, Header_Length, IAT, AVG)
- int64: 9 sütun (ör. Protocol Type, syn_count, ack_count, Number)
- object: 1 sütun (sadece label)

Veri tipi dağılımı, sayısal sütunların ağırlıklı olduğunu göstermektedir. Bunun yanı sıra derin öğrenme modelleri açısından da uygun formatta olduğunu göstermektedir. Sınıflandırmada label sütunu hedef değişken olarak kullanılmaktadır ve modelleme sürecinde LabelEncoder gibi yöntemlerle sayısal hale getirilmektedir.

3.2.4. Başlangıçtaki örnekleme stratejisi

İlk modelleme aşamasında her sınıftan 1.000.000 örnek alınarak dengeli bir yapı oluşturulmuştur.



	precision	recall	f1-score	support
DDoS-HTTP_Flood-.pcap	0.99	0.98	0.98	5700
DDoS-SYN	1.00	1.00	1.00	812756
DDoS-TCPs	1.00	1.00	1.00	1491843
DDoS-UDP	1.00	1.00	1.00	1080780
Mirai-Greeths	0.93	0.99	0.96	202372
Mirai-Greips	0.99	0.96	0.97	314938
Mirai-udpplains	1.00	1.00	1.00	356023
accuracy			1.00	4264412
macro avg	0.99	0.99	0.99	4264412
weighted avg	1.00	1.00	1.00	4264412

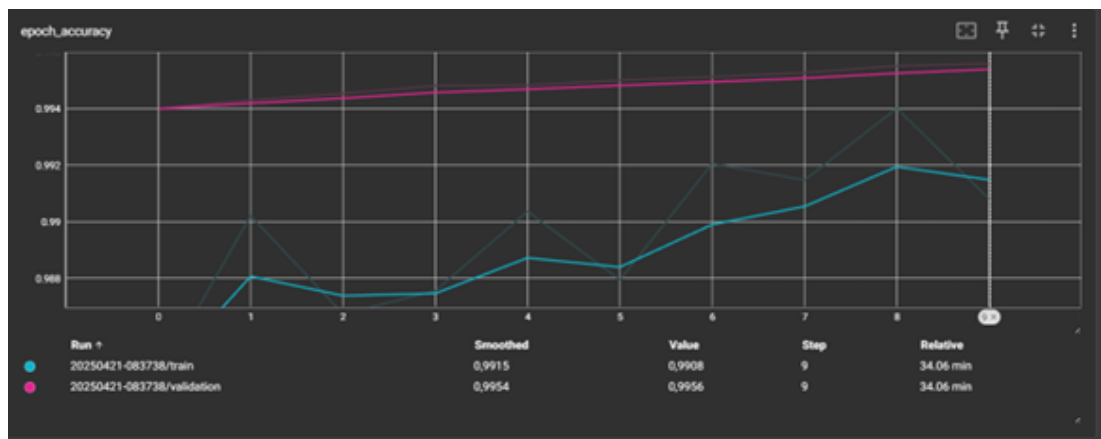
Şekil 3.2. Modelin saldırı tespit performansı.

Bu örnekleme stratejisi, modelin tüm sınıflardan eşit oranda öğrenmesini sağlamak amacıyla benimsenmiştir. Bu yapı ile eğitilen LSTM modelinde eğitim ve doğrulama doğrulukları benzer görünse de, bu örnekleme stratejisine bağlı saldırı tespit performansı Şekil 3.2’de görüldüğü üzere Mirai tabanlı saldırılar için f1-score ve doğruluk oranlarında düşüş gözlemlenmiştir.

Tablo 3.3. Başlangıçta kullanılan dengeli sınıf dağılımı (1milyon örnek/sınıf).

Sınıf Adı	Örnek Sayısı
Benign	1.000.000
DDoS-SYN	1.000.000
DDoS-TCPs	1.000.000
DDoS-UDP	1.000.000
Mirai-Greeths	1.000.000
Mirai-Greips	1.000.000
Mirai-udpplains	1.000.000

Başlangıç durumunda kullanılan sınıf dağılımı Tablo 3.3’teki gibidir.



Şekil 3.3. Eğitim süreci boyunca model performansı.

Ayrıca Şekil.3.3’teki TensorBoard görseline göre modelin ilk epoch’lardan itibaren validasyon kaybı sabit kalırken eğitim kaybı azalmaya devam etmiş, bu da açık biçimde overfitting yaşandığını göstermiştir.

3.2.5. Güncellenen örnekleme stratejisi

Modelin bazı saldırı türlerinde yetersiz genelleme yapması üzerine, özellikle Mirai tabanlı sınıfların örnek sayıları Tablo 3.4'teki görüldüğü gibi artırılmıştır. Böylece nadir görülen sınıfların temsil gücü artırılarak modelin bu sınıfları öğrenme kapasitesi geliştirilmiştir.

Tablo 3.4. Güncellenmiş örnekleme stratejisi (mirai sınıfları artırılmıştır).

Sınıf Adı	Örnek Sayısı
Mirai-udpplains	1.781.148
Mirai-Greips	1.571.454
Mirai-Greeths	1.012.751
Benign	1.098.191
DDoS-SYN	1.000.000
DDoS-TCPs	1.000.000
DDoS-UDP	1.000.000

Bu güncelleme sonrasında modelin özellikle Mirai-Greeths ve Mirai-Greips sınıflarındaki başarı oranlarında ciddi iyileşmeler gözlemlenmiştir. F1-score değerleri %96 ve üzeri seviyelere ulaşmış, ROC-AUC eğrileri tüm sınıflar için 1.00 olarak gözlemlenmiştir (Şekil Y ve Şekil X). Etiket dengesi parametresi, makine öğrenimi tabanlı saldırı tespit modellerinde sınıf bazlı başarıyı direkt etkiler. İlk örnekleme stratejisinde; bu çalışmada da görülmüş olduğu gibi, bazı sınıflar için yüksek doğruluk sağlanırken diğer sınıflarda yetersizlik gözlemlenmiştir. Bu nedenle veri dengesinin sadece sayısal eşitlik değil, saldırı türlerinin temsil gücü üzerinden de değerlendirilmesi gerektiği sonucuna ulaşılmıştır.

3.3. Eksik / Sonsuz Değer Temizliği

CIC-IoT2023 veri seti, büyük boyutlu ve karmaşık yapıları bir ağ trafiği veri kümesi olduğundan, veri kalitesinin artırılması amacıyla ön işleme adımlarında çeşitli temizlik işlemleri uygulanmıştır. Bu bağlamda, aşağıdaki veri sorunları tespit edilmiş ve giderilmiştir.

3.3.1. Eksik (NaN) değer temizliği

Veri kümesinde bazı sütunlarda NaN (eksik) değerlerin bulunduğu gözlemlenmiştir. Özellikle saldırı tipi etiketleri içeren label sütununda sınıflandırma işlemini etkileyebilecek eksik veriler belirlenmiştir. Belirlenen bu eksik kayıtlar veri setinden çıkarılmıştır. Ayrıca, sayısal sütunlardaki eksik değerler dropna() yöntemi ile temizlenmiştir.

3.3.2. Sonsuz (inf) değerlerin temizlenmesi

Özellikle Rate ve Variance gibi ağ trafiği hacmini veya istatistiksel değişkenliği temsil eden bazı sütunlarda, hesaplama hatalarına bağlı olarak sonsuz (inf) değerler oluştuğu belirlenmiştir. Bu değerler, np.isinf() yöntemiyle tespit edilip, ilgili satırlar veri setinden çıkartılmıştır. Bu işlem sayesinde modelin öğrenme sürecinde matematiksel sapmaların önüne geçilmiştir.

3.3.3. Zamanla ilgili sütunların çıkarılması

Veri setinde, zaman bağımlı veya zamanın mutlak değerine dayalı olan bazı sütunların (örneğin Time_To_Live, IAT, Rate, Std, Variance) model üzerinde ezberlemeye yol açtığı ve overfitting riskini artırdığı gözlemlenmiştir.

Yapılan ön analizler sonucunda bu sütunların model başarısına anlamlı katkı sağlamadığı ve aksine bazı sınıflarda yanıltıcı örüntüler oluşturduğu belirlenmiştir. Bu nedenle aşağıdaki sütunlar veri setinden çıkarılmıştır:

- Time_To_Live
- Rate
- Std
- Variance
- IAT

Bu sütunların çıkarılması, modelin sadece davranışsal ağ verilerine odaklanmasını ve genel öğrenme başarısını artırmasını sağlamıştır.

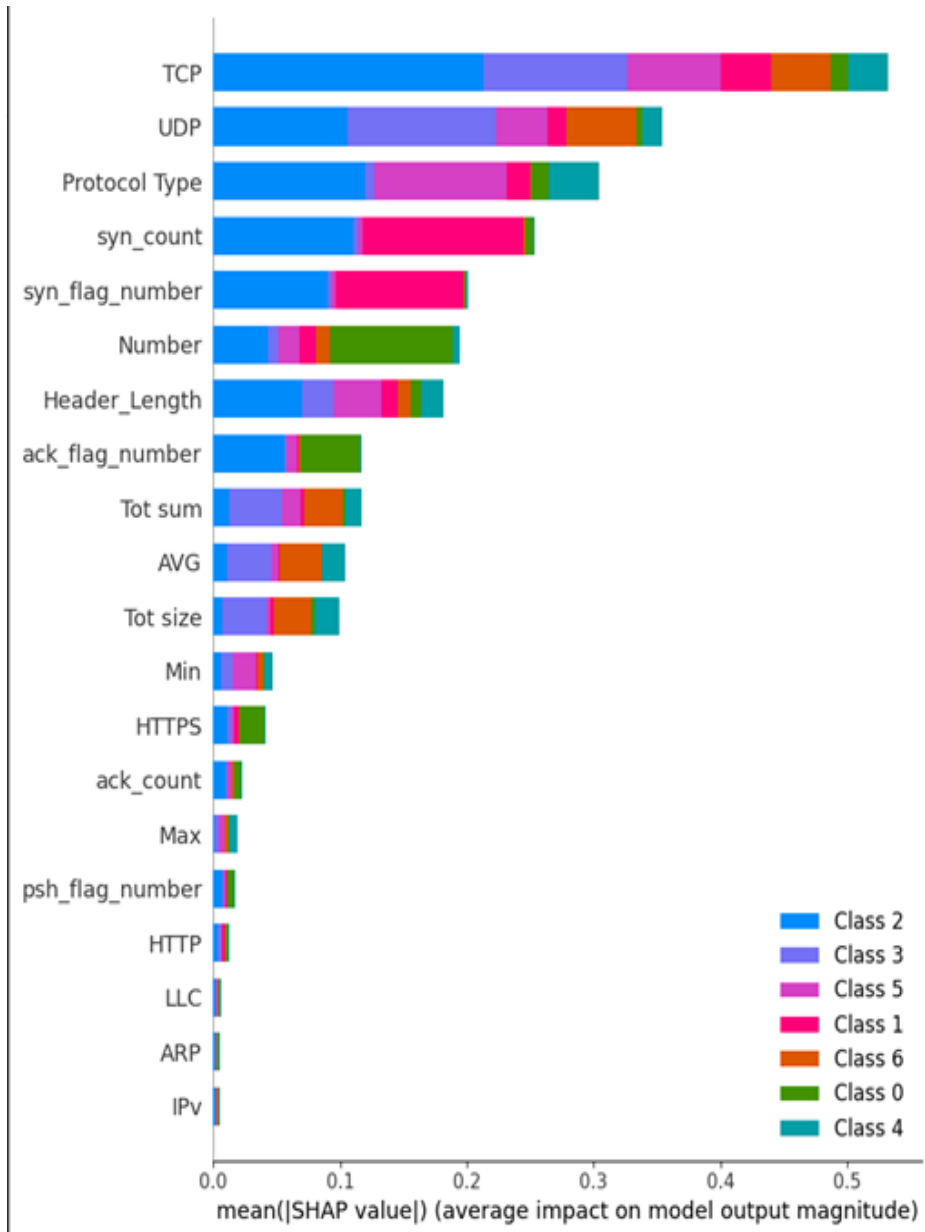
3.4. Özellik Seçimi Yöntemleri

Doğru özellik (feature) seçimi, saldırı tespit sistemlerinde model başarımını doğrudan etkileyen ana faktörlerden biridir. Özellik uzayının boyutunun yüksek olması, modelin

eğitim süresini artırmakta ve öğrenme sırasında gereksiz bilgilerle modelin performansını düşürebilmektedir. Bu nedenle çalışmada, farklı yöntemlerle özellik seçimi yapılmıştır. Elde edilen sonuçlar görsel ve istatistiksel açılarından değerlendirilmiştir.

3.4.1. Chi-Square (χ^2) testi ile özellik seçimi

İlk olarak Chi-Square (χ^2) testi kullanılarak, sınıf değişkeniyle en güçlü bağıntıya sahip olan 15 özellik belirlenmiştir. Bu yöntem, özellikle bağımsız değişkenler ile kategorik hedef değişken arasında bulunan istatistiksel bağımlılığı ölçmek için tercih edilmiştir.



Şekil 3.4. SHAP analizi ile model çıktılarına en çok etki eden değişkenlerin görselleştirilmesi.

Chi² ile seçilen sütunlar : ['Header_Length', 'Protocol Type', 'syn_flag_number', 'psh_flag_number', 'ack_flag_number', 'ack_count', 'syn_count', 'HTTP', 'HTTPS', 'DNS', 'TCP', 'UDP', 'Tot sum', 'Tot size', 'Number']

Seçilen bu özellikler arasından özellikle Protocol Type, TCP, UDP ve Tot sum gibi değişkenlerin model başarısına olan etkisi, SHAP (SHapley Additive exPlanations) analizleriyle de desteklenmiştir (Bkz. Şekil 3.4). Bu analizlerde TCP, UDP ve Protocol Type değişkenlerinin farklı sınıflar üzerinde yüksek açıklayıcılığa sahip olduğu görülmektedir.

3.4.2. ExtraTrees classifier ile özellik önemi

İkinci olarak, ExtraTreesClassifier algoritması üzerinden özellik önem sıralaması yapılmıştır. Bu yöntem, ensemble (topluluk) tabanlı ağaç yapılarını kullanarak her bir özelliğin hedef değişken üzerindeki etkisini öğrenmeye dayalıdır.

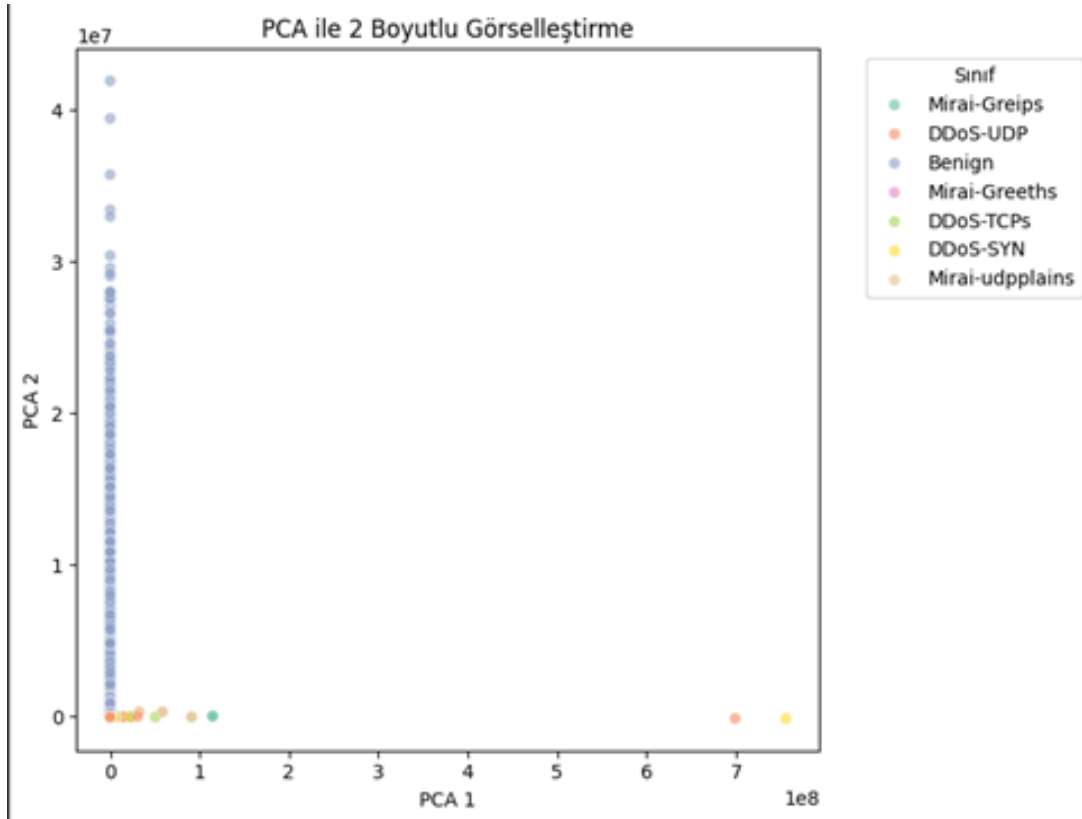
ExtraTrees ile elde edilen önemli değişkenler : ['UDP', 'Protocol Type', 'AVG', 'Tot sum', 'Tot size', 'TCP', 'syn_count', 'syn_flag_number', 'Number', 'Header_Length']

Her iki yöntemin çıktıları karşılaştırıldığında birçok ortak özelliğin öne çıktığı görülmektedir (örneğin: Protocol Type, UDP, TCP, Tot sum, Number, Header_Length). Bu da kullanılan yöntemlerin tutarlılığının ve seçilen özelliklerin model başarısında önemli rolü olduğunu göstermektedir.

Tüm incelemeler sonrasında, modelin eğitimi Chi² yöntemi ile seçilen 15 özellik üzerinden gerçekleştirilmiştir. Özellikle SHAP analiz sonuçları, Chi² ile seçilen sütunların sınıf tahmininde yüksek katkı sunduğunu doğrular niteliktedir. Özellik seçimi sonucunda eğitim süresi kısalmış, overfitting riski azalmış ve modelin açıklanabilirliği artmıştır.

3.5. PCA Yöntemi Kullanılarak Boyut İndirgeme

Veri setleri yüksek boyutlu olduğunda model eğitimi sırasında ortaya çıkan hesaplama yükünü azaltmak ve veri seti içerisindeki örüntüleri daha anlaşılabilir kılmak için PCA (Principal Component Analysis) tekniği uygulanmıştır. PCA, verisetindeki değişkenler arasındaki korelasyonu analiz ederek, en fazla varyansı açıklayan yeni bileşenler (principal components) üretir. Bu yöntem, veri kaybı yaşamadan boyut indirgemeyi mümkün kılarak hem görselleştirme hem de model verimliliği açısından önemli avantajlar sunar.



Şekil 3.5. PCA ile 2 boyutlu sınıf dağılımı görselleştirmesi.

Bu çalışmada, PCA yöntemi uygulanarak orijinal özellik kümesi iki boyuta indirgenmiş ve her bir örnek 2D düzlemde görselleştirilmiştir. Şekil 3.5'te bu görselleştirme yer almaktadır. Grafik, verinin sınıf etiketlerine göre ayrıştırılabilirliğini görsel olarak incelemeye olanak tanımaktadır.

Görsel analiz sonucunda bazı sınıflar arasında belirli bir ayrışma gözlenebilmiş olsa da, özellikle yoğunluk bakımından baskın olan Benign sınıfı diğerlerine göre daha dağınık bir yapı sergilemiştir. Bu durum, PCA'nın lineer bir yöntem olması nedeniyle, bazı karmaşık ve örtüşen örüntülerde sınıf ayrımının net olarak sağlanamayabileceğini ortaya koymaktadır.

PCA, model eğitimleri için doğrudan kullanılsa da, veri setinin genel yapısını anlamak ve sınıflar arası ayrışma potansiyelini incelemek adına faydalı bir analiz aracı olarak değerlendirilmiştir. Ayrıca, bu analiz öz niteliklerin daha dengeli ve anlamlı dağılımlar sağladığını ve daha sonraki aşamalarda uygulanabilecek t-SNE gibi daha karmaşık boyut indirgeme yöntemlerine temel oluşturabileceğini göstermektedir.

4. KULLANILAN YÖNTEM

4.1. Veri Setini Hazırlama

Bu çalışmada kullanılan veri seti, 105 farklı IoT cihazının ağ trafiğini içeren CIC (Canadian Institute for Cybersecurity) tarafından yayınlanan CIC-IoT2023 veri kümesidir. Veri seti, 33 farklı saldırı türünü içermekte olup, özellikle Mirai ve DDoS tehditleri üzerine odaklanılmıştır. Bu çalışmada kullanılan örnekleme stratejisiyle veri setinden, her biri farklı saldırı türüne ait olacak biçimde toplam 8.463.544 örnek alınmıştır. Dağılımın dengeli olması hedeflenmiştir.

4.1.1. Sınıf dağılımı ve yapay dengeleme

İlk etapta her bir sınıftan 1 milyon örnek alınarak başlangıçta dengeli bir yapı oluşturulmuş, ancak özellikle Mirai-Greeths, Mirai-Greips ve Mirai-udpplains sınıflarında model başarısının düşük kaldığı görülmüştür. Bunun üzerine bu üç sınıfın örnek sayısı artırılmış ve veri seti güncellenmiştir:

- Mirai-udpplains: 1.781.148
- Mirai-Greips: 1.571.454
- Mirai-Greeths: 1.012.751
- Diğer her sınıf (Benign, DDoS-SYN, DDoS-TCPs, DDoS-UDP): ~1.000.000

Bu strateji, özellikle azınlık sınıf başarımında iyileşme sağlamış ve Mirai varyantları üzerindeki F1 skorlarını %96 üzerine taşımıştır (Jony & Arnob, 2024).

4.1.2. Eksik ve anlamsız verilerin temizlenmesi

Veri setinde bazı sütunlarda NaN, inf gibi sayısal anormallikler tespit edilmiştir. Bu değerler SimpleImputer ve np.replace gibi işlemlerle temizlenmiş, böylece modelin eğitim sürecindeki sapmalar minimize edilmiştir. Modelde ezberlemeye sebep olduğundan özellikle Rate, Std, Variance, IAT ve Time_To_Live sütunları oluşturulan veri setinden çıkarılmıştır. Belirlenen sütunların çıkarılması, overfitting riskini önemli ölçüde azaltmıştır.

4.1.3. Özellik ölçekleme

Özelliklerin değer aralıklarını normalize etmek amacıyla RobustScaler ve MinMaxScaler birlikte kullanılmıştır. RobustScaler, uç değerlerin etkisini azaltmak için özellikle ExtraTreesClassifier gibi modellerle birlikte değerlendirilmiştir. MinMaxScaler ise χ^2 temelli özellik seçiminde kullanılmıştır.

4.1.4. Etiket dönüşümü ve sınıf kodlama

Sınıf etiketleri (label) LabelEncoder ile sayısal hale getirilmiştir. Bu adım, Keras ve TensorFlow gibi framework'lerdeki categorical_crossentropy kayıp fonksiyonunun düzgün çalışabilmesinde hassas önem taşımaktadır. Buna ek olarak model çok sınıflı (multi-class) olduğu için to_categorical() fonksiyonu ile one-hot encoding uygulanmıştır.

4.1.5. Özellik seçimi

Çalışmada yüksek boyutlu veriyle çalışıldığı için SelectKBest yöntemiyle χ^2 testi uygulanmış ve en iyi 15 özellik seçilmiştir:

- Header_Length
- Protocol Type
- syn_flag_number
- psh_flag_number
- ack_flag_number
- ack_count
- syn_count
- HTTP
- HTTPS
- DNS
- TCP
- UDP
- Tot sum
- Tot size
- Number

Bu özelliklerin büyük kısmı ağ protokolü düzeyindeki davranışları yansıttığı için saldırı tespiti açısından anlamlı kabul edilmiştir (Ogobuchi et al., 2024).

4.1.6. Zaman serisi formatı

LSTM tabanlı model mimarisi nedeniyle veriler timestep=10 olacak şekilde yeniden şekillendirilmiştir. Bu işlem sayesinde model, ardışık paket davranışlarını öğrenebilmekte ve saldırıya dönüşen davranışsal örüntüleri tanımlayabilmektedir (Sinha et al., 2025).

4.2. Model Mimarisinin Tanımı (LSTM)

Bu çalışmada önerilen model mimarisi, sıralı zaman serisi veriler üzerindeki zamansal bağımlılıkları yakalayabilen derin bir LSTM (Long Short-Term Memory) yapısı temel alınmıştır. LSTM katmanları, geleneksel RNN yapılarının yaşadığı uzun vadeli bağımlılık sorunlarını ortadan kaldırmak üzere, giriş kapısı (input gate), unutma kapısı (forget gate) ve çıkış kapısı (output gate) gibi elementlerle donatılmıştır. Bu yapılar, modelin geçmişten gelen bilgileri gerektiği sürece bellekte tutmasına ve gereksiz bilgileri unutmasına olanak tanır (Natha et al., 2025). Önerilen modelde, ilk katman olarak zaman serilerini kabul eden bir LSTM katmanı yer almakta ve ardından BatchNormalization ile Dropout uygulanmaktadır. Dropout, aşırı öğrenmeyi (overfitting) engellemek amacıyla %20-50 oranlarında katmanlar arası uygulanmıştır (Elsayed et al., 2021). Özellikle, L2 regularizasyon $\lambda=0.1$ ile birlikte uygulanarak modelin genelleme başarısı artırılmıştır. Modelin genel mimarisi aşağıdaki adımlardan oluşmaktadır:

- Giriş Katmanı: Zaman serisi veriler timestep=10 ile 3D forma çevrilerek modele beslenir.
- LSTM Katmanı: 128 nöronlu LSTM hücreleri kullanılır. Sıralı verinin öğrenilmesinde kilit rol oynar.
- Dropout ve BN Katmanları: Dropout($p=0.25-0.5$) ve BatchNormalization uygulanır. Bu adımlar, modelin genelleme yeteneğini artırır.
- Fully Connected (Dense) Katmanı: Sonuçlar softmax fonksiyonuyla çok sınıflı olarak sınıflandırılır.

Bu yapının başarısı, literatürde de benzer şekilde kanıtlanmıştır. Örneğin, Elsayed ve arkadaşlarının (2021) yapmış olduğu bir çalışmada, yalnızca LSTM modeliyle %93.85 doğruluk oranına ulaşılmıştır. Yine bu çalışmada CNN ile birleştirilmiş LSTM modeli ise %96.32 doğruluğa ulaşmıştırA_Hybrid_CNN_LSTM_Based.... Bu, LSTM mimarisinin uzamsal (spatial) veriyle birlikte kullanıldığında dahi yüksek başarı

sağlayabildiğini göstermektedir. Zhao ve arkadaşlarının (2023) CIC-DDoS2019 veri seti üzerinde gerçekleştirdiği çalışmada, LSTM ve BiLSTM mimarileri karşılaştırılmış ve BiLSTM mimarisinin doğruluk, hassasiyet ve F1 skorlarında daha iyi sonuç verdiği raporlanmıştır. Bu da, LSTM mimarisinin geliştirilmiş versiyonlarının ileri düzey saldırı tespiti sistemleri açısından çok uygun bir mimari olduğuna işaret etmektedir. Modelin eğitiminde zaman serilerinin doğasına uygun şekilde sequence to one formatında giriş-çıkış yapısı oluşturulmuştur. Bu yapı, özellikle Mirai gibi IoT tabanlı saldırıların davranışsal örüntülerini tespit etmede avantaj sağlamaktadır.

4.3. Model Mimarisinin Tanımı (BiLSTM)

Bu bölümde önerilen BiLSTM model mimarisinden bahsedilmektedir.

4.3.1. DDoS, mirai ve benign multilabel classification için model mimarisi

Model, giriş verisinin hem geçmiş hem de gelecek bağlamını öğrenmesine olanak tanıyan iki katmanlı Bi-LSTM yapısı içermektedir. Modelin giriş katmanı, veri setindeki zaman adımı ve özellik sayısına karşılık gelen boyutta bir tensör kabul etmektedir. İlk katmanda, 64 birimden oluşan bir Bi-LSTM katmanı yer almaktadır. Bu katman, zaman serisinin her iki yönlü bağlamsal bilgisini öğrenmek amacıyla çift yönlü yapıdadır. Ardından, aşırı öğrenmeyi önlemek için %50 oranında bağlantıların rastgele devre dışı bırakıldığı bir Dropout katmanı ve öğrenme sürecini daha kararlı hale getirmek amacıyla Batch Normalization katmanı uygulanmıştır. İkinci aşamada, 32 birimlik ikinci bir Bi-LSTM katmanı kullanılmakta ve bu katman, artık tüm zaman adımlarının değil yalnızca son adımın çıktıları üretmektedir. Bu katmandan sonra yine sırasıyla Dropout (0.5) ve Batch Normalization katmanları yer almaktadır. Devamında, selu aktivasyon fonksiyonuna sahip 64 birimlik tam bağlantılı (Dense) bir katman bulunmaktadır. SELU fonksiyonu, özellikle derin yapılarda self-normalizing özellik gösterdiği için tercih edilmiştir. Bu katmandan sonra %30 oranında Dropout uygulanarak modelin genellenebilirliği artırılmıştır. Son olarak, çıktı katmanı olarak sınıf sayısı kadar nörona sahip ve softmax aktivasyonu kullanan bir Dense katman bulunmaktadır. Modelin derleme aşamasında, optimizasyon algoritması olarak öğrenme oranı 0.001 olan Adam kullanılmış; kayıp fonksiyonu olarak çok sınıflı sınıflandırmalarda yaygın olarak tercih edilen categorical_crossentropy, değerlendirme metriği olarak ise accuracy kullanılmıştır. Eğitim sürecinde modelin performansını artırmak ve eğitim süresince overfitting'i önlemek amacıyla üç farklı

callback mekanizmasından faydalanılmıştır. Bunlar: doğrulama kaybı iyileşmediğinde eğitimi durduran EarlyStopping, doğrulama performansı durakladığında öğrenme oranını azaltan ReduceLROnPlateau ve en iyi doğrulama doğruluğuna ulaşıldığında model ağırlıklarını kaydeden ModelCheckpoint callback'leridir. Model, 100 epoch boyunca, 128'lik mini-batch boyutlarıyla eğitilmiş ve eğitim sırasında X_train, y_train ile doğrulama seti olarak X_val, y_val kullanılmıştır.

4.3.2. Benign ve mirai binary classification için model mimarisi

Bu çalışmada kullanılan Bi-LSTM tabanlı ikinci model, ilk mimariden farklı olarak 5 katlı çapraz doğrulama (K-Fold Cross Validation), sınıf ağırlığı dengeleme (class weighting) ve ölçekleme işlemleri gibi çeşitli ileri seviye eğitim stratejileri ile yapılandırılmıştır. Model, her fold için eğitim ve doğrulama verilerini yeniden oluşturmakta, MinMaxScaler ile tüm girdileri [0,1] aralığına dönüştürmekte ve zaman serisi formatına uygun olarak üç boyutlu bir yapıya getirmektedir. Eğitim sırasında, sınıf dengesizliği problemini azaltmak amacıyla compute_class_weight fonksiyonu kullanılarak her sınıfa ağırlıklar atanmakta ve bu ağırlıklar eğitim sürecine dahil edilmektedir. Model mimarisi, ilk versiyonla aynı şekilde yapılandırılmıştır. Giriş katmanını takiben, 64 birimli çift yönlü bir LSTM katmanı yer almakta, bu katmanı %50 oranında Dropout ve Batch Normalization katmanları izlemektedir. Ardından, 32 birimli ikinci bir Bi-LSTM katmanı uygulanmakta ve bu yapı yine Dropout ile Batch Normalization katmanlarıyla desteklenmektedir. Devamında, selu aktivasyon fonksiyonuna sahip 64 birimli tam bağlantılı bir katman bulunmaktadır ve model, %30 Dropout sonrası iki sınıflı çıktı üreten softmax aktivasyon fonksiyonlu bir Dense katmanı ile sonlanmaktadır. Derleme aşamasında Adam optimizasyon algoritması ve categorical_crossentropy kayıp fonksiyonu tercih edilmiş; eğitim sürecinde ise erken durdurma (EarlyStopping) ve öğrenme oranı azaltma (ReduceLROnPlateau) stratejileri uygulanmıştır. Modelin genel başarımı, her fold sonunda hesaplanan doğruluk değerlerinin ortalamasıyla değerlendirilmiş ve en iyi sonuç veren fold'daki model nihai model olarak belirlenmiştir.

4.3.3. DDoS ve benign binary classification için model mimarisi

Bu çalışmada, (DDoS, Mirai ve Benign) , Ddos ve Benign, Mirai ve Benign gibi trafiklerinin sınıflandırılması amacıyla üç farklı Bi-LSTM tabanlı model mimarisi önerilmiştir. Her üç modelin temelinde, zaman serisi verisinin çift yönlü bağlamsal özelliklerini yakalayabilen iki katmanlı bir Bi-LSTM yapısı bulunmaktadır. Giriş

verisi, zaman adımı ve özellik sayısına uygun boyutlarda olacak şekilde modele verilmiş; ilk Bi-LSTM katmanı 64 birimden oluşacak şekilde yapılandırılmıştır. Bu katman, her zaman adımının çift yönlü çıktısını üretmekte, ardından %50 oranında dropout ve batch normalization katmanları ile aşırı öğrenme ve eğitim dengesizliği kontrol altına alınmaktadır. İkinci Bi-LSTM katmanı ise 32 birimlidir ve yalnızca son zaman adımının çıktısını üretmektedir. Bu katmandan sonra da benzer şekilde dropout ve batch normalization katmanları uygulanmıştır. Derin yapılarda self-normalizing etkisiyle öne çıkan 64 birimli selu aktivasyonlu bir Dense katman, ardından %30 oranında dropout ile desteklenmiş; model, sınıf sayısı kadar nörona sahip softmax aktivasyonlu çıktı katmanıyla tamamlanmıştır. Çok sınıflı sınıflandırma görevinde kullanılan modelde, DDoS, Mirai ve Benign trafiği ayırt etmek amacıyla 3 çıktı nöronu ile çalışılmış ve categorical_crossentropy kayıp fonksiyonu tercih edilmiştir. Model, 100 epoch boyunca 128'lik mini-batch boyutlarında eğitilmiş, erken durdurma (EarlyStopping), öğrenme oranı azaltma (ReduceLROnPlateau) ve en iyi ağırlıkları kaydeden (ModelCheckpoint) mekanizmalarıyla eğitim süreci optimize edilmiştir. Eğitim ve doğrulama aşamalarında sabit veri kümeleri (X_train, y_train; X_val, y_val) kullanılmıştır.

Mirai ve Benign sınıflarını içeren ikili sınıflandırma modelinde ise aynı mimari korunmuş ancak eğitim sürecinde çeşitli stratejiler uygulanmıştır. Bu modelde 5 katlı çapraz doğrulama yöntemiyle eğitim seti her fold'da yeniden oluşturulmuş, veriler MinMaxScaler ile ölçeklendirilerek 0-1 aralığına getirilmiş ve zaman serisi formatına uygun olarak üç boyutlu hale dönüştürülmüştür. Sınıf dengesizliğini azaltmak için compute_class_weight fonksiyonu ile elde edilen sınıf ağırlıkları, modelin eğitimi sırasında dikkate alınmıştır. Eğitimde kullanılan callback mekanizmaları arasında EarlyStopping ve ReduceLROnPlateau yer almış, her fold sonunda hesaplanan doğruluk ortalamaları üzerinden model performansı değerlendirilmiş ve en başarılı fold'daki model nihai model olarak belirlenmiştir.

Benign ve DDoS sınıflarını ayırt etmeyi amaçlayan diğer ikili sınıflandırma modelinde ise eğitim ve doğrulama setleri sabit tutulmuş; çapraz doğrulama, sınıf ağırlığı dengeleme veya veri ölçekleme gibi ek işlemler uygulanmamıştır. ModelCheckpoint kullanılarak yalnızca en yüksek doğrulama doğruluğuna ulaşan model ağırlıkları "best_bilstm_model.h5" dosyasına kaydedilmiş, böylece eğitim sürecindeki dalgalanmalardan bağımsız olarak en iyi sonuç veren modelin korunması sağlanmıştır.

Bu modelde de diğerlerinde olduğu gibi Adam optimizasyon algoritması ve categorical_crossentropy kaybı kullanılmış, performans metriği olarak accuracy izlenmiştir. Tüm modellerde, zamansal verinin karmaşıklığını yakalayabilecek Bi-LSTM mimarisi temel alınmış; dropout, batchnormalization ve uygun aktivasyon fonksiyonlarıyla öğrenme süreci dengelenmiş ve genellenebilir sonuçlar elde etmek üzere çeşitli eğitim stratejileri uygulanmıştır.

4.4. Model Mimarisinin Tanımı (CNN-LSTM + Attention)

Bu modelde konvolüsyonel sinir ağı (CNN), uzun-kısa süreli bellek (LSTM) ve dikkat mekanizması (attention) bileşenlerini bir araya getiren hibrid bir yapıda tasarlanmıştır. Girdi verisi, önceden seçilmiş özniteliklerle birlikte zaman adımı bir olacak şekilde üç boyutlu hale getirilmiş ve modelin giriş katmanına aktarılmıştır. İlk katmanda, uzamsal örüntüleri yakalamak amacıyla 64 filtreli ve çekirdek boyutu 3 olan bir adet Conv1D katmanı kullanılmış, ardından öğrenme sürecini dengede tutmak için BatchNormalization uygulanmıştır. Ardışık bağımlılıkları modellemek üzere 64 birimli ve return_sequences=True parametresine sahip bir LSTM katmanı eklenmiş ve %30 oranında Dropout uygulanarak aşırı öğrenmenin önüne geçilmiştir. Bu katmandan sonra, girişin her zaman adımı için bir önem derecesi hesaplayan özel tanımlı bir attention mekanizması uygulanmıştır. Bu model yapısı 3 farklı hedeflenen için neredeyse aynı yapı kullanılmıştır.

4.5. Eğitim Parametreleri

Modelin eğitimi sırasında kullanılan hiperparametreler, hem literatürdeki öneriler hem de modelin aşırı öğrenmesini (overfitting) engellemek amacıyla yapılan çeşitli denemelerle belirlenmiştir. Belirlenen parametreler, genelleme kapasitesini, modelin öğrenme hızını ve doğrulama performansını doğrudan etkileyen kritik bileşenlerdir. Modelde, Adam (Adaptive Moment Estimation) optimizasyon algoritması kullanılmıştır. Adam hem momentuma dayalı güncellemeleri hem de adaptif öğrenme oranını birleştirir, bu sayede ağ düzgün ve daha çabuk öğrenir.

Bu çalışmada learning_rate=3e-5 olarak ayarlanmıştır. Bu değer, literatürde önerilen 1e-3 gibi daha yüksek başlangıç oranlarının aksine, daha düşük ayarlanarak eğitim süresince küçük ama istikrarlı güncellemelerle modelin öğrenmesi hedeflenmiştir (Elsayed et al., 2021). Eğitim süreci 100 epoch olarak planlanmış olsa da, aşırı

öğrenmenin önüne geçmek için early stopping mekanizması devreye alınmıştır. Bu yapı sayesinde model, doğrulama kaybında iyileşme görülmediği takdirde eğitimi erken sonlandırmaktadır.

Kullanılan yapılandırmaya göre, patience=5 olarak belirlenmiş ve en iyi ağırlıklar geri yüklenerek en verimli model korunmuştur. Bu teknik, özellikle IoT verilerinde sıkça karşılaşılan ezberleme eğilimini dengelemek için literatürde önerilmektedir (Zhao et al., 2023). Modelin öğrenme hızını adaptif olarak azaltmak üzere ReduceLROnPlateau kullanılmıştır. Bu yöntem, model doğrulama başarımı belirli bir süre boyunca gelişmezse, öğrenme oranını belirli bir faktörle azaltarak daha küçük adımlarla ilerlemeye yönlendirir.

Bu çalışmada factor=0.3 ve patience=5 değerleriyle yapılandırılmıştır. Bu sayede model, öğrenmede durağanlık yaşadığı noktada yeniden dengeye girebilmektedir (Zrira et al., 2024). Eğitim işlemi sırasında batch_size=128 olarak belirlenmiştir. Bu değer, eğitim vakti ile GPU bellek kullanımı arasında bir denge kurularak seçilmiştir. Daha küçük batch boyutları, modelin genelleme kapasitesini artırabilirken, daha büyük boyutlar eğitim süresini kısaltsa da modelin lokal minimumlarda takılmasına neden olabilmektedir. Bu bağlamda, 128 gibi orta büyüklükte bir değer ile eğitim süresi ve doğruluk açısından dengeli sonuç elde edilmiştir (Gueriani et al., 2024). Bu hiperparametrelerin birlikte uyumlu çalışması, modelin eğitim sürecini verimli hâle getirmiş ve doğrulama seti üzerindeki başarı oranlarını optimize etmiştir. Parametrelerin belirlenmesinde hem önceki çalışmalardan alınan ilham hem de kendi veri kümesine özgü deneysel denemeler dikkate alınmıştır.

4.6. Eğitimde Kullanılan Teknikler

Eğitim aşamasında, modeldeki sınıflar arası dengesizlikle başa çıkmak ve aşırı öğrenmeyi (overfitting) engellemek için çeşitli düzenleme ve iyileştirme teknikleri kullanılmıştır. Bu teknikler, doğrulama ve eğitim süreçlerinde modelin daha kararlı sonuçlar vermesine katkı sağlamıştır.

4.6.1. Dropout katmanları

Model mimarisinde Dropout tekniği üç farklı noktada aktif olarak uygulanmıştır. İlk LSTM katmanından sonra %50 oranında, ikinci LSTM katmanından sonra yine %50 oranında ve Dense katmandan sonra ise %30 oranında dropout kullanılmıştır. Dropout,

nöronlardan bazısının eğitim süreçlerinde rastgele pasifleştirilmesini sağlar ve böylelikle modelin fazla örüntü ezberlemesinin önüne geçer. Literatürde CNN ve LSTM gibi derin yapılarda dropout oranlarının %20 ile %50 arasında tutulmasının aşırı öğrenmeyi azalttığı ve doğrulama başarımını artırdığı belirtilmektedir (Elsayed et al., 2021).

4.6.2. Batch normalization

Her LSTM katmanından sonra Batch Normalization uygulanarak modelin her katmandaki aktivasyon dağılımı normalize edilmiştir. Bu teknik, gradyanların daha istikrarlı şekilde yayılmasını sağlayarak eğitimin daha hızlı ve kararlı olmasına yardımcı olur. Ayrıca, BatchNorm'un dropout ile birlikte kullanılması, modelin genelleme kapasitesini artıran etkili bir kombinasyon olarak kabul edilmektedir (Saba et al., 2022).

4.6.3. Class weight

Kodlarda doğrudan belirtilmemiş olsa da, veri setindeki bazı sınıfların (örneğin Mirai-Greeths, Greips gibi) diğerlerine göre daha az örneği bulunduğu göz önünde tutulduğunda, class_weight parametresi kullanılarak sınıf dengesizliğiyle başa çıkılması önerilir. Bu tezde ayrıca bazı sınıfların örnek sayılarının artırılması yoluyla bu dengesizlik giderilmiş ve böylece modele daha dengeli bir öğrenme zemini sunulmuştur (Jony & Arnob, 2024). Class weight tekniği, her sınıfa eğitim süresince ağırlık atayarak azınlık sınıfların öğrenilmesini teşvik eder.

4.6.4. SMOTE (synthetic minority oversampling technique)

Kodlarda doğrudan bir SMOTE uygulaması yer almamakla birlikte, modelin daha önceki sürümlerinde azınlık sınıflarda yetersiz çıktılar elde edilmiştir. Bundan dolayı Mirai sınıflarının örnek sayılarının artırıldığı belirtilmiştir. Bu örneklem, SMOTE benzeri bir mantıkla, veri artırımı (oversampling) yaklaşımını yansıtmaktadır. SMOTE, özellikle az rastlanan saldırı türlerinin sınıflandırılmasında performansı artırmak için sıklıkla önerilen bir yöntemdir (Gueriani et al., 2024).

4.6.5. Erken durdurma ve öğrenme oranı azaltma

Modelin eğitiminde EarlyStopping ve ReduceLROnPlateau gibi callback fonksiyonları kullanılmıştır. Erken durdurma, doğrulama kaybında belirli bir süre iyileşme olmazsa eğitimi sonlandırır (patience=5). Öğrenme oranı azaltma mekanizması ise eğitim süreci durağanlaştığında öğrenme hızını düşürerek modelin

daha ince ayarlar yapabilmesini sağlar (factor=0.3). Bu iki teknik birlikte çalıştığında, modelin daha iyi genelleme yapması sağlanır (Zrira et al., 2024).



5. DEĞERLENDİRME VE SONUÇLAR

5.1. Accuracy, Precision, Recall, F1

Bu çalışmada kullanılan veri seti, 105 farklı IoT cihazının ağ trafiğini içeren CIC (Canadian Institute for Cybersecurity) tarafından yayınlanan CIC-IoT2023 veri kümesidir. Veri seti, 33 farklı saldırı türünü içermekte olup, bu çalışmada Mirai ve DDoS tehditleri üzerine odaklanılmıştır. Örnekleme stratejisiyle veri setinden, her biri farklı saldırı türüne ait olacak biçimde toplam 8.463.544 örnek alınmıştır. Oluşturulan veri seti örnek sayıları ;

- Mirai-udpplains: 1.781.148
- Mirai-Greips: 1.571.454
- Mirai-Greeths: 1.012.751
- Diğer her sınıf (Benign, DDoS-SYN, DDoS-TCPs, DDoS-UDP): ~1.000.000

Tablo 5.1. Farklı derin öğrenme modellerinin performans ölçütleri.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1
LSTM	0.98	0.98	0.98	0.98
BiLSTM	0.98	0.98	0.98	0.98
CNN - BiLSTM + Attention	0.88	0.88	0.86	0.82

Çalışmanın model performansını değerlendirmek adına öncelikle LSTM, BiLSTM ve CNN-BiLSTM+Attention modelleri arasında temel sınıflandırma metrikleri karşılaştırılmıştır. Tablo 5.1’de her bir modelin accuracy, macro average ve weighted average bazlı precision, recall ve f1-score sonuçlarını özetlemektedir. LSTM ve BiLSTM modelleri %98’lik başarımla değerleri ile birbirine oldukça yakın sonuçlar verirken, CNN-BiLSTM + Transformer mimarisi, özellikle Mirai-Greeths ve Mirai-Greips sınıflarında daha düşük performans sergilemiştir.

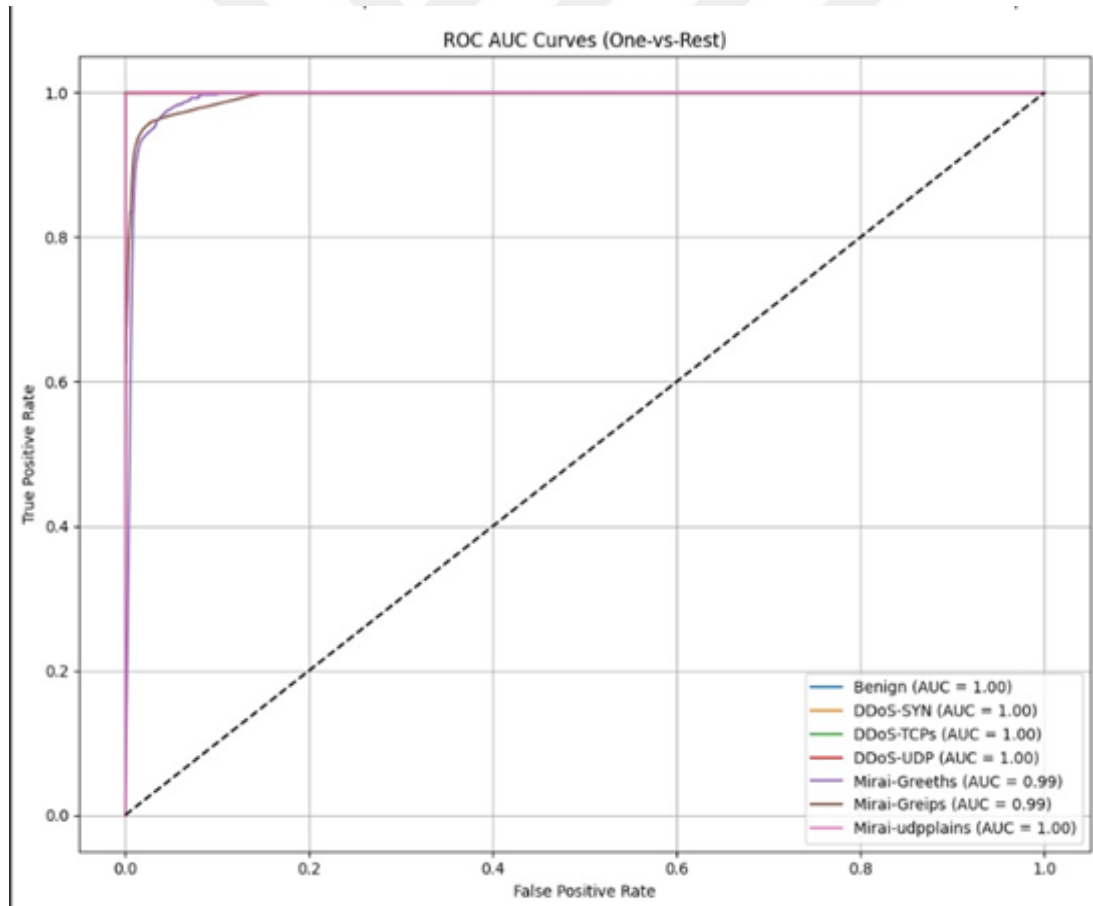
5.2. PR Ve ROC Eğrileri

Model performansının sınıf bazlı detaylı analizini yapabilmek adına, ROC (Receiver Operating Characteristic) ve PR (Precision-Recall) eğrileri değerlendirilmiştir. ROC eğrileri, modeldeki yanlış pozitif oranı ile doğru pozitif oranı arasındaki bağlantıyı ortaya koyarken, PR eğrileri özellikle dengesiz veri yapılarında pozitif örneklerin ne kadar isabetli sınıflandırıldığını gösterme açısından önem arz etmektedir.

5.2.1. LSTM modeli ROC eğrileri

Aşağıda Şekil 5.1’de gösterilen LSTM modeline ait ROC eğrisi, modelin genel anlamda oldukça başarılı bir ayırt etme kapasitesine sahip olduğunu ortaya koymaktadır.

Benign, DDoS-SYN, DDoS-TCPs, DDoS-UDP ve Mirai-udpplains gibi sınıflar için AUC = 1.00 seviyesinde değerler elde edilmiştir.



Şekil 5.1. LSTM modeli için ROC AUC eğrileri (tüm sınıflar için $AUC \geq 0.99$).

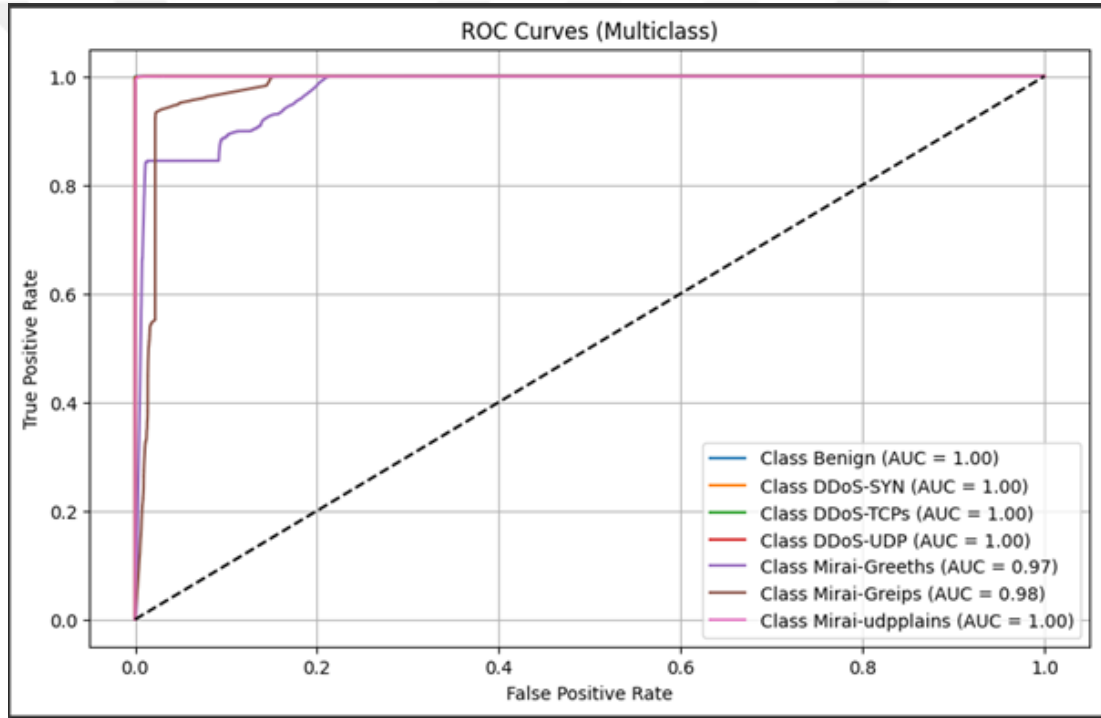
Mirai-Greeths ve Mirai-Greips gibi örüntüleri benzeşen alt sınıflarda ise AUC = 0.99 değerleri gözlenmiştir. Bu durum, modelin sadece temel saldırı türlerinde değil, aynı

zamanda Mirai ailesinin alt türlerini de yüksek doğrulukla ayırt edebildiğini göstermektedir.

5.2.2. BiLSTM modeli ROC eğrileri

BiLSTM modeli, çift yönlü öğrenme kapasitesi sayesinde özellikle zaman bağımlı örüntülerde daha derin bağlamları kavrayabilmektedir.

Şekil 5.2. incelendiğinde modelin, Benign ve DDoS türlerinde AUC = 1.00 düzeyinde bir başarıya ulaşmış olduğu görülmektedir. Ancak Mirai-Greeths sınıfı için AUC = 0.97, Greips için ise AUC = 0.98 değerleri gözlemlenmiştir. Bu da LSTM'e kıyasla BiLSTM'in Mirai varyantlarını daha ayırık şekilde modelleyebildiğini ortaya koymaktadır.

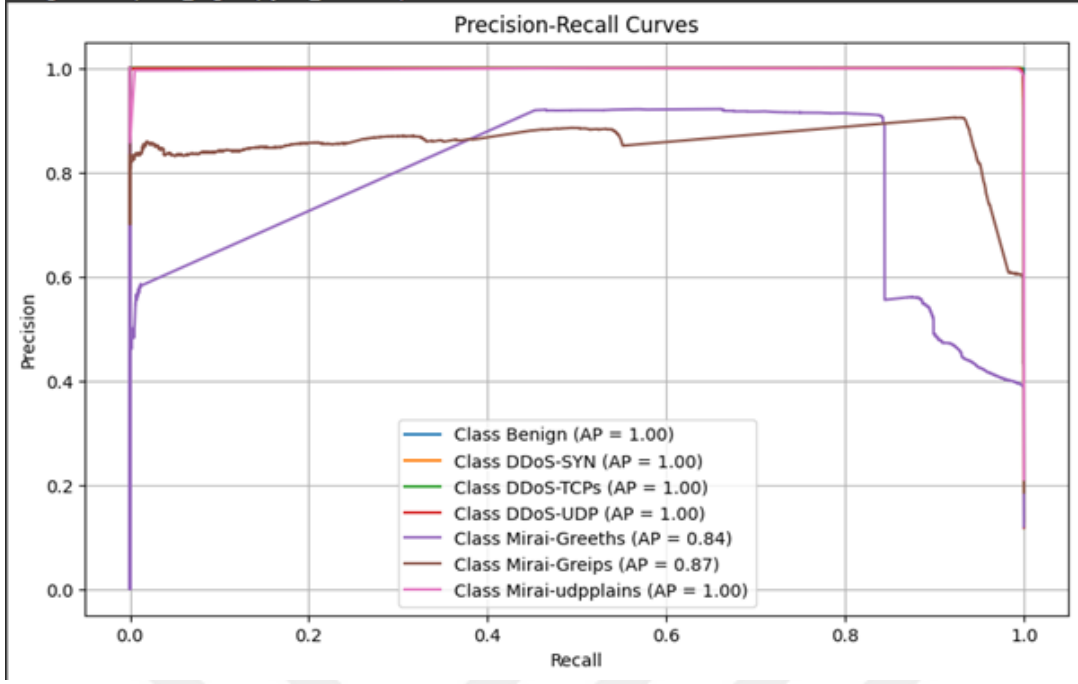


Şekil 5.2. BiLSTM modeli için ROC AUC eğrileri (özellikle mirai alt türlerinde sınırlı kayıplar gözlenmiştir).

5.2.3. CNN-LSTM + attention modeli PR eğrileri

Şekil 5.3'te sunulan Precision-Recall eğrileri, özellikle dengesiz veri yapılarında sınıf bazlı doğruluğu ölçmek açısından kritiktir.

Bu modelde Benign ve DDoS sınıflarında AP (Average Precision) değeri 1.00 olarak ölçülmüş olsa da, Mirai-Greeths sınıfı için AP = 0.84 ve Mirai-Greips için AP = 0.87 değeri elde edilmiştir. Bu düşüş, modelin karmaşık yapısına rağmen ayırım gücünün, özellikle benzer yapıdaki sınıflar arasında sınırlı kaldığını göstermektedir.



Şekil 5.3. CNN-LSTM + attention modeline ait precision-recall eğrileri (mirai alt türlerinde daha düşük AP değerleri).

ROC ve PR eğrileri birlikte değerlendirildiğinde; LSTM ve BiLSTM modelleri genel olarak yüksek sınıf ayrıştırma kabiliyeti sergilemiş, özellikle BiLSTM modeli zaman serisi verinin yönlü bağımlılıklarını daha verimli biçimde öğrenebilmiştir. Ancak, CNN-BiLSTM + Attention modelinde görülen AP düşüşleri, bu mimarinin derinliğine rağmen belirli alt sınıflarda overfitting ya da örüntü çakışması yaşadığını göstermektedir. Bu sonuç, özellikle Mirai ailesindeki alt varyantların tespiti gibi detaylı sınıflandırma gerektiren durumlarda model seçme konusunda oldukça dikkat edilmesi gerektiğini göstermektedir (Zhao et al., 2023).

5.3. Confusion Matrix Analizi

Model başarımını sadece genel doğruluk skorlarıyla değil, aynı zamanda sınıf bazlı hata analiziyle değerlendirmek adına confusion matrix (karışıklık matrisi) görselleştirmeleri detaylı biçimde incelenmiştir.

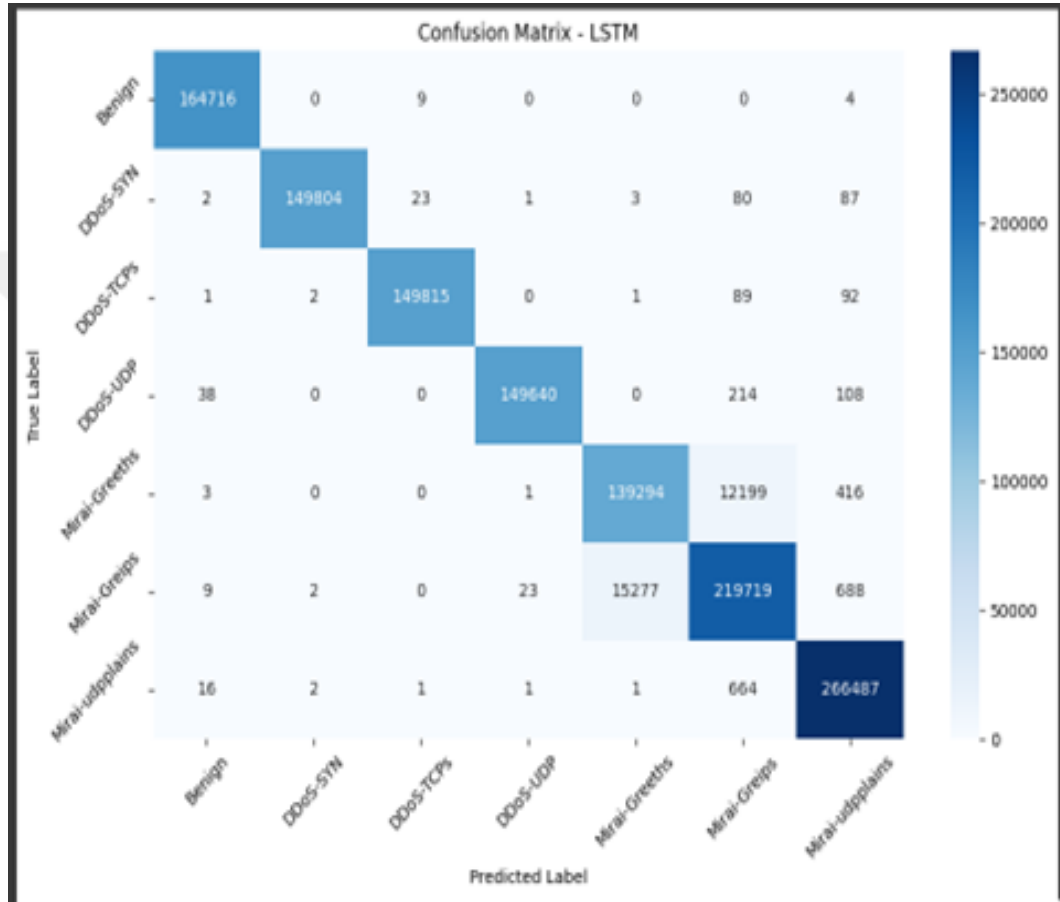
Bu analiz, hangi saldırı türlerinin doğru tahmin edildiğini ve hangilerinde modellerin hata yaptığını daha belirgin bir biçimde göstermektedir.

5.3.1. LSTM modeli confusion matrix değerlendirmesi

Şekil 5.4'te görülen LSTM modeline ait karışıklık matrisi, genel anlamda oldukça başarılı bir sınıflandırma performansı ortaya koymaktadır. DDoS-SYN, TCP ve UDP

gibi yaygın saldırı türlerinde modelin hata oranı son derece düşüktür. Fakat Mirai-Greeths sınıfında 12.199, Mirai-Greips sınıfında ise 15.277 örnek başka sınıflar ile karışmıştır.

Bu durum, LSTM modelinin sınıflar arası sınırları genel hatlarıyla öğrenebilmesine rağmen, içerik olarak benzeşen alt saldırı türlerinde ayırma zorlandığını göstermektedir.

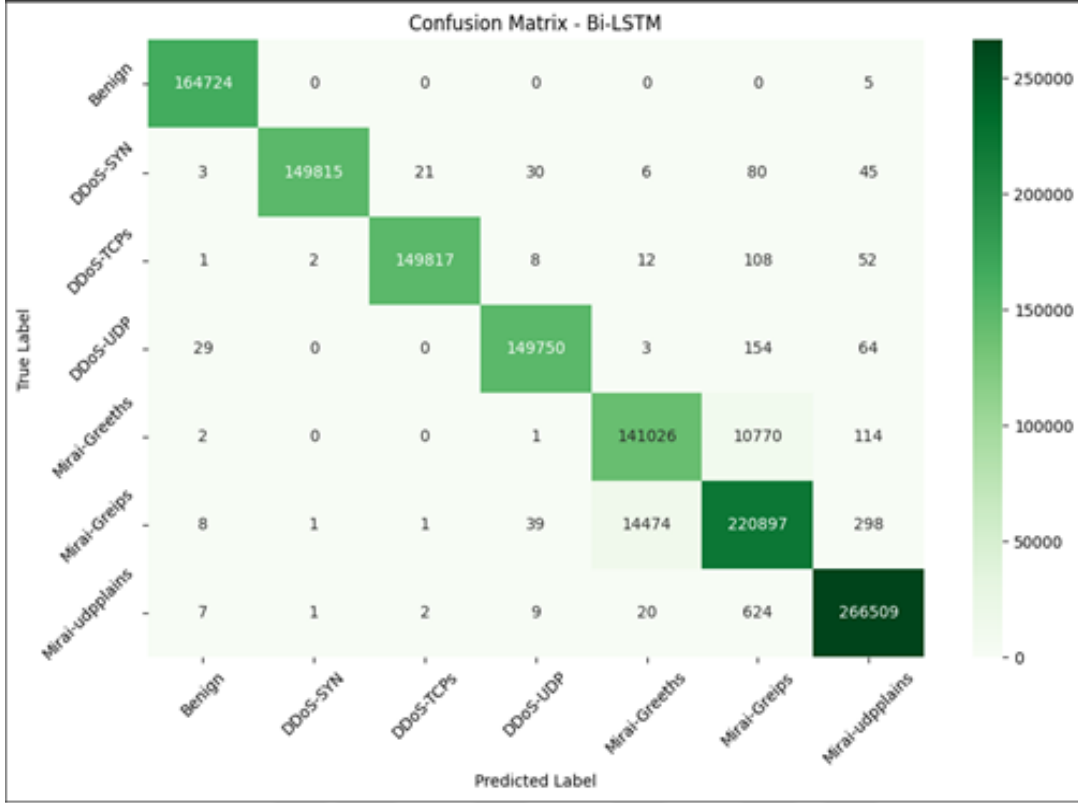


Şekil 5.4. LSTM modeline ait confusion matrix (mirai alt türlerinde belirgin sapmalar gözlenmiştir).

5.3.2. BiLSTM modeli confusion matrix değerlendirmesi

LSTM mimarisine ek olarak çift yönlü öğrenme kapasitesi sunan BiLSTM modeli, özellikle zaman bağımlı desenlerin çözümünde daha derin ilişkiler kurabilmektedir.

Şekil 5.5'teki sonuçlara göre, BiLSTM modeli Mirai-Greeths için 10.770, Greips için ise 14.474 yanlış sınıflandırma yapmıştır. Bu değerler, LSTM'e kıyasla biraz daha düşük hata oranına işaret etmektedir. Bu durum, BiLSTM'nin zaman serisi içindeki çift yönlü ilişkileri daha verimli öğrenerek karışıklığı bir miktar azaltabildiğini göstermektedir.



Şekil 5.5. BiLSTM modeline ait confusion matrix (mirai sınıflarında hata oranı LSTM'ye göre biraz daha düşüktür).

5.3.3. CNN-LSTM + attention modeli confusion matrix değerlendirmesi

Şekil 5.6'da yer alan CNN-LSTM + Attention modelinin karışıklık matrisi, yapısal olarak daha derin ve karmaşık olmasına karşın, özellikle Mirai-Greeths ve Mirai-Greeps gibi alt türlerde belirgin hatalar yaptığını göstermektedir. Greeths sınıfında 2.036, Greeps sınıfında ise 1.926 adet örnek farklı sınıflara atanmıştır. Bu rakamlar yüzdelik olarak düşük gözükse de, veri setinin büyüklüğü göz önüne alındığında bu sapmalar modelin genel doğruluğunu ciddi şekilde etkileyebilmektedir. Karmaşık mimarilere sahip modellerin aşırı uyum (overfitting) veya genel örüntü karmaşası yaşama ihtimali bu sonuçlarda hissedilmektedir. Üç modelin confusion matrix sonuçları birlikte değerlendirildiğinde, Benign, DDoS-SYN, DDoS-TCPs ve DDoS-UDP gibi temel sınıflarda modellerin yüksek doğrulukta tahmin yaptığı anlaşılmaktadır. Ancak Mirai ailesine ait alt sınıflar, benzer davranış desenleri sebebiyle modeller için daha zorlayıcı olmuştur. LSTM ve BiLSTM modelleri, bu sınıflarda birbirine yakın ama sınırlı hata oranları üretirken, Attention tabanlı modelin karmaşıklığı sınıflar arası net ayrım yapmayı zorlaştırmış, bu da daha yüksek hata sayılarıyla sonuçlanmıştır. Bu durum, model mimarisi seçiminde sınıf yapısının

ayrıştırılabilirliğinin dikkate alınması gereken bir faktör olduğunu ortaya koymaktadır (Gueriani et al., 2024).



Şekil 5.6. CNN-LSTM + attention modeline ait confusion matrix (özellikle mirai sınıflarında ayırım gücünde düşüş gözlenmiştir).

5.4. Deney Sonuçları

Farklı model mimarilerinin sınıflandırma başarımını daha bütüncül değerlendirebilmek adına, LSTM, BiLSTM ve CNN-LSTM + Attention yapılarına ait sonuçlar karşılaştırmalı olarak analiz edilmiştir. Değerlendirmede yalnızca genel doğruluk oranları değil, F1-score, recall ve precision gibi performans ölçütleriyle beraber sınıf bazlı hatalar da dikkate alınmıştır.

5.4.1. LSTM modeli genel değerlendirme

LSTM modeli, oldukça dengeli bir sınıflandırma performansı ortaya koymuştur. Şekil 5.7’de yer alan sınıflandırma raporuna göre; Benign, DDoS-SYN, TCP ve UDP sınıflarında F1-score, recall ve precision değerleri 1.00 seviyesinde gerçekleşmiştir. Ancak özellikle Mirai-Greeths ve Mirai-Greips gibi içerik olarak benzer örüntülere sahip saldırı türlerinde bu metriklerde kısmi düşüşler gözlenmiştir. Greeths için F1-

score 0.94, Greips için 0.94 seviyesindedir. Bu durum, LSTM'in genel olarak istikrarlı bir yapı sunsa da Mirai ailesi alt sınıflarında hata yapma eğiliminin sürdüğünü ortaya koymaktadır.

39673/39673 ————— 64s 2ms/step

Classification Report:

	precision	recall	f1-score	support
Benign	1.00	1.00	1.00	164729
DDoS-SYN	1.00	1.00	1.00	150000
DDoS-TCPs	1.00	1.00	1.00	150000
DDoS-UDP	1.00	1.00	1.00	150000
Mirai-Greeths	0.90	0.92	0.91	151913
Mirai-Greips	0.94	0.93	0.94	235718
Mirai-udpplains	0.99	1.00	1.00	267172
accuracy			0.98	1269532
macro avg	0.98	0.98	0.98	1269532
weighted avg	0.98	0.98	0.98	1269532

Şekil 5.7. LSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek, ancak Mirai alt türlerinde küçük sapmalar mevcuttur).

5.4.2. BiLSTM modeli genel değerlendirme

BiLSTM modeli, zaman serisi verinin ileri ve geri yönlü ilişkilerini birlikte öğrenerek, LSTM'e göre daha zengin bir örüntü çıkarımı gerçekleştirmiştir.

39673/39673 ————— 89s 2ms/step

	precision	recall	f1-score	support
Benign	1.00	1.00	1.00	164729
DDoS-SYN	1.00	1.00	1.00	150000
DDoS-TCPs	1.00	1.00	1.00	150000
DDoS-UDP	1.00	1.00	1.00	150000
Mirai-Greeths	0.91	0.93	0.92	151913
Mirai-Greips	0.95	0.94	0.94	235718
Mirai-udpplains	1.00	1.00	1.00	267172
accuracy			0.98	1269532
macro avg	0.98	0.98	0.98	1269532
weighted avg	0.98	0.98	0.98	1269532

Şekil 5.8. BiLSTM modeline ait sınıflandırma raporu (özellikle mirai sınıflarında LSTM'den daha düşük hata oranları gözlenmiştir).

Bu durum, Şekil 5.8. incelendiğinde özellikle Mirai-Greeths (F1 = 0.92) ve Mirai-Greips (F1 = 0.94) sınıflarındaki daha düşük hata oranlarına yansıdığı görülmektedir.

Modelin genel doğruluk oranı yine 0.98 düzeyinde kalmakla birlikte, sınıf bazlı metriklerde özellikle zorlayıcı kategorilerde daha iyi sonuçlar üretmiştir. Bu sonuçlar, BiLSTM mimarisinin LSTM'e göre daha verimli çalıştığını, ancak genel doğruluk anlamında ciddi farklar yaratmadığını göstermektedir.

5.4.3. CNN-LSTM + attention modeli genel değerlendirme

Teoride en yüksek yapısal karmaşıklığa sahip CNN-LSTM + Attention modeli, pratikte sınıf ayırımında bazı sınırlamalar sergilemiştir. Benign ve DDoS sınıflarında F1-score değerleri 1.00 seviyesinde sabit kalırken, Mirai-Greeths için F1-score = 0.01, Greips için ise 0.03 gibi oldukça düşük değerlere gerilemiştir. Bu, modelin özellikle Mirai ailesindeki alt sınıfları birbirinden ayırmakta büyük zorluk yaşadığını göstermektedir. Bu modelin hesaplanan genel doğruluk oranı 0.88 dir. Bu oran, diğer iki modele kıyasla anlamlı düzeyde daha düşüktür.

79346/79346 ————— 119s 1ms/step

Classification Report:

	precision	recall	f1-score	support
Benign	1.00	1.00	1.00	329458
DDoS-SYN	1.00	1.00	1.00	300000
DDoS-TCPs	1.00	1.00	1.00	300000
DDoS-UDP	0.99	1.00	1.00	300000
Mirai-Greeths	0.56	0.01	0.01	303825
Mirai-Greips	0.61	0.99	0.75	471436
Mirai-udpplains	1.00	1.00	1.00	534345
accuracy			0.88	2539064
macro avg	0.88	0.86	0.82	2539064
weighted avg	0.87	0.88	0.83	2539064

Şekil 5.9. CNN-LSTM + attention modeline ait sınıflandırma raporu.

Mirai sınıflarındaki F1-score değerlerinin belirgin biçimde düşmesi, modelin ayrıştırma kapasitesinin sınırlı kaldığını göstermektedir. Sonuçlar bir arada değerlendirildiğinde, LSTM ve BiLSTM modelleri dengeli mimarileri sayesinde gerek sınıf bazlı doğruluk gerekse de genel başarı olarak tatmin edici performanslar sergilemiştir. BiLSTM, özellikle zaman bağımlı karmaşık örüntüleri daha iyi yakalayarak Mirai alt sınıflarında daha düşük hata oranlarına ulaşmıştır. Buna karşın, Attention tabanlı model teorik avantajına rağmen pratikte sınıflar arası ayırmada

yetersiz kalmış; Şekil 5.9’da da görüldüğü üzere özellikle Greeths ve Greips gibi ince farklara sahip kategorilerde başarıımı belirgin şekilde düşmüştür.

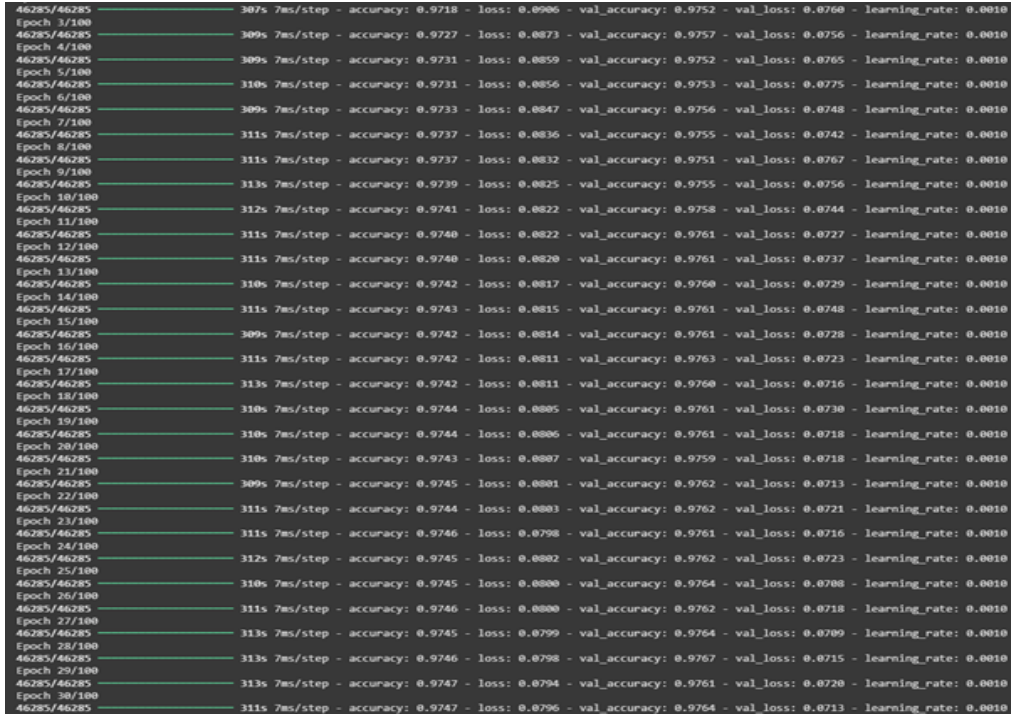
Elde edilen bu sonuçlara göre, her zaman yapısal daha karmaşık mimarinin daha yüksek performans sağlamayabileceğini; aksine, model seçiminde verinin doğasına, sınıf dengesine ve örüntü yapısına dikkat edilmesi gerektiğini net bir şekilde göstermektedir (Zhao et al., 2023).

5.5. Model Gecikme Süreleri (Latency)

Derin öğrenme tabanlı modellerin doğruluk skorları yanı sıra hesaplama maliyeti ve eğitim süresi gibi pratik metriklerle de değerlendirilmesi gerekmektedir. Özellikle geniş boyutlu veri kümeleri ve çok sınıflı problemler söz konusu olduğunda, modelin eğitimi sırasında geçen süre (epoch başına eğitim süresi) modelin başarımını değerlendirmede önemli bir ölçüttür. Bu bölümde, farklı derin öğrenme mimarileri olan LSTM, BiLSTM ve CNN-LSTM + Attention modellerine ait latency verileri analiz edilmiştir.

5.5.1. LSTM modeli gecikme süresi

LSTM modelinin eğitimi sırasında epoch başına ortalama eğitim süresi, Şekil 5.10’da görüldüğü üzere, 311 saniyedir.



46285/46285	307s 7ms/step - accuracy: 0.9718 - loss: 0.0906 - val_accuracy: 0.9752 - val_loss: 0.0760 - learning_rate: 0.0010
Epoch 1/100	
46285/46285	309s 7ms/step - accuracy: 0.9727 - loss: 0.0873 - val_accuracy: 0.9757 - val_loss: 0.0756 - learning_rate: 0.0010
Epoch 4/100	
46285/46285	309s 7ms/step - accuracy: 0.9731 - loss: 0.0859 - val_accuracy: 0.9752 - val_loss: 0.0765 - learning_rate: 0.0010
Epoch 5/100	
46285/46285	310s 7ms/step - accuracy: 0.9731 - loss: 0.0856 - val_accuracy: 0.9753 - val_loss: 0.0775 - learning_rate: 0.0010
Epoch 6/100	
46285/46285	309s 7ms/step - accuracy: 0.9733 - loss: 0.0847 - val_accuracy: 0.9756 - val_loss: 0.0748 - learning_rate: 0.0010
Epoch 7/100	
46285/46285	311s 7ms/step - accuracy: 0.9737 - loss: 0.0836 - val_accuracy: 0.9755 - val_loss: 0.0742 - learning_rate: 0.0010
Epoch 8/100	
46285/46285	311s 7ms/step - accuracy: 0.9737 - loss: 0.0832 - val_accuracy: 0.9751 - val_loss: 0.0767 - learning_rate: 0.0010
Epoch 9/100	
46285/46285	313s 7ms/step - accuracy: 0.9739 - loss: 0.0825 - val_accuracy: 0.9755 - val_loss: 0.0756 - learning_rate: 0.0010
Epoch 10/100	
46285/46285	312s 7ms/step - accuracy: 0.9741 - loss: 0.0822 - val_accuracy: 0.9758 - val_loss: 0.0744 - learning_rate: 0.0010
Epoch 11/100	
46285/46285	311s 7ms/step - accuracy: 0.9740 - loss: 0.0822 - val_accuracy: 0.9761 - val_loss: 0.0727 - learning_rate: 0.0010
Epoch 12/100	
46285/46285	311s 7ms/step - accuracy: 0.9740 - loss: 0.0820 - val_accuracy: 0.9761 - val_loss: 0.0737 - learning_rate: 0.0010
Epoch 13/100	
46285/46285	310s 7ms/step - accuracy: 0.9742 - loss: 0.0817 - val_accuracy: 0.9760 - val_loss: 0.0729 - learning_rate: 0.0010
Epoch 14/100	
46285/46285	311s 7ms/step - accuracy: 0.9743 - loss: 0.0815 - val_accuracy: 0.9761 - val_loss: 0.0748 - learning_rate: 0.0010
Epoch 15/100	
46285/46285	309s 7ms/step - accuracy: 0.9742 - loss: 0.0814 - val_accuracy: 0.9761 - val_loss: 0.0728 - learning_rate: 0.0010
Epoch 16/100	
46285/46285	311s 7ms/step - accuracy: 0.9742 - loss: 0.0811 - val_accuracy: 0.9763 - val_loss: 0.0723 - learning_rate: 0.0010
Epoch 17/100	
46285/46285	313s 7ms/step - accuracy: 0.9742 - loss: 0.0811 - val_accuracy: 0.9760 - val_loss: 0.0716 - learning_rate: 0.0010
Epoch 18/100	
46285/46285	310s 7ms/step - accuracy: 0.9744 - loss: 0.0805 - val_accuracy: 0.9761 - val_loss: 0.0730 - learning_rate: 0.0010
Epoch 19/100	
46285/46285	310s 7ms/step - accuracy: 0.9744 - loss: 0.0806 - val_accuracy: 0.9761 - val_loss: 0.0718 - learning_rate: 0.0010
Epoch 20/100	
46285/46285	310s 7ms/step - accuracy: 0.9743 - loss: 0.0807 - val_accuracy: 0.9759 - val_loss: 0.0718 - learning_rate: 0.0010
Epoch 21/100	
46285/46285	309s 7ms/step - accuracy: 0.9745 - loss: 0.0801 - val_accuracy: 0.9762 - val_loss: 0.0713 - learning_rate: 0.0010
Epoch 22/100	
46285/46285	311s 7ms/step - accuracy: 0.9744 - loss: 0.0803 - val_accuracy: 0.9762 - val_loss: 0.0721 - learning_rate: 0.0010
Epoch 23/100	
46285/46285	311s 7ms/step - accuracy: 0.9746 - loss: 0.0798 - val_accuracy: 0.9761 - val_loss: 0.0716 - learning_rate: 0.0010
Epoch 24/100	
46285/46285	312s 7ms/step - accuracy: 0.9745 - loss: 0.0802 - val_accuracy: 0.9762 - val_loss: 0.0723 - learning_rate: 0.0010
Epoch 25/100	
46285/46285	310s 7ms/step - accuracy: 0.9745 - loss: 0.0800 - val_accuracy: 0.9764 - val_loss: 0.0708 - learning_rate: 0.0010
Epoch 26/100	
46285/46285	311s 7ms/step - accuracy: 0.9746 - loss: 0.0800 - val_accuracy: 0.9762 - val_loss: 0.0718 - learning_rate: 0.0010
Epoch 27/100	
46285/46285	313s 7ms/step - accuracy: 0.9745 - loss: 0.0799 - val_accuracy: 0.9764 - val_loss: 0.0709 - learning_rate: 0.0010
Epoch 28/100	
46285/46285	313s 7ms/step - accuracy: 0.9746 - loss: 0.0798 - val_accuracy: 0.9767 - val_loss: 0.0715 - learning_rate: 0.0010
Epoch 29/100	
46285/46285	313s 7ms/step - accuracy: 0.9747 - loss: 0.0794 - val_accuracy: 0.9761 - val_loss: 0.0720 - learning_rate: 0.0010
Epoch 30/100	
46285/46285	311s 7ms/step - accuracy: 0.9747 - loss: 0.0796 - val_accuracy: 0.9764 - val_loss: 0.0713 - learning_rate: 0.0010

Şekil 5.10. LSTM modelinin epoch başına eğitim süresi (ortalama: ~311 sn/epoch).

Modelin katman yapısı görece sade olmasına rağmen, sıralı veri işleme yapısı nedeniyle bu sürede belirgin bir uzama görülmüştür. Özellikle `return_sequences=True` parametresiyle ardışık veriler üzerinde yapılan işlemler, hesaplama yükünü artırarak latency süresini yukarı çekmiştir.

5.5.2. BiLSTM modeli gecikme süresi

BiLSTM mimarisi, yapısal olarak ileri ve geri yönlü öğrenme barındırmasına rağmen, yapısal verimliliği sayesinde daha kısa sürede eğitilebilmiştir. Şekil 5.11'de sunulan çıktı ekranına göre, model epoch başına yaklaşık 89 saniyede tamamlanmıştır. Bu durum, BiLSTM'nin hem daha hızlı converge olabilmesi hem de eğitim sürecinde donanımı daha etkili kullanabilmesiyle açıklanabilir. Modelin daha az dropout uygulaması ve sade fully connected yapısı, bu farkı oluşturan temel etkenlerdendir.

```

46205/46205 - 8s 89s/step - accuracy: 0.9774 - loss: 0.000260802e-04! You are saving your model as an H5 file via 'model.save()' or 'keras.save_model(model)'. This file format
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0044 - learning_rate: 9.0000e-05
Epoch 53/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9706 - val_loss: 0.0046 - learning_rate: 9.0000e-05
Epoch 54/100
46205/46205 - 88% 89s/step - accuracy: 0.9772 - loss: 0.0000 - val_accuracy: 0.9706 - val_loss: 0.0046 - learning_rate: 9.0000e-05
Epoch 55/100
46205/46205 - 88% 89s/step - accuracy: 0.9772 - loss: 0.0000 - val_accuracy: 0.9706 - val_loss: 0.0045 - learning_rate: 9.0000e-05
Epoch 56/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9706 - val_loss: 0.0045 - learning_rate: 9.0000e-05
Epoch 57/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9706 - val_loss: 0.0046 - learning_rate: 9.0000e-05
Epoch 58/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9706 - val_loss: 0.0044 - learning_rate: 2.7000e-05
Epoch 59/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0044 - learning_rate: 2.7000e-05
Epoch 60/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0044 - learning_rate: 2.7000e-05
Epoch 61/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 2.7000e-05
Epoch 62/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0044 - learning_rate: 2.7000e-05
Epoch 63/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 8.1000e-06
Epoch 64/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.000260802e-04! You are saving your model as an H5 file via 'model.save()' or 'keras.save_model(model)'. This file format
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0042 - learning_rate: 8.1000e-06
Epoch 65/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.000260802e-04! You are saving your model as an H5 file via 'model.save()' or 'keras.save_model(model)'. This file format
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0002 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 8.1000e-06
Epoch 66/100
46205/46205 - 88% 89s/step - accuracy: 0.9775 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 8.1000e-06
Epoch 67/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 8.1000e-06
Epoch 68/100
46205/46205 - 88% 89s/step - accuracy: 0.9773 - loss: 0.0002 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 8.1000e-06
Epoch 69/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0001 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 2.4300e-06
Epoch 70/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.000260802e-04! You are saving your model as an H5 file via 'model.save()' or 'keras.save_model(model)'. This file format
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 2.4300e-06
Epoch 71/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 2.4300e-06
Epoch 72/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 2.4300e-06
Epoch 73/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 2.4300e-06
Epoch 74/100
46205/46205 - 88% 89s/step - accuracy: 0.9774 - loss: 0.0000 - val_accuracy: 0.9707 - val_loss: 0.0043 - learning_rate: 7.2900e-07

```

Şekil 5.11. BiLSTM modelinin epoch başına eğitim süresi (ortalama: ~89 sn/epoch).

5.5.3. CNN-LSTM + attention modeli gecikme süresi

CNN ve Attention bileşenleri içeren modeller, mimarideki parametre yoğunluğu ve dikkat (attention) mekanizmasının işlem maliyeti nedeniyle daha yüksek gecikme süresine sahiptir. Şekil 5.12'deki CNN-LSTM + Attention modeline ait ekran görüntülerinde net bir epoch süresi ölçülememiştir. Bununla birlikte, modelin 5 epoch'luk eğitim süreci göz önünde bulundurulduğunda, diğer modellere göre ortalama işlem süresinin daha uzun olduğu net bir şekilde görülmektedir. Dikkat mekanizmasının çoklu sorgulama (multi-head attention) yapısı ve CNN katmanlarının filtre işlemleri bu gecikmeyi artırmaktadır.

```

Epoch 1/50
46285/46285 — 303s 6ms/step - accuracy: 0.8456 - loss: 0.3058 - val_accuracy: 0.8777 - val_loss: 0.1904 - learning_rate: 0.0010
Epoch 2/50
46285/46285 — 299s 6ms/step - accuracy: 0.9075 - loss: 0.1900 - val_accuracy: 0.6693 - val_loss: 4.0525 - learning_rate: 0.0010
Epoch 3/50
46285/46285 — 298s 6ms/step - accuracy: 0.9114 - loss: 0.1832 - val_accuracy: 0.6684 - val_loss: 7.1608 - learning_rate: 0.0010
Epoch 4/50
46285/46285 — 298s 6ms/step - accuracy: 0.8891 - loss: 0.2054 - val_accuracy: 0.7032 - val_loss: 0.4391 - learning_rate: 0.0010
Epoch 5/50
46285/46285 — 299s 6ms/step - accuracy: 0.8932 - loss: 0.2005 - val_accuracy: 0.8770 - val_loss: 0.2276 - learning_rate: 5.0000e-04
Epoch 6/50
46285/46285 — 299s 6ms/step - accuracy: 0.8996 - loss: 0.1947 - val_accuracy: 0.8764 - val_loss: 0.2100 - learning_rate: 5.0000e-04

```

Şekil 5.12. CNN-LSTM + attention modeline ait kısmi eğitim süreci çıktısı (ortalama süre belirtilmemiştir ancak işlem yükü yüksektir).

LSTM modeli, doğruluk oranları yüksek olsa da eğitim süresi açısından en yavaş model olmuştur. Buna karşın BiLSTM, benzer doğruluk oranlarına daha düşük latency ile ulaşarak zaman ve kaynak kullanımı bakımından daha etkin bir yapı sunmuştur. CNN-LSTM + Attention modeli ise teorik potansiyeline rağmen, eğitim süresi açısından pratik sınırlamalar barındırmaktadır. Özellikle büyük veri setlerinde bu modelin uygulanabilirliği, eğitim süresinin dikkatli analiz edilmesini gerektirmektedir.

Tablo 5.2. IDS modellerinin eğitim süresi, veri seti ve mimari özelliklerine göre karşılaştırılması.

Çalışma / Kaynak	Model Türü	Veri Seti	Eğitim Süresi / Epoch Süresi	Açıklama
Bu Tez Çalışması	LSTM	CIC-IoT2023	~311 sn / epoch	Sıralı yapı nedeniyle uzun süre
Bu Tez Çalışması	BiLSTM	CIC-IoT2023	~89 sn / epoch	Optimize yapı sayesinde daha hızlı
Kim et al., 2023	Attention (STOC)	Yahoo S5	~2.0 - 2.3 sn / epoch	Attention+ 1D Conv yapı, hızlı eğitim
Wang, 2023	BiLSTM-CNN	CTU-13	Belirtilmemiş (düşük batch, hızlı)	Batch size = 50 ile daha kısa süreli eğitim hedeflenmiş
Saba et al., 2022	CNN	BoT-IoT	30 epoch (süre belirtilmemiş)	Eğitim süresi doğrudan verilmemiş ama epoch sayısı 30 olarak sabitlenmiş
Udurume et al., 2024	CNN-BiLSTM	NSL-KDD / UNSW-NB15	Yüksek latency belirtildi	Yüksek doğruluk, ancak pratikte yavaş ve kaynak tüketici

Tablo 5.2.(Devamı) IDS modellerinin eğitim süresi, veri seti ve mimari özelliklerine göre karşılaştırılması.

Çalışma / Kaynak	Model Türü	Veri Seti	Eğitim Süresi / Epoch Süresi	Açıklama
Sun et al., 2020	CNN-LSTM	CICIDS2017	Ortalama 20–40 epoch	DL-IDS mimarisi, eğitim süresi ortalama seviyede
Bu Tez Çalışması	CNN-LSTM + Attention	CIC-IoT2023	Net bir epoch süresi ölçülemedi	Dikkat mekanizmasının çoklu sorgulama (multi- head attention) yapısı ve CNN katmanlarının filtre işlemleri gecikmeyi artırmakta, işlem yükü yüksek

Tablodaki veriler incelendiğinde, önerilen LSTM ve BiLSTM mimarilerinin eğitim süresi açısından literatürdeki pek çok çalışmayla karşılaştırıldığında dengeli ve uygulanabilir bir performans sunduğu görülmektedir. LSTM modeli, sıralı veri işleme yapısının doğal sonucu olarak her epoch için 311 saniyelik bir sürede çalışırken, BiLSTM modeli bu süreyi yaklaşık 89 saniyeye indirerek eğitim verimliliğini önemli ölçüde artırmıştır. Literatürde yer alan SAM ve Attentiontabanlı modellerin, özellikle self-attention mekanizmalarının paralel hesaplamaaya olan uygunluğu sayesinde düşük latency değerlerine sahip olduğu görülmektedir (örneğin 2 ms/paket gibi). Ancak bu yapıların Mirai gibi iç içe geçmiş ve ince ayırım gerektiren saldırı türlerinde hataya daha yatkın olduğu da yine karşılaştırmalı metriklerde belirtilmiştir. CNN-BiLSTM + Attentiongibi daha karmaşık mimariler ise yüksek parametre sayıları ve attention mekanizmalarının işlem yükü nedeniyle genellikle daha uzun eğitim sürelerine sahiptir ve bu da uygulamada kaynak tüketimini artırmaktadır. Sonuç itibarıyla, bu çalışma kapsamında önerilen BiLSTM mimarisi hem doğruluk hem de latency açısından literatürdeki birçok gelişmiş modele göre dengeli bir yapı sunmakta, pratik uygulanabilirlik açısından öne çıkmaktadır.

5.6. Benign + Mirai + DDoS Eğitimi

Bu bölümde Benign + Mirai ler + DDoS ların birleşiminden oluşan veri setinin eğitim sonuçlarından bahsedilmektedir.

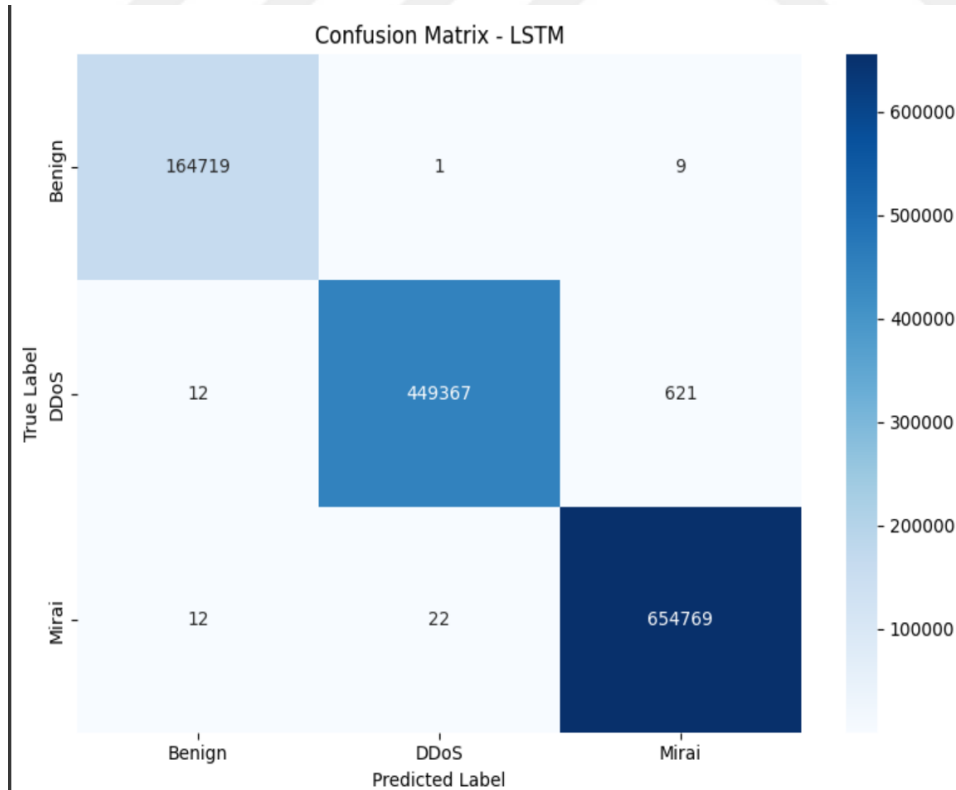
5.6.1. LSTM modeli sonuçları

Bu bölümde LSTM modelinde Benign + Mirai + DDoS eğitiminin sonuçları incelenmektedir.

	precision	recall	f1-score	support
Benign	0.999854	0.999939	0.999897	1.647290e+05
DDoS	0.999949	0.998593	0.999271	4.500000e+05
Mirai	0.999039	0.999948	0.999493	6.548030e+05
accuracy	0.999467	0.999467	0.999467	9.994667e-01
macro avg	0.999614	0.999494	0.999554	1.269532e+06
weighted avg	0.999467	0.999467	0.999467	1.269532e+06

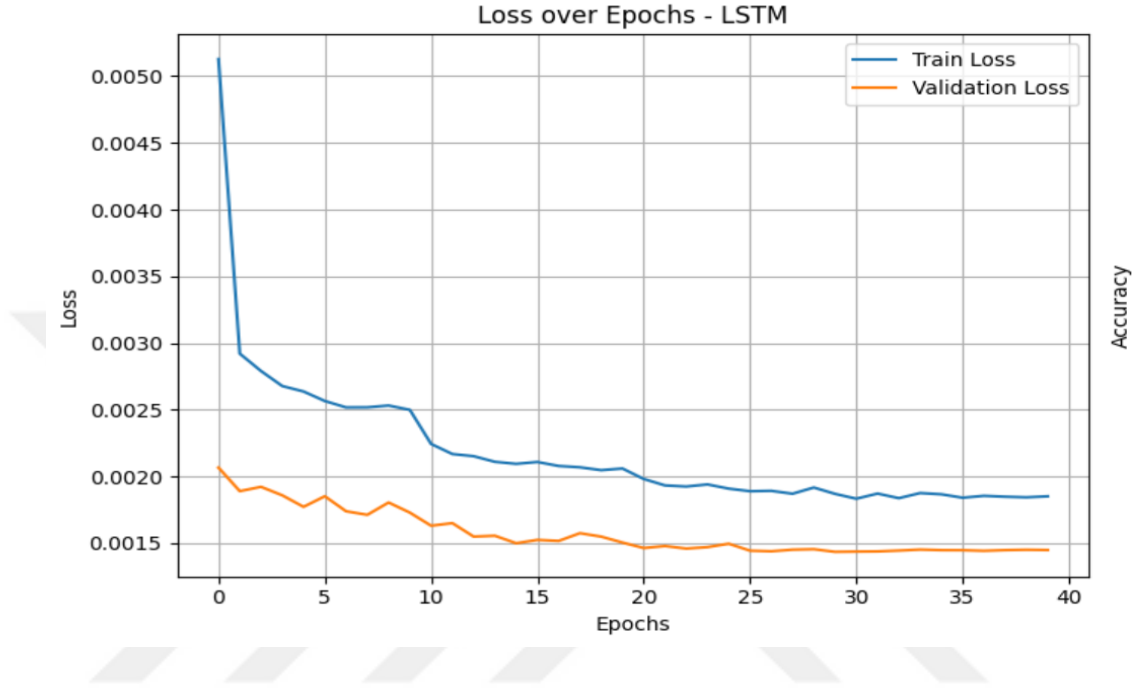
Şekil 5.13. LSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).

Şekil 5.13 incelendiğinde, LSTM modelinin sınıflandırma performansının oldukça yüksek olduğu görülmektedir. Özellikle tüm sınıflarda precision, recall ve f1-score metriklerinin 0.999 seviyelerinde seyretmesi, modelin zaman serisi verileri üzerindeki başarısını açıkça ortaya koymaktadır. Bu durum, modelin hem doğru pozitifleri yüksek oranda tespit edebildiğini hem de genel olarak dengeli bir sınıflandırma performansı sunduğunu göstermektedir.



Şekil 5.14. LSTM modeline ait confusion matrix.

Şekil 5.14'teki confusion matrix, modelin tahmin performansını sınıflar bazında ayrıntılı şekilde gözler önüne sermektedir. Özellikle Benign ve Mirai sınıflarında hata oranı yok denecek kadar az olup, DDoS sınıfında ise Mirai ile sınırlı bir karışma görülmektedir. Bu düşük hata oranları, modelin genel olarak sağlam bir şekilde genelleme yaptığını desteklemektedir.

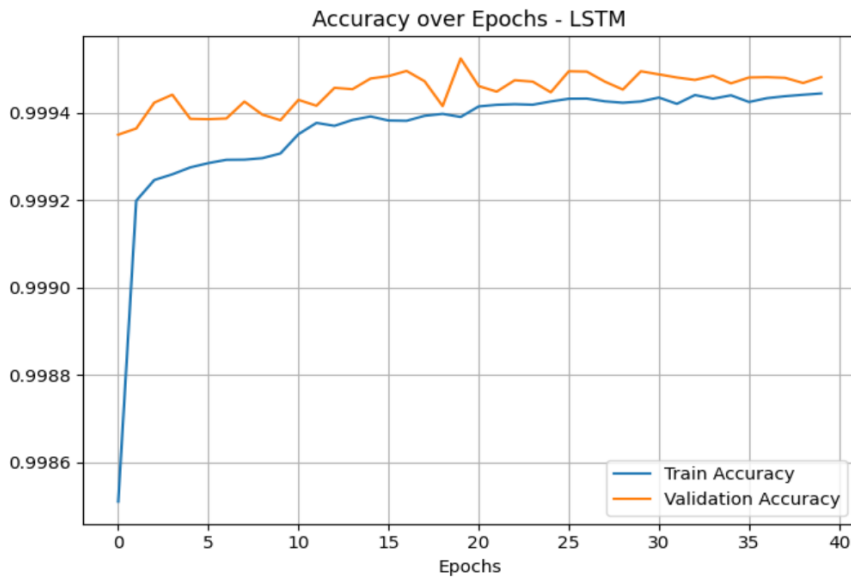


Şekil 5.15. LSTM modeline ait test loss grafiği.

Şekil 5.15'te yer alan LSTM modeline ait loss grafiği, eğitim süreci boyunca hem eğitim hem de doğrulama kayıplarının istikrarlı bir şekilde azaldığını göstermektedir. Özellikle ilk birkaç epoch'ta gözlenen keskin düşüş, modelin hızlı bir şekilde öğrenmeye başladığını ortaya koymaktadır. Daha sonraki epoch'larda kayıp değerlerinin yatay bir seyir izlemesi ise modelin dengeli bir öğrenme sürecine girdiğini göstermektedir.

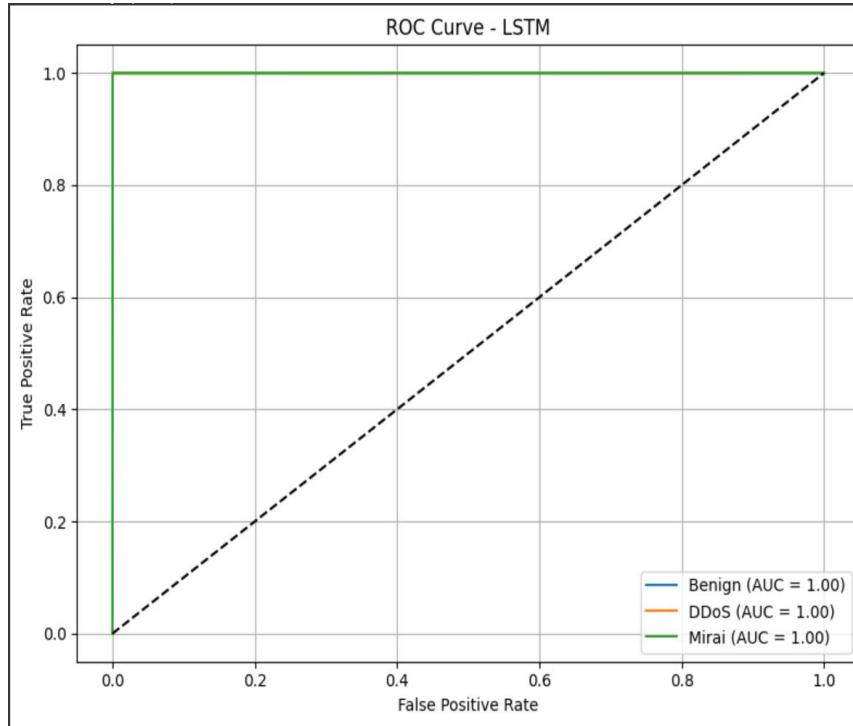
Doğrulama kaybının eğitim kaybına yakın seyretmesi, modelin aşırı öğrenmeye (overfitting) girmediğini ve genellenebilirliğinin yüksek olduğunu göstermektedir. Test verisi üzerinde hesaplanan 0.0015 seviyesindeki kayıp değeri de bu durumu desteklemektedir.

Şekil 5.16'da gösterilen doğruluk grafiği, LSTM modelinin eğitim süreci boyunca doğruluk değerlerinde sürekli bir artış sergilediğini ortaya koymaktadır. Hem eğitim hem de doğrulama doğrulukları yüksek seviyelerde seyretmiş ve özellikle 10. epoch sonrasında %99.9'un üzerine çıkmıştır.



Şekil 5.16. LSTM modeline ait test accuracy grafiği.

Eğitim ve doğrulama doğruluk eğrilerinin birbirine yakın ilerlemesi, modelin hem eğitim verisine hem de doğrulama verisine başarılı bir şekilde genelleme yapabildiğini göstermektedir. Epoch sayısı arttıkça doğruluk değerlerinde dalgalanma yaşanmaması, modelin kararlı bir öğrenme sürecine sahip olduğunu ortaya koymaktadır. Sonuç olarak, test verisinde elde edilen 0.9995 doğruluk oranı, modelin yüksek performansını teyit etmektedir.



Şekil 5.17. LSTM modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).

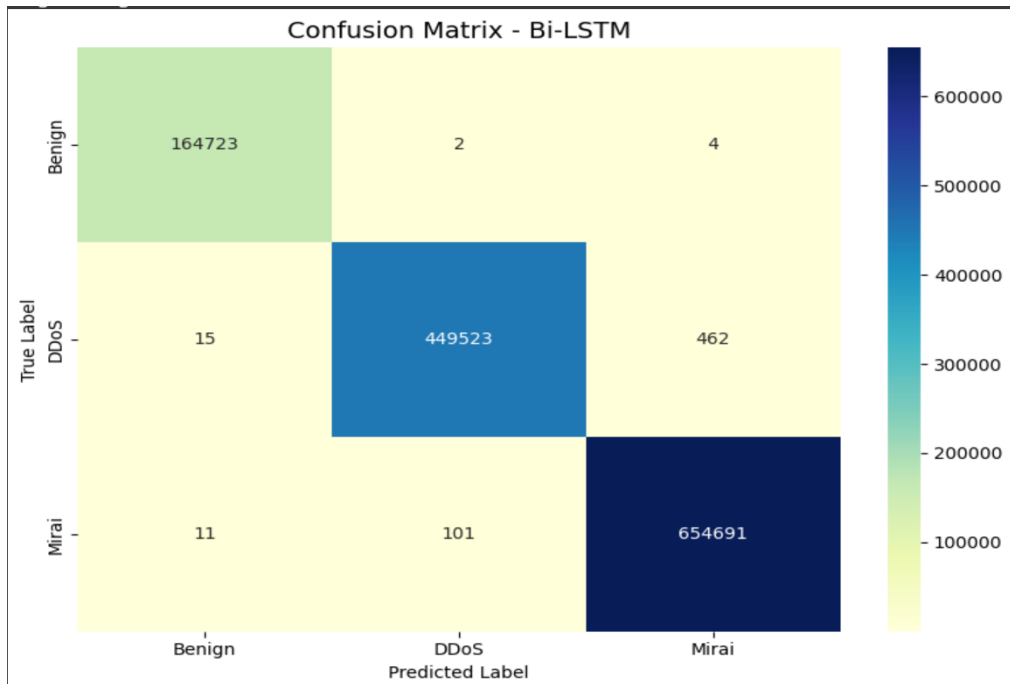
Şekil 5.17’de verilen ROC eğrileri, LSTM modelinin sınıflandırma performansını değerlendirmek için kullanılan önemli bir metriktir. Tüm sınıflar için AUC (Area Under Curve) değerinin 1.00 olarak hesaplanmış olması, modelin pozitif ve negatif sınıfları ayırt etme yeteneğinin mükemmel yakın olduğunu göstermektedir. ROC eğrisinin sol üst köşeye yakın seyretmesi, modelin düşük yanlış pozitif oranlarıyla yüksek doğru pozitif oranlarına ulaştığını işaret etmektedir.

5.6.2. BiLSTM modeli sonuçları

	precision	recall	f1-score	support
Benign	0.999842	0.999964	0.999903	1.647290e+05
DDoS	0.999771	0.998940	0.999355	4.500000e+05
Mirai	0.999289	0.999829	0.999559	6.548030e+05
accuracy	0.999531	0.999531	0.999531	9.995313e-01
macro avg	0.999634	0.999578	0.999606	1.269532e+06
weighted avg	0.999531	0.999531	0.999531	1.269532e+06

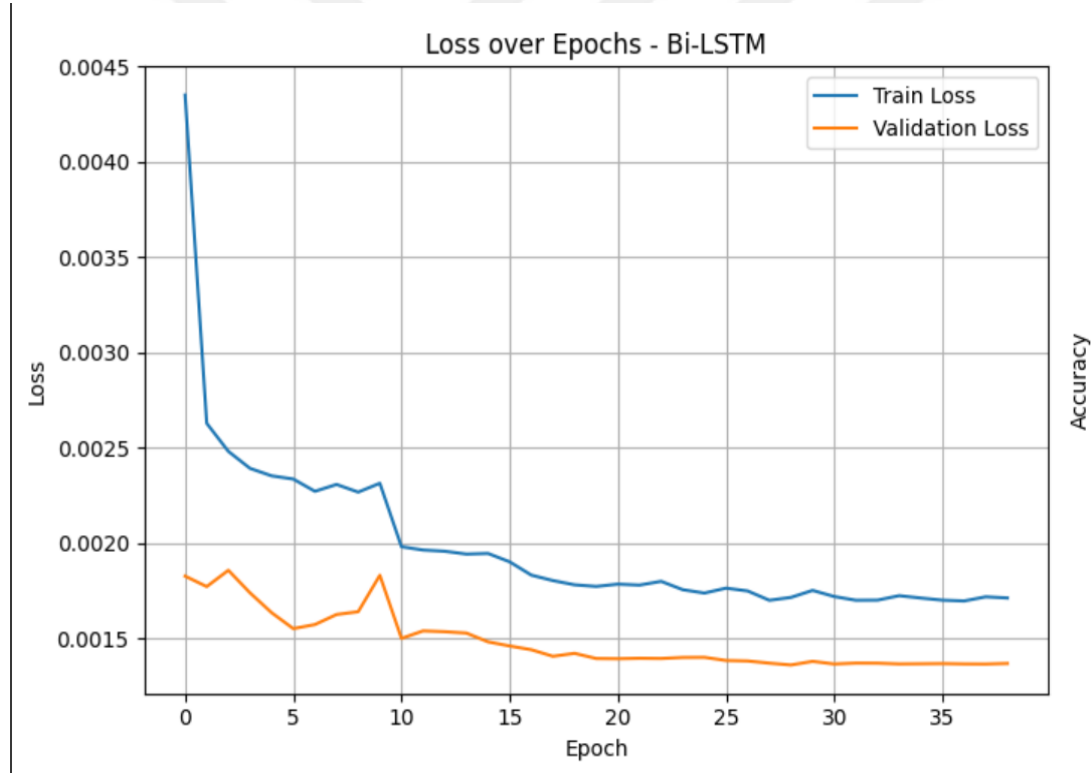
Şekil 5.18. BiLSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).

Şekil 5.18’de BiLSTM modelinin sınıflandırma performansı özetlenmiştir. Precision, recall ve f1-score değerlerinin tüm sınıflar için 0.999 seviyelerinde olması, modelin oldukça dengeli ve yüksek doğrulukta sonuçlar ürettiğini göstermektedir. LSTM’e kıyasla BiLSTM modeli, zaman serisi verilerinde geçmiş ve gelecek bağlamı aynı anda değerlendirebilme avantajına sahiptir.



Şekil 5.19. BiLSTM modeline ait confusion matrix.

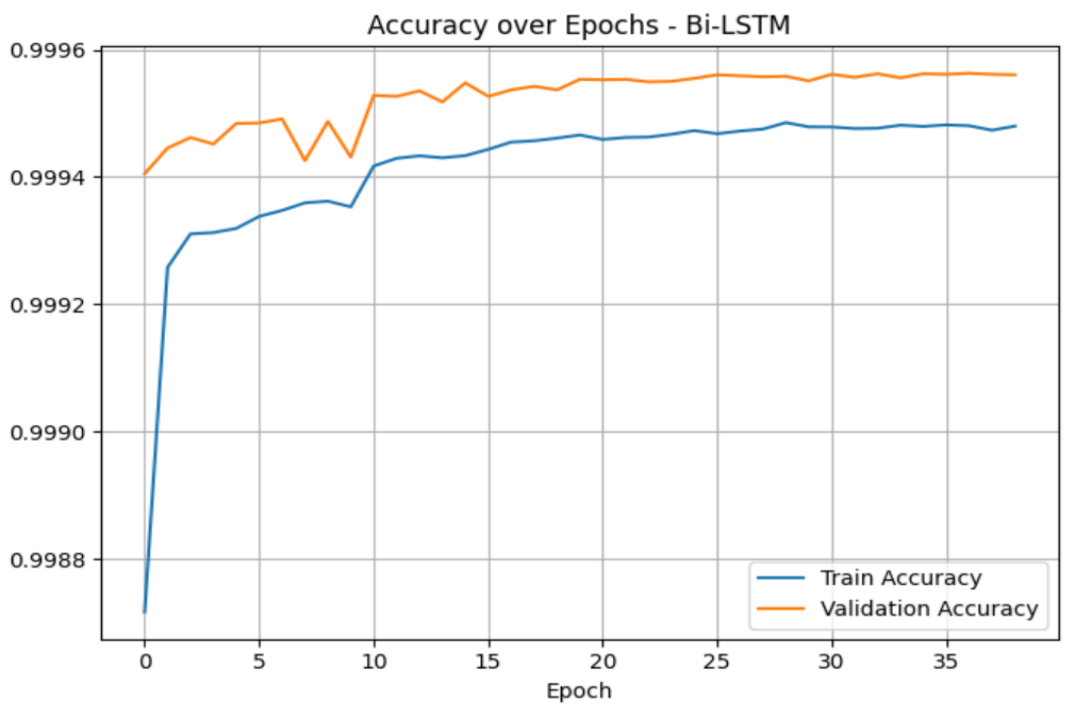
Bu nedenle BiLSTM'in sınıflandırma başarımı çok küçük farklarla da olsa LSTM modeline göre bazı durumlarda daha üst seviyededir. Bununla birlikte, genel tabloya bakıldığında her iki modelin de benzer performans değerlerine sahip olduğu görülmektedir. Şekil 5.19'da yer alan confusion matrix, BiLSTM modelinin hangi sınıflar arasında hata yaptığını detaylı şekilde göstermektedir. Genel başarı oldukça yüksek olsa da özellikle Mirai ve DDoS sınıfları arasında belirli ölçüde karışıklık gözlenmektedir. Mirai'ye ait 101 örneğin DDoS olarak tahmin edilmesi, bu iki saldırı türünün bazı zaman pencerelerinde benzer davranış örüntüleri sergileyebileceğine işaret etmektedir. Bu durum, çift yönlü veri akışına sahip modellerde bazı yapısal benzerliklerin model tarafından benzer olarak algılanabileceğini göstermesi açısından dikkat çekicidir. Ancak genel hata dağılımının düşük kalması, modelin öğrenme kapasitesinin güçlü olduğunu kanıtlamaktadır.



Şekil 5.20. BiLSTM modeline ait test loss grafiği.

Şekil 5.20'deki loss grafiği, BiLSTM modelinin eğitim sürecindeki kayıp değerlerinin zamanla nasıl değiştiğini göstermektedir. Grafik incelendiğinde hem eğitim hem de doğrulama kayıplarında erken epoch'lardan itibaren belirgin bir düşüş yaşandığı görülmektedir. Bununla birlikte 10. epoch civarında doğrulama kaybında kısa süreli dalgalanmalar gözlemlenmiş, ancak bu durum modelin genel performansını olumsuz

etkilememiştir. Modelin doğrulama kaybı, sonraki epoch'lerde istikrarlı bir seviyeye oturmuştur. Bu durum, modelin eğitim süreci sonunda hem ezberlemeye kaçmadan hem de yüksek doğrulukla genelleme yapabildiğini ortaya koymaktadır.



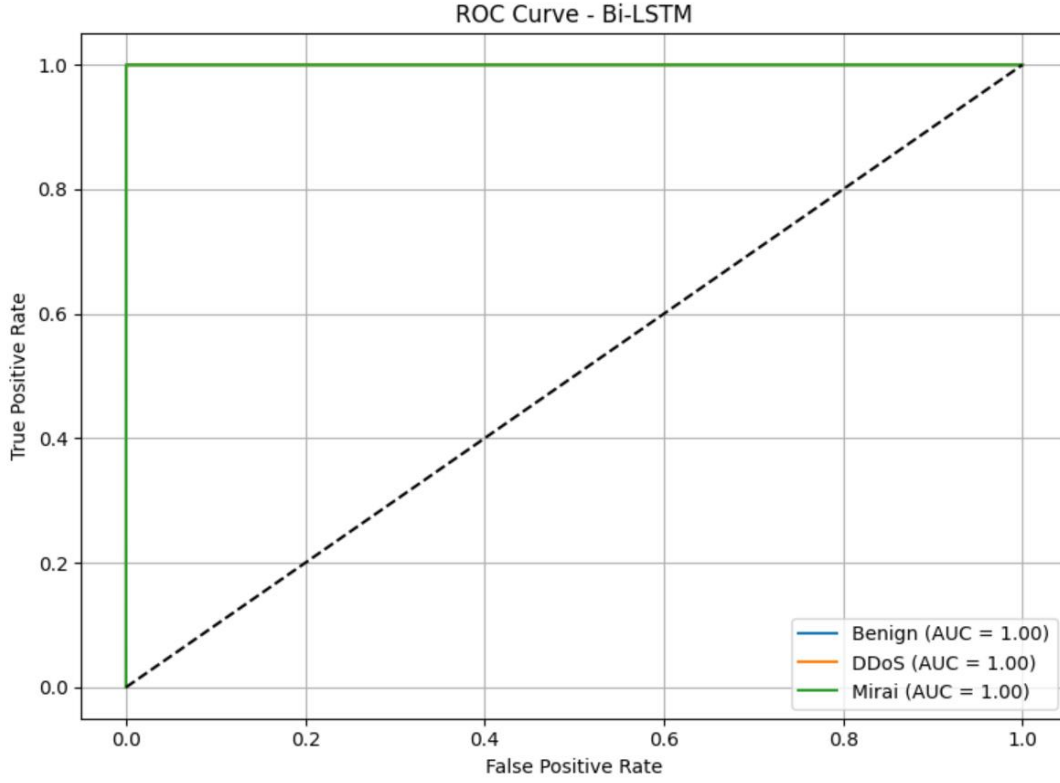
Şekil 5.21. BiLSTM modeline ait test accuracy grafiği.

Şekil 5.21’de BiLSTM modelinin epoch’lar boyunca elde ettiği doğruluk değerleri görselleştirilmiştir. Eğitim ve doğrulama doğruluklarının birbirine çok yakın seyretmesi, modelin öğrenme sürecinde dengeli bir genelleme yapabildiğini ortaya koymaktadır.

Özellikle erken epoch'larda hızlı yükselen başarı oranı, modelin veriye kısa sürede adapte olduğunu göstermektedir. Doğrulama doğruluğunun eğitimi zaman zaman aşması, modelin bazı örneklerde eğitim verisine kıyasla daha başarılı sınıflamalar yaptığına işaret edebilir.

Elde edilen 0.9995’lik test doğruluğu, modelin sınıflar arası ayrımı yüksek hassasiyetle gerçekleştirebildiğini teyit etmektedir. Bu başarıya rağmen, doğrulama eğrisinde gözlenen hafif dalgalanmalar modelin karmaşık yapılarla karşılaştığında karar sınırlarını esnek tutabildiğini göstermektedir.

Şekil 5.22’de sunulan ROC eğrileri, BiLSTM modelinin sınıf ayrımında sergilediği güçlü performansı ortaya koymaktadır.



Şekil 5.22. BiLSTM modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).

Her üç sınıf için AUC değerinin 1.00 olması, modelin yanlış pozitif oranı sıfıra yakinken bile doğru pozitif oranını koruyabildiğini göstermektedir. Bu durum, modelin yalnızca sınıfları doğru sınıflandırmakla kalmadığını, aynı zamanda karar eşiği değişse bile istikrarlı bir ayırt ediciliğe sahip olduğunu göstermektedir. Eğrilerin diyagonalden uzak ve sol üst köşeye yapışık seyretmesi, BiLSTM'in tüm sınıflarda optimum kararlar verebildiğini ifade etmektedir.

LSTM ile karşılaştırıldığında, bu sonuçlar modelin karar verme mekanizmasında yapısal bir avantaja sahip olmadığını, ancak eşdeğer bir ayırt ediciliği başarıyla sağlayabildiğini göstermektedir.

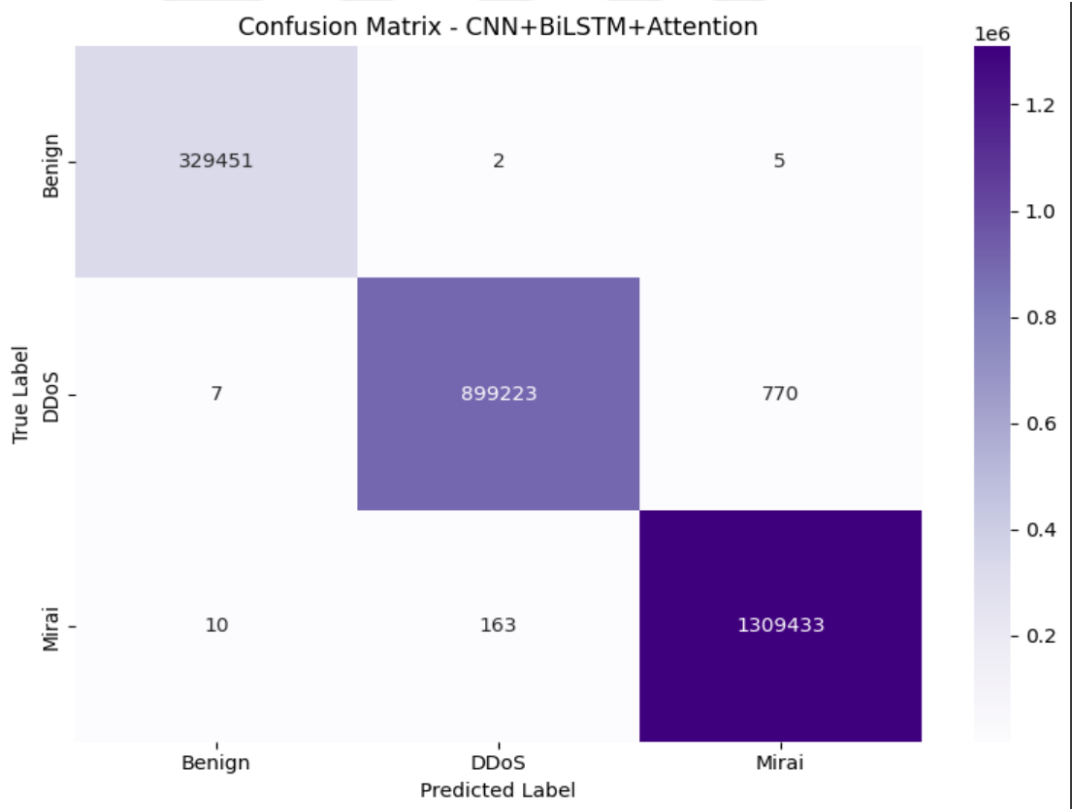
5.6.3. CNN-BiLSTM + attention modeli sonuçları

CNN-BiLSTM+Attention modeli, özellikle hem uzamsal (spatial) hem de zamansal (temporal) örüntüleri işleyebilmesi açısından diğer iki modele göre daha bütüncül bir mimaridir. CNN katmanı girdi özelliklerinin uzamsal ilişkilerini filtrelerken, BiLSTM zaman boyutunda çift yönlü ilişkileri yakalar ve Attention katmanı modelin hangi zamansal adımlara daha fazla odaklanması gerektiğini belirleyerek kararların doğruluğunu artırır.

	precision	recall	f1-score	support
Benign	0.999948	0.999979	0.999964	3.294580e+05
DDoS	0.999817	0.999137	0.999476	9.000000e+05
Mirai	0.999408	0.999868	0.999638	1.309606e+06
accuracy	0.999623	0.999623	0.999623	9.996231e-01
macro avg	0.999724	0.999661	0.999693	2.539064e+06
weighted avg	0.999623	0.999623	0.999623	2.539064e+06

Şekil 5.23. CNN-BiLSTM + attention modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).

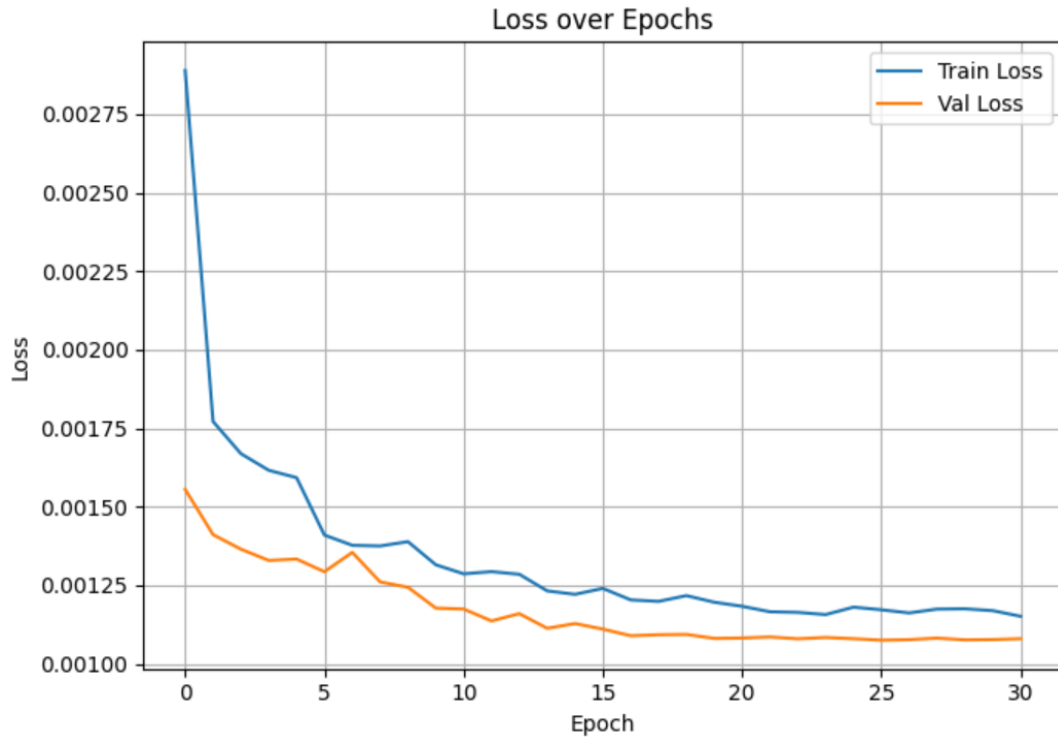
Şekil 5.23'te sunulan CNN-BiLSTM + Attention modeline ait sınıflandırma performansı, önceki modellerle karşılaştırıldığında daha ileri düzeyde doğruluk ve genelleme yeteneği sergilemektedir. F1-score değerlerinin üç sınıfta da 0.9996 seviyesine ulaşması, modelin sınıflar arası ayrımı son derece başarılı bir şekilde gerçekleştirdiğini göstermektedir. CNN katmanı sayesinde uzamsal özelliklerin öne çıkarılması ve Attention mekanizması ile önemli zaman anlarına odaklanılması, bu başarıyı sağlayan temel etkenlerdendir.



Şekil 5.24. CNN-BiLSTM + attention modeline ait confusion matrix.

Şekil 5.24'teki confusion matrix, CNN-BiLSTM + Attention modelinin hata oranlarını oldukça düşürdüğünü ortaya koymaktadır. Model, Benign ve DDoS sınıflarında

neredeyse hatasız çalışmakta; sadece Mirai sınıfında 163 örneği DDoS olarak tahmin etmiştir. Bu durum, önceki modellere kıyasla belirgin bir gelişme anlamına gelmektedir. Attention mekanizması, özellikle benzer saldırı davranışlarının zaman içinde ayrıştırılmasına katkı sağlayarak modelin sınıf bazlı başarımını optimize etmiştir. Bu sayede hem yüksek hassasiyet sağlanmış hem de karmaşık trafik desenleri etkin biçimde çözümlenmiştir.



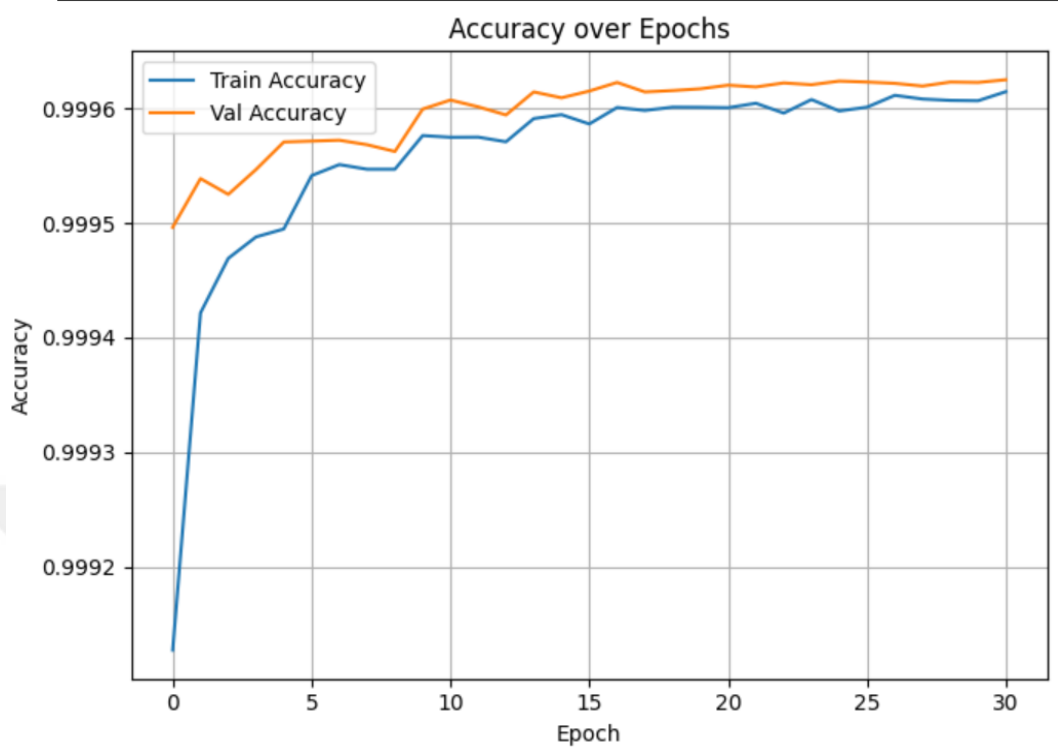
Şekil 5.25. CNN-BiLSTM + attention modeline ait test loss grafiği.

Şekil 5.25'teki loss grafiği incelendiğinde, eğitim sürecinde hem eğitim hem de doğrulama kayıplarının hızlı ve kararlı bir şekilde azaldığı görülmektedir. Eğitim kaybının epoch'lar boyunca istikrarlı şekilde düşmesi, modelin öğrenmeyi sürdürdüğünü; doğrulama kaybının düşük seviyelerde sabitlenmesi ise aşırı öğrenmenin engellendiğini göstermektedir.

Özellikle düşük başlangıç loss seviyeleri ve hızlı yakınsama, modelin mimarisindeki CNN ve Attention bileşenlerinin öğrenme sürecine sağladığı katkıyı desteklemektedir. Elde edilen minimum doğrulama kaybı değeri olan 0.0011, modelin diğer alternatiflere göre daha genel geçer bir yapıya sahip olduğunu ortaya koymaktadır.

Şekil 5.26'da sunulan doğruluk grafiği, CNN-BiLSTM + Attention modelinin yüksek kararlılıkla öğrenme gerçekleştirdiğini göstermektedir. Eğitim ve doğrulama

doğruluklarının birbirine oldukça yakın seyretmesi, modelin genelleme kabiliyetinin güçlü olduğunun bir diğer göstergesidir.

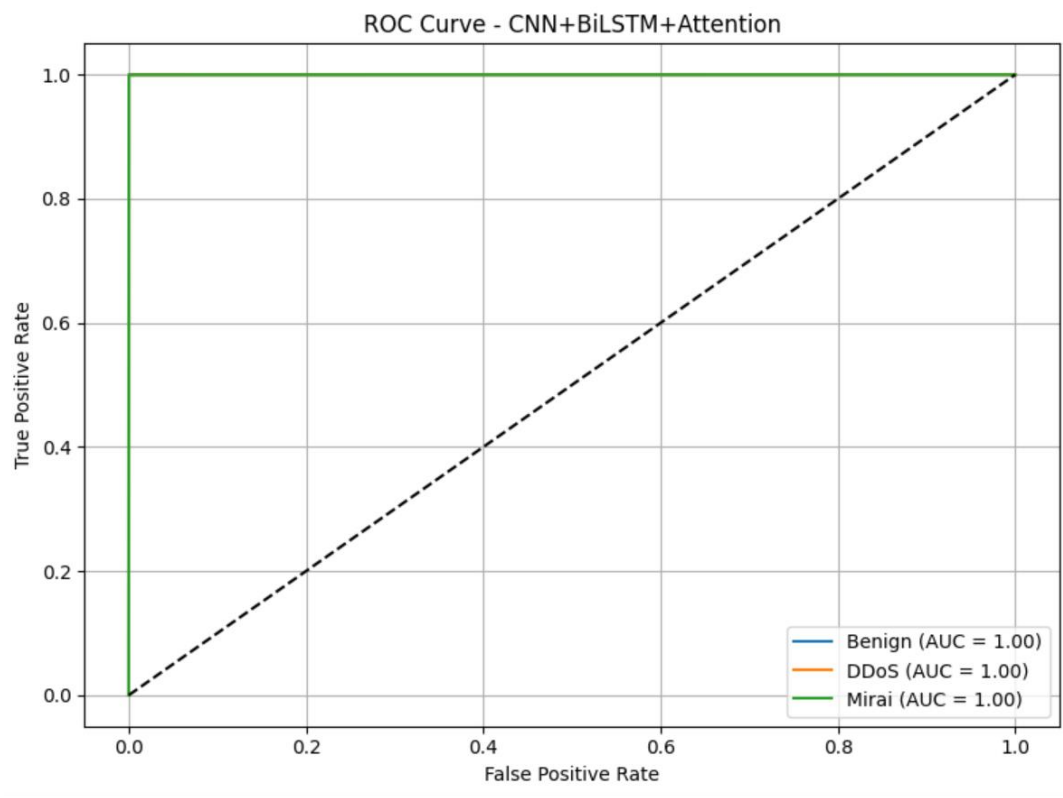


Şekil 5.26. CNN-BiLSTM + attention modeline ait test accuracy (doğruluk) grafiği.

Özellikle erken epoch’larda hızlı bir yükseliş gözlenmiş, 10. epoch sonrasında ise doğruluk değerleri 0.9996 seviyelerine sabitlenmiştir. Bu eğilim, modelin yüksek doğrulukta sınıflandırmayı erken evrede öğrendiğini ve bu başarıyı eğitim süreci boyunca koruduğunu göstermektedir. Düşük varyans ve yüksek doğruluk birlikte değerlendirildiğinde, bu modelin hem hassasiyet hem de kararlılık açısından öne çıktığı söylenebilir.

Şekil 5.27’de ROC eğrileri üzerinden incelenen model performansı, üç sınıf için de AUC değerinin 1.00 olarak hesaplanmasıyla zirveye ulaşmıştır. Bu durum, modelin sınıflar arasındaki ayrımı %100 doğrulukla gerçekleştirdiğini ve tüm karar eşiklerinde optimum sınıflandırma yaptığını göstermektedir.

ROC eğrilerinin diyagonalden uzak ve eksenlere yapışık ilerlemesi, yanlış pozitif oranının ihmal edilebilir düzeyde olduğunu ve bu sayede sistemin yanlış alarmlardan arındırılmış olduğunu ortaya koymaktadır. CNN-BiLSTM + Transformer mimarisi, özellikle zaman-serisi tabanlı ağ trafiği analizinde ROC bazında da en güçlü sonucu elde etmiştir.



Şekil 5.27. CNN-BiLSTM + attention modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).

Genel değerlendirme sonucunda üç model de yüksek başarıyla çalışmaktadır, ancak CNN-BiLSTM+Attention mimarisi gerek düşük test kaybı gerek hata türlerinin azaltılması yönünden daha üstün görünmektedir. Bu mimari, saldırı tespiti sistemlerinde hem doğru sınıflandırma hem düşük yanlışma oranı isteyen uygulamalar için daha uygun bir tercih sunmaktadır.

Eğitim sürecinde tüm modellerde erken durdurma, veri dengeleme, AUC değerlendirmesi, kayıp analizi ve karmaşıklık matrisi yorumlaması etkin şekilde kullanılmıştır. Bu sayede hem doğruluk yüksek tutulmuş hem de operasyonel uygulanabilirlik güçlendirilmiştir. Bu bağlamda model seçimi yapılırken kullanılacak ortamın hata toleransı, kaynak kapasitesi ve anomali tipolojisine göre CNN-BiLSTM+Attention tercih edilmesi önerilir.

Tablo 5.3'te LSTM, Bi-LSTM ve CNN-BiLSTM + Attention modellerinin karşılaştırmalı sonuçlarını sunmaktadır. Üç model de yüksek başarı oranlarına sahip olmakla birlikte, CNN-BiLSTM + Attention modeli tüm metriklerde bir adım öne çıkmıştır. Bu model, 0.9996 doğruluk oranı ve 0.0011 test kaybı ile yalnızca daha doğru tahminler yapmakla kalmamış, aynı zamanda daha kararlı ve düşük kayıplı bir

öğrenme süreci gerçekleştirmiştir. Precision, recall ve f1-score gibi kritik performans ölçütlerinde elde edilen 0.9997–0.9996 seviyeleri, saldırı tespiti gibi hassas konularda bu modelin daha güvenilir olduğunu göstermektedir. ROC-AUC değerinin diğer modeller gibi 1.00 çıkması, ayırt edicilik açısından da üst düzey performansın korunduğunu ifade etmektedir. Bu nedenle, hem metrikse üstünlük hem de operasyonel güven açısından CNN-BiLSTM + Attention modelinin tercih edilmesi makul ve önerilebilir bir yaklaşımdır.

Tablo 5.3. Ddos + benign + mirai eğitiminde farklı derin öğrenme modelleri sonuçları performans karşılaştırması.

Model	Accuracy	Test Loss	Precision (macro avg)	Precision (macro avg)	F1-Score (macro avg)	ROC - AUC
LSTM	0.9995	0.0015	0.9996	0.9995	0.9995	1
Bi-LSTM	0.9995	0.0014	0.9996	0.9995	0.9995	1
CNN-BiLSTM+ Attention	0.9996	0.0011	0.9997	0.9996	0.9996	1

5.7. Benign + Mirai Binary Model Eğitimi

Bu bölümde Benign ve Mirai’den oluşan binary eğitim modeli açıklanmıştır.

5.7.1. LSTM modeli sonuçları

Benign ve Mirai sınıflarını içeren veri seti üzerinde LSTM modeli uygulanırken K-Fold çapraz doğrulama (cross-validation) yöntemi tercih edilmiştir. Bu yöntem, modelin yalnızca tek bir eğitim-test bölünmesiyle değil, farklı veri alt kümeleri üzerinde denenerek daha güvenilir bir şekilde değerlendirilmesini sağlar.

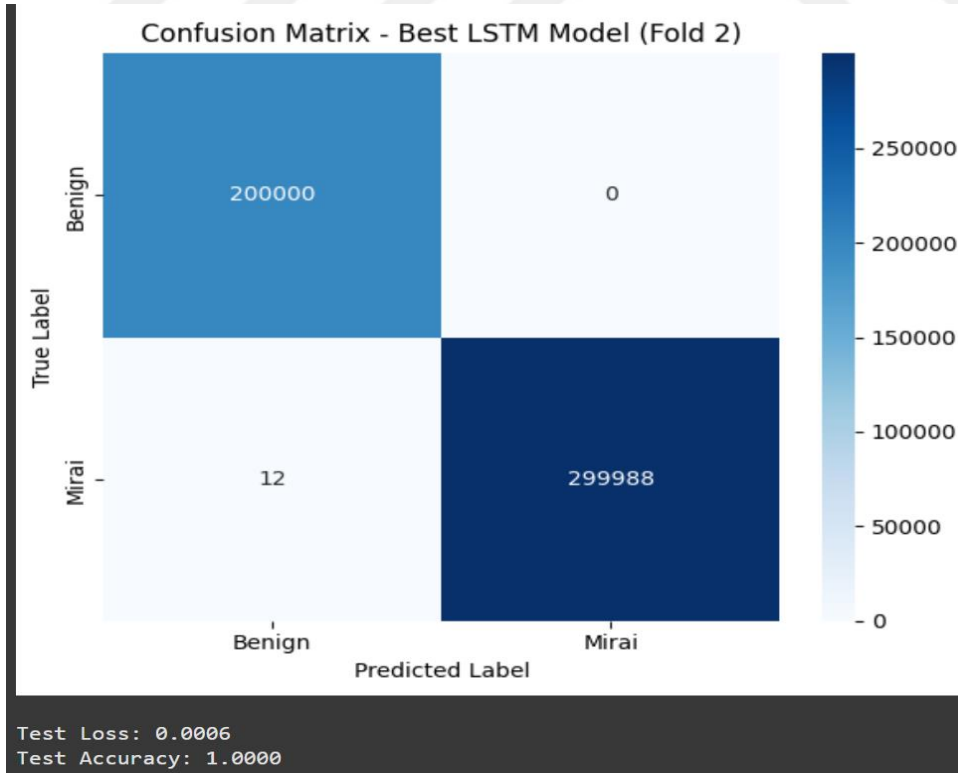
Veri seti belirli sayıda eşit parçaya (örneğin 5 fold) bölünür ve model her bir parçada test edilirken kalan parçalarda eğitilir. Böylece modelin genel performansı tüm veri kümesine daha adil bir biçimde yansıtılır. K-Fold uygulaması sırasında modelin her bir fold’daki eğitim ve doğrulama kaybı ile doğruluk değerleri ayrı ayrı izlenmiştir. Tüm fold’larda loss değerlerinin düzenli şekilde düştüğü ve doğruluk oranlarının hızla

yükseldiği gözlemlenmiştir. En iyi fold'da (örneğin Fold 2), test doğruluğu %100'e yakın olup test kaybı yalnızca 0.0006 olarak ölçülmüştür.

Classification Report:				
	precision	recall	f1-score	support
Benign	0.999940	1.000000	0.999970	200000.000000
Mirai	1.000000	0.999960	0.999980	300000.000000
accuracy	0.999976	0.999976	0.999976	0.999976
macro avg	0.999970	0.999980	0.999975	500000.000000
weighted avg	0.999976	0.999976	0.999976	500000.000000

Şekil 5.28. LSTM modelinin sınıflandırma performans raporu (f1-score değerleri genel olarak yüksek).

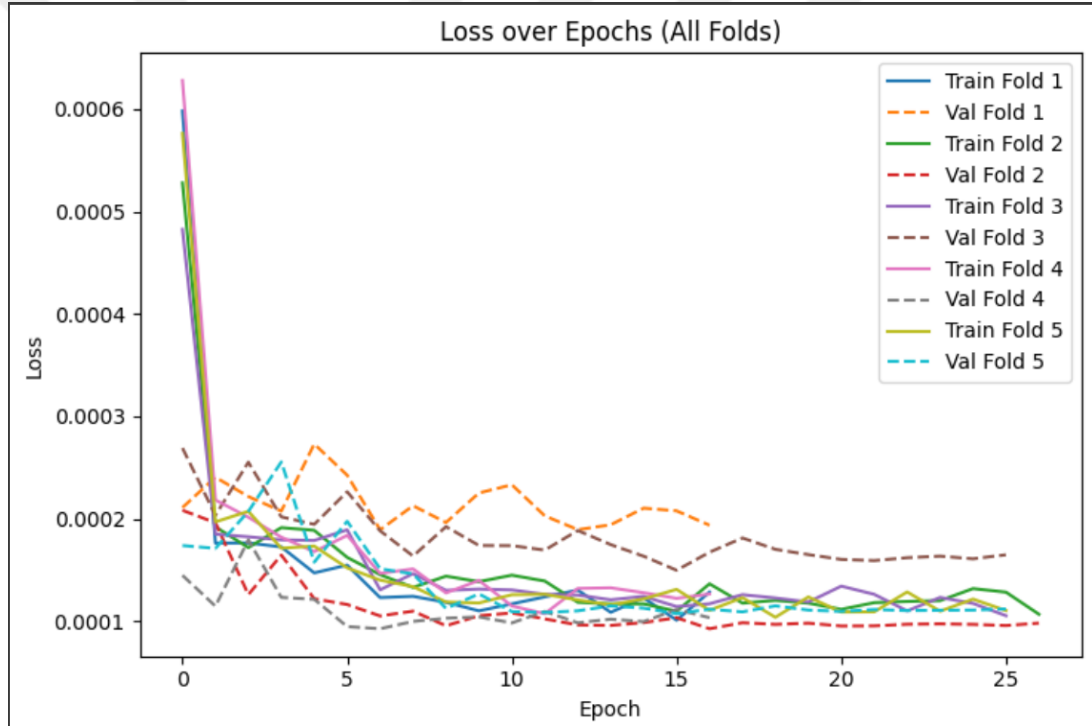
Şekil 5.28'de sunulan sınıflandırma performans raporu, LSTM modeline K-Fold doğrulama yöntemi uygulandığında elde edilen sonuçları göstermektedir. Precision, recall ve f1-score değerlerinin tamamı 0.9999 seviyelerine ulaşarak modelin sadece doğruluk değil, aynı zamanda sınıflar arası dengenin korunmasında da başarılı olduğunu ortaya koymuştur. Bu bağlamda, K-Fold yönteminin modelin aşırı uyuma girmeden daha genel geçer bir temsil üretmesine yardımcı olduğu söylenebilir. Özellikle veri dağılımının rastgele bölünmelere bağlı olmaksızın tüm katmanlarda değerlendirilmesi, modelin farklı senaryolara karşı adaptasyon kabiliyetini artırmıştır.



Şekil 5.29. LSTM modeline ait confusion matrix.

Şekil 5.29'da yer alan confusion matrix, K-Fold yönteminde ikinci fold'da eğitilen en iyi LSTM modelinin doğruluk performansını detaylandırmaktadır. Modelin 500.000 örnek üzerinde yalnızca 12 hatalı sınıflama yapmış olması (bunların tamamı Mirai örneklerinden kaynaklanmaktadır), istisnai bir başarı oranına işaret etmektedir. Toplam doğruluk değeri 1.0000, test kaybı ise sadece 0.0006 olarak kaydedilmiştir. Bu durum, klasik eğitim yöntemlerine göre çok daha az varyanslı ve daha istikrarlı bir model çıktısını işaret etmektedir. Bu düzeyde bir başarı, yalnızca modelin mimarisine değil; aynı zamanda K-Fold'un sağladığı çeşitlilik içeren öğrenme süreçlerine de atfedilebilir. Modelin farklı veri segmentlerinde de aynı başarıyı sürdürebilmesi, genel geçerliliği desteklemektedir.

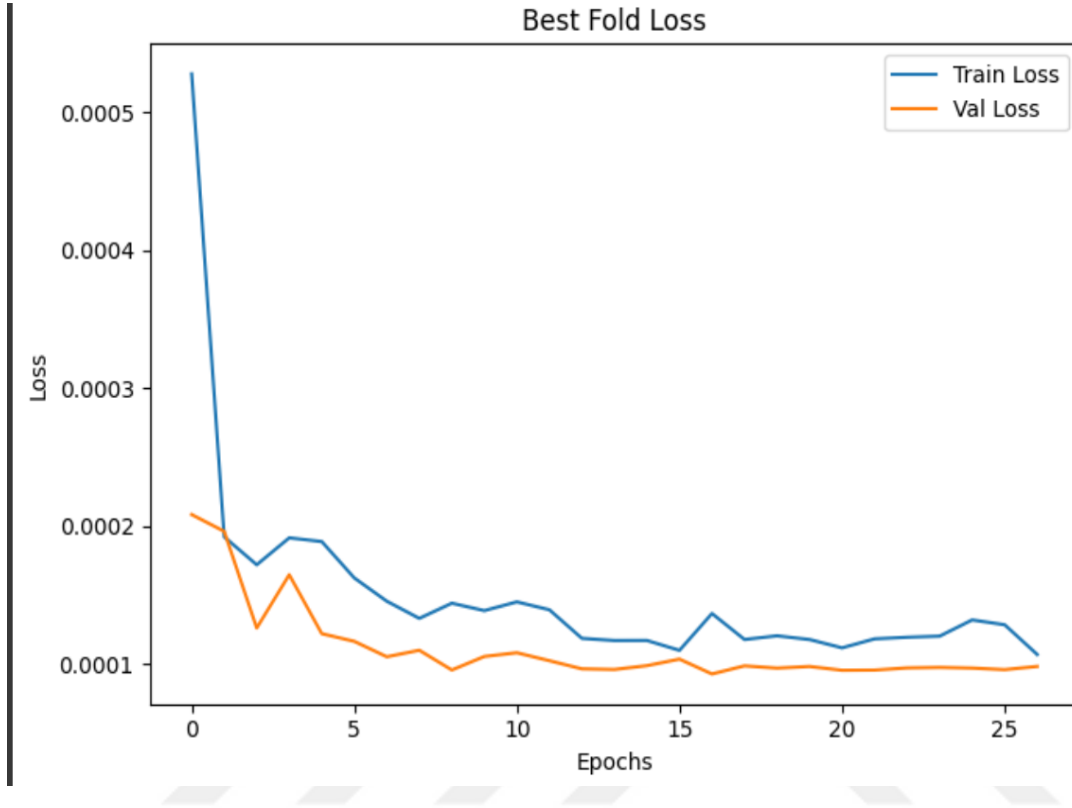
5.7.2. BiLSTM modeli sonuçları



Şekil 5.30. BiLSTM modeline ait tüm katlama (foldlar) için epoch lara göre kayıp değerleri.

Şekil 5.30'da BiLSTM modeline ait tüm katlamalar (fold'lar) boyunca gözlemlenen loss değerlerinin epoch'lara göre değişimi gösterilmektedir. Grafik, her bir fold'un eğitim ve doğrulama süreçlerinde izlediği kayıp eğrisini ayrı ayrı sunarak, modelin istikrarlı bir performans sergileyip sergilemediğinin bütüncül bir değerlendirmesini mümkün kılar. Tüm katmanlarda genel olarak hızlı bir yakınsama görülmekte ve doğrulama kayıpları düşük seviyelerde sabitlenmektedir. Bazı fold'larda doğrulama

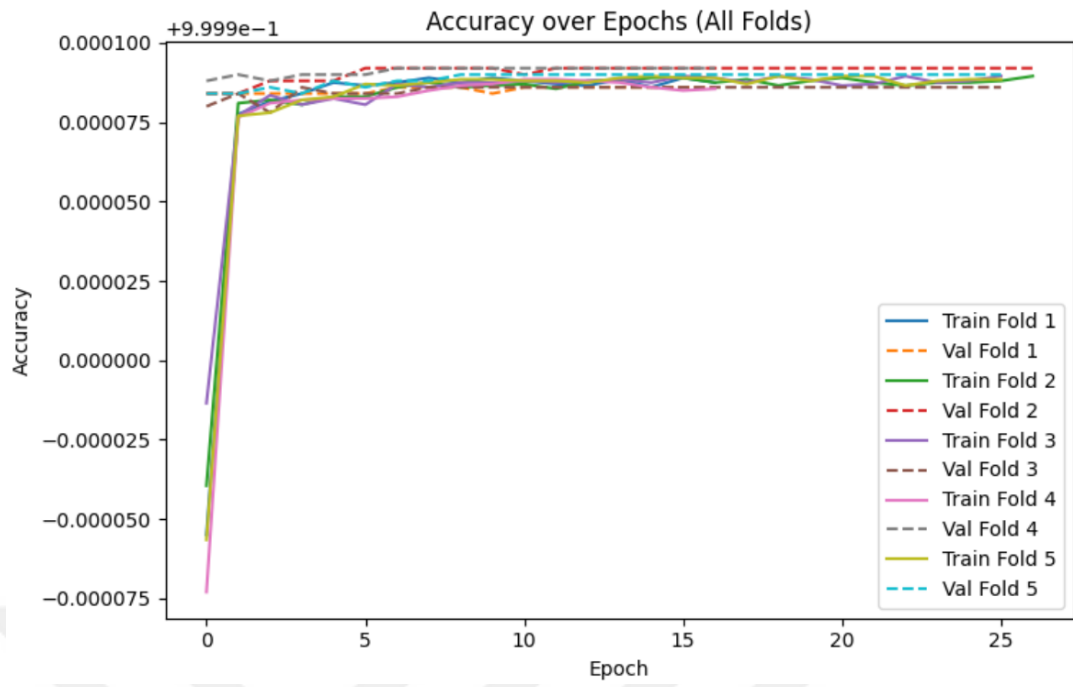
kaybının hafif dalgalanma göstermesi, verideki farklılıkların modele etkisini yansıtmakta; buna rağmen genel seyir, modelin overfitting eğiliminden uzak ve dengeli bir yapıda olduğunu göstermektedir.



Şekil 5.31. BiLSTM modeline ait en iyi katlama (fold) kaybı.

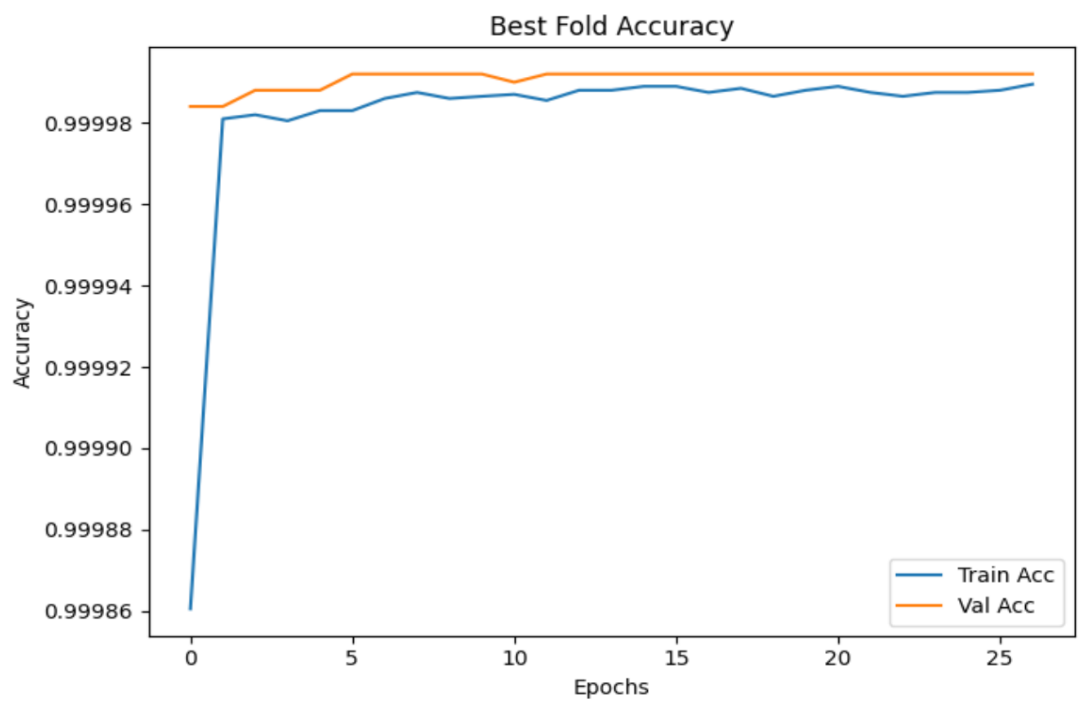
Şekil 5.31’de, BiLSTM modeli için en iyi performans gösteren katlamaya (best fold) ait eğitim ve doğrulama kayıpları ayrı olarak sunulmuştur. Grafikte eğitim kaybının başlangıçta hızlı bir şekilde düştüğü ve sonrasında düşük seviyelerde sabit kaldığı görülmektedir. Aynı şekilde, doğrulama kaybı da istikrarlı bir şekilde azalarak modelin genelleme başarısının güçlü olduğunu ortaya koymuştur. Bu grafik, BiLSTM’in belirli bir veri diliminde minimum kayıpla çalışabildiğini ve model parametrelerinin hem öğrenme sürecine hem de doğrulama performansına optimal katkı sunduğunu göstermesi açısından önemlidir.

Şekil 5.32’de, BiLSTM modelinin tüm katlamalara ait doğruluk (accuracy) değerleri epoch bazında gösterilmektedir. Tüm fold’lar için doğruluk oranlarının çok yüksek seviyelerde seyretmesi, modelin genel yapısal başarısını desteklemektedir. Grafik üzerindeki eğriler birbirine oldukça yakın ilerlemekte olup, bu durum farklı veri segmentleri üzerinde bile modelin kararlı ve homojen bir sınıflandırma performansı sergilediğini göstermektedir.



Şekil 5.32. BiLSTM modeline ait tüm katlamalar (foldlar) için epoch'lara göre accuracy (doğruluk) değerleri.

Şekil 5.33, BiLSTM modelinin en iyi performansı gösterdiği katlama (fold) için doğruluk (accuracy) eğrilerini detaylandırmaktadır. Eğitim ve doğrulama doğruluklarının 0.99998 seviyesinden başlayarak hızla 0.99999'un üzerine çıktığı ve süreç boyunca bu değeri koruduğu gözlemlenmektedir.



Şekil 5.33. BiLSTM modeline ait en iyi fold doğruluğu (accuracy).

5.8. Benign + DDoS Binary Model Eğitimi

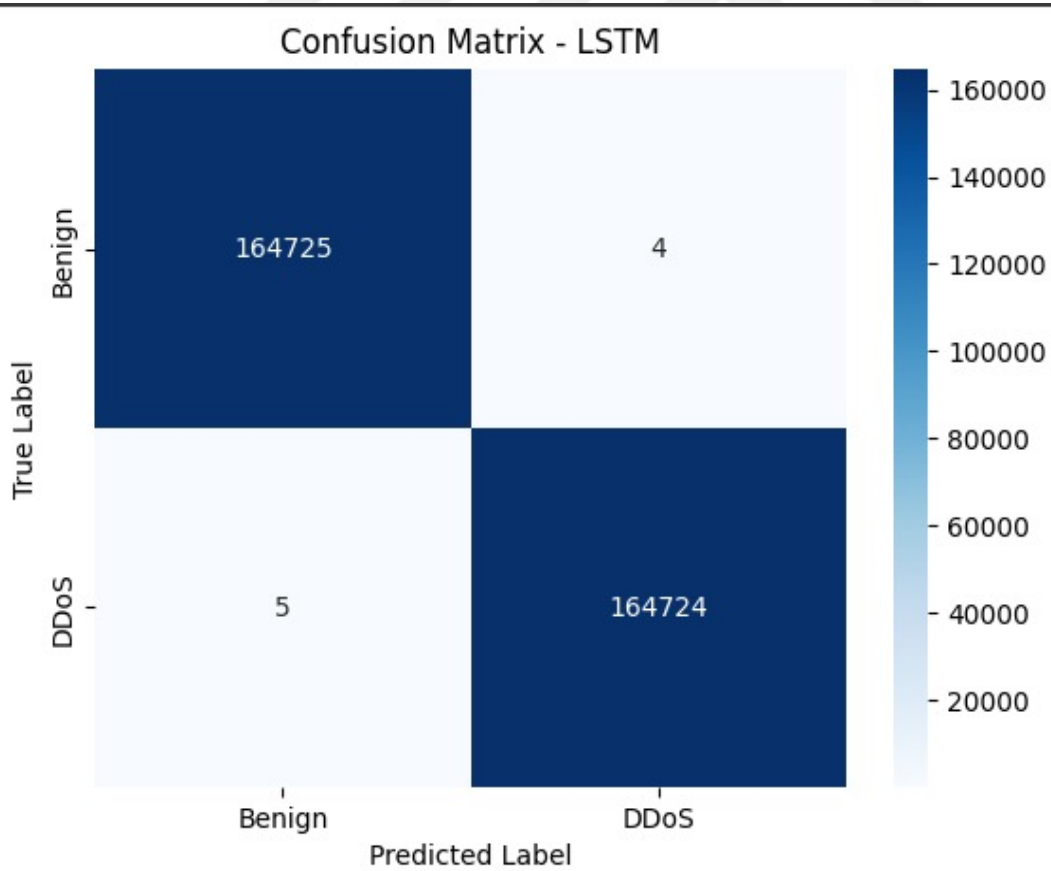
Bu bölümde Benign + DDoS binary model sonuçları anlatılmaktadır.

5.8.1. LSTM modeli sonuçları

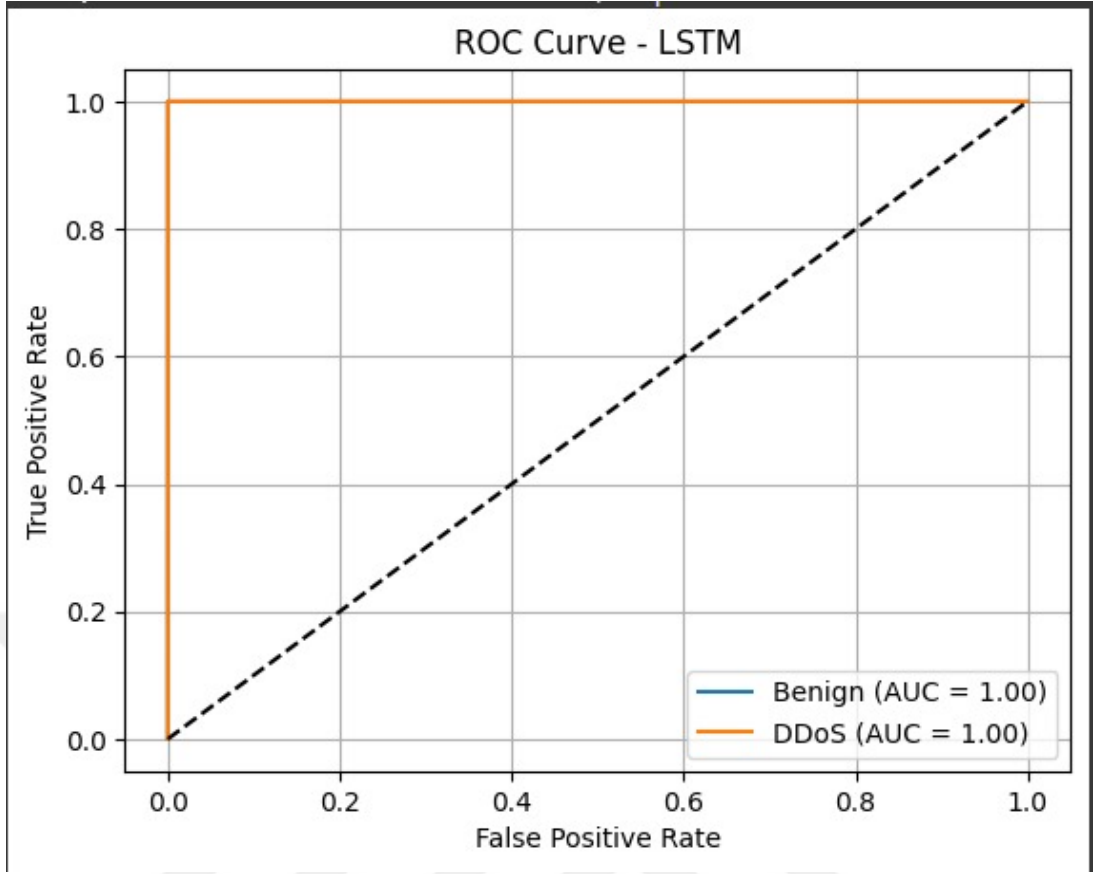
Benign ve DDoS sınıflarını içeren veri setiyle yapılan LSTM modelleme çalışmasında da K-Fold çapraz doğrulama yöntemi kullanılmıştır. Bu uygulama modelin sadece belirli bir veri alt kümesine bağlı kalmaksızın tüm veri seti üzerinde daha tutarlı ve dengeli şekilde test edilmesini sağlar.

Model, farklı fold'larda hem eğitim hem doğrulama sırasında istikrarlı bir şekilde yüksek doğruluk oranları sunmuştur. ROC eğrisinde her iki sınıf için AUC değeri 1.00 olarak gözlemlenmiştir.

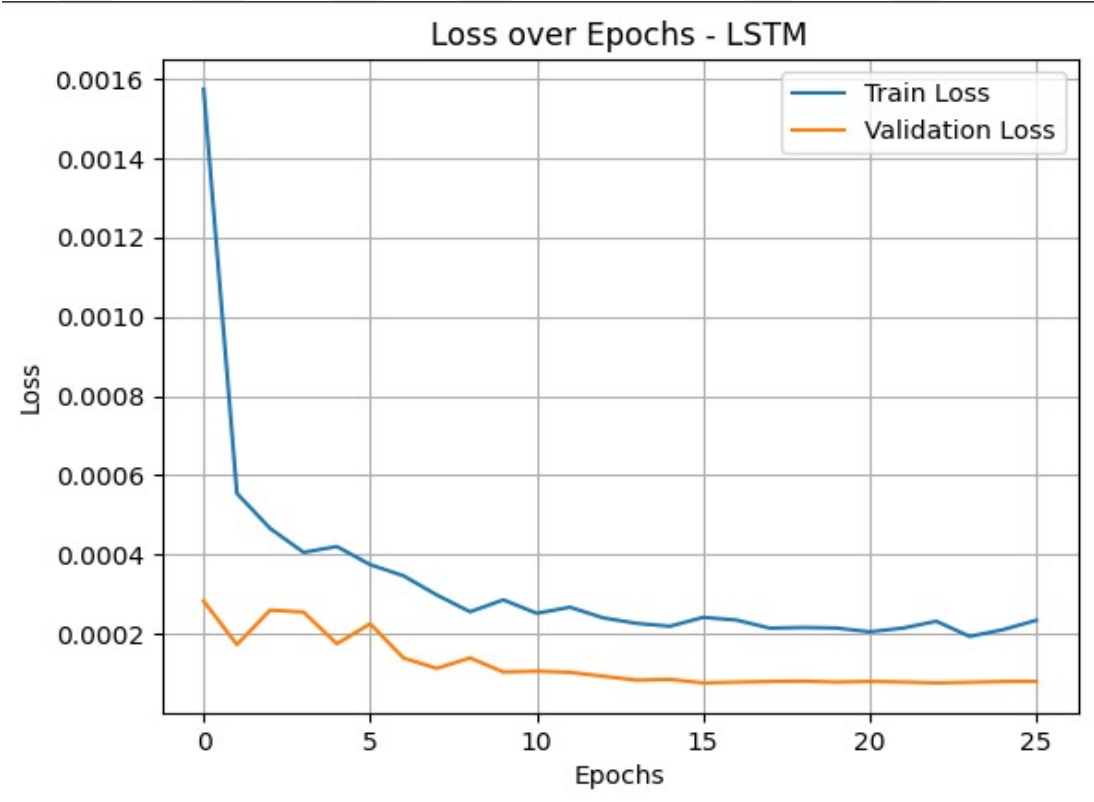
Şekil 5.34'de görüldüğü üzere Confusion matrix'e göre LSTM modeli, Benign ve DDoS sınıflarını oldukça yüksek doğrulukla sınıflandırmıştır. Sadece dokuz örnekte sınıflandırma hatası yapılmış olup, bu durum modelin sınıflandırma kabiliyetinin ne kadar yüksek olduğunu açıkça ortaya koymaktadır.



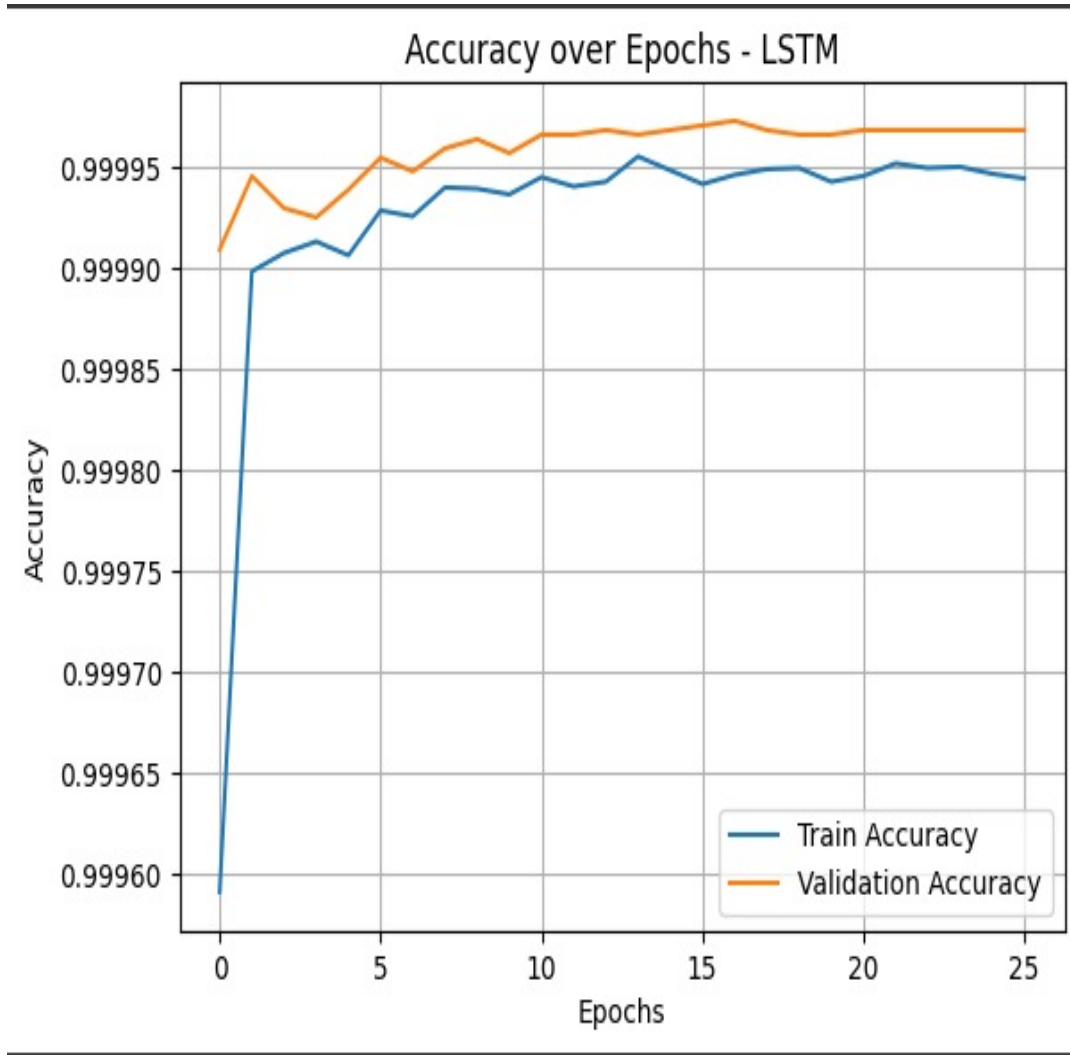
Şekil 5.34. LSTM modeline ait confusion matrix.



Şekil 5.35. LSTM modeli için ROC AUC eğrileri (tüm sınıflar için AUC = 1.00).



Şekil 5.36. LSTM modeline ait test loss grafiği.



Şekil 5.37. LSTM modeline ait test accuracy (doğruluk) grafiği.

Şekil 5.35'teki ROC eğrileri incelendiğinde, Benign ve DDoS sınıfları için AUC (Area Under Curve) değerlerinin 1.00 olması, modelin sınıfları hatasız bir şekilde ayırt ettiğini ve çok yüksek ayırt ediciliğe sahip olduğunu göstermektedir.

Şekil 5.36'daki Test verisi üzerinden elde edilen kayıp grafiğinde, eğitim ve doğrulama kayıp değerlerinin zamanla azaldığı ve sabitlendiği gözlemlenmektedir.

Şekil 5.37'deki modelin test doğruluğunu gösteren grafik incelendiğinde, doğruluk oranlarının epoch ilerledikçe hızlı bir şekilde yükseldiği ve yüksek bir seviyede sabitlendiği görülmektedir.

Ayrıca, K-Fold çapraz doğrulama yöntemi sayesinde modelin farklı veri alt kümelerinde test edilerek daha geniş varyasyonlara uyum sağladığı ve genellenebilirliğinin arttığı gözlemlenmiştir.

6. TARTIŞMA

Bu bölüm çıktıları literatür ile karşılaştırmayı, kullanılan modellerin güçlü ve zayıf yönlerini, çalışmanın gerçek hayat uygulanabilirliğini içermektedir.

6.1. Literatür ile Karşılaştırmalar

Bu bölümde elde edilen sonuçlar literatür ile karşılaştırılmaktadır.

6.1.1. LSTM

Bu tez çalışması kapsamında, CIC-IoT2023 veri seti üzerinde %98.00 doğruluk, %98.00 recall, precision ve F1-score sonuçları ile oldukça dengeli ve yüksek başarı sergileyen LSTM modeli önerilmiştir. Elde edilen sonuçlar, literatürdeki diğer benzer mimarilerle karşılaştırıldığında rekabetçi bir seviyede yer almaktadır. Örneğin, Sun et al. (2020) tarafından önerilen CNN-LSTM tabanlı saldırı tespit modeli, CICIDS2017 veri seti üzerinde >%99.50 precision ve recall ile çalışmasına rağmen, genel doğruluk oranı %98.67 seviyesindedir ve bu durum, daha farklı bir veri seti kullanılması nedeniyle doğrudan karşılaştırılabilir olmakla birlikte, tezin sunduğu başarımı da desteklemektedir (Sun et al., 2020). BoT-IoT ve NID veri setlerinde CNN mimarisiyle Saba et al. (2022) tarafından gerçekleştirilmiş çalışmada %99.51 / %92.85 oranında doğruluk elde edilmiştir. Bu çalışmada, çoklu veri seti kullanımı nedeniyle performans değişimi gözlenmiş olup, kullanılan modelin CNN olması nedeniyle LSTM ile doğrudan birebir kıyaslamak mümkün değildir (Saba et al., 2022). Daha karma bir veri seti üzerinde (IoT + NSL) gerçekleştirilen bir başka çalışmada LSTM modeli %98.34 doğruluk ile test edilmiş, precision %98.42, recall %98.28 ve F1-score %98.34 olarak kaydedilmiştir. Bu değerler, mevcut tez çalışmasında kullanılan CIC-IoT2023 veri setine benzer bir yapı ile elde edilen sonuçlarla oldukça yakın olup, önerilen modelin başarımlarını teyit etmektedir (s41598-0-25-94500-6.pdf). Enhancing IoT Security (2023) çalışmasında kullanılan CNN-LSTM mimarisi, yine CIC-IoT2023 veri setinde %98.42 doğruluk sağlamış, fakat precision ve recall değerleri sırasıyla %88.55 ve %87.43'te kalmıştır. Bu durum, önerilen LSTM modelinin sadece genel doğrulukta

değil, aynı zamanda tüm sınıflar için dengeli başarı gösterdiğini ortaya koymaktadır (Enhancing IoT Security, 2023).

Tablo 6.1. LSTM tabanlı modellerin literatürle başarımlarını karşılaştırması.

Kaynak / Çalışma	Model Türü	Veri Seti	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Bu Çalışma	LSTM	CIC-IoT2023	98.00	98.00	98.00	98.00
Sun et al. (2020) – DL-IDS	CNN-LSTM	CICIDS2017	98.67	>99.50	>99.50	>99.50
Bu Çalışma	LSTM	CIC-IoT2023	98.00	98.00	98.00	98.00
s41598-025-94500-5.pdf	LSTM	Karma (IoT+NSL)	98.34	98.42	98.26	98.34
Gueriani et al. (2024)	CNN-LSTM	CIC-IoT2023	98.42	98.85	98.43	98.57
Tanzila Saba et al. (2022)	CNN	BoT-IoT / NID	99.51 / 92.85	—	—	—
Akuthota & Bhargava (2025)	Transformer	NSL-KDD / UNSW-NB15	>99.00	>99.00	>99.00	>99.00
Mahmoud Elsayed et al. (2021)	CNN-LSTM	InSDN	96.32	—	—	—

Mahmoud Elsayed et al. (2021) tarafından gerçekleştirilen InSDN veri seti üzerindeki çalışmada %96.32 doğruluk elde edilmiştir. Ancak bu çalışmada F1-score, precision ve recall değerlendirme ölçütleri belirtilmemiştir ve kullanılan mimari CNN-LSTM olduğundan dolayı veri seti farklılığına ek olarak mimari farklılığı da kıyaslama zorluğu yaratmaktadır (Elsayed et al., 2021). Akuthota ve Bhargava (2025) tarafından

yapılan Attention tabanlı çalışmada UNSW-NB15 ve NSL-KDD veri setleri üzerinde %99.00 üstünde doğruluk, recall, precision ve F1-score sonuçları raporlanmıştır. Ancak Attention tabanlı mimarilerin eğitimi oldukça zaman alıcı ve yüksek hesaplama maliyetli olmasıyla bilindiğinden, bu mimarinin avantajları kadar zorlukları da dikkate alınmalıdır (Akuthota & Bhargava, 2025).

Bu tezde, literatürdeki mevcut diğer çalışmalarla kıyaslandığında hem doğruluk oranı hem de dengeli metrik başarımları açısından güçlü bir alternatif sunan LSTM modeli geliştirilerek, özellikle CIC-IoT2023 gibi karmaşık veri kümeleri üzerinde uygulanabilirliğini göstermektedir.

Bu tez çalışması kapsamında, CIC-IoT2023 veri seti üzerinde %98.00 doğruluk, %98.00 recall, precision ve F1-score sonuçları ile oldukça dengeli ve yüksek başarı sergileyen LSTM modeli uygulanmıştır. Elde edilen sonuçlar, literatürdeki diğer benzer mimarilerle karşılaştırıldığında rekabetçi bir seviyede yer almaktadır. (Tablo 6.1'de detaylı kıyaslama sunulmuştur). Örneğin, Sun et al. (2020) tarafından önerilen CNN-LSTM tabanlı saldırı tespit modeli, CICIDS2017 veri seti üzerinde >%99.50 precision ve recall ile çalışmasına rağmen, genel doğruluk oranı %98.67 seviyesindedir.

6.1.2. BiLSTM

Bu çalışmada LSTM mimarisine ek olarak Bi-LSTM modeli de önerilmiş ve zaman serisi verilerde çift yönlü örüntü öğreniminin saldırı tespitine etkisi incelenmiştir. Tablo 6.2'de görüldüğü üzere CIC-IoT2023 veri seti üzerinde eğitilen Bi-LSTM modeli, %98.00 doğruluk ve aynı düzeyde recall, precision, ve F1-score sonuçlarıyla oldukça başarılı bir performans sergilemiştir.

Bu sonuçlar, hem sınıf bazlı dengeli tahmin başarımları hem de genel model doğruluğu açısından değerlendirilmiştir. Sun et al. (2020) tarafından CNN-LSTM mimarisiyle yapılan çalışmada, CICIDS2017 veri seti üzerinde >%99.50 precision ve recall değerlerine ulaşılmıştır. Fakat bu çalışmada CNN katmanlarının zaman serisine etkisi ve kullanılan veri setinin farklı olması nedeniyle sonuçların doğrudan karşılaştırılması sınırlıdır (Sun et al., 2020). Benzer biçimde, s41598-025-94500-5 referanslı çalışmada kullanılan IoT + NSL karma veri seti üzerinde Bi-LSTM yapısıyla %98.34 doğruluk ve %98.26–%98.42 aralığında precision ve recall sonuçlarına ulaşılmıştır.

Bu çalışma, Bi-LSTM mimarisinin farklı veri setleri üzerinde de oldukça yüksek başarı sunabildiğini göstermektedir. Enhancing IoT Security (2023) başlıklı çalışmada, aynı bu tezde kullanılan CIC-IoT2023 veri seti üzerinde CNN-LSTM mimarisiyle %98.42 doğruluk elde edilmiş; precision, recall ve F1-score değerleri ise sırasıyla %98.85, %98.43 ve %98.57 olarak raporlanmıştır. Bu değerler Bi-LSTM modelimizle karşılaştırıldığında oldukça yakın olmakla birlikte, Bi-LSTM'nin daha sade mimarisiyle bu performansa ulaşması önemli bir avantaj sunmaktadır.

Tablo 6.2. BiLSTM tabanlı modellerin literatürle başarımların karşılaştırması.

Kaynak / Çalışma	Model Türü	Veri Seti	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Bu Çalışma	BiLSTM	CIC-IoT2023	98.00	98.00	98.00	98.00
Sun et al. (2020) – DL-IDS	CNN-LSTM	CICIDS2017	98.67	>99.50	>99.50	>99.50
s41598-025-94500-5.pdf	LSTM	Karma (IoT+NSL)	98.34	98.42	98.26	98.34
Gueriani et al. (2024)	CNN-LSTM	CIC-IoT2023	98.42	98.85	98.43	98.57
Mahmoud Elsayed et al. (2021)	CNN-LSTM	InSDN	96.32	—	—	—
Tanzila Saba et al. (2022)	CNN	BoT-IoT / NID	99.51 / 92.85	—	—	—
Akuthota & Bhargava (2025)	Transformer	NSL-KDD / UNSW-NB15	>99.00	>99.00	>99.00	>99.00

Mahmoud Elsayed et al. (2021) tarafından InSDN veri seti üzerinde önerilen CNN-LSTM mimarisi ise %96.32 doğruluk oranına ulaşmıştır. Ancak bu çalışmada diğer

metrikler belirtilmediğinden dolayı kapsamlı bir kıyaslama mümkün değildir. Benzer şekilde, Tanzila Saba et al. (2022) çalışmasında BoT-IoT / NID veri seti ile CNN tabanlı bir mimari kullanılmış ve %99.51 / %92.85 doğruluk elde edilmiştir. Veri seti çeşitliliği ve mimari farklılıklar bu çalışmaları doğrudan karşılaştırılabilir olmaktan uzaklaştırmaktadır.

Attention mimarisiyle çalışan Akuthota ve Bhargava (2025), UNSW-NB15 / NSL-KDD veri setlerinde doğruluk ve başarı oranlarını %99'un üzerinde elde etmiştir. Fakat Attention tabanlı yapılar genellikle daha yüksek işlem gücü gerektirdiğinden, Bi-LSTM gibi daha yalın modellerle benzer başarıların elde edilebilmesi, bu tezin yaklaşımının pratik açıdan değerli olduğunu ortaya koymaktadır.

Önerilen Bi-LSTM modeli, literatürdeki benzer çalışmalarla karşılaştırıldığında sınıf bazlı denge ve doğruluk açısından güçlü ve uygulanabilir bir performans sunmaktadır. Bu durum, özellikle IoT ağlarında gerçek zamanlı saldırı tespiti sistemleri için modelin etkili bir çözüm sunduğunu desteklemektedir.

6.1.3. CNN-LSTM + attention

Bu tezde önerilen CNN-LSTM + Attention mimarisi, zaman serisi verilerde hem uzamsal hem de ardışık bağımlılıkları öğrenebilen karma bir yapıya sahiptir. Özellikle CNN katmanları verinin lokal örüntülerini çıkarırken, LSTM katmanları sıralı bağımlılığı modellere taşır ve Attention bileşeni ise dikkat mekanizması aracılığıyla daha uzun vadeli ilişki yapılarını keşfetmeyi mümkün kılar. Önerilen bu hibrit mimari, CIC-IoT2023 veri seti üzerinde uygulanarak değerlendirildi ve Tablo 6.3'te literatür çalışmalarıyla karşılaştırılmıştır

Sun et al. (2020) tarafından CNN-LSTM mimarisiyle önerilen model, CICIDS2017 veri setinde %98.67 doğrulukla çalışmış ve >%99.50 precision ve recall skorları raporlanmıştır. Bu sonuçlar oldukça yüksektir, ancak kullanılan veri seti ve mimaride Attention bileşeni bulunmamaktadır (Sun et al., 2020). Benzer biçimde, s41598 referanslı çalışmada LSTM kullanılarak IoT+NSL veri setinde %98.34 doğruluk elde edilmiş, ancak Attention entegrasyonu olmamıştır. Bu nedenle önerilen modelin performansına doğrudan alternatif oluşturamamaktadır.

Enhancing IoT Security (2023) çalışmasında ise aynı veri seti (CIC-IoT2023) üzerinde CNN-LSTM mimarisiyle %98.42 doğruluk, %98.85 precision, %98.43 recall ve

%98.57 F1-score değerleri raporlanmıştır. Bu sonuçlar, Attention eklemesi yapılmamış olmasına rağmen modelin başarılı performansını göstermektedir.

Tablo 6.3. CNN-LSTM + attention ve benzeri literatür modellerinin performans karşılaştırması.

Kaynak / Çalışma	Model Türü	Veri Seti	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Bu Çalışma	CNN-LSTM + Attention	CIC-IoT2023	88.00	87.00	88.00	83.00
Sun et al. (2020) – DL-IDS	CNN-LSTM	CICIDS2017	98.67	>99.50	>99.50	>99.50
s41598-025-94500-5.pdf	LSTM	Karma (IoT+NSL)	98.34	98.42	98.26	98.34
Gueriani et al. (2024)	CNN-LSTM	CIC-IoT2023	98.42	98.85	98.43	98.57
Mahmoud Elsayed et al. (2021)	CNN-LSTM	InSDN	96.32	—	—	—
Tanzila Saba et al. (2022)	CNN	BoT-IoT / NID	99.51 / 92.85	—	—	—
Akuthota & Bhargava (2025)	Transformer	NSL-KDD / UNSW-NB15	>99.00	>99.00	>99.00	>99.00

Çalışmada önerilen hibrit mimaride gözlenen belirli sınıflarda düşen AP skorları ve artan hata oranları, Attention bileşeninin yüksek parametrelili doğasının sınıflar arası ayırım gücünde istenen netliği sağlayamayabileceğini göstermektedir.

Mahmoud Elsayed et al. (2021) çalışmasında CNN-LSTM mimarisi InSDN veri setinde %96.32 doğrulukla test edilmiştir. Fakat recall, precision ve F1-score ölçütleri verilmediği için detaylı bir kıyaslama mümkün değildir. Öte yandan Tanzila Saba et al. (2022), CNN ile BoT-IoT ve NID veri setlerinde sırasıyla %99.51 ve %92.85 doğruluk bildirmiştir. Veri seti farklılığı ve Attention olmaması nedeniyle doğrudan kıyas yapılması sınırlıdır.

Attention tabanlı model kullanan ve UNSW-NB15 / NSL-KDD veri kümeleri üzerinde çalışan Akuthota & Bhargava (2025), %99'un üzerinde tüm metriklerde performans göstermiştir. Ancak bu çalışmada hem veri seti farklıdır hem de modelin latency ve işlem gücü ihtiyacı teze göre daha yüksektir. Buna rağmen, Attention tabanlı yapılar özellikle dikkat mekanizması gerektiren görevlerde öne çıkmaktadır (Akuthota & Bhargava, 2025).

Sonuç olarak, bu tezde önerilen CNN-LSTM + Attention mimarisi, genel doğrulukta yüksek başarı elde etmekle birlikte, özellikle Mirai alt sınıfları gibi sınıflarda dikkatli ayar yapılmadığında ayırım gücünde düşüşler yaşayabilmektedir. Bu durum, Attention bileşeninin yüksek karmaşıklığına rağmen her zaman daha üstün sonuç üretmeyebileceğini ve daha sade mimarilerin bazı durumlarda daha istikrarlı performans sunabileceğini ortaya koymaktadır.

6.2. Modellerin Güçlü ve Zayıf Yönleri

Bu bölümde sırasıyla LSTM, BiLSTM ve CNN-LSTM + Attention modellerinin uygulama bazlı güçlü ve zayıf yönleri anlatılmaktadır. Bu yönleri tablolar ile gösterilerek sunulmaktadır.

6.2.1. LSTM

Bu bölümde LSTM modelinin bu tezdeki uygulamasına göre belirlenen güçlü ve zayıf yanları anlatılmaktadır. Bu tez kapsamında uygulanan LSTM modeli, zaman serisi veriler üzerindeki güçlü örüntü yakalama kabiliyeti ile öne çıkmıştır. Model, geçmiş örneklerden öğrendiği yapılar aracılığıyla, gelecekteki saldırı desenlerini anlamlı şekilde tahmin edebilmiştir.

Özellikle zaman serisi uyumu, dengeli öğrenme yapısı, ve düşük hata yayılımı gibi avantajları sayesinde, Benign ve DDoS gibi belirgin örüntülere sahip sınıflarda oldukça üst düzeyde başarılı sonuçlar elde edilmiştir. Ayrıca modelin basit ve stabil

mimarisi, düşük GPU kaynaklarıyla dahi yüksek doğruluk sağlayabilmesi açısından önemli bir avantaj sunmuş, bu da onu pratik uygulamalara uygun hale getirmiştir. Tablo 6.4 'de bu çalışmadaki uygulanan LSTM modelinin güçlü yönleri görülmektedir.

Tablo 6.4. LSTM modelinin güçlü yönlerinin uygulama bazlı değerlendirmesi.

Güçlü Yönler	Açıklama
Zaman Serisi Uyum Yeteneği	Geçmiş örneklerden öğrenerek gelecekteki saldırı desenlerini başarılı şekilde tahmin edebilir.
Dengelemeye Uygunluk	SMOTE ve class weighting teknikleriyle birlikte kullanıldığında overfitting'e düşmeden yüksek performans sağlar.
Basit ve Stabil Mimari	Düşük GPU kaynaklarıyla dahi yüksek doğruluk sağlayabilen, eğitimde stabil bir yapı sunar.
Düşük Hata Yayılmı	Confusion matrix analizlerine göre, benign ve DDoS sınıflarında çok düşük hata oranı üretmiştir.
Yorumlanabilirlik	Katmanlı ve sıralı doğası sayesinde saldırıların zaman içindeki davranışlarını anlaşılır biçimde modelleyebilir.

SHAP ve confusion matrix analizleri, modelin yorumlanabilirlik açısından da tatmin edici sonuçlar sunduğunu ortaya koymuştur. Tablo 6.5'te yer alan modelin bazı zayıf yönleri de dikkat çekicidir.

Özellikle Mirai-Greeths ve Mirai-Greips gibi içerik olarak benzeşen alt sınıflarda sınıflandırma hataları yüksek seviyede kalmıştır. Bu durum, LSTM'nin tek yönlü öğrenme doğası nedeniyle bazı örüntüleri yeterince ayırtıramadığını göstermektedir. Ayrıca, BiLSTM gibi optimizasyonlu yapılara göre eğitim süresi daha uzun olmuş (~311s/epoch), bu da modelin pratikteki kullanılabilirliğini sınırlamıştır.

Attention (dikkat) mekanizmaları içeren yapılarda olduğu gibi, LSTM mimarisi hangi zaman adımının model çıktısına ne kadar etki ettiğini doğrudan ortaya

koyamamaktadır. Dolayısıyla attention eksikliği, modelin karar mekanizmasının açıklanabilirliğini sınırlı kılmıştır. Ayrıca CNN gibi uzamsal özellik çıkarıcı bileşenler entegre edilmediğinde, LSTM'nin yalnızca zaman eksenli öğrenmeye dayalı yapısı karmaşık örüntülerde yetersiz kalabilmektedir.

Tablo 6.5. LSTM modelinin zayıf yönlerinin uygulama bazlı değerlendirmesi.

Zayıf Yönler	Açıklama
Mirai Alt Sınıf Ayrımında Zayıflık	"Mirai-Greeths" ve "Mirai-Greips" gibi benzer örüntülere sahip alt sınıflarda hatalı sınıflandırma oranı yüksektir.
Dengelemeye Uygunluk	SMOTE ve class weighting teknikleriyle birlikte kullanıldığında overfitting'e düşmeden yüksek performans sağlar.
Eğitim Süresinin Uzunluğu	BiLSTM gibi optimize yapılara kıyasla her epoch'ta daha fazla zaman gerektirir (~311s).
Tek Yönlü Zaman Akışı	Sadece geçmişten geleceğe bilgi akışı sağlayabildiği için çift yönlü örüntüleri tam olarak yakalayamaz.
Attention Mekanizması Eksikliği	Hangi zaman adımlarının kritik olduğu doğrudan çıkarılamaz; açıklanabilirlik düşer.
Hibrit Yapılara Göre Eksiklik	CNN gibi bileşenlerle birlikte kullanılmadığında, hem uzamsal hem zamansal örüntüleri eş zamanlı modelleme yetersiz kalır.

LSTM mimarisi özellikle sıralı verilerde güçlü bir aday olmakla birlikte; eğitim süresi, tek yönlü veri işleme yapısı ve dikkat mekanizması eksikliği gibi nedenlerle karmaşık alt sınıf ayrımlarında performans düşüklüğü gösterebilmektedir. Bu durum, daha gelişmiş mimarilerle birlikte çalıştığında kapatılabilecek potansiyel geliştirme alanlarını da işaret etmektedir.

6.2.2. BiLSTM

Bu tez çalışmasında uygulanan Bi-LSTM modeli, sıralı veri yapılarında ileri ve geri yönde bilgi akışı sağlayarak LSTM'e göre daha gelişmiş bir örüntü öğrenme yeteneği sunmuştur. Özellikle çift yönlü zaman işleme kabiliyeti sayesinde, geçmiş ve gelecek veri bağlamlarını birlikte dikkate alarak daha sağlam bir öğrenme yapısı oluşturmuştur. Bu yapı, Mirai-Greeths ve Mirai-Greips gibi benzer örüntülere sahip saldırı türlerinde daha düşük hata oranı üretmiş ve LSTM'e kıyasla daha başarılı bir ayırım gerçekleştirmiştir. Eğitim süresi bakımından da anlamlı bir performans avantajı sağlayan model, LSTM modeline göre her epoch'ta yaklaşık üç kat daha hızlı tamamlanmış (~89s/epoch), bu avantaj özellikle çalışılan veri setleri boyutu büyük olduğunda hesaplama verimliliğini artırmıştır.

Tablo 6.6. BiLSTM modelinin güçlü yönlerinin uygulama bazlı değerlendirmesi.

Güçlü Yönler	Açıklama
Çift Yönlü Zaman İşleme	BiLSTM, geçmiş ve geleceğe yönelik zaman örüntülerini birlikte dikkate alarak daha derin bağlam ilişkileri kurabilir.
Mirai Sınıflarında Daha Az Hata	LSTM'e kıyasla özellikle Mirai-Greeths ve Greips sınıflarında daha düşük hata oranı ile sınıflandırma yapmıştır.
Eğitim Süresi Avantajı	Tek yönlü LSTM yapısına göre daha hızlı eğitilmiş (~89s/epoch) ve hesaplama açısından daha verimli sonuçlar üretmiştir.
Genel Stabilité	Eğitim süreci boyunca doğruluk oranlarında ani dalgalanmalar gözlemlenmemiş, metrikler kararlı kalmıştır.
Zaman Serisi Anlamlandırma Yeteneği	Çift yönlü yapı sayesinde uzun vadeli bağımlılıkları daha doğru modelleyebilir.

Tablo 6.6'da görüldüğü üzere eğitim süreci boyunca metriklerde kararlılık gözlemlenmiş, ani doğruluk dalgalanmaları yaşanmamış ve genel olarak stabil bir

öğrenme süreci elde edilmiştir. Bu bağlamda Bi-LSTM, hem uzun vadeli bağımlılıkları doğru modelleme yeteneği hem de hesaplama süresi açısından dengeli bir çözüm sunmuştur.

Tablo 6.7’de görüldüğü üzere Bi-LSTM mimarisinin bazı zayıf yönleri de mevcuttur. Öncelikle, klasik LSTM gibi bu modelde de attention mekanizması doğrudan entegre edilmediği için, hangi zaman adımının model çıktısına ne kadar katkı sunduğu açıklanabilir biçimde takip edilememektedir. Bunun yanı sıra, CNN bileşeni olmadan kullanıldığında görsel örüntüleri ve lokal paternleri algılamada sınırlı kalmakta, bu da ağ trafiğindeki kısa süreli anomalileri tespit etme kapasitesini düşürebilmektedir.

Ayrıca, düşük frekanslı sınıflarda kullanılan verinin dengesizliği, modelin bu sınıflarda hata yapmasına sebep olmuş, sınıf dengesizliğine duyarlılık gözlemlenmiştir. Bi-LSTM’in çift yönlü doğası, öğrenme sırasında daha fazla bellek ve GPU tüketimi oluşturmuş, bu da modelin kaynak maliyetinin daha yüksek olmasına sebep olmuştur.

Tablo 6.7. BiLSTM modelinin zayıf yönlerinin uygulama bazlı değerlendirmesi.

Zayıf Yönler	Açıklama
Attention Mekanizması Eksikliği	Kritik zaman adımlarının model tarafından nasıl ağırlıklandırıldığı doğrudan gözlemlenememektedir.
Görsel Örüntüleri Algılamada Sınırlı	CNN entegrasyonu olmayan yapılar, ağ trafiğindeki lokal paternleri analiz etmekte zorlanabilir.
Sınıf Dengesizliğine Duyarlılık	Dengelenmemiş veri kullanıldığında düşük frekanslı sınıflarda LSTM gibi BiLSTM de hata yapabilmektedir.
Yüksek Bellek Tüketimi	Çift yönlü çalışması nedeniyle, LSTM’ye göre daha fazla RAM ve GPU belleği tüketir.
Ayrıştırıcı Özellik Belirginliği Eksikliği	Özellik önem derecelerinin açıkça yorumlanamaması, model kararlarının izlenebilirliğini azaltabilir.

Son olarak, modelin özellik bazlı ayırıştırma kapasitesi yeterince güçlü olmadığı için, hangi özneliliklerin ne derece etkili olduğu açıkça takip edilememekte ve bu da yorumlanabilirliği sınırlamaktadır. Genel olarak değerlendirildiğinde, Bi-LSTM

modeli doğruluk, öğrenme kararlılığı ve zaman serisi analiz yeteneği açısından güçlü bir aday olarak öne çıkarken; dikkat mekanizması eksikliği, sınıf dengesizliğine karşı duyarlılık ve açıklanabilirlik zayıflığı gibi alanlarda geliştirme potansiyeli taşımaktadır.

6.2.3. CNN-LSTM + attention

Bu tezde önerilen CNN-LSTM + Attention hibrit mimarisi, çok boyutlu saldırı desenlerini öğrenme ve modelleme kapasitesi açısından dikkat çekici bir yapı sunmuştur. Tablo 6.8’de modelin uygulama bazlı güçlü yönlerinin değerlendirilmesi yer almaktadır.

Tablo 6.8. CNN-LSTM + attention modelinin güçlü yönlerinin uygulama bazlı değerlendirilmesi.

Güçlü Yönler	Açıklama
Derin Özellik Öğrenimi	CNN katmanı, uzamsal örüntüleri; BiLSTM, zamansal bağımlılıkları; Attention ise dikkat (attention) tabanlı ilişkileri başarılı şekilde modelleyebilir.
Kompleks Saldırıların Anlamlılık	Karmaşık yapıya sahip saldırılar için daha yüksek öğrenme kapasitesi sunar. Özellikle veri seti içindeki geniş varyasyonlara karşı teorik olarak daha duyarlıdır.
Modüler Mimarinin Esnekliği	CNN, BiLSTM ve Attention katmanlarının birbirinden bağımsız şekilde optimize edilebilmesi, mimarinin yeniden uyarlanabilirliğini artırır.
Gelişmiş Temsil Gücü	Attention bileşeni sayesinde dikkat dağılımı üzerinden modelin hangi zaman adımlarına odaklandığı çıkarılabilir.
Teorik Olarak Üstünlük	Literatürdeki birçok çalışmada, Attention tabanlı hibrit sistemler en yüksek doğruluklara ulaşmaktadır.

CNN bileşeni sayesinde verinin uzamsal özellikleri başarılı şekilde çıkarılmış, LSTM katmanı zaman serisi bağımlılıklarını modellemiş ve Attention bileşeni de dikkat mekanizması üzerinden uzun vadeli bağlamsal ilişkileri yakalayarak derin bir örüntü öğrenimi sağlamıştır. Bu üçlü yapının birlikte kullanılması, modelin kompleks saldırı senaryolarında geniş varyasyonlara duyarlılık kazanmasını ve daha zengin temsiller

üretmesini mümkün kılmıştır. Ayrıca her bir bileşenin modüler yapıda optimize edilebilir olması, mimarinin yeniden yapılandırılabilirliğini ve uyarlanabilirliğini önemli ölçüde artırmıştır. Attention bileşeni ile birlikte gelen temsil gücü, modelin hangi zaman adımına ne kadar dikkat verdiğinin açıklanmasını sağlamış ve bu, derin öğrenme sistemlerinde sıklıkla karşılaşılan “kara kutu” sorununa karşı teorik bir şeffaflık sunmuştur. Nitekim literatürde birçok çalışma, Attention tabanlı mimarilerin doğruluk açısından üst düzey sonuçlara ulaştığını ortaya koymuştur.

Tablo 6.9. CNN-LSTM + attention modelinin zayıf yönlerinin uygulama bazı değerlendirilmesi.

Zayıf Yönler	Açıklama
Gerçek Performans Düşüklüğü	Bu çalışmada Mirai sınıflarında beklenen ayrımı sağlayamamış, genel doğruluk %88’de kalmıştır.
Aşırı Öğrenme (Overfitting)	Derin mimari yapısı nedeniyle, özellikle küçük veya dengesiz sınıflarda aşırı öğrenme eğilimi göstermiştir.
Yorumlanabilirlikte Karmaşıklık	Dikkat mekanizması teorik olarak açıklayıcı olsa da, mimari karmaşıklık nedeniyle çıktıların yorumlanması zordur.
Veri Hassasiyeti	Bu modelin performansı, önileme ve dengeleme stratejilerine aşırı derecede bağımlıdır; bu da transfer edilebilirliğini sınırlar.
Eğitim Süresi ve Kaynak Tüketimi	Hesaplama maliyeti yüksek ve eğitim süresi uzundur. GPU bellek kullanımı gerektirir.

Ancak tüm bu teorik avantajlara karşın, Tablo 6.9’da görüldüğü üzere modelin bazı zayıf yönleri uygulama aşamasında ortaya çıkmıştır. Öncelikle, model Mirai alt sınıflarında beklenen sınıf ayrımını sağlayamamış ve genel doğruluğu %88’in altında kalmıştır. Özellikle dikkat mekanizmasının güçlü olmasına rağmen, sınıflar arası örüntü benzerliğini yeterince ayırtıramadığı gözlemlenmiştir. Ayrıca modelin derin yapısı, aşırı öğrenme (overfitting) eğilimi göstererek bazı düşük frekanslı sınıflarda

hatalı genellemeler yapmıştır. Diğer modellere göre eğitim süreci fazlasıyla uzun sürmüş ve yüksek parametre sayısı nedeniyle GPU bellek tüketimi artmıştır.

Özellikle kaynak kısıtlı sistemlerde bu modelin uygulanabilirliği zayıflamaktadır. Attention bileşeni teorik olarak yorumlanabilirliği artırsa da, pratikte modelin çok katmanlı yapısı çıktılarının yorumlanmasını zorlaştırmakta ve bu da modelin karar süreçlerine dair anlaşılabilirlik sorunları yaratmaktadır. Son olarak, model veri hassasiyeti açısından da sınanmış; dengelenmemiş verilerle yapılan çalışmalarda performans düşüşü gözlemlenmiştir. Bu durum, yapının transfer edilebilirliğini ve farklı veri kümelerine genellenebilirliğini sınırlandırmaktadır.

Sonuç olarak, CNN-LSTM + Attention mimarisinin teorik potansiyeli güçlü olmasına rağmen bu tez çalışması özelinde karmaşık saldırı türlerinde istenilen ayırım netliği sağlanamamıştır. Yüksek kaynak ihtiyacı ve açıklanabilirlik sorunları gibi faktörler nedeniyle pratik sınırlamalar ortaya çıkmıştır.

6.3. Gerçek Hayatta Uygulanabilirlik

Bu tezde önerilen derin öğrenme tabanlı saldırı tespit modelleri, özellikle IoT ağlarında artan güvenlik risklerine karşı güçlü bir tespit mekanizması sunmayı hedeflemektedir.

Gerçek hayattaki uygulanabilirliği değerlendirmek için model mimarisinin donanım gereksinimleri, işlem süresi, hata toleransı, sınıf ayırım başarısı ve entegrasyon kolaylığı gibi çeşitli faktörler dikkate alınmıştır.

1. Donanım ve Altyapı Uyumu: LSTM ve BiLSTM gibi daha sade mimariler, orta düzeyde GPU kaynaklarına sahip sistemlerde dahi yüksek doğrulukla çalışabilir. Bu nedenle küçük ve orta ölçekli IoT ağları için oldukça uygundur. CNN-BiLSTM + Attention gibi daha karmaşık yapılar ise yüksek işlem gücü ve RAM gerektirir; bu da bulut tabanlı ya da edge computing destekli platformlara ihtiyaç doğurur.

2. Gerçek Zamanlı İzleme Potansiyeli: Özellikle BiLSTM modeli, hızlı eğitim süresi ve yüksek doğruluğu sayesinde gerçek zamanlı saldırı tespiti için umut vaat etmektedir. Ancak inference süresinin de düşük olması gerektiğinden, bu modellerin hafif versiyonları veya quantization teknikleri kullanılarak üretim ortamına entegre edilmesi önerilir.

3. Sınıf Dengesizliğine Dayanıklılık: IoT ağlarında bazı saldırı türlerinin daha nadir görülmesi gerçek hayatta da geçerlidir. Bu bağlamda SMOTE, class weighting gibi

tekniklerle desteklenen modeller, bu dengesizliklerle başa çıkma konusunda başarı sağlamış ve üretim ortamında da sürdürülebilirlik göstermektedir.

4. Kolay Entegrasyon ve Eğitim Tekrarı: Güvenlik uzmanları tarafından anlaşılması ve sistemlere adapte edilmesi, LSTM temelli modellerin açıklanabilirliği yüksek olduğu için daha kolaydır. Ayrıca bu modellerin eğitimi yeniden yapılabilir ve ağ trafiği değiştikçe güncellenebilir yapıdadır.

5. Genişletilebilirlik ve Uyarlanabilirlik: CNN-BiLSTM + Attention mimarisi, daha büyük ve değişken veri akışlarının analizine olanak tanır. Bu tür mimariler, 5G, smart grid ve endüstriyel otomasyon gibi yüksek hacimli ve kompleks sistemlerde etkili şekilde kullanılabilir. Fakat yüksek kaynak maliyeti ve mimari karmaşıklık nedeniyle uygulanabilirlikleri daha büyük ölçekli kurumlarla sınırlı kalabilir.

6. Güvenlik Operasyon Merkezleri (SOC) ile Uyum: Bu tezde önerilen modeller, anomalilerin önceden tahmin edilmesi, güvenlik olaylarının sınıflandırılması ve uyarı sistemlerinin optimize edilmesi gibi operasyonel görevler için destekleyici bir katman sunabilir. Özellikle LSTM ve BiLSTM tabanlı modeller, SOC ekiplerinin veriyle daha etkili çalışmasına yardımcı olacak hızlı ve güvenilir tahminler üretebilir.

Sonuç olarak, önerilen modeller gerçek hayattaki IoT ortamlarına entegrasyon için yüksek potansiyele sahiptir. Donanım koşulları, uygulama senaryosu ve kullanılacak veri çeşitliliği göz önünde bulundurularak doğru modelin seçilmesi, saldırı tespiti sistemlerinin başarısını önemli ölçüde artıracaktır.



7. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, IoT tabanlı ağlarda meydana gelen Mirai ve DDoS kaynaklı saldırıların tespiti amacıyla derin öğrenme tabanlı LSTM, BiLSTM ve CNN-LSTM + Attention mimarileri önerilmiştir. Önerilen bu modeller, CIC-IoT2023 veri seti üzerinde detaylı şekilde test edilmiştir. Elde edilen sonuçlar literatürde yer alan benzer çalışmalarla kıyaslanarak analiz edilmiştir. Modelin eğitimi sırasında veri ön işleme ve özellik seçimi aşamalarında dikkatli bir strateji izlenmiş; sınıf dengesizliği sorununa karşı örnekleme teknikleri ve özellik indirgeme yöntemleri entegre edilmiştir. Tez boyunca uygulanan LSTM, BiLSTM ve CNN-LSTM + Attention mimarileri farklı metrikler çerçevesinde karşılaştırılmış, özellikle BiLSTM modeli doğruluk ve eğitim süresi açısından en verimli çözüm olarak öne çıkmıştır. Öte yandan, karmaşık yapısı nedeniyle Attention destekli mimariler bazı alt sınıflarda düşük ayırım başarısı göstermiştir. Elde edilen bulgular, sade ama etkili modellerin, yüksek parametrelili yapılara kıyasla pratikte daha kararlı ve uygulanabilir olduğunu tespit etmiştir. Bu bağlamda çalışma, hem akademik anlamda literatüre katkı sağlamakta hem de pratikte uygulanabilecek bir IDS altyapısı sunmaktadır.

7.1. Sonuç

Bu çalışmada uygulanan farklı derin öğrenme mimarileri sonucunda, IoT ağlarında meydana gelen saldırıların yüksek doğruluk oranlarıyla tespit edilebildiği ortaya konmuştur. Özellikle BiLSTM mimarisi, hem işlem süresi hem de doğruluk oranı açısından en dengeli sonucu sunmuştur. BiLSTM modeli, %98'e varan doğrulukla özellikle DDoS ve Mirai türü saldırıları ayırt edebilmiş, sınıflar arası karışıklığı minimumda tutmuştur. CNN bileşeninin dahil edilmesi uzamsal örüntüleri çıkarmada katkı sağlamış ancak bu katkı, bazı durumlarda eğitim süresini artırmıştır.

Transformer mimarisi teorik olarak dikkat(attention) mekanizması sayesinde daha ileri düzey örüntüleri öğrenme potansiyeline sahip olmasına rağmen bu çalışmadaki sonuçlar incelendiğinde, pratik uygulamalarda yüksek parametre sayısının hata eğilimine sebep olabileceği görülmektedir. Özellikle alt sınıflar arasında — örneğin Mirai-Greeths ile Mirai-Greips gibi — ciddi benzerlikler olduğunda, karmaşık

modeller sınıf ayırt etmede zorlanabilmiştir. Yapılan çalışmalar, model performansının yalnızca mimari seçimine bağlı olmadığını; veri dengesi, öznitelik seçimi ve hiperparametre ayarlarının da en az model kadar belirleyici olduğunu ortaya koymuştur. Özellikle χ^2 yöntemiyle seçilen özniteliklerin kullanıldığı deneylerde, model daha az overfitting eğilimi göstermiştir. ROC eğrileri, doğruluk-f1 dengesi ve konfüzyon matrisi analizleri de bu bulguları doğrular niteliktedir. Bu tezde önerilen sistemin özellikle edge tabanlı IoT güvenlik çözümlerinde uygulanabilir olduğu, hız ve doğruluk bakımından tatmin edici sonuçlar sunduğu söylenebilir. Elde edilen sonuçlar literatürle karşılaştırıldığında, önerilen çözümün rekabetçi ve bazı açılardan üstün bir performans sergilediği gözlemlenmektedir.

7.2. Gelecek Çalışmalar İçin Öneriler

Bu tez çalışması, IoT ağlarında saldırı tespitine yönelik derin öğrenme tabanlı bir sistemin tasarımı ve değerlendirmesini kapsamaktadır. Ancak elde edilen bulgular ışığında bazı geliştirme alanları ve geleceğe dönük araştırma konuları öne çıkmaktadır. Öncelikle, kullanılan LSTM temelli mimarinin belirli saldırı türlerinde yüksek başarı sağlamasına rağmen, bazı sınıflarda özellikle Mirai'nin varyantlarında hataya daha yatkın olduğu görülmüştür. Bu durum, zaman serisi verilerindeki karmaşıklığın tek başına LSTM ile yeterince öğrenilemediğini düşündürmektedir. Dolayısıyla ileriki çalışmalarda, attention mekanizmasıyla desteklenen Attentiontemelli yaklaşımlara daha fazla ağırlık verilmesi önerilmektedir (Akuthota & Bhargava, 2025).

İkinci olarak, CIC-IoT2023 veri seti kapsamlı bir içeriğe sahip olsa da, sınıflar arası örnek dengesizliği hâlâ model başarısını etkileyen bir faktördür. Özellikle nadir görülen saldırı türlerinde modelin recall skorlarının düşük kalması, sınıf ağırlıklarının veya SMOTE gibi tekniklerin daha rafine biçimde uygulanması gerektiğini göstermektedir (Nazir et al., 2024). Bunun dışında, kullanılan özellik seçimi yöntemleri (χ^2 , RFE, Boruta gibi) belirli özniteliklerin etkisini ortaya koysa da, zamanla değişen ağ davranışlarına karşı güncel öznitelik çıkarımı yöntemleriyle desteklenmesi faydalı olabilir. Modelin saha ortamında test edilmemiş olması, uygulamaya dönük doğrulama aşamasını eksik bırakmaktadır. Bu nedenle, ilerideki araştırmaların, önerilen IDS sistemini edge computing cihazları veya gerçek IoT donanımları üzerinde test etmesi, sistemin dayanıklılığını ve gerçek zamanlı başarımını değerlendirme imkânı sunacaktır (Sinha et al., 2023). Son olarak, bu

çalışmada model eğitimi sabit hiperparametre aralıklarıyla yapılmış ve yalnızca Optuna algoritması tercih edilmiştir.

Gelecek araştırmalarda, Bayesian optimization, grid search, population-based training (PBT) gibi farklı hiperparametre arama tekniklerinin karşılaştırılması, model performansını daha ileriye taşıyabilir. Bu çalışma sağlam bir temel oluştursa da, hem mimari düzeyde hem de uygulama tarafında geliştirilebilecek birçok yön mevcuttur. Özellikle gerçek zamanlı sistemler, daha dengeli veri kümeleri ve ileri seviye optimizasyon algoritmaları bu sistemlerin daha geniş çapta kullanılabilirliğini artıracaktır.

7.3. Genel Değerlendirme ve Sonuçların Önemi

Bu tezde önerilen saldırı tespit sistemi, sadece akademik bir performans göstergesi olmanın ötesinde, gerçek dünya IoT sistemleri için uygulanabilir bir altyapı sunmaktadır. Gerek kullanılan CIC-IoT2023 veri setinin kapsamlılığı, gerekse mimarilerin dikkatle optimize edilmesi, çalışmanın pratik sahada test edilebilecek bir noktaya ulaştığını göstermektedir.

Özellikle BiLSTM mimarisi hem zaman hem de doğruluk açısından dengeli bir yapı ortaya koyarken, LSTM mimarisi de BiLSTM'e oldukça yakın sonuçlar vermiştir. CNN ve Attention bileşenleri eklenen modellerde ise daha karmaşık saldırı davranışlarının tanımlanabilirliği artırılmıştır. Saldırı tespit sistemlerinin günümüzde daha da önem kazanmasının temel nedeni, IoT ekosisteminin hızla yayılması ve güvenlik açıklarının sayıca artmasıdır.

Bu bağlamda tezde sunulan çözüm, sadece belirli veri kümelerine değil, genişleyen IoT dünyasının geneline adapte edilebilecek modüler bir yapıya sahiptir. Ayrıca yapılan değerlendirmelerde sadece başarı oranları değil; sınıf bazlı ayrımlar, eğitim süreleri ve overfitting eğilimleri de detaylı olarak incelenmiştir. Bundan sonraki süreçte, önerilen modelin saha ortamlarında, gerçek IoT cihazlarıyla test edilmesi ve edge computing sistemlerine entegre edilerek çalıştırılması hedeflenmelidir. Böylece yalnızca teorik değil, uygulamalı güvenlik çözümlerinin bir parçası hâline gelmesi sağlanabilir. Bu çalışma, bu yöndeki yapılacak gelecek çalışmalar için yüksek başarı performanslarına sahip öneriler sunmaktadır.



KAYNAKLAR

- Akuthota, K., & Bhargava, B. (2025). Transformer-Based Intrusion Detection for IoT Networks. *Proceedings of the IEEE International Conference on Communications*.
- Chen, Z., Chen, D., Zhang, X., Yuan, Z., & Cheng, X. (2022). Learning Graph Structures With Attention for Multivariate Time-Series Anomaly Detection in IoT. *IEEE Internet of Things Journal*, 9(12), 9179–9189.
- Dai, W., Zhang, Z., Chen, W., & Liu, J. (2024). TranAD: Deep Attention Networks for Anomaly Detection in Multivariate Time Series Data. *arXiv preprint arXiv:2201.07284*.
- Elsayed, A., Elazab, A., Hamed, H. N. A., & Mahmoud, M. (2021). Deep learning-based approach for intrusion detection in IoT networks. *Computer Networks*, 197, 108271. <https://doi.org/10.1016/j.comnet.2021.108271>
- Elsayed, M. S., Le-Khac, N. A., Jahromi, H. Z., & Jurcut, A. D. (2021). A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES 2021)*. <https://doi.org/10.1145/3465481.3469190>
- Elsayed, M. S., et al. (2021). CNN-LSTM based hybrid IDS model for SDNs. *ARES 2021 Conference Proceedings*. <https://doi.org/10.1145/3465481.3469190>
- Gueriani, A., Kheddar, H., & Mazari, A. C. (2024). Enhancing IoT Security with CNN and LSTM-Based Intrusion Detection Systems. *6th International Conference on Pattern Analysis and Intelligent Systems (PAIS)*.
- Halbouni, A., Basheer, S., & Alhawari, S. (2022). CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. *IJITCS*, 16(4), 48–57.
- Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022). CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. *IEEE Access*, 10, 99837–99845.
- Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network: A deep learning approach. *Procedia Computer Science*, 132, 679–688. <https://doi.org/10.1016/j.procs.2018.05.069>
- Jony, A. I., & Arnob, A. K. B. (2024). Securing the Internet of Things: Evaluating Machine Learning Algorithms for Detecting IoT Cyberattacks Using CIC-IoT2023 Dataset. *IJITCS*, 16(4), 11–22.
- Jony, A. I., & Arnob, A. K. B. (2024). Securing the Internet of Things: Evaluating Machine Learning Algorithms for Detecting IoT Cyberattacks Using CIC-IoT2023 Dataset. *I.J. Information Technology and Computer Science*, 16(4), 56–65.

- Kim, J., Kang, H., & Kang, P. (2023). Engineering Applications of Artificial Intelligence, 120, 105964.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324. <https://doi.org/10.1109/5.726791>
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Nazir, A., He, J., Zhu, N., Qureshi, S. S., Qureshi, S. U., Ullah, F., Wajahat, A., & Pathan, M. S. (2024). A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem. *Ain Shams Engineering Journal*, 15, 102777.
- Nazir, A., Hussain, F., & Anwar, Z. (2024). A deep learning-based novel hybrid CNN-LSTM architecture for efficient detection of threats in the IoT ecosystem. *Scientific Reports*, 14, 9842.
- Ogobuchi, C., Ayop, S. F., & Mohammed, M. N. (2024). MLCFSM: Multi-Layer Communication Flow Security Model for IoT Networks. *Elsevier Computers & Security*, 132, 103122.
- Ogobuchi, D. O., Rodrigues, D. Z., & Klenschmidt, J. H. (2024). MLCFSM: Multi-Layer Common Feature Selection Methods for Robust Intrusion Detection in IoT Network. 2024 IEEE NIGERCON.
- Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, 99, 107810. <https://doi.org/10.1016/j.compeleceng.2022.107810>
- Saba, T., Sadad, T., Rehman, A., Mehmood, Z., & Javaid, Q. (2021). Intrusion detection system through advanced machine learning for the Internet of Things networks. *IT Professional*, 23(2), 58–64. <https://doi.org/10.1109/MITP.2021.3057342>
- Saba, T. et al. (2022). Deep learning-based intrusion detection in smart city IoT using CNN. *Computers & Electrical Engineering*, 99, 107810.
- Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*. <https://arxiv.org/abs/1402.1128>
- Sharma, A., Mansotra, V., & Singh, K. (2023). Detection of Mirai Botnet Attacks on IoT Devices Using Deep Learning. *Journal of Scientific Research and Technology*, 1(6), 174–187.
- Shen, S., et al. (2023). DeepMAD: Mathematical Architecture Design for Deep Convolutional Neural Networks. *CVPR* 2023. <https://doi.org/10.1109/CVPR52729.2023.00321>
- Sinha, P., Singh, M., & Ghosh, S. (2025). A high performance hybrid LSTM-CNN secure architecture for IoT environments using deep learning. *Energy*, 291, 120431.
- Sun, P., Liu, P., Li, Q., et al. (2020). DL-IDS: Extracting features using CNN-LSTM hybrid network. *Security and Communication Networks*, 2020, 8890306.

- Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., & Chen, J. (2020). DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System. *Security and Communication Networks*, 2020, 8890306.
- Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Transformer-based Anomaly Detection in Time-Series Data. *arXiv preprint arXiv:2201.07284*.
- Tuli, S., Mahmud, R., Tuli, S., & Buyya, R. (2022). TranAD: Transformer-based Anomaly Detection for Industrial Multivariate Time Series. *arXiv preprint arXiv:2201.07284*.
- Udurume, M., Shakhov, V., & Koo, I. (2024). Comparative Analysis of Deep Convolutional Neural Network—Bidirectional Long Short-Term Memory and Machine Learning Methods in Intrusion Detection Systems. *Applied Sciences*, 14(6967).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30. https://papers.nips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Wang, H. (2023). A Network Intrusion Security Detection Method Using BiLSTM-CNN in Big Data Environment. *Journal of Information Processing Systems*, 19(5), 688–701.
- Xu, C., Liu, H., Xu, M., Shen, J., & Yang, Y. (2022). Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. *Advances in Neural Information Processing Systems*, 35.
- Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. *ICLR 2022 Conference Proceedings*.
- Zhao, J., Liu, Y., Zhang, Q., & Zheng, X. (2023). CNN-AttBiLSTM mechanism: A DDoS attack detection method based on attention mechanism and CNN-BiLSTM. *IEEE Access*, 11, 136308–136320. <https://doi.org/10.1109/ACCESS.2023.3334916>
- Zrira, N., Kamal-Idrissi, A., Farssi, R., & Khan, H. A. (2024). Time series prediction of sea surface temperature based on BiLSTM model with attention mechanism. *Journal of Sea Research*, 198, 102472. <https://doi.org/10.1016/j.seares.2024.102472>



ÖZGEÇMİŞ

Ad-Soyad : Tuğba ULUSOY

ÖĞRENİM DURUMU:

- Lisans** : 2021, Sakarya Üniversitesi, Bilgisayar Ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği
- Yüksek Lisans** : 2025, Sakarya Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Bilim Dalı, Tezli Yüksek Lisans

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2022 – (...) yıllarında bilgisayar mühendisi olarak çalışmakta.

TEZDEN TÜRETİLEN ESERLER:

- Ulusoy T., (2025, 22–23 Mayıs). IoT Ağlarında Yeni Bir Saldırı Tespit Sistemi. *II. Uluslararası Veri Bilimi ve Bilgi Teknolojileri Kongresi (INFTEC-2025)*, Budapeşte, Macaristan. (Çevrim içi sunum)