KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

NESNELERİN İNTERNETİNDE MAKİNE ÖĞRENMESİ TABANLI SALDIRI TESPİTİ İÇİN GELİŞMİŞ ÖZNİTELİK MÜHENDİSLİĞİ VE VERİ DENGESİZLİĞİ ÇÖZÜMLERİNİN TASARIMI VE ANALİZİ

KAZIM KIVANÇ EREN

KOCAELİ 2026

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

DOKTORA TEZİ

NESNELERİN İNTERNETİNDE MAKİNE ÖĞRENMESİ TABANLI SALDIRI TESPİTİ İÇİN GELİŞMİŞ ÖZNİTELİK MÜHENDİSLİĞİ VE VERİ DENGESİZLİĞİ ÇÖZÜMLERİNİN TASARIMI VE ANALİZİ

KAZIM KIVANÇ EREN

Prof. Dr. Kerem KÜÇÜK **Danışman, Kocaeli Üniversitesi**	.....
Doç. Dr. Orhan AKBULUT **Jüri Üyesi, Kocaeli Üniversitesi**	.....
Doç. Dr. Mehmet Zeki KONYAR **Jüri Üyesi, Kocaeli Üniversitesi**	.....
Prof. Dr. Ali ÇALHAN **Jüri Üyesi, Düzce Üniversitesi**	.....
Doç. Dr. Fatih ÖZYURT **Jüri Üyesi, Fırat Üniversitesi**	.....

Tezin Savunulduğu Tarih: 23.01.2026

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez/proje çalışmasında,

- Bu tezin/projenin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamasında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmamın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin/projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez/proje çalışması olarak sunmadığımı,

beyan ederim.

☐ Bu tez/proje çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☑ Bu tez/proje çalışması kapsamında üretilen veri ve bilgiler TÜBİTAK tarafından 124E307 no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

.....

Kazım Kıvanç EREN

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin/projemin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin/projemin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanımı bana ait olacaktır.
Tezin/projenin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin/projenin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.
Yükseköğretim kurulu tarafından yayınlanan ***"Lisanüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.
☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin/projemin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.
☑ Tezim/projem ile ilgili gizlilik kararı verilmemiştir.

.....

Kazım Kıvanç EREN

ÖNSÖZ VE TEŞEKKÜR

Bu doktora tezi, Nesnelerin İnterneti (IoT) ortamlarında güvenliğin sağlanmasına yönelik olarak makine öğrenmesi ve derin öğrenme temelli saldırı tespit sistemlerinin tasarımı, geliştirilmesi ve kapsamlı biçimde değerlendirilmesini amaçlamaktadır. Çalışma kapsamında, klasik makine öğrenmesi yaklaşımlarının uygun öznitelik mühendisliği ve yeni öznitelik üretim algoritmalarıyla derin öğrenme modelleriyle rekabet edebilecek düzeye taşınabileceği gösterilmiş; dağılım temelli bir yeniden örnekleme yöntemi olan KLSampler algoritması ile dengesiz veri problemlerine yenilikçi bir çözüm sunulmuştur. Ayrıca, farklı IoT veri setleri arasında yayılım kayması ve yönelim kayması etkileri analiz edilerek, çok ortamlı ve küresel ölçekte genellenebilir bir IoT saldırı tespit sistemi oluşturulabilirliğine yönelik kapsamlı deneysel bulgular elde edilmiştir. Bu yönüyle tez, yalnızca yüksek başarımlı modeller önermekle kalmayıp, aynı zamanda güvenilir, açıklanabilir ve sürdürülebilir IoT-IDS mimarilerinin nasıl inşa edilebileceğine dair bütüncül bir metodolojik çerçeve sunmayı hedeflemektedir.

Bu çalışma, uzun ve yoğun bir araştırma sürecinin, sayısız deneyin, başarısız denemelerin, yeniden kurulan modellerin ve sürekli sorgulanan varsayımların bir ürünüdür. Akademik açıdan olduğu kadar kişisel gelişim açısından da öğretici olan bu süreçte, bilimin sabır, tutarlılık ve eleştirel düşünce gerektiren doğası bir kez daha tecrübe edilmiştir. Tezin ortaya koyduğu sonuçların, IoT güvenliği alanında çalışan araştırmacılara, özellikle öznitelik mühendisliği, veri dengesizliği ve dağılım kayması gibi temel problemlere odaklanan çalışmalara katkı sağlaması ve gelecekte geliştirilecek daha ölçeklenebilir ve genellenebilir saldırı tespit sistemlerine ışık tutması en büyük temennimdir.

Tez çalışmam boyunca, hem akademik bilgisiyle hem de insanlığıyla bana kattıklarından dolayı, danışmanım Prof. Dr. Kerem Küçük'e teşekkürü bir borç bilirim. Hiçbir fedakarlıktan kaçınmayarak, akademik tecrübesini bilgi birikimini, her şeyden önemlisi değerli zamanını benimle paylaşarak, iyi niyeti ve anlayışıyla, değerli destekleriyle bu tez çalışmasının tamamlanabilmesini sağlamıştır. Bu uzun, yorucu ama hem keyifli hem de öğretici maceramda, doktora sürecimin her anını benimle paylaşan, destekleriyle ve en az benim kadar bu çalışmanın çıkmasına öncülük eden, annem Selma Eren ve babam Eşref Eren, ablam Gizem Eren Uysal ile eniştem Yasin Uysal'a teşekkürlerin ve minnetin en büyüğünü sunuyorum. Doktora sürecimin başlarında hayatıma giren, motivasyonumu yükselten, neşe kaynağım, yeğenim Gülce Uysal'a da teşekkür ederim. Bana hem akademik zenginlikleriyle ufkumu açan, kendimi geliştirmeme sebep olan, hem de arkadaşlıkları sayesinde geçirdiğim zamanın kalitesini arttıran, çok değerli arkadaşlarım Can Özbey, Dr. Esra Çınar, ve Dr. Olympia M. Hardy'ye de teşekkürü bir borç bilirim.

Ocak - 2026 Kazım Kıvanç EREN

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
ÖNSÖZ VE TEŞEKKÜR iii
İÇİNDEKİLER iv
ŞEKİLLER DİZİNİ vi
TABLOLAR DİZİNİ vii
SİMGELER VE KISALTMALAR DİZİNİ viii
ÖZET xi
ABSTRACT xii
1. GİRİŞ 1
2. IOT SALDIRI TESPİT SİSTEMLERİNE BAKIŞ 4
2.1. Etki Şeklinde Göre Sınıflandırma 4
2.1.1. Pasif Saldırılar 4
2.1.2. Aktif Saldırılar 5
2.2. Kaynağına Göre Sınıflandırma 6
2.2.1. Dış Saldırılar 6
2.2.2. İç Saldırılar 6
2.2.3. Kaynak Karmaşası 7
2.3. Mimari Katmanlara Göre Kategorizasyon 7
2.3.1. Algılama (Perception/Physical) Katman Saldırıları 7
2.3.2. Ağ Katmanı Saldırıları 8
2.3.3. Uygulama Katmanı Saldırıları 9
2.4. IoT Saldırı Tespit Sistemleri (IoT-IDS) 10
2.4.1. Makine Öğrenmesi ve Derin Öğrenmeye Dayalı Sistemler 11
2.4.1.1. Geleneksel Makine Öğrenmesi Yaklaşımları 12
2.4.1.2. Derin Öğrenme Temelli Yaklaşımlar 13
2.4.1.3. Hibrit Yaklaşımlar 13
3. ARAŞTIRMA SORULARI 15
3.1. ML ve DL Tabanlı Sistemlerin Detaylı İncelemesi 15
3.2. Araştırma Sorusu 1 20
3.3. Araştırma Sorusu 2 21
3.4. Araştırma Sorusu 3 23
4. IOT AĞLARI İÇİN GELİŞMİŞ BİR SALDIRI TESPİT SİSTEMİ 24
4.1. Uygun Veri Setinin Seçilmesi 24
4.2. Eğitim ve Test Verisinin Hazırlanması 31
4.3. Kayıp Verilerin Analizi 36
4.4. Sabit ve Kısmi-Sabit Öznitelikler 39
4.5. Kategorik Verilerin İşlenmesi 40
4.6. Çoklu Korelasyon Analizi 42
4.7. İteratif Bir Öznitelik Üretim Algoritması 47
4.8. Model Eğitimi 51
4.9. Sonuçlar 56
4.9.1. Orjinal ve Üretilmiş Özniteliklerin Karşılaştırılması 56
4.9.2. Literatürdeki Diğer Yöntemlerle Karşılaştırma 59

4.10. Sonuçlar 63
5. AĞAÇ TABANLI BİR ÖZNİTELİK ÜRETME ALGORİTMASI 65
5.1. Ağaç Tabanlı Yorumlanabilir Bir Öznitelik Üretme Algoritması 65
5.2. Model Eğitimi 72
5.2.1. Orjinal ve Üretilmiş Özniteliklerin Kıyaslanması 73
5.2.2. Karışıklık Matrisi Analizi 77
5.2.3. Üretilmiş Özniteliklerin Model Perfomansına Etkisi 79
5.2.4. Literatürdeki Diğer Yöntemlerle Karşılaştırma 82
5.3. Hesaplama Karmaşıklığı 88
5.4. Sonuçlar 90
6. IOT VERİ SETLERİNDE DENGESİZ DAĞILIM PROBLEMİ 92
6.1. Ağ Saldırı Veri Setlerinin İncelemesi 94
6.2. KLSampler Algoritması 97
6.2.1. Algoritmanın Betimleyici Açıklaması 97
6.2.2. KLSampler Sözde Kodu 101
6.3. KLSampler Algoritmasının Karmaşıklık Analizi 106
6.4. Deneysel Sonuçlar 108
6.4.1. Çeşitli Veri Setlerinde Alınan Sonuçlar 108
6.4.2. KLSampler'ın Ağ Saldırı Verisetlerinde Test Edilmesi 118
6.5. Sonuçlar 123
7. GLOBAL BİR IOT-IDS SİSTEMİ OLUŞTURMA ÇALIŞMALARI 124
7.1. Farklı Veri Setlerinin Standardizasyonu 124
7.2. Veri Ön İşleme Aşamaları 125
7.2.1. Sabit ve Kısmi-Sabit Öznitelikler 127
7.2.2. Uygun Saldırı Grubunun Seçilmesi 128
7.2.3. Veri Setleri Arası Öznitelik Yayılımı 129
7.2.4. Eğitim-Test Kombinasyonlarının Oluşturulması 134
7.2.5. Çoklu Korelasyon Analizi 134
7.2.6. Özniteliklerin Tekil Analizi 137
7.3. Yayılım Kayması Analizi 150
7.3.1. Yayılım Kayması Analizi İçin Kullanılan Yöntemler 152
7.3.2. Yayılım Kayması Analizleri 156
7.4. Geleneksel Yöntemler için Sonuçlar 166
7.5. Derin Öğrenme için Sonuçlar 171
7.6. Sonuçlar 175
8. SONUÇLAR VE ÖNERİLER 177
8.1. Tez Çıkarımları 177
8.2. Araştırma Sorularının Cevapları 179
KAYNAKLAR 182
EKLER 195
KİŞİSEL YAYIN VE ESERLER 212
ÖZGEÇMİŞ 213

ŞEKİLLER DİZİNİ

Şekil 4.1. DNS İndikatör Değerleri için Sınıf Dağılımları 38
Şekil 4.2. Üretilmiş Öznitelikler için Karışıklık Matrisi 57
Şekil 5.1. EBLR Algortimasının İş Akışı 66
Şekil 5.2. Karışıklık Matrisleri (Orijinal + Üretilmiş Öznitelikler) 78
Şekil 5.3. En Önemli 20 Öznitelik 80
Şekil 6.1. KL-Sampler Algoritmasına Ait İş Akışı 99
Şekil 6.2. Churn Veri Seti için KDE Kıyaslamaları 113
Şekil 6.3. Adult Veri Seti için KDE Kıyaslamaları 114
Şekil 6.4. Bank Marketing Veri Seti için KDE Kıyaslamaları 115
Şekil 6.5. Online Shoppers için KDE Kıyaslamaları 117
Şekil 6.6. KLSampler İterasyon Sonuçları 122
Şekil 7.1. Eğitim Kombinasyonları için Flow Duration Dağılımları 138
Şekil 7.2. BoT-UNSW ve ToN_IoT için Yayılım Metrikleri 158
Şekil 7.3. BoT-UNSW ve ToN_IoT için Histogramlar 161
Şekil 7.4. BoT-UNSW ve ToN_IoT için ECDF Grafikleri 162
Şekil 7.5. BoT-UNSW ve ToN_IoT için KDE Grafikleri 163
Şekil 7.6. BoT-UNSW Eğitim ve ToN_IoT için KDE Farkları 164
Şekil A.1. ToN-UNSW ve BoT_IoT için Yayılım Metrikleri 197
Şekil A.2. ToN-UNSW ve BoT_IoT için Histogramlar 198
Şekil A.3. ToN-UNSW ve BoT_IoT için ECDF Grafikleri 199
Şekil A.4. ToN-UNSW ve BoT_IoT için KDE Grafikleri 200
Şekil A.5. ToN-UNSW ve BoT_IoT için KDE Farkları 201
Şekil A.6. BoT-ToN ve UNSW-NB15 için Yayılım Metrikleri 203
Şekil A.7. BoT-ToN ve UNSW-NB15 için Histogramlar 204
Şekil A.8. BoT-ToN ve UNSW-NB15 için ECDF Grafikleri 205
Şekil A.9. BoT-ToN ve UNSW-NB15 için KDE Grafikleri 206
Şekil A.10. BoT-ToN ve UNSW-NB15 için KDE Farkları 207
Şekil A.11. BoT-ToN Eğitim için Katışklık Matrisleri 209
Şekil A.12. BoT-UNSW Eğitim için Katışklık Matrisleri 210
Şekil A.13. ToN-UNSW Eğitim için Katışklık Matrisleri 211

TABLOLAR DİZİNİ

Tablo 3.1. Literatürde kullanılan IDS modellerinin karşılaştırılması 16
Tablo 3.2. Literatürdeki IoT-IDS Çalışmalarının Karşılaştırması 18
Tablo 4.1. Çalışmada Değerlendirilen Güncel IoT Veri Setleri 26
Tablo 4.2. Bağlantı Öznitelikleri 28
Tablo 4.3. İstatistiksel Öznitelikler 29
Tablo 4.4. Kullanıcı Merkezli Grup Öznitelikleri 29
Tablo 4.5. İhlal Öznitelikleri 30
Tablo 4.6. Etiket Öznitelikleri 30
Tablo 4.7. Örnek IP-Port Kayıtları 33
Tablo 4.8. Yüksek Kayıplı Öznitelikler için Hedef Korelasyonlar 37
Tablo 4.9. Orta Düzeyde Kayıp Özniteliklerin Hedef Korelasyonu 38
Tablo 4.10. Protokol Türüne Göre Etiket Alanının Dağılımı 40
Tablo 4.11. Bağlantı Durunlarının Sınıflara Göre Dağılım Oranları 42
Tablo 4.12. Yüksek Korelasyon Gösteren Öznitelik Grupları 47
Tablo 4.13. Eğitim Veri Seti İçin Sınıf Başına Düşen Örnek Sayıları 52
Tablo 4.14. Rastgele Orman Modeli için Hiperparametre Uzayı 55
Tablo 4.15. Test Verisi Üzerinde Alınan Sonuçlar 58
Tablo 4.16. Diğer Yöntemler ile Karşılaştırmalı Sonuçlar 62
Tablo 5.1. Önerilen Sistem İçin Kullanılan Öznitelik Listesi 70
Tablo 5.2. Elde Edilen Sonuçlar 74
Tablo 5.3. Diğer Yöntemler ile Karşılaştırmalı Sonuçlar 85
Tablo 6.1. Veri Setlerinde Saldırı Türlerinin Dağılımı 95
Tablo 6.2. Veri Kümeleri İçin Yeniden Örnekleme İstatistikleri 110
Tablo 6.3. Seçili saldırı sınıflarının veri setlerine göre dağılımı 120
Tablo 6.4. KLSampler Sonrası Sınıf Dağılımları 121
Tablo 7.1. NetFlow v2 Öznitelikleri 126
Tablo 7.2. Veri Setlerinde Saldırı Türlerinin Dağılımı 128
Tablo 7.3. Öznitelik Seçim Kararları 133
Tablo 7.4. Veri Setleri Arası Eğitim ve Test Senaryoları 134
Tablo 7.5. DURATION_IN Özniteliğine ait İstatistikler 139
Tablo 7.6. NUM_PKTS_UP_TO_128_BYTES İstatistikleri 146
Tablo 7.7. Yayılım Kayması Ölçütlerinin Karşılaştırılması 155
Tablo 7.8. Yayılım Şiddet Düzeyleri için Eşik Değerler 156
Tablo 7.9. Öznitelik Düzeyinde Yayılım Metrikleri 157
Tablo 7.10. ExtraTrees için Hiperparametre Arama Uzayı 167
Tablo 7.11. Benign ve DoS için Test Sonuçları 168
Tablo 7.12. Benign ve DDoS için Test Sonuçları 168
Tablo 7.13. Benign ve DDoS-DoS için Test Sonuçları 169
Tablo 7.14. Benign ve Reconnaissance için Test Sonuçları 170
Tablo 7.15. Benign ve DoS için DL Sonuçları 173
Tablo 7.16. Benign ve DDoS için DL Sonuçları 173
Tablo 7.17. Benign ve DoS+DDoS için DL Sonuçları 174
Tablo 7.18. Benign ve Reconnaissance için DL Sonuçları 174

SİMGELER VE KISALTMALAR DİZİNİ

$\arg\max$	:	En büyük değeri veren argüman
$\arg\min$	:	En küçük değeri veren argüman
$best_kl$	:	En düşük KL ayrışım değeri
C	:	Sınıf kümesi
C_j	:	j. küme
C'_j	:	Alt örnekleme sonrası j. küme
D	:	Veri kümesi
D_i	:	i. sınıfa ait veri alt kümesi
D'	:	Dengelenmiş veri kümesi
F_k	:	k. kata atanmış grup kümesi
$FC_\pi(\Delta)$	:	Operatörler uygulanarak elde edilen öznitelik kümesi
G	:	Tüm grupların kümesi
g	:	Grup etiketleri vektörü
h	:	KDE bant genişliği parametresi
$h_{g,c}$	:	g grubunda c sınıfına ait örnek sayısı
$\mathbf{h}_g$	:	g grubuna ait sınıf histogram vektörü
$\mathbb{I}(\cdot)$	:	Gösterge (indikatör) fonksiyonu
K	:	Kat (fold) sayısı
$K(\cdot)$	:	Çekirdek (kernel) fonksiyonu
k	:	Küme veya kat indeksi
$KL(f)$	:	f özniteliği için Kullback–Leibler ayrışımı
m	:	Öznitelik sayısı
n	:	Veri kümesindeki toplam örnek sayısı
$P_f(x)$	:	Çoğunluk sınıfı için olasılık yoğunluk fonksiyonu
$Q_f(x)$	:	Alt örneklenmiş veri için olasılık yoğunluk fonksiyonu
p_c	:	c sınıfının veri kümesindeki oranı
$\mathbf{p}$	:	Global sınıf oranları vektörü
π	:	Matematiksel operatörler kümesi
Σ	:	Korelasyon grupları kümesi
σ	:	Tekil korelasyon grubu
$\sum$	:	Toplama işleci
$total_kl$	:	Tüm öznitelikler üzerinden hesaplanan toplam KL ayrışımı
X	:	Özniteliklerden oluşan veri kümesi / veri çerçevesi
y	:	Sınıf etiketleri vektörü
Δ	:	Öznitelik kümesi
Δ'	:	Genişletilmiş öznitelik kümesi
$\lvert\cdot\rvert$	:	Küme kardinalitesi (eleman sayısı)
$\lVert\cdot\rVert_1$	:	L_1 normu
$\lVert\cdot\rVert_2$	:	L_2 (Öklidyen) normu

Kısaltmalar

6LoWPAN	:	IPv6 over Low-Power Wireless Personal Area Networks (Düşük Güçlü Kablosuz Kişisel Ağlar Üzerinde IPv6)
AE	:	Autoencoder (Oto-Kodlayıcı)

ANN	:	Artificial Neural Networks (Yapay Sinir Ağları)
AUC	:	Area Under the Curve (Eğri Altındaki Alan)
BLSTM	:	Bidirectional Long Short-Term Memory (Çift Yönlü Uzun Kısa Süreli Bellek)
Bluetooth	:	Bluetooth Technology (Bluetooth Haberleşme Teknolojisi)
cNN	:	Condensed Nearest Neighbour (Yoğunlaştırılmış En Yakın Komşu)
CNN	:	Convolutional Neural Networks (Evrişimli Sinir Ağları)
DBN	:	Deep Belief Networks (Derin İnanç Ağları)
DDoS	:	Distributed Denial of Service (Dağıtık Hizmet Engelleme)
DL	:	Deep Learning (Derin Öğrenme)
DNS	:	Domain Name System (Alan Adı Sistemi)
DoS	:	Denial of Service (Hizmet Engelleme)
DT	:	Decision Trees (Karar Ağaçları)
EDA	:	Exploratory Data Analysis (Keşifsel Veri Analizi)
FTP	:	File Transfer Protocol (Dosya Aktarım Protokolü)
GMM	:	Gaussian Mixture Model (Gauss Karışım Modeli)
HTTP	:	Hypertext Transfer Protocol (Hiper Metin Aktarım Protokolü)
ICMP	:	Internet Control Message Protocol (İnternet Denetim Mesaj Protokolü)
IDS	:	Intrusion Detection Systems (Saldırı Tespit Sistemleri)
IoT	:	Internet of Things (Nesnelerin İnterneti)
IoT-IDS	:	Internet of Things Intrusion Detection Systems (IoT için Saldırı Tespit Sistemleri)
IP	:	Internet Protocol (İnternet Protokolü)
KA	:	Knowledge-Aware Transformation (Bilgiye Duyarlı Dönüşüm)
KDE	:	Kernel Density Estimation (Çekirdek Yoğunluk Tahmini)
KL	:	Kullback–Leibler Divergence (Kullback–Leibler Ayrışımı)
KNN	:	K-Nearest Neighbors (K-En Yakın Komşu)
L7	:	Layer 7 Protocols (Uygulama Katmanı Protokolleri)
LR	:	Logistic Regression (Lojistik Regresyon)
LSTM	:	Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MITM	:	Man-In-The-Middle (Ortadaki Adam Saldırısı)
ML	:	Machine Learning (Makine Öğrenmesi)
MLP	:	Multi Layered Perceptron (Çok Katmanlı Algılayıcı)
NB	:	Naive Bayes (Naif Bayes)
OHE	:	One-Hot Encoding (Tek-Sıcak Kodlama)
OSS	:	One-Sided Selection (Tek Taraflı Seçim)
PCA	:	Principal Component Analysis (Temel Bileşenler Analizi)
PCAP	:	Packet Capture (Paket Yakalama)
RF	:	Random Forests (Rastgele Orman)
RFID	:	Radio Frequency Identification (Radyo Frekansı ile Tanımlama)
RL	:	Reinforcement Learning (Pekiştirmeli Öğrenme)
RNN	:	Recurrent Neural Networks (Yinelemeli Sinir Ağları)
ROC	:	Receiver Operating Characteristic (Alıcı İşletim Karakteristiği)
RPL	:	Routing Protocol for Low-Power and Lossy Networks (Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü)
SMO	:	Sequential Minimal Optimization (Ardışık En Küçük Optimizasyon)
SQL	:	Structured Query Language (Yapısal Sorgu Dili)
SVM	:	Support Vector Machines (Destek Vektör Makineleri)

TCP	:	Transmission Control Protocol (İletim Kontrol Protokolü)
TCP Flags	:	Transmission Control Protocol Flags (TCP Bayrakları)
TTL	:	Time To Live (Yaşam Süresi)
UDP	:	User Datagram Protocol (Kullanıcı Datagram Protokolü)
Wi-Fi	:	Wireless Fidelity (Kablosuz Ağ)
XSS	:	Cross-Site Scripting (Siteler Arası Betik Çalıştırma)
YZ	:	Yapay Zeka
ZigBee	:	ZigBee Protocol (ZigBee Haberleşme Protokolü)

NESNELERİN İNTERNETİNDE MAKİNE ÖĞRENMESİ TABANLI SALDIRI TESPİTİ İÇİN GELİŞMİŞ ÖZNİTELİK MÜHENDİSLİĞİ VE VERİ DENGESİZLİĞİ ÇÖZÜMLERİNİN TASARIMI VE ANALİZİ

ÖZET

Nesnelerin İnterneti (Internet of Things, IoT) kavramı, 1990'lı yılların sonunda ortaya çıkan, internete bağlı farklı cihazların bir araya gelerek, veri toplama, iletişim, kontrol ve ölçme gibi fonksiyonları bir arada gerçekleştirildiği sistemler olarak ifade edilebilir. Bu sistemler, genellikle, küçük boyutlu, enerji tasarufu yüksek ve uzaktan kontrol edilebilir cihazlar olarak karşımıza çıkmaktadır. Nesnelerin internetine dayalı sistemlerin gün geçtikçe kullanımın artması, cihazların doğası gereği internet ağında sürekli bağlı kalması, az enerji tüketimi kaygılarıyla geleneksel saldırı tespit sistemlerinin kullanılamaması gibi sebeplerden dolayı güvenlik açıklarına ve buna bağlı saldırılara oldukça açık bir yapı taşımaktadır. Nesnelerin interneti için, son teknolojiler, günümüzde kullanımı son derece yüksek olan, yapay zeka araçlı sistemler üzerinde yoğunlaşmaktadır. Yapay zekaya dayalı saldırı tespit sistemleri, yeni veri setlerinşin yayınlanması, hataya karşı dirençlilik, yüksek başarımlar elde edilmesi ile ilk defa karşılaşılan yeni (zero-day) saldırılara karşı oldukça iyi sonuçlar vermesi sebebiyle güçlü savunma sistemleri olarak karşımıza çıkmaktadırlar.

Bu tezde, IoT sistemler üzerinde makine öğrenmesine ve derin öğrenmeye dayalı saldırı tespit sistemleri geliştirilmiştir. Literatürde yer alan eksiklikler göz önüne alınarak, geleneksel makine öğrenmesi modellerinden faydalanan, basit ve etkili sistemlerin kurulmasına yönelik, uygun eğitim-test şemalarının belirlenmesi, nitelikli öznitelik mühendisliğinin gerçekleştirilmesi, yeni öznitelik oluşturma algoritmalarının kullanımı ile güçlü, hızlı eğitilebilir, yönetilebilir sistemler oluşturulmuştur. IoT saldırı tepsit sistemleri için büyük bir problem olan, veri dengesizliği problemleri de, bu tez kapsamında tasarlananan yeninden örnekleme algoritması ile giderilmiş ve literatüre dağılım tabanlı bir yeniden örnekleme metodolojisi sunulmuştur. Son olarak, bu tez kapsamında, global bir IoT saldırı tespit sistemi oluşturulması için çalışmalar yapılmış ve yayılım kayması gibi farklı dinamikler içeren IoT ağ ortamlarının, global bir IoT saldırı tespit sistemi oluşturulması için kullanıp kullanılamayacakları tartışılmıştır.

Anahtar Kelimeler: Derin Öğrenme, Makine Öğrenmesi, Nesnelerin İnterneti, Öznitelik Mühendisliği, Saldırı Tespit Sistemleri.

DESIGN AND ANALYSIS OF ADVANCED FEATURE ENGINEERING AND DATA IMBALANCE SOLUTIONS FOR MACHINE LEARNING–BASED INTRUSION DETECTION IN THE INTERNET OF THINGS

ABSTRACT

The concept of the Internet of Things (IoT) emerged in the late 1990s and can be defined as systems in which heterogeneous devices connected to the internet collaboratively perform functions such as data collection, communication, control, and monitoring. These systems typically consist of small-scale, energy-efficient, and remotely controllable devices. The rapid growth in the adoption of IoT-based systems, the inherent requirement for devices to remain continuously connected to network infrastructures, and strict energy consumption constraints render the use of traditional intrusion detection systems impractical. Consequently, IoT environments exhibit a highly vulnerable structure that is prone to a wide range of security threats and cyber-attacks.

Recent technological advancements in IoT security have increasingly focused on artificial intelligence–driven solutions, which have gained widespread adoption in contemporary applications. AI-based intrusion detection systems have demonstrated strong defensive capabilities due to the availability of newly published datasets, robustness against noise and errors, and their ability to achieve high detection performance. Moreover, these systems have shown promising results in identifying previously unseen (zero-day) attacks, making them a powerful alternative to conventional security mechanisms.

In this thesis, machine learning– and deep learning–based intrusion detection systems for IoT environments are developed. By addressing the limitations identified in the existing literature, this study emphasizes the design of simple yet effective systems based on classical machine learning models. Within this scope, appropriate training–testing schemes are established, advanced and high-quality feature engineering processes are conducted, and novel feature generation algorithms are employed to construct robust, fast-trainable, and easily manageable detection systems. One of the major challenges in IoT intrusion detection—namely, data imbalance—is addressed through a newly designed resampling algorithm proposed in this thesis, introducing a distribution-based resampling methodology to the literature. Finally, this study investigates the feasibility of constructing a global IoT intrusion detection system by analyzing heterogeneous IoT network environments that exhibit complex dynamics such as covariate and propagation shifts, and discusses whether such environments can be effectively leveraged for global intrusion detection purposes.

Keywords: Deep Learning, Machine Learning, Internet of Things, Feature Engineering, Intrusion Detection Systems.

1. GİRİŞ

IoT kavramı, 1990'lı yılların sonlarında ortaya çıkan, internete bağlı farklı cihazların bir araya gelerek, veri toplama, iletişim, kontrol ve ölçme gibi fonksiyonları bir arada gerçekleştirdiği sistemler olarak ifade edilebilir. Günümüzde, IoT kavramı büyüyerek, karşımıza akıllı ev ve bina sistemleri, sağlık, akıllı arabalar gibi alanlarda kullanımı yaygınlaşmıştır (Sanchez ve diğ., 2022). IoT sektörünün 2025 sonunda 50 milyar cihazdan oluşan ve 3.9 ile 11.1 trilyon dolar ekonomik etki yaratan bir pazar haline gelmesi beklenmektedir (Abushwereb ve diğ., 2022). Business Insider raporuna göre 2027'ye kadar yıllık 2.4 trilyon dolarlık büyüme beklenmektedir (Gubbi ve diğ., 2013). IoT cihazları, genellikle küçük boyutu, enerji tasarruflu ve uzaktan kontrol edilebilirdir. Teknolojinin hızlı gelişmesi, artan rekabet, kısıtlı kaynaklar ve maliyet kaygıları gibi sebepler, IoT cihazların normal ağlara kıyasla daha savunmasız olmasına ve dışarıdan gelecek saldırılara karşı daha savunmasız olmasına sebep olmaktadır. Bir IoT saldırısı temelde, IoT ekosistemine zarar vererek, bu cihazlara yetkisiz elde etmeyi amaçlayan ve bilgi sistemlerinin, gizliliğini, bütünlüğünü ve kullanılabilirliğini yüksek oranda tehlikeye atan eylemlerdir (Khraisat ve Alazab, 2021; Manivannan, 2024; Santos ve diğ., 2018). IoT cihazlar kısıtlı kaynaklara (düşük işlem gücü, sınırlı bellek kapasitesi, az enerji kullanımı) sahip olduğunda dolayı, geleneksel güvenlik yazılımları çalıştırılamamaktadır (Hamouda ve diğ., 2021; Mliki ve diğ., 2021; Zarpelão ve diğ., 2017). Ayrıca, bazı cihazlardaki varsayılan şifrelerin değiştirilememesi, aygıt yazılım güncellemelerinin geç veya hiç yapılamaması, şifreleme eksikliklikeri ile cihazların doğası gereği sürekli internet ağına bağlı kalmak zorunda oluşu, bu saldırılar için temel bir zemin hazırlar (Inayat ve diğ., 2022; Mishra ve Pandya, 2021; Roopak, 2021).

Kaspersky'ın araştırmasına göre, 2019'un ilk 6 ayında 276.000'den fazla IP adresinden toplamda 105 milyona yakın saldırı, IoT cihazlara gerçekleştirilmiştir (Mohammadi ve diğ., 2018). IoT cihazlarının sayısı arttıkça, farklı ağ trafiği desenleri ortaya çıkmakta ve ilk-gün (zero-day) denilen saldırıların tespitini zorlaşmaktadır. Makine öğrenmesi, bu türden hızla evrimleşen problemlerin çözümü için IoT cihazları için yaygın olarak kullanılan bir yöntem haline gelmektedir.

Literatürdeki son teknoloji (state-of-the-art) saldırı tespit çalışmalarının, yeni veri setlerinin yayınlanması, hataya karşı dirençlilik, yüksek başarımlar elde edilmesi ve ilk defa karşılaşılan yeni (zero-day) saldırılara karşı iyi sonuçlar vermesinden dolayı makine öğrenmesine dayalı yöntemlerin yaygınlaştığı görülmüştür. Makine öğrenmesine dayalı IoT saldırı tespit sistemleri (IoT-IDS), ağ davranışının içerisinde yer alan desenleri, bu ağ sistemlerinin dinlenmesiyle toplanılan verileri kullanarak, gelecekte olabilecek potansiyel saldırıları, makine öğrenmesi yöntemlerini kullanarak çözmeyi amaçlar.

Bu kapsamda ilgili tez çalışmasında, IoT sistemler içini, makine öğrenmesi ve derin öğrenme tabanlı saldırı tespit sistemi geliştirilmesi amaçlanmıştır. Bu kapsamda, tez çalışmasında, IoT-IDS sistemler için, makine öğrenmesine dayalı başarılı modeller kurulmasına yönelik olarak, tez kapsamında yapılan literatüre yapılan katkılar şu şekilde özetlenebilir. Bölüm 2'de ve 3'te, IoT-IDS sistemlerine yönelik olarak sistematik bir literatür çalışması gerçekleştirilmiştir. Bu sayede okuyucuya, literatürdeki saldırı türlerinin taksonomisi, saldırı türlerinin detaylı açıklamaları, literatürde kullanılan farklı saldırı tespit mimarilerinin anlaşılması, ve makine öğrenmesi (machine learning, ML) ile derin öğrenmeye (deep learning, DL) dayalı son teknoloji trendler hakkında anlayış kazandırılması amaçlanmıştır. Bu bölümün sonucunda okuyucu derin öğrenme tabanlı saldırı tespit sistemlerinin genel yönünün derin öğrenme odaklı olduğu okuyucu tarafından anlaşılacaktır. Bu bölümleri takiben, Bölüm 4'te ise derin öğrenme modelleriyle yarışabilecek, basit ama etkili, geleneksel makine öğrenmesine dayalı modeller kurulabilmesine yönelik çalışmalar gerçekleştirilmiştir. Bu bölümde, yanlılıktan (bias) ve başarılı modellerin kurulabilmesi adına, uygun bir eğitim ve test şemasının belirlenmesi, özniteliklerin yapısına ve karakteristiğine uygun olarak ön işleme aşamalarının tanımlanması, tez kapsamında geliştirilen, model başarımını arttırmaya yönelik olarak yenilikçi bir iteratif bir öznitelik üretme algoritmasını da içeren bir klasik öğrenme iş akışı dizayn edilmiştir. Bu iş akışı üzerinden oluşturulan, makine öğrenmesine dayalı model, literatürde yer alan derin öğrenme algoritmaları ile kıyaslanarak, klasik yöntemlerin hala iyi sonuçlar verdiği test edilmeye çalışılmıştır. Model başarımının arttırılmasına yönelik olarak, Bölüm 5'te, daha önce tasaralanan iteratif bir öznitelik algoritması yerine, yine bu tez kapsamında var olan bir öznitelik üretim algoritması temel

alınarak tasarlanan, yenilikçi bir ağaç tabanlı iteratif bir öznitelik algoritması üzerinden, sınıflandırma problemleri başta olmak üzere, IoT-IDS sistemlerinde kullanılabilecek, yorumlanabilir bir öznitelik algoritması tasarlanmış ve test edilmiştir. Bu bölümde de, Bölüm 4'de olduğu gibi son teknoloji DL modelleri ile kıyaslama yapılarak, açıklanabilir özniteliklerle, ve daha az sayıda öznitelikle oluşturulmuş, daha basit, geleneksel makine öğrenmesine dayalı IDS sistemlerin hala DL tabanlı modellerle yarışabilir olduğu gösterilmiştir. Bölüm 6.2.'da ise, literatürde yer alan ve hem ML-DL problemlerinde hem de IoT-IDS literatüründe büyük bir problem olan, dengesiz veri setlerinin dengeli hale getirilmesine yönelik, dağılım temelli bir yeniden örnekleme (resampling) metodu tasarlanmıştır. Son olarak, Bölüm 7.3.'te ise global bir IoT IDS sistemleri oluşturulmasına yönelik olarak, farklı veri setlerinin birleşitirlmesi, bu veri setleri üzerinden yayılım kayma (drift shift) analizleri, ve bu aşamaları takiben hem ML hem de DL tabanlı modeller oluşturularak, global bir IoT-IDS oluşturulabilmesine yönelik çalışmalar yapılmış, teknik zorluklar ve böyle bir sistemin oluşturulabilirliği üzerine derinlemesine tartışılmıştır. Bu sayede okuyucuya, literatürde de genel olarak trend haline geleceği düşünülen, global bir IoT-IDS sistemi tasarlama düşüncesinde olan araştırmacılar için iyi bir başlangıç noktası oluşturması amaçlanmıştır.

2. IOT SALDIRI TESPİT SİSTEMLERİNE BAKIŞ

Bu bölümde, IoT Saldırılarının doğası, kullanılan yöntemler ele alınacak ve tez çalışmasında gerçekleştirilen çalışmaların temel sebepleri ve araştırma soruları ele alınarak, genel bir anlayışın okuyucuya kazandırılması amaçlanmaktadır.

Analojik olarak, IoT güvenliği, pencereleri (sensörler) her zaman açık olan bir eve benzetilebilir. Burada bilgi çalacak hırsızlar, doğrudan kapıdan girmeye çalışabileceği gibi (dışsal saldırı), içeri giren bir misafir tarafından anahtarın çalınması (iç saldırı, tampering), veya evin su hattının tıkanması (DoS), veya ev sahiplerinin kandırılarak kilitlerin kendileri tarafından (oltalama-phishing, sosyal mühensislik-social engineering) açılması sağlanabilir. Her pencere veya kapı (mimari katman), içeriye farklı şekillerde sızma yöntemleri sağlayacak açık kapı niteliğindedir.

2.1. Etki Şeklinde Göre Sınıflandırma

IoT saldırılarını, etki şekline göre sınıflandırıldığında olursak, pasif ve aktif saldırılar olarak iki gruba ayrılabilir (Anitha ve Arockiam, 2022; Benkhelifa ve diğ., 2018). Bu sınıflandırma, temel olarak, saldırıyı gerçekleştiren saldırganın, ağdaki veriye müdahale edip etmemesine, sistemin işleyişini bozup bozmamasına dayanmaktadır (Benkhelifa ve diğ., 2018; Heidari ve Jamali, 2023).

2.1.1. Pasif Saldırılar

Pasif saldırılar, sistem kaynaklarına zarar vermek veya veri akışını değiştirmek yerine, sadece bilgiyi izlemeyi, dinlemeyi veya analiz etmeyi amaçlamaktadır (Anitha ve Arockiam, 2022; Heidari ve Jamali, 2023). Buradaki temel amaç ise, bilgi hırsızlığı yaparak, gizli verilere erişmek veya sistemin zaafiyetlerini belirlemek adına keşifsel analiz yapmaktır (Moustafa ve diğ., 2023). Sistemde işleyişe dair herhangi bir kesinti oluşturmadıklarından tespit edilmeleri oldukça zordur (Anitha ve Arockiam, 2022).

Aşağıda pasif saldırı türlerine ait yaygın örnekler listelenmiştir:

- Dinleme (Eavsedropping): İletişim kanallarındaki veri akışının gizlice dinlenmesidir. Bu şekilde, şifreleme anahtaları, konfigürasyon ayarları, veya

uygulama verieri ele geçirilmeye çalışılır (Zarpelão ve diğ., 2017).

- Trafik Analizi: İlgilenilen cihazdaki iletişim içeriği şifreli dahi olsa, paket sıklığı, boyutları ve paket akışının yönü gibi materyaller analiz edilerek ilgili ağın yapısı ve kullanıcı alışkanlıkları hakkında bilgi toplanır (Heidari ve Jamali, 2023; Roopak, 2021).
- Koklama (Sniffing): Ağdaki veri paketleri çeşitli yazılımsal araçlarla yakalanarak, kullanıcı adları, parolalar gibi hassas içerikli bilgilerin elde edilmesidir (Ashraf ve diğ., 2022; Rajkarnikar, 2023).

2.1.2. Aktif Saldırılar

Aktif saldırıları, veri akışını doğrudan değiştiren, sisteme sahte veri enjekte edebilen, çeşitli servislerin kullanılabilirliğini engelleyen eylemler olarak adlandırabiliriz (Anitha ve Arockiam, 2022; Benkhelifa ve diğ., 2018). Aktif saldırılar, temel olarak, sistemin bütünlüğünü bozmak, yetkisiz erişim elde etmek ve hizmetleri tamamen durdurmaya odaklanmaktadır (Benkhelifa ve diğ., 2018; Roopak, 2021). Aktif saldırılar sisteme doğrudan müdahale ettiklerinden dolayı, sistem performansında görünür etkiler bırakmaları olasıdır (Anitha ve Arockiam, 2022). Sistemin hem bütünlüğünü (integrity) hem de erişilebilirliğni (availability) hedef alan saldırı türleridir (Albulayhi ve diğ., 2021; Benkhelifa ve diğ., 2018).

Aktif saldırıların en yaygın örnekleri aşağıda gösterilmiştir.

- Hizmet Engelleme (Denial of Service, DoS) ve Dağıtık Hizmet Engelleme (Distributed Denial of Service, DDoS): Sistem kaynaklarını (bant genişliği, işlemci gücü gibi), sahte bir trafikle boğarak, kullanıcıların hizmete erişimlerinin engellenmesi olarak ifade edilebilir (Ashraf ve diğ., 2022; Mliki ve diğ., 2021).
- Ortadaki Adam (Man-In-The-Middle, MITM): Saldırganin iletişim kuran iki cihazın arasındaki iletişim kanalına sızarak, verileri değiştirme veya farklı bir kaynağa yönlendirmesidir (Chaabouni ve diğ., 2019; Mliki ve diğ., 2021).
- Sahtecilik (Spoofing): Saldırganın kendisini yetkili bir cihaz gibi sisteme tanıtarak sisteme sızması ve kötü amaçlı yazılım (malvare) yaymasıdır (Mliki ve diğ., 2021;

Rajkarnikar, 2023).

- Müdahale (Tampering): IoT cihazların donanımlarına veya yazılımlarına gerek fiziksel gerekse yazılımsal olarak müdahale edilmesidir (Mliki ve diğ., 2021; Roopak, 2021).

Bu bölümü takiben, bir diğer sınıflandırma grubu olan, kaynağına göre sınıflandırma incelencektir.

2.2. Kaynağına Göre Sınıflandırma

IoT saldırıları, saldıganın ağ üzerindeki konumuna ve bu ağ üzerindeki yetki düzeyine göre temel olarak Dış Saldırılar (External Attacks) ve İç Saldırılar (Internal/Insider attacks) olmak üzere iki kategoriye ayrılmaktadırlar (Ashraf ve diğ., 2022; Zarpelão ve diğ., 2017). Bu sınıflandırma biçimi, ağa ait güvenlik duvarlarının (firewall) ve saldırı tespit sistemlerinin (Intrusion Detection-IDS) nereye odaklanması gerektiğini belirlemek açısından kritiktir (Anitha ve Arockiam, 2022; Khraisat ve Alazab, 2021).

2.2.1. Dış Saldırılar

Dış saldırılar, hedeflenen ağın veya IoT ekosisteminin dışındaki yetkisiz kullanıcılar tarafından internet üzerinden başlatılan zararlı eylemler olarak düşünülebilir (Anitha ve Arockiam, 2022; Zarpelão ve diğ., 2017). Saldırganlar saldırı yöntemi olarak, cihazlara uzaktan erişim sağlamak için internete açık kapıları, zayıf şifreleri veya güncellenmemiş yazılım açıklarını kullanırlar (Anitha ve Arockiam, 2022; Ashraf ve diğ., 2022). Genellikle iç ağdan bir kullanıcı (insider) gibi görünerek veri iletişimi sırasında kötü amaçlı kod enjekte (code injection) ederler. Buradaki temel amaç sisteme sızarak, yetkisiz erişim elde etmek ve ağın bütünlüğünü bozmaktır.

2.2.2. İç Saldırılar

Bu saldırı türünde ise, ağın içinde bulunan ve bir dereceye kadar meşru erişim yetkisine sahip olan kullanıcılar veya cihazlar tarafından gerçekleştirilen saldırılardır (Anitha ve Arockiam, 2022; Zarpelão ve diğ., 2017). Bu türden saldırılarda ise, yetkili kişiler sahip oldukları kullanım haklarını kötüye kullanarak, bu hakları çeşitli yetkisiz işlemler ve

ihlaller üzerinden kötü amaçlara hizmet etmeyi hedef olarak belirlerlrer. İç saldırıların tespiti, dış saldırılara göre çok daha zordur, çünkü bu saldırganlar halihazırda güvenilen kanallardan iletişime geçer ve geçerli kimlik bilgilerini kullanılırlar (Anitha ve Arockiam, 2022).

2.2.3. Kaynak Karmaşası

IoT saldırıları bazı durumlarda, iç ve dış saldırıların arasında bir yerde konumlanabilirler. Örneğin bir dış saldırgan, zayıf bir şekilde korunan bir IoT cihazını ele geçirerek onu bir zombi veya bot haline getirebilir (Khraisat ve Alazab, 2021; Roopak, 2021). Bu durumda saldırgan esasen ağa göre dış bir kaynakta olsa da, saldırıyı ağın içindeki bir cihazdan yani içten tetiklemiş olur ki Mirai Botneti gibi yapılar, binlerce iç cihazı kullanarak büyük ölçekli DDoS saldırılarını bu şekilde koordine etmektedirler (Albulayhi ve diğ., 2021; Roopak, 2021).

2.3. Mimari Katmanlara Göre Kategorizasyon

IoT mimarisi, heterojen cihazlar ve protokollerin karmaşıklığını yönetmek amacıyla genellikle katmanlı bir yaklaşım kullanılarak analiz edilmektedir (Benkhelifa ve diğ., 2018; Mliki ve diğ., 2021). Literatürder, üç, beş, veya altı katmanlı olmak üzere çeşitli modeller önerilmiş olsa da en temel sınıflandırma Algılama (Perception), Ağ (Network) ve Uygulama (Application) katmanları üzerinden yapılmaktadır (Kikissagbe ve Adda, 2024; Santos ve diğ., 2018). Her katman, kendine ait çeşitli işlevlere ve bu işlevleri hedef alan spesifik saldırı türlerine sahiptir.

2.3.1. Algılama (Perception/Physical) Katman Saldırıları

Bu katman, IoT ekosisteminde fiziksel bileşenlere tekabül eden ve RFID (Radyo Frekansı ile Tanımlama Teknolojisi, Radio Frequency Identification) etiketleri gibi veriyi toplayan ve fiziksel ortamla etkileşen katman olarak ifade edilebilir (Albulayhi ve diğ., 2021; Rajkarnikar, 2023; Saadouni ve diğ., 2024; Santos ve diğ., 2018). Bu katmanda, çevresel verileri (sıcak, nem, konum vb.) toplama ve bu verilerin dijital forma dönüştürülerek üst katmanlara iletilmesi sağlanmaktadır (Albulayhi ve diğ., 2021; Rajkarnikar, 2023; Santos ve diğ., 2018).

Bu katmandaki yaygın saldırılar;

- Düğüm Yakalama (Node Capture): Bir saldırgan cihazı fiziksel olarak ele geçirerek, bu cihaz içerisindeki şifreleme anahtalarını veya çeşitli hassas bilgileri çalmasıdır (Khraisat ve Alazab, 2021).
- Donanıma Müdahale (Tampering): Cihaz donanımın fiziksel olarak değişitirlmesi veya bozulmasıdır (Roopak, 2021).
- Radyo Frekansı Bozma (Jamming): Kablosuz iletişimi engellemek adına yüksek gürültü sinyallerin cihaza gönderilerek çeşitli sevislerin kullanılamaz hale getirilmesidir (Roopak, 2021).
- Uyku Yoksunluğu (Sleep Deprivation): Pil ile çalışan kısıtlı kaynakalra sahip cihazların sürekli olarak yüksek yoğunlukta çalışmaya zorlanılarak enerjilerinin kasti olarak bitirlmesidir (Chaabouni ve diğ., 2019).
- Yan Kanal Saldırıları (Side-Channel): Şifreleme işlemlerini kırmak amacıyla cihaza ait güç tüketimi veya işlem süresi gibi fiziksel sızıntıları elde ederek analiz edilmesidir (Mliki ve diğ., 2021).

şeklinde özetlenebilmektedir.

2.3.2. Ağ Katmanı Saldırıları

Ağ (Network-Transmission) katmanı, genel olarak IoT mimarisinin merkezinde yer alır, ve bir alt katmandan gelen verilerin işlenmesi, yönlendirilmesi ve ilgili birimlere (bulut sunucuları, ağ geçitleri vb.) iletilmesinden sorumludur (Kikissagbe ve Adda, 2024; Santos ve diğ., 2018). İşlevsel olarak, veri iletimi, yönlendirme (routing), ve ağ yönetimini gerçekleştirirken Wi-Fi, Blueetooth, ZigBee, 6LoWPAN ve RPL gibi çeşitli protokoller yine bu katmanda çalışmaktadır (Kikissagbe ve Adda, 2024; Saadouni ve diğ., 2024; Santos ve diğ., 2018).

Ağ Katmanındaki yaygın saldırılar:

- Hizmet Engelleme (Denial of Service, DoS) ve Dağıtık Hizmet Engelleme (Distributed Denial of Servce, DDoS): Sistem kaynaklarını (bant genişliği, işlemci gücü gibi), sahte bir trafikle boğarak, kullanıcıların hizmete erişimlerinin

engellenmesi olarak ifade edilebilir (Ashraf ve diğ., 2022; Mliki ve diğ., 2021).

- Ortadaki Adam (Man-In-The-Middle, MITM): Saldırganin iletişim kuran iki cihazın arasındaki iletişim kanalına sızarak, verileri değiştirme veya farklı bir kaynağa yönlendirmesidir (Chaabouni ve diğ., 2019; Mliki ve diğ., 2021).
- Sahtecilik (Spoofing): Saldırganın kendisini yetkili bir cihaz gibi sisteme tanıtarak sisteme sızması ve kötü amaçlı yazılım (malvare) yaymasıdır (Mliki ve diğ., 2021; Rajkarnikar, 2023).
- Yönlendirme (Routing) Saldırıları: Spoofing (veri trafiğini yanlış yöne çekme), Solucan Deliği (Wormhole) (iki uzak nokta arasında gizli bir tünel kurma), Karadelik (BlackHole) (paketleri yutma), ile sahte kimlikler oluşturan Sybil gibi verinin ağda yönlendirilmesine dair bütünlüğü bozan saldırılardır.(Mliki ve diğ., 2021; Roopak, 2021).

şeklinde özetlenebilir.

2.3.3. Uygulama Katmanı Saldırıları

IoT mimarisindeki en üst düzey katman olup, kullanıcılar akıllı hizmetler veren ve arayüzler sunan iş mantığının bulunduğu yer olarak ifade edilebilir. Verileri işleyerek, akıllı ev, akıllı şehir, veya sağlık izleme hizmetleri gibi çeşitli sevisler sunarak, son kullanıcının IoT cihazları ile etkileşim kurmasın sağlanmaktadır (Rajkarnikar, 2023).

Aşağıda sıklıkla karşılaşılan uygulama katmanı saldırıları sunulmuştur:

- Kötü Amaçlı Yazılımlar (Malware): Virüsler, solucanlar (worms), truva atları (trojan), fidye yazılımları (ransomware) gibi çeşitli zararlı yazılımlarla sistemlerin enfekte edilmesidir (Roopak, 2021).
- Kimlik Avı (Phishing): Sahte e-posta, web sayfaları aracılığıyla kullanıcılara ait kimlik verilerinin çalınmasıdır (Santos ve diğ., 2018).
- Enjeksiyon (Injection) Saldırıları: SQL enjeksiyonu (SQL injection) veya komut enjeksiyonu gibi yöntemlerle, uygulama veritabanlarına veya mantığına yetkisiz ve zararlı müdahalelerin bulunulmasıdır (Zarpelão ve diğ., 2017).
- Siteler Arası Betik Çalıştırma (XSS): Web tabanlı IoT arayüzlerine kötü amaçlı

betikler enjekte edilerek kullanıcı tarayıcalarının kontrol edilmesidir (Roopak, 2021).

2.4. IoT Saldırı Tespit Sistemleri (IoT-IDS)

IoT Saldırı Tespit Sistemleri, bir önceki bölümlerde de bahsi geçen, bir ağdaki veya bilgisayar sistemindeki etkinlikleri izleyerek, kötü amaçlı eylemleri, güvenlik politikalarında yapılmış ihlalleri belirleyen, raporlayan, gerek donanım gerekse yazılımsal tabanlı güvenlik araçlarıdır (Anitha ve Arockiam, 2022; Khraisat ve Alazab, 2021; Zarpelão ve diğ., 2017). Bu sistemler, IoT ekosisteminde şifreleme ve kimlik doğrulama gibi temel güvenlik tedbirlerinin aşıldığı durumlarda devreye giren ekstra bir katman olarak ifade edilebilir. Yüzeysel olarak temel bileşenler, veri toplayan sensörler, verileri analiz eden bir analiz motoru ve uyarıları oluşturan bir raporlama sistemidir (Ashraf ve diğ., 2022).

IoT saldırı tespit sistemleri, cihazların düşük işlem gücü, sınırlı bellek ve enerji kısıtları nedeniyle, geleneksel ağlara yönelik ağır güvenlik yazılımları yerine IoT'ye uyarlanmış hafif (lightweight) çözümler sunmayı hedeflemektedir (Anitha ve Arockiam, 2022).

IoT-IDS sistemleri yerleşim stratejilerine göre merkezi, dağıtık veya hibrit yapılarda tasarlanabilmektedir. Merkezi yaklaşımlarda ağ trafiği genellikle bir sınır yönlendiricisi veya ana sunucu üzerinden izlenirken, dağıtık yapılarda her IoT cihazına yerleştirilen hafif ajanlar aracılığıyla yerel ve komşu trafik denetlenir. Hibrit yöntemler ise bu iki yaklaşımı küme tabanlı bir yapı altında birleştirir (Moustafa ve diğ., 2023).

IoT-IDS'ler saldırılar üzerinde kullandıkları tespit mekanizmasına göre temel olarak 5 grupta incelenebilir. Aşağıda bu gruplar verilmiştir.

- İmzaya Dayalı (Signature-Based): Bilinen saldırı kalıpları veritabanında daha önce kaydedilmektedir. Yeni bir veri akışı geldiğinde, veritabanındaki kalıplarla karşılaştırılarak, saldırı olup olmadığı veya saldırının türü tespit edilmeye çalışılır. Burada veritabanında hazır kalıplar bulunduğundan dolayı, bilinen tehditlere karşı başarılı olsa da, tamamen yeni saldırıları (sıfıncı gün, zero-day) tespit edememektedir (Liao ve diğ., 2013).

- Anomaliye Dayalı (Anomaly-Based): Sistemin normal yani zararsız durumdaki çalışma profili öğrenilir ve bu profilden sapan her hareket birer anomali yani saldırı olarak işaretlenir. Bilinmeyen saldırıları yakalamakta oldukça etkili bir yöntemdir (Chaabouni ve diğ., 2019).
- Spesifikasyona Bağlı (Specification-Based): Uzmanlar tarafından tanımlanan belirli protokol ve cihaz kurallarını temel alan, anomali tabanlı sistemlere benzer ancak manuel kurallar tanımlandığından dolayı yanlış alarm oranı (false positive rate, FPR) daha düşük olan sistemlerdir (Chaabouni ve diğ., 2019).
- Hibrit Sistemler: Yukarıda bahsi geçen sistemleri birleştiren IDS sistemleri olarak ifade edilebilir.
- Öğrenme Tabanlı Sistemler (Learning Based Systems): Günümüzdeki çoğu IoT-IDS, büyük veriyi analiz etmek için Makine Öğrenmesi (Machine Learning, ML) ve Derin Öğrenme (Deep Learning, DL) tekniklerini yoğun olarak kullanmaktadır. Bu yöntemler, verideki çeşitli desenleri ve saldırılara ait farklı karakteristikleri doğrudan verinin kendisinden öğrenerek, insan müdahalesini minimum seviyede tutan ve literatürde de en yüksek başarıyı gösteren sistemlerdir.

Bir sonraki bölümde makine öğrenmesi ve derin öğrenmeye dayalı sistemler detaylı olarak incelenecektir.

2.4.1. Makine Öğrenmesi ve Derin Öğrenmeye Dayalı Sistemler

Bu tez çalışmasında Makine Öğrenmesi ve Derin Öğrenmeye dayalı bir IoT-IDS sistemi gerçekleştirileceğinden, bu türden IoT-IDS sistemlerinin nasıl kurulduğunun anlaşılması amacıyla, bu sistemler hakkında daha detaylı bir bilgi verilmesi amaçlanmıştır.

IoT ekosisteminin hızla büyümesi ve cihazların kaynak kısıtlı doğası (düşük bellek, işlem gücü ve enerji), geleneksel güvenlik yöntemlerini yetersiz kılmaktadır. Bu bağlamda, ML ve DL tabanlı IDS'ler, büyük veri setlerinden karmaşık kalıpları öğrenme, sıfırıncı gün saldırılarını tespit etme ve insan müdahalesine gerek duymadan kararlar verme yetenekleri sayesinde literatürde olarak öne çıkmaktadır (Anitha ve Arockiam, 2022; Liao ve diğ., 2024; Maheswaran ve diğ., 2023; Zarpelão ve diğ., 2017).

2.4.1.1. Geleneksel Makine Öğrenmesi Yaklaşımları

Bu yaklaşımlar genellikle yapılandırılmış veriler üzerinde çalışır ve manuel olarak gerçekleştirilmiş öznitelik mühendisliğine (feature engineering) ihtiyaç duymaktadırlar. Temel olarak IoT-IDS literatüründe kullanılan geleneksel yöntemler, 2 alt gruba ayrılabilir.

Denetimli Öğrenme Tabanlı Yaklaşımlar (Supervised Learning):

- Destek Vektör Makineleri (Support Vector Machines, SVM): Yüksek boyutlu verilerde güçlü sınıflandırma yeteneğine sahip olan bu metodolojide, IoT ağlarında özellikle yönlendirme saldırılarında (sinkhole, blackhole) tespitinde %99,8'e varan doğruluk oranlarına ulaşılmıştır (Jing ve Chen, 2019). Ancak SVM, yüksek değişkenlik gösteren ve dinamik IoT verilerinde problem yaratabilirken, yüksek hesaplama maliyeti yüzünden efektif bir çözüm olmaktan uzaktır (Liao ve diğ., 2024).
- K-En Yakın Komşu Algoritması (K-Nearest Neighbor, KNN): Basit ve etkili bir yöntemdir, ancak büyük veri setlerinde hesaplama maliyeti yüksek olup, aykırı durumlardan (outliers) negatif olarak etkilenebilir (Abdaljabar ve diğ., 2021; Zhu ve diğ., 2020).
- Karar Ağaçları ve Rastgele Ormanlar (Decision Trees-DT, Random Forests (RF)): Karar Ağaçları, hızlı tahmin süreleri, aşırı öğrenmeye (overfitting) karşı güçlü olmaları, yorumlanabilme (interpretability) kolaylıkları ve yüksek başarımları nedeniyle klasik yöntemler arasında en güçlü modellerdendir (Espinosa, 2023; Saadouni ve diğ., 2024).
- Naif Bayes Sınıflandırıcı (Naive Bayes, NB): Olasılıksal bir modeldir; hızlı çalışmasına rağmen öznitelikler arasındaki bağımsızlık varsayımı nedeniyle karmaşık saldırılarda performansı düşürebilir (Saadouni ve diğ., 2024).

Denetimsiz Öğrenme Yaklaşımları:

- K-Ortalama Kümeleme (K-Means): Etiketlenmemiş verilerdeki benzerlikleri bulur ve Sybil gibi kimlik sahteciliği saldırılarını tespit etmekte kullanılmaktadır (Jarrah

ve diğ., 2018; Muniyandi ve diğ., 2012).

- Temel Bileşenler Analizi (Principal Component Analysis, PCA): Genellikle bir saldırı tespit yönteminden ziyade, verideki boyut sayısını azaltarak sistemin karmaşıklığını ve hesaplama yükünü düşürmeye yönelik bir ön işleme adımı olarak ifade edilebilir (Bhattacharya ve diğ., 2020).

2.4.1.2. Derin Öğrenme Temelli Yaklaşımlar

Derin Öğrenmeye dayalı yöntemler ise, geleneksel makine öğrenmesine dayalı yöntemlerin aksine, ham verilerden otomatik olarak öznitelik çıkarımı yapabilir ve yüksek boyutlu verilerde oldukça iyi çalışan sistemler sunmasıyla bilinmektedirler. IoT-IDS perspektifinden incelendiğinde ise DL tabanlı yaklaşımlar, aşağıdaki gruplara ayrılabilir.

- Evrişimli Sinir Ağları (Convolutional Neural Networks, CNN): Görüntü tanıma için tasarlanmış olsa da, IoT ağ trafiğini görüntülere veya 2D matrislere dönüştürerek DDoS saldırılarını ve kötü amaçlı yazılımları (malware) yüksek doğrulukla tespit edebilmektedir (Bhattacharya ve Kumar, 2025; Liao ve diğ., 2024).
- Yinelemeli Sinir Ağları (Recurrent Neural Network, RNN), Uzun Kısa Süreli Bellek (Long-Short Term Memory, LSTM): aman serisi verilerini işleme yeteneği sayesinde trafik akışındaki zamansal bağımlılıkları öğrenir; botnet aktivitelerinin tespitinde son derece etkilidir (Kim ve diğ., 2020; Park ve diğ., 2020).
- Oto-Kodlayıcılar (Autoencoders, AE): Denetimsiz bir yöntemdir; normal trafiği yeniden yapılandırmayı öğrenir ve yapılandırma hatası yüksek olan verileri anomali olarak işaretlemektedir (Binitha ve Sathya, 2012).
- Derin İnanç Ağları (Deep Belief Networks, DBN): Katmanlı yapısıyla karmaşık özellikleri öğrenebilir; özellikle etiketli verinin az olduğu durumlarda ön eğitim (pre-training) yeteneğiyle avantaj sağlamaktadır (Ferrag ve diğ., 2020).

2.4.1.3. Hibrit Yaklaşımlar

Bu yaklaşımlar birden fazla yaklaşımın bir arada kullanarak, farklı yöntemlerin avantajlarını birleştirmeyi, v e buna bağlı olarak da tek bir metodolojinin eksikliklerini

azaltmayı amaçlar.

- Topluluk Öğrenme (Ensemble): Bu metodoloji birden fazla zayıf sınıflandırıcıyı (weak learner) - DT, NB, RF gibi- birleştirerek, daha güçlü ve hata payı düşük bir sistem oluşturur (Aburomman ve Reaz, 2016, 2017).
- Federatif Öğrenme (Federated Learning): Veriyi tek bir merkeze toplamadan, cihazların kendi yerel verileriyle modelleri eğitip sadece ağırlıkları merkeze göndermesi esasına dayanır; bu sayede veri gizliliği korunur
- Biyo-ilhamlı Algoritmalar (Bio-inspired Algorithms): Genetik Algoritmalar (Genetic Algorithms, GA), Parçacık Sürü Optimizasyonu (Partical Swarm Optimization, PSO), Gri Kurt Optimizasyonu (Gray Wolf Optimization, GWO) gibi doğadan esinlenen yöntemler, özellik seçimi veya model parametrelerinin ince ayarı (fine-tuning) için ML modelleriyle birleştirilir (Kilichev ve Kim, 2023) .

Bu bölümde, IoT-IDS literatürüne genel bir giriş yapılarak, temel olarak IDS sistemlerinin IoT perspektifinde nasıl incelendiği, bu türden sistemlere neden ihtiyaç duyulduğu, olası saldırı türleri ve bu saldırı türlerinin farklı açılardan (etki şekline göre, kaynağına göre, mimari katmana göre) nasıl incelendiği ile alakalı genel bir anlayış sunulması amaçlanmıştır. Bunu takiben literatürdeki IoT tabanlı IDS sistemleri için genel bir sınıflandırma yapılarak, bu problemin hangi şekillerde ve hangi farklı açılardan ele alındığı ve nasıl çözümler sunulduğunun anlaşılması sağlanmıştır. Bu sayede tez kapsamında geliştirilmiş olan IoT IDS sisteminin tam olarak ne konumda bulunduğunun okuyucu tarafından anlaşılması amaçlanmıştır.

3. ARAŞTIRMA SORULARI

Bu bölümde tez kapsamında yapılan çalışmalarının bulunduğu pozisyonun anlaşılması, hangi sorulara çözüm oluşturulmaya çalışıldığının anlaşılabilmesi için öncelikli olarak, Bölüm 2.4.1.'de bahsedilen ML ve DL üzerine yapılan IoT-IDS çalışmalarının daha detaylı bir analizi sunulacak ve literatürden elde edilen çıkarımlarla elde edilen araştırma sorularının detayları açıklanacaktır.

3.1. ML ve DL Tabanlı Sistemlerin Detaylı İncelemesi

ML ve DL tabanlı IoT-IDS sistemlerine ait tasarım ve uygulanma süreçlerinde, yapılan güncel çalışmaların doğru bir şekilde kavranabilmesi; kullanılabilecek yöntemlerin belirlenmesi, karşılaştırılacak çalışmaların seçilmesi ve değerlendirme aşamasında başvurulacak performans ölçütlerinin tespiti açısından kritik öneme sahiptir. Bu bağlamda, ilgili tez çalışması için mantıklı bir başlangıç zemini oluşturmak ve sağlıklı bir akademik çerçeve sunabilmek amacıyla, güncel literatürde yer alan üç farklı sistematik literatür taraması (systematic literature review) ayrıntılı biçimde incelenmiştir (Ahmad ve Alsmadi, 2021; Idrissi ve diğ., 2020; Khan ve diğ., 2022) .

Söz konusu çalışmalarda ele alınan yöntemler, kullanılan veri setleri, değerlendirme ölçütleri ve elde edilen performans sonuçları karşılaştırmalı olarak analiz edilmiş; bu doğrultuda, güncel bir IDS çalışmasında tercih edilebilecek yöntemler tartışılmıştır. Özellikle Ahmad ve Alsmadi, 2021 çalışmasında, yazarlar farklı araştırma soruları tanımlamış ve bu sorulara, belirlenen çeşitli değerlendirme kriterleri doğrultusunda yanıt aramışlardır. İlgili araştırma sorularını ve kriterleri karşılayan çalışmalar Tablo 3.1'de sunulmuştur.

Tablo 3.1'de yer alan Ref. sütunu çalışmanın referansını, DL ve ML sütunları sırasıyla kullanılan Derin Öğrenme ve Makine Öğrenmesi modellerini göstermektedir. Acc., Prec., Rec. ve F1 sütunları ise modellerin değerlendirilmesinde kullanılan doğruluk (accuracy), kesinlik (precisicion), hassasiyet (recall) ve F1-skoru (F1-Score) gibi performans ölçütlerini temsil etmektedir.

Tablo 3.1. Literatürde kullanılan IDS modellerinin karşılaştırılması

Ref.	DL	DL Performans				ML	ML Performans			
		Acc.	Prec.	Rec.	F1		Acc.	Prec.	Rec.	F1
(Ioannou ve Vassiliou, 2019)	-	–	–	–	–	SVM	58.81	–	–	–
(Chaudhary ve Gupta, 2019)	-	–	–	–	–	RF	88.17	–	–	–
(Kumar ve Lim, 2019)	-	–	–	–	–	KNN	88.44	86.62	88.90	88.76
(Dwyer ve diğ., 2019)	-	–	–	–	–	RF	88.59	88.09	88.08	88.58
(Soe ve diğ., 2019b)	ANN	96.63	96.18	96.64	96.38	–	–	–	–	–
(Soe ve diğ., 2019a)	CNN	99.84	–	–	–	–	–	–	–	–
(Fatayer ve Azara, 2019)	MLP	99.84	–	–	–	–	–	–	–	–
(Hwang ve diğ., 2019)	LSTM	88.75	88.75	88.75	88.75	–	–	–	–	–
(Goyal ve diğ., 2019)	ANN	99.74	99.56	100	99.78	SVM	99.86	99.98	100	99.92
(Roopak ve diğ., 2019)	CNN+LSTM	99.76	99.41	99.80	99.23	SVM	99.56	98.72	98.82	98.82
(Wehbi ve diğ., 2019)	ANN	99.88	–	–	–	KNN	99.89	–	–	–
(MacDermott ve diğ., 2019)	Yazarlar k-Ortalama kullansada, DL modellerin daha uygun olduğunu ifade etmişlerdir.									
(Roopak ve diğ., 2020)	CNN+LSTM	99.68	99.69	99.68	99.68	SVM	99.48	99.58	99.48	99.47
(Khatun ve diğ., 2019)	ANN	87.51	–	–	–	SVM	87.57	–	–	–
(Liang ve Znati, 2019)	LSTM	99.89	100	99.80	99.88	SVM	98.18	96.13	98.10	95.13
(Ullah ve diğ., 2019)	CNN	98.47	98.47	98.46	98.44	SVM	96.10	96.16	96.10	96.13

İncelenen çalışmaların ilk dördü sadece geleneksel makine öğrenmesi yaklaşımlarını temel alırken, beşinci ile sekizinci çalışmalar derin öğrenme tabanlı modelleri kullanmaktadır. Fatayer ve Azara (2019), Khatun ve diğ. (2019), Soe ve diğ. (2019a) ve Wehbi ve diğ. (2019) çalışmalarında değerlendirme yalnızca doğruluk ölçütü üzerinden gerçekleştirilmiştir. Buna karşılık, Liang ve Znati (2019) tarafından yapılan çalışmada doğruluk yerine hassasiyet, kesinlik ve F1-skoru gibi daha kapsamlı performans kriterleri dikkate alınmıştır. Tüm çalışmalar birlikte değerlendirildiğinde, yalnızca Goyal ve diğ. (2019) tarafından yapılan çalışmada ise, doğruluk ölçütü açısından Destekçi Vektör Makineleri (Support Vector Machines, SVM) modelinin yapay sinir ağlarına (Artificial Neural Netrowks, ANN) modelinden daha iyi sonuç verdiği görülmektedir; ancak diğer performans ölçütlerinde ANN'nin üstünlük sağlaması nedeniyle, bu çalışmada da derin öğrenme modelinin genel performans açısından daha başarılı olduğu söylenebilir.

Genel olarak literatür incelendiğinde, derin öğrenme modellerinin klasik makine öğrenmesi yaklaşımlarına kıyasla daha yüksek performans sergilediği görülmektedir. Bu doğrultuda, yapay öğrenme tabanlı IDS uygulamalarında derin öğrenme modellerinin daha etkili sonuçlar sunduğu sonucuna varılabilir. Literatürde yaygın olarak kullanılan yöntemler incelendiğinde, derin öğrenme tarafında özellikle ANN, CNN, ve CNN ile birleşitirilmiş LSTM (uzun kısa zamanlı bellek) mimarilerinin; klasik makine öğrenmesi tarafında ise SVM ve Rastgele Orman (Random Forest, RF) modellerinin öne çıktığı görülmektedir.

Tablo 3.2'de ise daha çok saldırı tespit sistemleri içerisinde özellikler IoT üzerine yoğunlaşmış olan çalışmalar özetlenmiştir.

Tablo 3.2'de yer alan modelleme yaklaşımları incelendiğinde, ağırlıklı olarak ANN ve CNN modellerinin ön plana çıktığı; bunları LSTM, MLP (Multi Layered Perceptron), otokodlayıcı (Autoencoder), DBN (Deep Belief Networks) ve yinelemeli sinir ağları (RNN) gibi farklı derin öğrenme tabanlı modellerin takip ettiği görülmektedir. Makine öğrenmesi tarafında ise RF ve SVM modellerinin literatürde sıklıkla tercih edilen yöntemler arasında yer aldığı gözlemlenmiştir.

Tablo 3.2. Literatürdeki IoT-IDS Çalışmalarının Karşılaştırması

Referans	Yıl	Model	Veri Seti	Doğruluk (Acc.)
(Kasongo ve Sun, 2020)	2020	WFEU-FFDNN	UNSW-NB15	İkili sınıf: %99.66 Çoklu sınıf: %99.77
(Li ve diğ., 2020)	2020	CNN	NSL-KDD	İkili sınıf: %86.95 Çoklu sınıf: %81.33
(Ferrag ve diğ., 2020)	2020	ANN, RNN, CNN	CICIDS2017, BoT-IoT	CICIDS-CNN: %97.376 BoT-IoT-CNN: %98.371
(Ge ve diğ., 2019)	2019	FNN	BoT-IoT	İkili sınıf: %99.414 Çoklu sınıf: %82
(Lopez-Martin ve diğ., 2019)	2019	MLP, CNN, AdaBoost, SVM, RF Lineer Model + KA Transformasyon	NSL-KDD, UNSW-NB15	Lineer + KA: NSL-KDD: %82.6 UNSW-NB15: %89.8
(Nagisetty ve Gupta, 2019)	2019	MLP, CNN, ANN, Otokodlayıcı	UNSW-NB15, NSL-KDD	ANN (NSL-KDD): %99.21
(Liang ve diğ., 2019)	2019	ANN	NSL-KDD	%97
(Roy ve Cheung, 2018)	2019	Çift-yönlü LSTM (BLSTM RNN)	UNSW-NB15	%95
(Otoum ve diğ., 2022)	2019	SMO, SVM, KNN, DT	NSL-KDD	SMO: %99.02
(Ferrag ve Maglaras, 2019)	2019	RNN	CICIDS2017, BoT-IoT	CICIDS2017: %99.811 BoT-IoT: %99.822
(Zhang ve diğ., 2019)	2019	Genetik Algoritmayla optimize edilmiş DBN	NSL-KDD	%99
(Tian ve diğ., 2019)	2019	SVM ve Derin Otokodlayıcı	UNSW-NB15	AUC: 0.97
(Nsunza ve diğ., 2018)	2018	CNN	NSL-KDD	%82.83
(Pratomo ve diğ., 2018)	2018	Otokodlayıcı	UNSW-NB15	Z-score: 65%
(Zhou ve diğ., 2018)	2018	ANN	NSL-KDD, UNSW-NB15	NSL-KDD: %99.29 UNSW-NB15: %93.13

Literatürde gerçekleştirilen çalışmalar incelendiğinde, özellikle derin öğrenme tabanlı yaklaşımların büyük çoğunlukla son teknoloji yöntemler olarak benimsendiği görülmektedir. Bununla birlikte, bu çalışmaların önemli bir kısmının öznitelik mühendisliğinden ziyade model mimarisi ve modelleme odaklı yaklaşımlar benimsediği dikkat çekmektedir. Modern IoT veri setleri üzerinde gerçekleştirilen çalışmalarda, ön işleme süreçleri değerlendirildiğinde, genellikle kategorik özniteliklerin One-Hot

Encoding (OHE) yöntemiyle kodlandığı ve sayısal özniteliklerin Min–Max veya Z-skor normalizasyonu ile ölçeklendirildiği görülmektedir.

Klasik makine öğrenmesi tabanlı yaklaşımlar incelendiğinde, Ioannou ve Vassiliou (2019) çalışmasında verilerin yalnızca $[-1,1]$ aralığına normalize edilmesinin ardından SVM tabanlı bir modelin eğitildiği raporlanmıştır. Chaudhary ve Gupta (2019) ise özel olarak oluşturulan veri seti üzerinde paket boyutu ve paketler arası zaman gibi sınırlı sayıda öznitelik çıkarılmış; SVM, RF, LR (Lojistik Regresyon) ve DT (Karar Ağaçları) gibi modeller doğrudan karşılaştırılmıştır. Benzer şekilde, Kumar ve Lim (2019), yalnızca dört öznitelik kullanılarak herhangi bir ön işlembge adımı uygulanmaksızın farklı sınıflandırma modellerini kıyasladıkları görülmektedir.

Modern IoT veri setleri arasında yer alan BoT-IoT veri setini kullanan çalışmalarda, Soe ve diğ. (2019b), flag, proto ve state gibi kategorik özniteliklerin doğrudan sayısal değerlere dönüştürüldüğü ve ardından Z-skor normalizasyonunun uygulandığı belirtmiştir. Ge ve diğ. (2019) ise BoT-IoT veri setindeki bazı çakışmayan özniteliklerin birleştirildiği, kategorik değişkenlerin OHE ile kodlandığı ve sonrasında bir normalizasyon adımının uygulandığı ifade edilmekle birlikte, kullanılan yöntemin detayları açıkça belirtilmemiştir. Soe ve diğ. (2019a), N-BaIoT veri seti üzerinde yalnızca Min–Max normalizasyonu uygulanarak model eğitiminin gerçekleştirildiği raporlanmıştır.

CICIDS2017 veri setini kullanan çalışmalara bakıldığında, Roopak ve diğ. (2019), herhangi bir ön işleme adımı uygulanmadan CNN, MLP, LSTM, SVM ve RF gibi farklı modellerin doğrudan performans karşılaştırmasının yapıldığı görülmektedir. Diğer bir çalışmada ise, ön işleme adımının uygulandığı belirtilmiş olsa da detaylara yer verilmemiş, normalizasyon aşamasında ise $[0,1]$ aralığında Min–Max ölçekleme kullanılmıştır (Roopak ve diğ., 2020). Yazarlar tarafından oluşturulan özel bir veri seti üzerinde çeşitli öznitelikler çıkarılmış; ancak herhangi bir veri ön işleme adımına değinilmemiştir. Benzer şekilde, (Khatun ve diğ., 2019) çalışmasında, UNSW-NB15 veri seti yalnızca Z-normalizasyonu kullanılarak ölçeklendirilmiş ve ek bir ön işleme süreci uygulanmamıştır. Liang ve Znati (2019)' ise CICIDS2017 veri seti kullanılarak bir

DDoS tespit sistemi geliştirilmiş olsa da, ek bir veri ön işleme adımının uygulanmadığı belirtilmiştir. Lopez-Martin ve diğ. (2019), NSL-KDD ve UNSW-NB15 veri setleri üzerinde gerçekleştirilen çalışmada da doğrudan modelleme aşamasına geçildiği ve kapsamlı bir veri ön işleme sürecine yer verilmediği gözlemlenmiştir.

Bu çalışmalar topluca değerlendirildiğinde, literatürde genel olarak modelleme odaklı yaklaşımların baskın olduğu; veri ön işleme ve öznitelik mühendisliği süreçlerine ise sınırlı ölçüde odaklanıldığı sonucuna varılmaktadır. Son olarak, Tablo 3.1 ve Tablo 3.2 birlikte incelendiğinde, farklı çalışmalar arasında raporlanan başarım değerlerinin önemli ölçüde değişkenlik gösterdiği görülmektedir. Bu durumun temel nedenleri arasında, kullanılan veri setlerinin hem içerik hem de köken açısından farklılık göstermesi ve bazı çalışmalarda ön işleme adımlarının uygulanmasına karşın diğerlerinde bu adımların tamamen ihmal edilmesi yer almaktadır. Bu bağlamda, tablolarda vurgulanmak istenen temel nokta, literatürde model merkezli yaklaşımların ön planda olması ve veri ön işleme süreçleriyle ilgili belirgin eksikliklerin bulunmasıdır.

3.2. Araştırma Sorusu 1

Bir önceki bölümde tartışıldığı üzere, literatürdeki çoğu çalışma doğrudan model odaklı bir yaklaşımı tercih ederek, genellikle kompleks modeller ile, herhangi bir öznitelik mühendisiliği (feature engineering) yapmadan, doğrudan model mimarisi üzerine odaklanmıştır. Bu durum IoT-IDS kapsamında yapılacak çalışmalarda gereksiz yere, model karmaşıklığının yükselmesine, eğitim süresinin uzunluğuna bağlı olarak yeni model versiyonlarında gecikmeye yol açabilir. Dolayısıyla tez kapsamındaki ilk araştırma sorusu şu şekildedir:

- Araştırma Sorusu 1: İyi bir önişleme, eğitim-şeması ve öznitelik mühendisliği ile basit modeller ile hala derin öğrenme modellerine eş veya yakın performanslar alınabilir mi?

Öncelikli olarak, uygun bir eğitim test şeması tanımlayarak, modelin eğitim ve test aşamalarının yanlılıktan (bias) uzak olmasını amaçlıyoruz. Bu sayede modelin iyi öğrendiğinden emin olacak ve kuşkuların minimuma indirgeneceği bir model eğitim ve

değerlendirme aşaması tasarlıyoruz.

İkinci olarak da, model eğitimi öncesinde, özniteliklerin uygun olarak kendi başlarına ve birbirleriyle etkileşimlerini hesaba katarak, literatürde yer almayan ve detaylı, her özniteliğin kendi karakteristiğine uygun olacak, en iyi öznitelik ön işleme sürecini tespit etmeyi yine bu tez kapsamında amaçlıyoruz. Bu sayede bilgi kaybını en aza indiriecek ve modele sunulacak özniteliklerin kalitesini arttırmak tezin temel hedeflerinden biri olarak karşımıza çıkmaktadır.

Son olarak da, modeldeki kompleks özniteliklerin kullanılarak daha kompleks ilişkilerin ön plana çıkartılmasını amaçlanmıştır. Literatürdeki mevcut çalışmalar incelendiğinde, çoğunlukla mevcut özniteliklerin yapısının büyük ölçüde korunarak, görece standart ön işleme şemaları üzerinden ilerlenip, kullanılan makine öğrenmesi modelinin performansına odaklanıldığı görülmektedir. Bununla birlikte, öznitelik mühendisliği aşaması, özellikle sınıflar arası ayrılabilirliğin artırılması açısından kritik bir öneme sahiptir. Bu bağlamda, model tarafından öğrenilebilirliği yüksek ve aynı zamanda açıklanabilir nitelikte özniteliklerin tasarlanması büyük önem taşımaktadır. Bu tez kapsamında, çeşitli öznitelik oluşturma algoritmalarının problem alanına uygun biçimde uyarlanması ve geliştirilmesi yoluyla, daha zengin ve açıklanabilir öznitelik temsillerinin elde edilmesi hedeflenmiştir. Elde edilen bu özniteliklerin modelleme sürecine entegre edilmesiyle, sınıflar arası ayrımın güçlendirilmesi ve nispeten daha basit makine öğrenmesi modelleri kullanılarak, derin öğrenme tabanlı yaklaşımlarla karşılaştırılabilir ya da daha üstün performans seviyelerine ulaşılması amaçlanmıştır.

3.3. Araştırma Sorusu 2

IDS veri setlerinde genel olarak veri dengesizliği (data imbalance) sıklıkla karşılaşılan bir durum olup, sınıflar arasındaki örnek sayısının aşırı farklı olması durumunu ifade eder ve IoT güvenliği için geliştirilen modellerin performansını doğrudan etkileyen önemli bir faktördür.

Veri dengesizliği sınıflar arasında, 1:1000 oranına kadar oldukça yüksek seviyelere çıkabilmektedir (Manivannan, 2024). Örneğin BoT-IoT veri setinde (Koroniotis ve

diğ., 2019), toplamda 73.370.443 örneğin toplamda 73.360.900'ü saldırı iken, 9543'ü normal ağ akışına ait veriler içermektedir (Mohy-Eddine ve diğ., 2023). Diğer bir veri seti CICIDS2017 (Sharafaldin ve diğ., 2018) ise, yaklaşık %19,70 seviyesinde saldırı örneğine sahipken, yaklaşık olarak %80 oranında normal ağ akışı desenine sahiptir. CICIoT2023 (Neto ve diğ., 2023) için bu durum %97,65 saldırı, ve %2,75 normal (benign) şeklindedir. KDDCUP (KDD Cup, 1999) veri seti de yaklaşık olarak hem eğitim hem de test setinde %80'e yakın saldırı içerirken, %20 oranında normal ağ akışı verisine sahiptir.

Bu durumu bir makine öğrenmesi ile kurulmıuş IDS sisteminde şu şekilde problemlere yol açabilir:

- Gerçek Dünya Dinamikleri: Gerçek bir ağda normal trafik çoğunluklayken, nadiren saldırılar oluşur. Bu da saldırı verisinin toplanmasına yönelik çalışmaların gerçek dünya durumlarını tam ifade edememesine yol açabilir.
- Model Yanlılığı (Bias): Modeller çoğunlukla fazla sayıda olan sınıfa ait örnekleri daha iyi öğrenirken, azınlık sayısı çok azsa burada durumların kaçırmasına veya iyi öğrenememesine yol açabilir. Bu durumda, modelin bir sınıfı iyi öğrenmesine ve diğerlerini göz ardı etmesine (bias) yol açabilir. Bu azınlık sınıflarda aşağı öğrenme (underfitting) problemlerine yol açabilir.
- Yanlış Alarm Oranları (False Alarm Rate): Azınlık sınıflardan bilgi alınamadığında sistemler, normal sınıfı saldırı sanarken, saldırıları da normal sınıf örnekleri gibi değerlendirebilir. Bu durumda da model yanlış alarmlar vererek, kullanıcının pratikte bu sistemleri kullanabilirliğini olumsuz olarak etkileyebilir.

Bu kapsamda belirlenen araştırma sorusu şu şekildedir:

- Araştırma Sorusu 2: IoT-IDS sistemlerinde kullanılabilecek, veri dengesizliği problemini iyi bir şekilde çözebilen, bir yeniden örnekleme (resampling) algoritması dizayn edilebilir mi?

Literatürde pek çok yeniden örnekleme algoritması bulunmaktadır. Bu algoritmalar çeşitli şekillerde veri dengesizliğini gidermeye çalışsalar da temel olarak veri örneklemesi gerçekleştirirken, verinin dağılımsal ve istatistiksel karakterlerini dikkate

almamaktadırlar. Bu yöntemler uygulandığı veri setinin temel karakteristiğini bozabilir ve eğitilecek makine öğrenmesi modellerinin farklı desenleri yakalamasını güçleştirebilir. Yeniden örnekleme algoritmaları detaylı olarak Bölüm 6'da incelenecektir. Burada çözülmeye çalışılan problem, verinin dağılımını genel karakteristiğini koruyacak bir yeniden örnekleme algoritması dizayn edilmesidir. Bu sayede gelecekte yapılacak çalışmalar için, IoT-IDS problemlerinde oldukça kritik ve sık karşılaşılan bir problem olan veri dengesizliği probleminin, matematiksel temeli güçlü, veri karakteristiğini koruyacak, bu tez kapsamında tasarlanmış bir örnekleme algoritması ile giderilmesi amaçlanmaktadır.

3.4. Araştırma Sorusu 3

Literatürde yer alan saldırı tespit sistemi çalışmalarında sıklıkla bir veri setinde model eğitimi yapılırken, yine aynı veri seti üzerinde model başarısı test edilmektedir. Bunun temel sebebi ise, farklı çalışmalarda üretilen veri setlerinin farklı test ortamlarında (testbed) simüle edilmesi ve farklı ağ analiz araçlarının kullanılmasından ötürü tamamen farklı öznitelk uzaylarına, farklı saldırı sınıflarına ve dolayısıyla tamamen farklı veri setleri olmalarına yol açmaktadır. Buna istinaden tez kapsamında incelenecek üçüncü ve son araştırma sorusu şu şekildedir:

- Araştırma Sorusu 3: IoT-IDS sistemlerinde bütünleşik bir sınıflandırma modeli oluşturulabilir mi?

Bu tez kapsamında farklı veri setleri birleştirilerek, geleneksel makine öğrenmesi algoritmaları veya derin öğrenme yöntemleri ile birden fazla test ortamından elde edilmiş farklı veri setleri ile bütünleşik bir model yapılmaya çalışılarak, birleşmiş ve farklı ortamlardaki saldırılarda çalışabilecek daha global bir model oluşturabilir mi sorusuna cevap aranmıştır.

4. IOT AĞLARI İÇİN GELİŞMİŞ BİR SALDIRI TESPİT SİSTEMİ

Literatürde de görülebileceği üzere, IoT saldırı tespit sistemleri için, pek çok ileri seviye model ve teknikler, derin öğrenme tabanlı sistemlerin kurulmasına ve kompeks modellerin oluşturulması temeline dayanmaktadır. Ancak literatür incelendiğinde, pek çok çalışma, derin öğrenme üzerinde kurulduğu ve klasik makine öğrenme modellerinde oldukça standart yaklaşımlar ve temel öznitelik mühendisliği işlemleri gerçekleştirerek, doğrudan bir klasik öğrenme modellerinin kurulması şeklindedir. Bu bölümün temel motivasyonu, uygun eğitim-test şeması belirlendiğinde, detaylı ve titiz bir şekilde öznitelik mühendisliği yapıldığında, eğer gerekiyorsa yeni özniteliklerin var olan özniteliklerden oluşturulmasıyla, derin öğrenme modellerine yakın ve yer yer daha iyi çalışacak, daha basit, hesaplama maliyeti düşük, ve başarılı modellerin oluşturulabileceğidir. Dolayısıyla, uygun bir işlemler bütünü bir araya getirilerek, klasik makine öğrenmesi modelleriyle dahi, derin öğrenme ile yarışabilecek güçlü sistemlerin oluşturulabileceği gerçeği test edilmek istenmiştir. Buna istinaden, bu bölümde, klasik makine öğrenmesine dayalı bir IoT-IDS oluşturulabilmesi için, uygun mühendislik adımlarını uygulayan, derin öğrenme modelleri ile rekabet edebilecek, klasik makine öğrenmesine dayalı bir saldırı tespit sistemi akışı tasarlanmıştır.

4.1. Uygun Veri Setinin Seçilmesi

Makine öğrenmesi ve derin öğrenme gibi, veri odaklı çözümlerin en kritik noktalarından birisi, uygun veri setinin seçimidir. Bir veri setinde, ilgili probleme ait, farklı durumlara ait çeşitliliğin olması, yeterli sayıda veri noktası içermesi, dengeli bir veri yapısına sahip olunması, ve farklı senaryoların yeterince temsili, eğitilecek modellerin ne kadar fazla değişken durumu öğreneceğini ve ölçebileceğini doğrudan etkileyen bir faktördür. IoT saldırı tespit sistemi açısından değerlendirildiğinde, seçilecek veri setinin temel kriterleri aşağıdaki gibi özetlenebilir:

- Saldırı Çeşitliliği: Bir veri seti farklı saldırı çeşitlerine sahip olmalıdır. Bu daha fazla saldırı deseninin öğrenilmesini sağlar.
- Saldırı İçi Çeşitlilik: Farklı saldırılar için yeterince geniş sayıda veri noktası olmalıdır. Bu sayede bir saldırı türünün farklı alt desenlerinin öğrenilmesi mümkün

olabilir.

- Sınıf Dağılımının Dengesi: Veri setinde azınlık ve çoğunluk içeren farklı saldırı türleri arasında uçurum olması, azınlık saldırıların öğrenilmesini zorlaştırabilir.
- Toplam Kayıt Sayısı: Veri boyutu iyi bir eğitim yapmaya elverişli olacak şekilde geniş olmalıdır.
- Yaygınlık ve Bilinirlik: İyi bilinen, daha önce test edilmiş br veri seti tercih edilmelidir. Bu hem çalışmanın güvenilirliğini, hem de değerlendirme aşamasında diğer yöntemlerle kıyaslanabilirliğine doğrudan etki edecektir.

Bu kapsamda literatürde yer alan veri setleri araştırılmış ve veri seti seçimi nihai bir karara varılmıştır.

İlk aşamada, literatürde yaygın biçimde kullanılan ancak güncellik ve uygulama alanı açısından sınırlılıkları bulunan bazı veri setleri değerlendirme dışı bırakılmıştır. Bu kapsamda DARPA98 (McHugh, 2000), KDDCUP (KDD Cup, 1999), NSL-KDD (Tavallaee ve diğ., 2009), CICIDS2017 (Sharafaldin ve diğ., 2018), ve UNSW-NB15 (Moustafa ve Slay, 2015) veri setleri, ilk etapta elenen veri setleri olarak belirlenmiş ve bu bölümde ayrıntılı bir incelemeye dâhil edilmemiştir. Bu tercihin temel nedenleri; söz konusu veri setlerinin günümüz saldırı tekniklerini yeterince yansıtmaması, IoT odaklı senaryoları kapsamaması ve saldırı türleri açısından yeterli çeşitliliğe (diversity) sahip olmamasıdır. Bu tez çalışması kapsamında, bir IoT Sistemi için Saldırı tespit sistemi oluşturulması önceliklidir. Bu yüzden, IoT ortamlarından elde edilmiş veri setlerine odaklanılmasına karar verilmiştir. Bu nedenle, söz konusu veri setleri mevcut çalışmaların kapsamı dışında tutulmuştur. Bu çalışmada, literatürde yaygın olarak kullanılan ve IoT tabanlı saldırı tespit sistemleri için önerilen çeşitli güncel veri setleri detaylı biçimde incelenmiştir. İnceleme kapsamında ToN-IoT, BoT-IoT, IoT-23 ve N-BaIoT veri setleri ele alınmış; her bir veri seti mimari yapısı, saldırı çeşitliliği, veri dengesi ve deneysel uygunluğu açısından değerlendirilmiştir. Güncel ve ileri seviye IoT veri setleri ise Tablo 4.1'de sunulmaktadır.

ToN-IoT veri seti (Moustafa, 2021), heterojen ve karmaşık IoT senaryolarını temsil edebilecek şekilde tasarlanmış, dağıtık mimariye sahip kapsamlı bir veri setidir. Veri

setinin temel amacı, yetkin ve temsil gücü yüksek bir IoT veri seti oluşturmaktır.

Tablo 4.1. Çalışmada Değerlendirilen Güncel IoT Veri Setleri

Veri Seti	Kayıt Sayısı	Öznitelik Sayısı	Araç	Sınıf Sayısı
ToN-IoT (Moustafa, 2021)	223.390.210	46	Zeek	9
BoT-IoT (Koroniotis ve diğ., 2019)	73.370.443	48	Argus	6
IoT-23 (Garcia ve Plohmann, 2020)	325.397.990	23	Zeek	8
N-BaIoT (Meidan ve diğ., 2018)	849.234	115	–	2

Bu doğrultuda, edge, fog ve cloud katmanlarını içeren gerçekçi bir deney ortamı kurulmuş ve veri toplama süreci bu çok katmanlı yapı üzerinden gerçekleştirilmiştir. ToN kısaltması; Telemetri, İşletim Sistemi (Operating System) ve Ağ Trafiği (Network) kavramlarını temsil etmekte olup, veri seti bu üç temel kaynaktan elde edilen bilgilerle oluşturulmuştur. Çalışmada dört ana öznitelik grubu tanımlanmıştır: (i) bağlantı öznitelikleri (kaynak/hedef IP adresleri, port numaraları ve protokol bilgileri), (ii) istatistiksel öznitelikler (akış bazlı istatistikler), (iii) kullanıcı merkezli öznitelikler (DNS ve üst seviye servis etkileşimleri) ve (iv) ihlal öznitelikleri (paket iletimi sırasında gözlemlenen anormal davranışlar). Bu yapı sayesinde veri seti, IDS, kötü amaçlı yazılım tespiti ve adli analiz gibi yapay zekâ tabanlı güvenlik uygulamaları için oldukça zengin bir içerik sunmaktadır. ToN-IoT veri seti toplamda dokuz farklı saldırı türünü içermekte olup; Scanning, DoS, DDoS, Ransomware, Backdoor, Injection, XSS, Password Cracking ve Man-in-the-Middle (MITM) saldırıları bu kapsamda etiketlenmiştir.

BoT-IoT veri seti (Koroniotis ve diğ., 2019) ise, özellikle botnet tabanlı IoT saldırı senaryolarına odaklanan bir çalışmadır. Bu veri seti, IoT saldırılarını büyük veri problemi olarak ele almakta ve çeşitlilik, hız ve hacim olmak üzere üç temel unsuru gözeterek oluşturulmuştur. Mevcut veri setlerinde sıkça karşılaşılan; yetersiz etiketlenmiş veri, sınırlı saldırı çeşitliliği, trafik tekrarları ve eksik ground-truth gibi sorunlara çözüm sunmayı amaçlamaktadır. Öznitelik seçimi sürecinde Matthews Correlation Coefficient (MCC) ve entropi tabanlı değerlendirme yöntemleri kullanılmıştır. Argus aracıyla çıkarılan ağ akış öznitelikleri ve istatistiksel özellikler, kayan pencere (sliding window) yaklaşımıyla hesaplanarak uzun süreli saldırıların da yakalanması hedeflenmiştir. Veri setinde; tarama ve bilgi toplama saldırıları (servis tarama, işletim sistemi parmak izi),

DoS/DDoS saldırıları (TCP, UDP, HTTP) ve bilgi hırsızlığı saldırıları (veri hırsızlığı, keylogging) yer almaktadır.

IoT-23 veri seti (Garcia ve Plohmann, 2020), Avast AIC Laboratuvarı ve Avast Software tarafından desteklenen bir veri toplama projesi kapsamında oluşturulmuştur. 2018–2019 yılları arasında sekiz farklı zararlı yazılım senaryosu ve normal ağ trafiği toplanmıştır. Normal trafik, gerçek IoT cihazlarından (Philips Smart HUE LED, Amazon Echo ve Somfy akıllı kapı kilidi) elde edilmiştir. Zararlı senaryolar Raspberry Pi üzerinde gerçekleştirilmiş ve her saldırı için uzun süreli (yaklaşık 24 saat) veri toplanmıştır. Veri seti gerçekçi olması açısından değerli olmakla birlikte, literatürde çoğunlukla referans amaçlı kullanıldığı ve doğrudan model eğitimi için sınırlı tercih edildiği gözlemlenmiştir.

N-BaIoT veri seti (Meidan ve diğ., 2018) ise, belirli bir saldırı tespit problemi için veri seti üretmekten ziyade, IoT cihazlar üzerinde otomatik kodlayıcılar kullanarak bir IDS geliştirmeyi amaçlayan öncü bir çalışmanın parçası olarak oluşturulmuştur. Veri seti Mirai ve Bashlite olmak üzere iki saldırı türünü içermektedir. Dokuz farklı gerçek IoT cihazından elde edilen veriler üzerinden, farklı zaman pencerelerinde hesaplanan istatistiksel ölçümlerle toplam 115 öznitelik üretilmiştir. Ancak saldırı çeşitliliğinin sınırlı olması sebebiyle kapsamlı karşılaştırmalar için yeterli görülmemiştir.

Yapılan değerlendirmeler sonucunda, bu çalışmada kullanılacak birincil veri seti olarak ToN-IoT tercih edilmiştir. BoT-IoT veri setinde normal trafik kayıtlarının son derece az olması (yaklaşık 400 kayıt) ciddi bir sınıf dengesizliği problemine yol açmakta ve normal ile saldırı trafiğinin ayırt edilmesini güçleştirmektedir. IoT-23 veri seti literatürde daha çok betimleyici analizlerde kullanıldığından, doğrudan model karşılaştırmaları için uygun bulunmamıştır. N-BaIoT ise yalnızca iki saldırı türü içermesi nedeniyle elenmiştir. ToN-IoT veri seti; daha dengeli sınıf dağılımı, daha geniş saldırı çeşitliliği, uç-sis-bulut (edge–fog–cloud) mimarisini kapsayan zengin test ortamı ve güncelliği sayesinde bu çalışmanın hedefleriyle en uyumlu veri seti olarak belirlenmiştir.

Veri seti üzerinde gerçekleştirilecek analizlerde, öncelikle veri setine ait temel bilgilerin (künye) bilinmesi, kullanılan özniteliklerin doğru şekilde anlaşılması ve keşifçi veri analizi sürecinin sağlıklı bir biçimde yürütülebilmesi açısından kritik bir öneme sahiptir.

Bu nedenle, bu bölümde çalışmada kullanılan ToN-IoT veri setinin künye bilgileri ayrıntılı olarak incelenmiştir. Burada sunulan künye bilgileri, veri setinin orijinal dokümantasyonunda yer alan bilgilerden derlenmiştir (Moustafa, 2021).

ToN-IoT veri seti, ikisi etiket bilgisi olmak üzere toplam 46 adet öznitelikten oluşmaktadır. Bu öznitelikler, IoT ortamlarında gerçekleşen ağ trafiğini ve sistem davranışlarını çok boyutlu bir şekilde temsil edecek biçimde tasarlanmıştır. Aşağıda, veri setinde yer alan öznitelik grupları ve bu gruplara ait temel açıklamalar sunulmaktadır.

Bağlantı Öznitelikleri (Connection Features): Ağ akışlarını tanımlayan temel bilgileri içermektedir. Bu öznitelikler; kaynak ve hedef IP adresleri, bağlantı noktaları (portlar), kullanılan protokol türleri gibi akışa özgü tanımlayıcı nitelikleri kapsamaktadır. İlgili bağlantı öznitelikleri Tablo 4.2'de ayrıntılı olarak gösterilmiştir.

Tablo 4.2. Bağlantı Öznitelikleri

Öznitelik	**Tip**	**Açıklama**
ts	Zaman (Time)	Bağlantı akışına ait zaman damgası.
src_ip	Metin (String)	Kaynak IP adresi.
src_port	Sayısal (Number)	Kaynak TCP/UDP port numarası.
dst_ip	Metin (String)	Hedef IP adresi.
dst_port	Sayısal (Number)	Hedef TCP/UDP port numarası.
proto	Metin (String)	Akış sırasında kullanılan taşıma katmanı protokolü (TCP, UDP vb.).
service	Metin (String)	DNS, HTTP ve SSL gibi dinamik olarak tespit edilen uygulama katmanı servisleri.
duration	Sayısal (Number)	İlk ve son paket arasındaki zaman farkı.
src_bytes	Sayısal (Number)	Kaynak tarafından gönderilen toplam bayt sayısı.
dst_bytes	Sayısal (Number)	Hedef tarafından alınan toplam bayt sayısı.
conn_state	Metin (String)	Bağlantı durumu bilgisi (S0: bağlantı tamamlanmadı, S1: bağlantı sağlandı, REJ: bağlantı isteği reddedildi).
missed_bytes	Sayısal (Number)	İçerik boşluklarında eksik olan bayt sayısı.

İstatistiksel Öznitelikler (Statistical Features): Ağ akışlarına ait temel tanımlayıcıların yanı sıra bu tanımlayıcılardan türetilen istatistiksel ölçümleri içeren öznitelik grubudur. Bu öznitelikler, paket ve bayt bazlı dağılımlar üzerinden ağ trafiğinin davranışsal

özelliklerini yansıtmaktadır. İlgili özniteliklerin detayları Tablo 4.3'te sunulmuştur.

Tablo 4.3. İstatistiksel Öznitelikler

Öznitelik	**Tip**	**Açıklama**
src_pkts	Sayısal (Number)	Kaynak sistem tarafından gönderilen paket sayısı.
src_ip_bytes	Sayısal (Number)	Kaynak IP başlık alanlarının toplam bayt uzunluğu.
dst_pkts	Sayısal (Number)	Hedef sistem tarafından alınan paket sayısı.
dst_ip_bytes	Sayısal (Number)	Hedef IP başlık alanlarının toplam bayt uzunluğu.

Kullanıcı Merkezli Grup (User-Centric Group): Son kullanıcılar ile doğrudan etkileşim halinde olan üst seviye ağ servislerine ilişkin bilgileri içeren öznitelik olup, Tablo 4.4'te listelenmiştir.

Tablo 4.4. Kullanıcı Merkezli Grup Öznitelikleri

Öznitelik	**Tip**	**Açıklama**
dns_query	Metin	DNS sorgusuna ait alan adı bilgisi.
dns_qclass	Sayısal	DNS sorgu sınıfını temsil eden değer.
dns_qtype	Sayısal	DNS sorgu türü.
dns_rcode	Sayısal	DNS yanıt kodu.
dns_AA	Mantıksal	DNS yetki bayrağı.
dns_RD	Mantıksal	Özyinelemeli DNS isteği bilgisi.
dns_RA	Mantıksal	DNS sunucusunun özyinelemeyi destekleme durumu.
dns_rejected	Mantıksal	DNS sorgusunun reddedilip reddedilmediği bilgisi.
ssl_version	Metin	SSL sürüm bilgisi.
ssl_cipher	Metin	SSL şifreleme paketi.
ssl_resumed	Mantıksal	SSL oturumunun yeniden kullanımı.
ssl_established	Mantıksal	SSL bağlantısının kurulup kurulmadığı bilgisi.
ssl_subject	Metin	Sertifika konu alanı.
ssl_issuer	Metin	Sertifikayı imzalayan otorite bilgisi.
http_method	Metin	HTTP istek metodu.
http_status_code	Sayısal	HTTP durum kodu.
http_user_agent	Metin	HTTP user-agent bilgisi.

İhlal Nitelikleri (Violation Attributes): Paket iletimi sırasında ortaya çıkan olağan dışı durumlar, protokol sapmaları ve anomali göstergelerine ilişkin bilgileri içeren öznitelik grubudur. Bu nitelikler, özellikle yapay zekâ tabanlı güvenlik uygulamalarında (saldırı tespit sistemleri, kötü amaçlı yazılım analizi ve adli bilişim çalışmaları gibi) kritik öneme sahiptir. İlgili ihlal özniteliklerine ait ayrıntılar Tablo 4.5'te sunulmaktadır.

Tablo 4.5. İhlal Öznitelikleri

Öznitelik	**Tip**	**Açıklama**
weird_name	Metin	Tespit edilen protokol anomalisinin adı.
weird_addl	Metin	Anomaliye ilişkin ek açıklamalar.
weird_notice	Mantıksal	Anomali durumunun bildirim oluşturup oluşturmadığı.

Burada açıklanan bağlantı, istatistiksel, kullanıcı merkezli ve ihlal özniteliklerine ek olarak, veri setinde denetimli öğrenme sürecinde kullanılan etiket bilgileri de yer almaktadır. Bu etiket öznitelikleri, her bir ağ akışının normal ya da saldırı içerip içermediğini ve saldırı durumunda hangi saldırı türüne ait olduğunu ifade etmektedir. Modelin hem ikili (binary) hem de çok sınıflı (multi-class) sınıflandırma görevlerinde eğitilebilmesi ve değerlendirilmesi açısından bu etiket bilgileri kritik öneme sahiptir. İlgili etiket özniteliklerine ait detaylar Tablo 4.6'da sunulmuştur.

Tablo 4.6. Etiket Öznitelikleri

Öznitelik	**Açıklama**
label	0 değeri normal ağı, 1 değeri saldırı içeren ağı temsil eder.
type	Saldırı türü kategorisini belirtir (DoS, DDoS, vb.).

Bu bölümde, ToN-IoT veri setine ait öznitelik grupları ayrıntılı bir biçimde incelenmiş, bağlantı, istatistiksel, kullanıcı merkezli, ihlal ve etiket özniteliklerinin içerdiği bilgiler açıklanmıştır. Bu inceleme, veri setinin yapısının anlaşılması, özniteliklerin anlamlandırılması ve sonraki aşamalarda uygulanacak ön işleme ve modelleme adımlarının sağlıklı bir şekilde kurgulanabilmesi açısından kritik bir temel oluşturmaktadır.

Sonraki bölümlerde, model oluşturmanın ilk adımı olarak veri setinin eğitim ve test

kümelerine ayrılması ele alınacaktır. Bu kapsamda, veri sızıntısını (data leakage) önlemek ve modelin genellenebilirliğini arttırmak için uygulanan ayrım stratejileri açıklanacak; model eğitimi öncesinde izlenen yöntemler ve eğitim sürecine dair detaylar sunulacaktır.

4.2. Eğitim ve Test Verisinin Hazırlanması

Makine öğrenmesi problemlerinde, uygun eğitim ve test veri setlerinin oluşturulması, model başarımını ve elde edilen sonuçların güvenilirliğini doğrudan etkileyen en kritik aşamalardan biridir. Eğitim ve test kümelerinin hatalı biçimde ayrıştırılması; veri sızıntısına (data leakage), sınıflar arası dağılım dengesizliğinin bozulmasına ve gerçek dünya koşullarını yansıtmayan iyimser performans tahminlerine yol açabilmektedir. Özellikle zamansal, hiyerarşik ya da doğal gruplar içeren veri setlerinde, rastgele ve kontrolsüz bölme işlemleri, aynı gözlem grubuna ait örneklerin hem eğitim hem de test kümelerinde yer almasına neden olarak modelin genelleme yeteneğini yapay biçimde artırabilmektedir. Bunun sonucunda, modelin test verisi üzerindeki başarımı gerçeği yansıtmamakta ve dağılım kaymaları ya da sınıf dengesizlikleri karşısında modelin performansı ciddi biçimde düşebilmektedir. Bu nedenle, veri setlerinin bölünme sürecinde sınıf dağılımlarının korunması ve olası grup bağımlılıklarının dikkate alınması, güvenilir ve tekrarlanabilir makine öğrenmesi modelleri geliştirilmesi açısından hayati önem taşımaktadır.

Makine öğrenmesi modellerinin güvenilir biçimde eğitilebilmesi ve elde edilen performans sonuçlarının gerçeği yansıtabilmesi için, eğitim ve test veri setlerinin oluşturulma sürecinde herhangi bir yanlılık (bias) oluşmadığından emin olunması büyük önem taşımaktadır. Buna istinaden literatürde de çokça kullanılan, grup temelli rastgele bölümleme (group shuffle split) kullanılmıştır (Sklearn, 2016). Makine öğrenmesi modellerinin güvenilir biçimde eğitilebilmesi ve elde edilen performans sonuçlarının gerçeği yansıtabilmesi için, eğitim ve test veri setlerinin oluşturulma sürecinde herhangi bir yanlılık (bias) oluşmadığından emin olunması büyük önem taşımaktadır. Grup temelli rastgele bölümlee metodolojisi, aynı gruba ait kayıtların hem eğitim hem de test veri setinde eş zamanlı olarak yer almasını engelleyerek, veri sızıntısı (data leakage) riskini ortadan kaldırmayı hedeflemektedir. Bu kapsamda, her bir grup yalnızca tek bir veri

kümesine atanmakta; başka bir ifadeyle, bir gruba ait tüm kayıtlar ya eğitim veri setinde ya da test veri setinde bulunmaktadır. Bu yapı sayesinde, modelin daha önce gördüğü grup özelliklerini test aşamasında dolaylı olarak yeniden kullanmasının önüne geçilmekte ve değerlendirme süreci tamamen bağımsız gruplar üzerinden gerçekleştirilmektedir.

Bu çalışmada, grup-temelli veri bölme yaklaşımının etkin bir biçimde uygulanabilmesi amacıyla, veri setindeki her bir kayıt için ait olduğu grup tanımı, ağ iletişimini temsil edecek şekilde özel olarak oluşturulmuştur. Bu kapsamda, her bir örneğin grup bilgisi, Denklem (4.1)'de gösterilmiştir:

$$src_port : src_ip \rightarrow dst_port : dst_ip \tag{4.1}$$

Burada, src_ip ve src_port öznitelikleri sırasıyla kaynak makinenin IP adresi ve port bilgisini temsil ederken, dst_ip ve dst_port öznitelikleri ise hedef makineye ait IP adresi ve port bilgisini ifade etmektedir. Bu yapı sayesinde, bir kaynak–hedef makine çifti arasında belirli bir port üzerinden gerçekleşen her bir veri iletim kanalı, benzersiz bir grup olarak modellenmiştir.

Bu grup tanımının temel motivasyonu, ağ trafiğinde aynı IP ve port çiftleri üzerinden gerçekleşen iletişimlerin, içerik ve davranış örüntüleri bakımından, özellikle ToN-IoT gibi kontrollü ortamlarda oluşturulmuş olan test ortamlarında (testbed), yüksek düzeyde benzerlik göstermesidir. Dolayısıyla, aynı iletişim kanalına ait kayıtların hem eğitim hem de test veri setinde yer alması, modelin test aşamasında daha önce öğrendiği iletişim örüntülerini dolaylı olarak yeniden kullanmasına ve bu durumun performans metriklerini yapay biçimde yükseltmesine yol açabilmektedir. Bu tür bir senaryo, özellikle saldırı tespit problemlerinde modelin gerçek genelleme kabiliyetini maskeleyen önemli bir yanlılık kaynağı olarak değerlendirilmektedir.

Bu çalışmada, söz konusu yanlılığın önüne geçebilmek amacıyla, her bir iletişim kanalı yalnızca eğitim ya da yalnızca test veri setinde yer alacak şekilde konumlandırılmıştır. Böylece, test aşamasında modelin, eğitim sürecinde hiç karşılaşmadığı kaynak–hedef–port kombinasyonları üzerinden değerlendirilmesi sağlanmış ve modelin tamamen bağımsız iletişim kanalları üzerindeki başarımı

ölçülmüştür. Bu yaklaşım, geliştirilen saldırı tespit sisteminin gerçek dünya senaryolarındaki yeni ve daha önce gözlemlenmemiş ağ iletişimleri karşısındaki performansını daha güvenilir ve tarafsız bir biçimde analiz etmeye olanak tanımaktadır.

Veri setinde yer alan ip-port çiftlerine ait bir kesit Tablo 4.7'de gösterilmiştir.

Tablo 4.7. Örnek IP-Port Kayıtları

ts	src_ip	src_port	dst_ip	dst_port
1554198358	3.122.49.24	1883	192.168.1.152	52976
1554198358	192.168.1.79	47260	192.168.1.255	15600
1554198359	192.168.1.152	1880	192.168.1.152	51782
1554198359	192.168.1.152	34296	192.168.1.152	10502

Önreğin Tablo 4.7'te ki ilk kayıt için bağlı olduğu grup 3.122.49.24 : 1883_192.168.1.152 : 52976 olacaktır. Benzer şekilde, 2. satır içinde 192.168.1.79 : 47260_192.168.1.255 : 15600 şeklinde hesaplanır. Bu şekilde bir şema ile aynı ip:port çiftlerinin sadece eğitim veya test veri setinde olmasını sağlayarak aynı iletim kanalıyla eğittiğimiz modelimizin aynı iletim kanalıyla test edilmesinin önüne geçerek, veri sızıntısı ve yanlılık problemlerinin azaltılmasına katkı gösterir.

Eğitim sürecinde de grup temelli bölme yaklaşımı korunarak 10-kat çapraz geçerleme uygulanmıştır. Bu yöntemde veri her iterasyonda 10 parçaya ayrılmakta; 9 parça eğitim, 1 parça doğrulama (validation) için kullanılmaktadır. Toplam 10 iterasyon sonunda elde edilen doğrulama performanslarının ortalaması alınarak model değerlendirilmiştir. Eğitim aşamasında da yanlılığı önlemek amacıyla veri katlara ayrılmış ve ek olarak tabakalı bölme (stratification) uygulanarak her bir katmanda sınıf oranlarının korunması sağlanmıştır.

Örneğin iki sınıflı bir problemde veri setinde %60'a %40 oranında bir sınıf dağılımı düşünüldüğünde, bu oranın her bir eğitim kat grubu, değerlendirme grubu ve test veri setinde benzer dağılım göstermesi sağlanmaktadır. Burada bahsedilen tabakalı bölme işlemi çapraz geçerlemenin her adımında hem eğitimin yapıldığı 9 parçada %60'a %40 oranı korunurken hem de valide edilecek değerlendirme setinde de %60'a %40 oranı korunarak eğitim yapılmasıdır. Bu şekilde verinin her bölümündeki (her 10-kat

için) dağılımında her bir parça için benzer olması amaçlanmıştır. Buna istinaden, tez kapsamında kullanılan tabakalı grup-temelli k-katlı çapraz geçerleme yaklaşımının sözde kodu Algoritma 1'de gösterilmiştir.

Algoritma 1: Tabakalı Grup-Temelli K-Katlı Çapraz Geçerleme (Stratified Group K-Fold)

Gerekli:

X: Özniteliklerden oluşan veri çerçevesi ($n \times m$)
y: Sınıf etiketleri vektörü (n)
g: Grup etiketleri vektörü (j)
K: Kat (fold) sayısı (bu çalışmada $K = 10$)

Ensure:

$\{(X_{\text{train}}^{(k)}, y_{\text{train}}^{(k)}), (X_{\text{val}}^{(k)}, y_{\text{val}}^{(k)})\}_{k=1}^{K}$

Başlatma:

$G \leftarrow \{g_1, g_2, \ldots, g_M\}$
$F_1, \ldots, F_K \leftarrow \emptyset$

Her grup için sınıf histogramını hesapla:

for each group $g \in G$ **do**
    for each class $c \in \{1, \ldots, C\}$ **do**
        $h_{g,c} \leftarrow \sum_{i=1}^{n} \mathbb{I}(g_i = g \wedge y_i = c)$
    end for
    $\mathbf{h}_g \leftarrow [h_{g,1}, h_{g,2}, \ldots, h_{g,C}]$
end for

Global sınıf oranlarını hesapla:

for each class $c \in \{1, \ldots, C\}$ **do**
    $p_c \leftarrow \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(y_i = c)$
end for
$\mathbf{p} \leftarrow [p_1, p_2, \ldots, p_C]$

Grupları tabakalı biçimde katlara ata:

for each group $g \in G$ **do**
    $k^* \leftarrow \arg\min_{k \in \{1,\ldots,K\}} \left\| \frac{\mathbf{H}_k + \mathbf{h}_g}{\|\mathbf{H}_k + \mathbf{h}_g\|_1} - \mathbf{p} \right\|_2$
    $F_{k^*} \leftarrow F_{k^*} \cup \{g\}$
    $\mathbf{H}_{k^*} \leftarrow \mathbf{H}_{k^*} + \mathbf{h}_g$
end for

K-katlı çapraz geçerleme bölmelerini oluştur:

for $k = 1$ **to** K **do**
    $X_{\text{val}}^{(k)} \leftarrow \{x_i : g_i \in F_k\}$
    $y_{\text{val}}^{(k)} \leftarrow \{y_i : g_i \in F_k\}$
    $X_{\text{train}}^{(k)} \leftarrow X \setminus X_{\text{val}}^{(k)}$
    $y_{\text{train}}^{(k)} \leftarrow y \setminus y_{\text{val}}^{(k)}$
end for

Algoritmanın girdileri; $n \times m$ boyutundaki öznitelik matrisini temsil eden X, her bir örneğe karşılık gelen sınıf etiketlerini içeren y vektörü, her örneğin ait olduğu iletişim kanalını tanımlayan grup etiketleri vektörü g ve çapraz geçerleme sürecinde kullanılacak kat sayısını ifade eden K parametresidir. Bu çalışmada kat sayısı $K = 10$ olarak belirlenmiştir. Algoritmanın çıktısı ise her bir kat için oluşturulan eğitim ve doğrulama veri kümelerinden oluşan $\{(X_{\text{train}}^{(k)}, y_{\text{train}}^{(k)}), (X_{\text{val}}^{(k)}, y_{\text{val}}^{(k)})\}_{k=1}^{K}$ çiftleridir.

Başlatma adımında, veri setinde yer alan tüm benzersiz grup etiketleri $G = \{g_1, g_2, \ldots, g_M\}$ kümesi altında toplanmaktadır. Burada M, veri setindeki benzersiz grup sayısını ifade etmekte olup genel durumda $M \leq n$ koşulu sağlanmaktadır. Ardından, her bir kat için grupların atanacağı boş grup kümeleri $F_1, \ldots, F_K$ oluşturulmaktadır.

Algoritmanın devamında, her bir grup için sınıf dağılımını temsil eden histogram vektörleri hesaplanmaktadır. Bu kapsamda, her bir grup $g \in G$ ve her bir sınıf etiketi $c \in \{1, \ldots, C\}$ için, ilgili gruba ve sınıfa ait örnek sayısı $h_{g,c}$ olarak tanımlanmaktadır. Elde edilen bu değerler kullanılarak, her grup için sınıf dağılımını temsil eden $\mathbf{h}_g$ vektörü oluşturulmaktadır. Bu adım, grupların sınıf yapılarının açık biçimde modellenmesini sağlamaktadır.

Bir sonraki aşamada, tüm veri seti üzerindeki global sınıf oranları hesaplanmaktadır. Bu amaçla, her bir sınıf için veri setindeki örnek oranı p_c olarak belirlenmekte ve bu oranlar $\mathbf{p} = [p_1, p_2, \ldots, p_C]$ vektörü altında toplanmaktadır. Elde edilen bu global dağılım, katlara yapılacak grup atamalarında referans dağılım olarak kullanılmaktadır.

Grupların katlara atanması aşamasında, her bir grup $g \in G$ için, grubun hangi kata atanması durumunda sınıf dağılımlarının global dağılıma en yakın olacağı hesaplanmaktadır.

Bu amaçla, her bir kat için mevcut sınıf histogramı $\mathbf{H}_k$ ile ilgili grubun histogramı $\mathbf{h}_g$ birleştirilmekte ve ortaya çıkan dağılımın global sınıf oranlarından sapması ℓ_2 normu kullanılarak ölçülmektedir. Söz konusu sapmayı minimize eden kat k^* seçilerek, ilgili grup bu kata atanmakta ve katın sınıf histogramı güncellenmektedir. Bu süreç sayesinde, hem grup bütünlüğü korunmakta hem de sınıf oranlarının katlar arasında dengeli biçimde

dağılması sağlanmaktadır.

Son adımda, oluşturulan katlara göre k-katlı çapraz geçerleme bölmeleri tanımlanmaktadır. Her bir iterasyon için, ilgili kata atanmış gruplara ait örnekler doğrulama veri seti olarak belirlenirken, geri kalan tüm örnekler eğitim veri setini oluşturmaktadır. Bu yapı sayesinde, aynı gruba ait örneklerin eğitim ve doğrulama aşamalarında eş zamanlı olarak yer almasının önüne geçilmekte ve modelin performansı, daha önce görülmemiş iletişim kanalları üzerinden değerlendirilmektedir.

Özetle, Algoritma 1, grup-temelli veri sızıntısını engelleyen ve sınıf dağılımlarını koruyan bütüncül bir çapraz geçerleme stratejisi sunmakta; geliştirilen saldırı tespit modelinin genelleme kabiliyetinin yanlılıktan arındırılmış ve güvenilir bir biçimde ölçülmesine olanak tanımaktadır.

Bu işlemler sonucunda ise, yanlılıktan uzak, hem katlar hem de eğitim ve test veri setleri arasındaki, sınıf dağılımını koruyan bir yapı elde edilmiştir.

4.3. Kayıp Verilerin Analizi

ToN_IoT veri seti üzerinde gerçekleştirilen keşifsel veri analizi (Exploratory Data Analysis, EDA) sürecinin ilk adımlarından biri olarak, veri setindeki kayıp değerlerin yapısı ve dağılımı incelenmiştir. Kayıp verilerin uygun şekilde ele alınmaması, makine öğrenmesi modellerinin performansını olumsuz etkileyebilmekte; yanlı tahminlere, hatalı çıkarımlara ve genelleme kabiliyetinin zayıflamasına yol açabilmektedir. Literatürde yapılan çalışmalar, kayıp verilerin göz ardı edilmesinin sınıflandırma doğruluğunu düşürdüğünü, varyans tahminlerini bozduğunu ve model sonuçlarının güvenilirliğini azalttığını açıkça ortaya koymaktadır (Akbar ve diğ., 2025; Choudhury ve Kosorok, 2020; Karlaš ve diğ., 2025; Zhou ve diğ., 2024).

İncelenen ToN_IoT veri setinde kayıp değerler, özgün veri formatında ’-’ sembolü ile temsil edilmektedir. Ön işleme aşamasında bu semboller, standart eksik değer temsili olan null değerlerine dönüştürülmüştür. Ardından öznitelikler, kayıp değer oranlarına göre iki ana grupta sınıflandırılmıştır: yüksek oranda kayıp içeren öznitelikler (kayıp oranı %95’in üzerinde olanlar) ve orta düzeyde kayıp içeren öznitelikler (Patel, 2016). Bu

ayrım, kayıp verilerin doğrudan elenmesi yerine, potansiyel bilgi içeriklerinin sistematik olarak değerlendirilmesini amaçlamaktadır.

Yüksek oranda kayıp içeren öznitelikler, ilk aşamada doğrudan veri setinden çıkarılmamıştır. Bunun yerine, bu özniteliklerdeki kayıp olma durumunun hedef değişken (Normal-Benign / Saldırı-Attack) ile anlamlı bir ilişki içerip içermediği analiz edilmiştir. Bu amaçla, her bir yüksek kayıp oranına sahip öznitelik için ikili bir gösterge (indicator) değişkeni oluşturulmuştur. Bu gösterge değişkenleri, ilgili öznitelikteki değerin kayıp olması durumunda 1, aksi durumda 0 değerini almaktadır. Örneğin, ssl_version özniteliği için ssl_version_indicator adlı yeni bir değişken türetilmiştir.

Oluşturulan gösterge değişkenleri ile ikili hedef etiket arasındaki ilişki, Pearson korelasyon katsayısı kullanılarak hesaplanmıştır. Pearson korelasyonu, ikili değişkenler arasındaki doğrusal ilişkinin ölçülmesinde uygun bir yöntem olarak literatürde yaygın biçimde kullanılmaktadır (Moraes ve diğ., 2024). Elde edilen korelasyon değerleri Tablo 4.8'de sunulmaktadır.

Tablo 4.8. Yüksek Kayıplı Öznitelikler için Hedef Korelasyonlar

Öznitelik	**Korelasyon**
ssl_version_indicator	0,016
ssl_cipher_indicator	0,016
ssl_resumed_indicator	0,025
ssl_established_indicator	0,025
ssl_subject_indicator	0,001
ssl_issuer_indicator	0,001
http_trans_depth_indicator	0,004
http_method_indicator	0,003
http_uri_indicator	0,003
http_version_indicator	0,005

Tablo 4.8 incelendiğinde, yüksek oranda kayıp içeren özniteliklere ait tüm gösterge değişkenlerinin hedef etiket ile olan korelasyonlarının oldukça düşük olduğu (korelasyon < 0.1) görülmektedir. Bu bulgu, söz konusu özniteliklerdeki kayıp olma durumunun saldırı veya normal sınıf ayrımında ayırt edici bir bilgi taşımadığını göstermektedir. Bu

nedenle, yüksek oranda kayıp içeren bu öznitelikler veri setinden tamamen çıkarılmıştır.

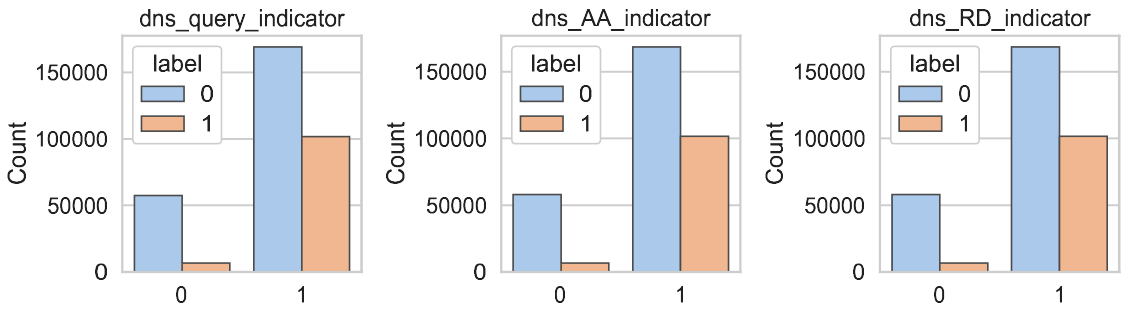
Bir sonraki aşamada, orta düzeyde kayıp içeren öznitelikler için benzer bir analiz gerçekleştirilmiştir. Bu özniteliklere ait gösterge değişkenleri ile hedef etiket arasındaki korelasyon değerleri Tablo 4.9'da sunulmaktadır. Tabloya göre, service alanına ait gösterge değişkeninin hedef değişken ile olan korelasyonu görece düşük bulunmuş ve bu öznitelik de veri setinden çıkarılmıştır.

Tablo 4.9. Orta Düzeyde Kayıp Özniteliklerin Hedef Korelasyonu

Öznitelik	**Korelasyon**
service_indicator	0,089
dns_query_indicator	0,228
dns_AA_indicator	0,229
dns_RD_indicator	0,229
dns_RA_indicator	0,229
dns_rejected_indicator	0,229

Buna karşın, DNS ile ilişkili özniteliklerin kayıp olma durumlarına ait gösterge değişkenlerinin hedef değişken ile orta düzeyde korelasyon değerleri sergilediği gözlemlenmiştir. Bu durum, DNS tabanlı özelliklerdeki eksikliklerin saldırı davranışları ile ilişkili olabileceğine işaret etmektedir. Bu nedenle, DNS ile ilişkili gösterge değişkenleri doğrudan elenmemiş; daha ayrıntılı bir inceleme yapılması amacıyla hedef etiketin bu özniteliklerin varlık/yokluk durumlarına göre dağılımı ayrıca analiz edilmiştir. Bu dağılımlar Şekil 4.1'de sunulmaktadır.

Sonuç olarak, bu aşamada uygulanan sistematik kayıp veri analizi sayesinde, hem bilgi



Şekil 4.1. DNS İndikatör Değerleri için Sınıf Dağılımları

içeriği düşük olan özniteliklerin veri setinden elenmesi sağlanmış hem de saldırı tespiti açısından potansiyel olarak anlamlı olabilecek kayıp örüntüleri korunmuştur. Bu yaklaşım, gereksiz boyut artışının önüne geçerken, modelin genelleme kabiliyetini destekleyen daha tutarlı ve bilgi açısından zengin bir öznitelik uzayı oluşturulmasına katkı sağlamaktadır.

4.4. Sabit ve Kısmi-Sabit Öznitelikler

Makine öğrenmesi tabanlı sistemlerde dikkate alınması gereken bir diğer önemli husus, sabit (constant) veya kısmi-sabit (quasi-constant) özniteliklerin varlığıdır. Bu tür öznitelikler, ya tamamen sıfır varyansa sahip olmakta ya da çok düşük varyans göstererek sınıflar arasında ayırt edici bir bilgi sunmamaktadır. Literatürde, sabit veya kısmi-sabit özniteliklerin modele dâhil edilmesinin gereksiz öznitelik fazlalığına yol açtığı, hesaplama maliyetini artırdığı ve modelin genelleme kabiliyetini olumsuz etkileyebileceği belirtilmektedir (Chandrashekar ve Sahin, 2014).

ToN_IoT veri seti özelinde, kısmi-sabit özniteliklerin tespit edilmesi amacıyla %95 eşik değeri kullanılmıştır. Buna göre, bir öznitelikteki en sık tekrar eden değerin tüm gözlemlerin %95'inden fazlasını oluşturması durumunda, ilgili öznitelik kısmi-sabit olarak tanımlanmıştır. Tüm örneklerde aynı değeri alan öznitelikler ise sabit öznitelikler olarak değerlendirilmiştir. Yapılan analizler sonucunda, veri setinde tamamen sabit herhangi bir öznitelik tespit edilmemiştir. Ancak missed_bytes, http_status_code, http_request_body_len ve http_response_body_len öznitelikleri kısmi-sabit olarak belirlenmiştir.

Bu özniteliklerin hedef değişken ile olan ilişkileri ayrıca incelenmiş; missed_bytes özniteliği haricindeki kısmi-sabit özniteliklerin hedef etiket ile anlamlı bir korelasyon sergilemediği gözlemlenmiştir. Bu nedenle, ayırt edici bilgi taşımadığı değerlendirilen söz konusu öznitelikler veri setinden çıkarılmıştır. Bu adım, hem öznitelik uzayının sadeleştirilmesini hem de modelin daha etkin ve genellenebilir bir yapı ile eğitilmesini amaçlamaktadır.

4.5. Kategorik Verilerin İşlenmesi

Önceki bölümlerde açıklanan ön işleme adımlarının ardından, ToN_IoT veri setinde yalnızca iki adet kategorik öznitelik kalmıştır. Bunlardan ilki olan proto özniteliği, ağ iletişiminde kullanılan protokol türünü temsil etmekte olup tcp, udp ve icmp olmak üzere üç farklı değer alabilmektedir. Bu protokol türlerinin hedef değişken (label) ile olan dağılımları Tablo 4.10'da sunulmaktadır.

Tablo 4.10. Protokol Türüne Göre Etiket Alanının Dağılımı

Protokol	Etiket	Kayıt Sayısı	Oran
ICMP	0	4843	0,9826
	1	86	0,0174
TCP	0	112327	0,5256
	1	101394	0,4744
UDP	0	109361	0,9406
	1	6906	0,0594

Tablo 4.10 incelendiğinde, ICMP protokolüne ait kayıtların büyük çoğunluğunun normal trafik (etiket 0) olduğu görülmektedir. UDP protokolünde saldırı oranı görece daha yüksek olmakla birlikte, saldırı kayıtlarının mutlak çoğunluğunun TCP protokolü ile ilişkili olduğu dikkat çekmektedir. Bu bulgular, proto özniteliğinin saldırı tespiti açısından potansiyel olarak ayırt edici bilgi içerdiğini göstermektedir. Bu nedenle, proto özniteliği model eğitiminde tutulmuş ve modelden bağımsız (model-agnostic) kodlama yöntemleri ile temsil edilmesi uygun görülmüştür.

İkinci kategorik öznitelik olan conn_state, iki makine arasındaki iletişimin durumunu, ilgili IP:port çiftlerine dayalı olarak tanımlamaktadır. Bu özniteliğin alabileceği değerler ve anlamları aşağıda özetlenmiştir:

- SF: Başarılı bağlantı (Success) – İletişim başarıyla kurulmuştur.
- S0: Gizli bağlantı denemesi (Stealth) – Bağlantı girişimi yapılmış ancak herhangi bir yanıt alınmamıştır.
- REJ: Reddedilen bağlantı (Rejected) – Bağlantı isteği hedef tarafından reddedilmiştir.
- SHR: Eksik paket iletimi (Shrink) – Kaynak makine eksik veya hatalı bir paket

göndermiştir.

- SH: Yarı-açık bağlantı (Half-open) – Bağlantı başlatılmış ancak el sıkışma (handshake) tamamlanmamıştır.
- S1: Zaman aşımı (Timeout) – Bağlantı girişimi zaman aşımına uğramıştır.
- S3: Kapatılan bağlantı (Closed) – Bağlantı kaynak makine tarafından kapatılmıştır.
- RSTR: Bağlantı sıfırlama (Reset) – Bağlantı sıfırlanmıştır.
- RSTRH: Hedef tarafından sıfırlanan bağlantı (Reset Host) – Bağlantı hedef makine tarafından sıfırlanmıştır.
- RSTO: Kaynak tarafından sıfırlanan bağlantı (Reset Origin) – Bağlantı kaynak makine tarafından sıfırlanmıştır.
- S2: SYN gönderildi (Syn Sent) – Kaynak makine bağlantıyı başlatmak için SYN paketi göndermiştir.
- RSTOS0: Gizli sıfırlama (Reset Origin Stealth) – El sıkışma tamamlanmadan bağlantı kaynak tarafından sıfırlanmıştır.
- OTH: Diğer bağlantı durumları (Other) – Tanımlı kategorilere girmeyen bağlantı durumları.

Bu özniteliğin, hem yüksek kardinaliteye sahip olduğu hem de kategoriler arasında ciddi bir dengesizlik barındırdığı gözlemlenmiştir. Bu durumun modele olası olumsuz etkilerini azaltmak ve daha dengeli bir temsil elde etmek amacıyla, bağlantı durumları hem dağılımsal özellikleri hem de anlamsal benzerlikleri dikkate alınarak gruplandırılmıştır. Bu gruplandırma aşağıdaki şekilde gerçekleştirilmiştir:

- Başarısız Bağlantılar (FAIL): REJ, RSTR, RSTRH, RSTO, RSTOS0
- Kısmi Bağlantılar (PARTIAL): SHR, SH, S1, S2
- Diğer/Bilinmeyen Bağlantılar (OTHER): OTH

Bu yeniden gruplandırma işlemi sonucunda elde edilen conn_state özniteliğinin güncellenmiş yapısı, hedef değişken ile birlikte Tablo 4.11'de gösterilmiştir.

Tablo 4.11'de, FAIL ve PARTIAL kategorilerinde saldırı (etiket 1) oranının görece yüksek olduğu dikkat çekmektedir. Bu durum, başarısız ya da tamamlanmamış bağlantı girişimlerinin kötü niyetli davranışlarla ilişkili olabileceğine işaret etmektedir.

Tablo 4.11. Bağlantı Durunlarının Sınıflara Göre Dağılım Oranları

conn_state	Etiket	Toplam Kayıt	Oran
FAIL	1	37941	0,9817
	0	748	0,0183
OTHER	0	89407	0,8907
	1	10968	0,1093
PARTIAL	0	20371	0,5548
	1	16357	0,4452
S0	0	62596	0,8173
	1	13972	0,1827
SUCCESS	0	53409	0,6470
	1	29148	0,3530

Benzer şekilde, S0 ve OTHER durumları da bazı saldırı türleriyle ilişkili olabilecek anormal ağ davranışlarını yansıtabilmektedir. SUCCESS durumunda ise sınıf dağılımı daha dengeli olmakla birlikte, bu kategoride de hatırı sayılır miktarda saldırı trafiği bulunmaktadır. Dolayısıyla, bağlantı durumlarının dağılımının analiz edilmesi, ağ trafiğinin karakterize edilmesi ve etkili bir saldırı tespit sistemi geliştirilmesi açısından önemli bilgiler sunmaktadır.

4.6. Çoklu Korelasyon Analizi

Makine öğrenmesi tabanlı modellerin güvenilir, kararlı ve genellenebilir sonuçlar üretebilmesi için, modelleme aşamasından önce öznitelikler arasındaki ilişkilerin dikkatli bir biçimde incelenmesi gerekmektedir (Liu ve diğ., 2025). Bu bağlamda korelasyon analizi, veri ön işleme sürecinin kritik bileşenlerinden biri olarak öne çıkmaktadır (Liu ve Yu, 2005). Bir özniteliğin hedef değişken ile yüksek düzeyde korelasyon göstermesi, genellikle o özniteliğin tahmin gücünün yüksek olduğuna işaret ederken; özniteliklerin kendi aralarında yüksek korelasyona sahip olması, özellikle çoklu korelasyon (multicollinearity) problemi açısından önemli riskler barındırmaktadır (Dormann ve diğ., 2013). Çoklu korelasyon, modele benzer veya tekrarlı bilgilerin dâhil edilmesine yol açarak model katsayılarının kararsızlaşmasına, küçük veri değişimlerinin tahminlerde büyük dalgalanmalara neden olmasına ve modelin yorumlanabilirliğinin zayıflamasına sebep olabilmektedir (Zou ve Hastie, 2005).

Özellikle doğrusal ve doğrusal temelli makine öğrenmesi modellerinde, yüksek derecede

birbiriyle ilişkili özniteliklerin birlikte kullanılması, modelin gereksiz yere karmaşık hale gelmesine ve sağlanan bilginin tekrarlanmasına neden olmaktadır (Motamedi ve Villányi, 2025). Bu durum yalnızca hesaplama maliyetini artırmakla kalmamakta, aynı zamanda modelin eğitim verisine aşırı uyum (overfitting) riskini de beraberinde getirmektedir (Guyon ve Elisseeff, 2003). Bu nedenle, yüksek korelasyona sahip öznitelik kümelerinin belirlenmesi ve bu kümeler içerisinden en bilgilendirici özniteliklerin seçilmesi, sağlam ve genellenebilir bir model tasarımı açısından kritik öneme sahiptir (Chandrashekar ve Sahin, 2014).

Bu bölümde, çoklu korelasyon problemini ele almak amacıyla, öznitelikler arasındaki ikili korelasyon ilişkileri sistematik olarak analiz edilmiş ve birbirleriyle yüksek derecede korelasyon gösteren öznitelik grupları belirlenmiştir. Ardından, her bir grup içerisinden hedef değişken ile en güçlü ilişkiye sahip olan öznitelikler korunarak, geri kalan tekrarlı öznitelikler veri setinden çıkarılmıştır. Bu yaklaşım sayesinde, hem öznitelik uzayı sadeleştirilmiş hem de modelin kararlılığı, genelleme performansı artırılmaya çalışılmıştır.

Bu süreçte, öznitelikler öncelikle aralarındaki korelasyon katsayılarına göre gruplandırılmıştır. Birbirleriyle olan mutlak korelasyon değeri %80'in üzerinde olan öznitelikler, aynı grup içerisinde değerlendirilmiştir. Bu eşik değerin seçilmesindeki amaç, yüksek derecede benzer bilgi taşıyan özniteliklerin tespit edilerek çoklu korelasyon probleminin önüne geçilmesidir.

Oluşturulan her bir öznitelik grubu için, özniteliklerin hedef değişken üzerindeki göreli önemini değerlendirmek amacıyla bir Rastgele Orman (Random Forest, RF) modeli kullanılmıştır. RF modeli tarafından hesaplanan öznitelik önem skorları doğrultusunda, her bir grup içerisinden hedef değişken ile en yüksek ilişkiye sahip olan öznitelik korunmuş; geri kalan öznitelikler ise bilgi tekrarına yol açtıkları gerekçesiyle veri setinden çıkarılmıştır. Bu yaklaşım, hem modelin gereksiz karmaşıklığını azaltmayı hem de yalnızca en bilgilendirici özniteliklerin korunmasını hedeflemektedir.

Uygulanan bu seçici öznitelik eleme süreci sonucunda, ileri analizlerde kullanılmak üzere toplam dokuz adet sayısal öznitelik veri setinde tutulmuştur. Bu öznitelikler sırasıyla

src_port, dst_port, duration, src_bytes, missed_bytes, src_ip_bytes, dst_ip_bytes ve son

Algoritma 2: Çoklu Korelasyona Dayalı Öznitelik Seçimi

Gerekli:

X_{train}: Eğitim veri seti
y: Sınıf etiketleri
X_{test}: Test veri seti
τ_1: Alt korelasyon eşiği (bu çalışmada $\tau_1 = 0,8$)
τ_2: Üst korelasyon eşiği (bu çalışmada $\tau_2 = 1$)

Ensure:

$X_{\text{train_filtered}}$, $X_{\text{test_filtered}}$: Seçilen öznitelikleri içeren filtrelenmiş veri setleri

Başlatma:
$CorrList \leftarrow [\emptyset]$

Yüksek korelasyonlu öznitelik çiftlerini belirle:
for each öznitelik çifti (f_1, f_2) in X_{train} **do**
    $corr_value \leftarrow compute_correlation(f_1, f_2)$
    if $\tau_1 \leq |corr_value| < \tau_2$ **then**
        AppendList($CorrList$, $(f_1, f_2, corr_value)$)
    end if
end for

Korelasyon gruplarını oluştur:
$correlated_groups \leftarrow [\,]$
for each f_1 in unique_features($CorrList$) **do**
    $group \leftarrow [f_1]$
    for each $(g_1, g_2, _)$ in $CorrList$ **do**
        if $g_1 == f_1$ **then**
            AppendList($group$, g_2)
        end if
    end for
    AppendList($correlated_groups$, $group$)
end for

Her grup için en bilgilendirici özniteliği seç:
$SelectedFeatures \leftarrow [\,]$
for each $group$ in $correlated_groups$ **do**
    $model \leftarrow RandomForestClassifier()$
    $model.fit(X_{\text{train}}[group], y)$
    $importance_scores \leftarrow perm_importance(model, X_{\text{train}}[group], y)$
    $best_feature \leftarrow group[\arg\max(importance_scores)]$
    AppendList($SelectedFeatures$, $best_feature$)
end for

Seçilen özniteliklerle veri setlerini filtrele:
$X_{\text{train_filtered}} \leftarrow X_{\text{train}}[SelectedFeatures]$
$X_{\text{test_filtered}} \leftarrow X_{\text{test}}[SelectedFeatures]$

olarak, dns_qtype şeklinde belirlenmiştir. Söz konusu işlemin adımlarını özetleyen sözde kod, Algoritma 2'de sunulmaktadır.

Algoritma 2'de sunulan sözde kod, Galli (2024) tarafından sunulan GitHub deposundan esinlenilerek geliştirilmiştir. İlgili sözde kod, bir veri setindeki öznitelikler arasındaki korelasyon ilişkilerini sistematik bir biçimde analiz etmeyi ve özellikle çoklu korelasyon problemini tespit edip ele almayı amaçlayan bir yaklaşımı tanımlamaktadır. Çoklu korelasyon, iki veya daha fazla özniteliğin yüksek derecede korelasyon göstermesi durumunda ortaya çıkmakta olup, özellikle özniteliklerin bağımsız olduğu varsayımına dayanan doğrusal regresyon gibi modellerde performansın ve yorumlanabilirliğin ciddi biçimde zarar görmesine neden olabilmektedir. Bu prosedür, yüksek derecede korelasyon sergileyen öznitelik çiftlerini tespit ederek, gereksiz ve tekrarlı bilgilerin modele dâhil edilmesini önlemeyi ve böylece model performansını optimize etmeyi hedeflemektedir.

İlk aşamada, eğitim veri seti X_{train} üzerinde yer alan tüm öznitelikler için ikili (pairwise) korelasyon değerleri hesaplanmaktadır. Oluşturulan korelasyon matrisi, her bir öznitelik çifti arasındaki doğrusal ilişkinin derecesini nicel olarak ifade etmektedir. Bu aşamada, yalnızca mutlak korelasyon değeri $0,8 \leq |correlation| < 1$ aralığında olan öznitelik çiftleri seçilerek CorrList adlı bir liste içerisinde saklanmaktadır. Bu filtreleme işlemi, çoklu korelasyon problemi açısından risk oluşturan, yani yüksek derecede ilişkili özniteliklerin belirlenmesini sağlamaktadır.

Belirlenen yüksek korelasyonlu öznitelik çiftleri, her bir çiftin ilk özniteliği (feature_1) temel alınarak gruplandırılmaktadır. Her bir feature_1 için, onunla yüksek korelasyon gösteren tüm feature_2 öznitelikleri aynı grup altında toplanmakta ve feature_1 de bu gruba dâhil edilmektedir. Bu şekilde oluşturulan ve correlated_feature_group olarak adlandırılan gruplar, birbirleriyle yüksek derecede ilişkili öznitelik kümelerini temsil etmektedir. Bu adım, hangi özniteliklerin bilgi tekrarına yol açtığının belirlenmesi ve potansiyel olarak elenebilecek özniteliklerin tespit edilmesi açısından kritik öneme sahiptir.

Her bir korelasyon grubuna ait özniteliklerin hedef değişken üzerindeki göreli önemini değerlendirmek amacıyla, her grup için ayrı bir Rastgele Orman (Random Forest)

sınıflandırıcısı eğitilmektedir. Model eğitiminin ardından, gruptaki her bir öznitelik için permütasyon önemi (permutation importance) hesaplanmaktadır. Permütasyon önemi, ilgili özniteliğin değerleri rastgele karıştırıldığında model performansında meydana gelen düşüşü ölçmekte olup, bu düşüşün büyüklüğü özniteliğin model açısından taşıdığı bilginin önemini yansıtmaktadır. Her bir korelasyon grubunda, model performansında en büyük düşüşe neden olan öznitelik en önemli öznitelik olarak belirlenmekte ve SelectedFeatures listesine eklenmektedir. Bu adım sayesinde, her yüksek korelasyonlu grup içerisinden yalnızca en bilgilendirici öznitelik korunmaktadır.

Son aşamada, eğitim (X_{train}) ve test (X_{test}) veri setleri, yalnızca SelectedFeatures listesinde yer alan öznitelikleri içerecek şekilde filtrelenmektedir. Bu işlem, tekrarlı ve gereksiz özniteliklerin veri setinden çıkarılmasını sağlayarak öznitelik uzayını sadeleştirmekte ve çoklu korelasyon kaynaklı aşırı uyum (overfitting) riskini azaltmaktadır. Böylece model, daha özlü ve bilgi açısından daha zengin bir öznitelik kümesi üzerinde eğitilmekte; bu durum hem genelleme kabiliyetini hem de hesaplama verimliliğini artırmaktadır.

Çoklu korelasyon analiz çerçevesi, model eğitiminden önce çoklu korelasyon problemini ele alarak güçlü doğrusal ilişkilere sahip tekrarlı özniteliklerin tespit edilmesini ve elenmesini sağlamaktadır. Bu yaklaşım, özellikle doğrusal regresyon ve ridge regresyon gibi çoklu korelasyona duyarlı algoritmalarda model doğruluğunu ve yorumlanabilirliğini artırmaktadır.

ToN_IoT veri seti üzerinde gerçekleştirilen ön işleme adımlarının ardından, önceki bölümlerde de tartışıldığı üzere, öznitelikler arasında güçlü doğrusal (lineer) korelasyon ilişkilerinin bulunmadığı gözlemlenmiştir. Ancak korelasyon ilişkileri yalnızca doğrusal uzayda ortaya çıkmak zorunda değildir; öznitelikler arasında doğrusal olmayan (non-lineer) bağımlılıklar da söz konusu olabilmektedir. Bu tür ilişkilerin yalnızca klasik korelasyon analizleri ile tespit edilmesi çoğu zaman mümkün olmamaktadır.

Bu nedenle, doğrusal olmayan bağımlılıkların daha hızlı ve etkili bir biçimde ortaya çıkarılabilmesi amacıyla, korelasyon matrisi hesaplanmadan önce özniteliklere logaritmik dönüşüm uygulanmıştır. Logaritmik dönüşüm, özellikle çarpımsal ilişkiler, ölçek farklılıkları ve çarpık dağılımlar içeren özniteliklerde, doğrusal olmayan örüntülerin daha

belirgin hale gelmesini sağlamaktadır. Bu yaklaşım sayesinde, öznitelikler arasındaki olası gizli bağımlılıkların korelasyon analizi yoluyla yakalanması hedeflenmiştir.

Gerçekleştirilen bu genişletilmiş korelasyon analizi sonucunda, birbirleriyle yüksek derecede ilişkili üç farklı öznitelik grubu tespit edilmiştir. Bu gruplar, doğrusal ve doğrusal olmayan bağımlılıkları birlikte yansıtan öznitelik kümelerini temsil etmekte olup, çoklu doğrusal bağlantı problemine yol açabilecek potansiyel öznitelik fazlalıklarını işaret etmektedir. Belirlenen yüksek korelasyonlu öznitelik grupları Tablo 4.12'de özetlenmiştir.

Tablo 4.12. Yüksek Korelasyon Gösteren Öznitelik Grupları

Grup	**Öznitelik 1**	**Öznitelik 2**	**Korelasyon**
1	dst_ip_bytes	dst_pkts	0,994350
		dst_bytes	0,812308
2	src_pkts	src_ip_bytes	0,994323
3	dns_qtype	dns_query	0,974765
		dns_qclass	0,954726

Belirlenen bu gruplar arasından, hedef değişken (label) ile en yüksek korelasyona sahip olan dst_ip_bytes, src_pkts ve dns_qtype öznitelikleri veri setinde tutulmuş; diğer öznitelikler ise elenmiştir. Bu seçici yaklaşım sayesinde, elde edilen öznitelik kümesinin yalnızca doğrusal bağımlılıklardan arındırılması değil, aynı zamanda olası doğrusal olmayan korelasyonların da en aza indirilmesi amaçlanmıştır. Sonuç olarak oluşturulan öznitelik setine sahip bir veri seti, öznitelik uzayındaki bilgi tekrarını azaltarak hem modelin yorumlanabilirliğini hem de tahmin performansını destekleyen daha sade yapıya ulaşmıştır.

4.7. İteratif Bir Öznitelik Üretim Algoritması

Öznitelik üretimi (feature generation), bir veri setinde mevcut özniteliklerden yeni ve daha bilgilendirici öznitelikler türetmeyi amaçlayan güçlü bir tekniktir. Bu süreç, veri analizi ve makine öğrenmesi uygulamalarında kritik bir öneme sahiptir; zira uygun özniteliklerin tanımlanması ve tasarlanması, model performansını önemli ölçüde artırabilmektedir. Üretilen yeni öznitelikler, veri seti içerisindeki örüntüleri ve ilişkileri daha etkin bir

biçimde temsil ederek, modelin daha isabetli kararlar almasına olanak tanımaktadır. Öznitelik üretimi genellikle veri odaklı yaklaşımlara dayanmakta olup, literatürde yaygın olarak kullanılan bazı temel yöntemler aşağıda özetlenmiştir:

- Polinomsal Öznitelikler (Polynomial Features): Mevcut özniteliklerin belirli bir polinom derecesine yükseltilmesiyle yeni öznitelikler üretilebilir. Örneğin, birinci dereceden bir öznitelik (x) için, ikinci veya daha yüksek dereceden terimler (x^2, x^3) türetilerek modele eklenebilir (Pedregosa ve diğ., 2011).
- Etkileşim Öznitelikleri (Interaction Features): Mevcut öznitelikler arasındaki etkileşimler dikkate alınarak yeni öznitelikler oluşturulabilir. İki özniteliğin çarpımı, toplamı veya farkı gibi işlemlerle elde edilen etkileşim öznitelikleri, bu ilişkilerin modele yansıtılmasını sağlar (Kanter ve Veeramachaneni, 2015).
- İstatistiksel Öznitelikler (Statistical Features): Veri setinin istatistiksel özelliklerinden yararlanılarak yeni öznitelikler üretilebilir. Ortalama, standart sapma, minimum ve maksimum gibi istatistiksel ölçütler, veri setine ek öznitelikler olarak dâhil edilebilir (Christ ve diğ., 2018).
- Zaman Serisi Öznitelikleri (Time Series Features): Zaman serisi verileriyle çalışırken, geçmiş gözlemler veya belirli zaman aralıkları üzerinden hesaplanan istatistiksel değerler gibi zamana dayalı öznitelikler üretilebilir (Montero-Manso ve Hyndman, 2021).
- Alan-Özgü Öznitelikler (Domain-Specific Features): Veri setinin ait olduğu alana özgü bilgi kullanılarak yeni öznitelikler tasarlanabilir. Örneğin, görüntü işleme uygulamalarında kenar tespiti veya renk histogramları gibi alan-özgü öznitelikler sıklıkla kullanılmaktadır (Ugale ve Potgantwar, 2025).

Bu yöntemlerin uygulanması, veri setinin yapısına ve analiz amacına bağlı olarak farklı düzeylerde değişkenlik (variability) ortaya çıkarabilmektedir. Öznitelik üretimi, veri setinin boyutunu ve öznitelik uzayının karmaşıklığını artırabileceğinden, daha karmaşık modellerin kullanımını gerektirebilir. Uygun özniteliklerin seçilmesi, modelin hem tahmin performansı hem de genelleme kabiliyeti önemli ölçüde iyileştirilebilmektedir. Bu nedenle, öznitelik üretimi süreci, etkili ve başarılı bir makine öğrenmesi modelinin temel bileşenlerinden biri olarak değerlendirilmektedir.

Bu kapsamda, daha karmaşık özniteliklerin elde edilmesi ve bu özniteliklerin bir saldırı tespit sistemi (Intrusion Detection System, IDS) üzerindeki etkilerinin incelenmesi amacıyla otomatik öznitelik oluşturma temelli bir yaklaşım benimsenmiştir. Geleneksel yaklaşımlarda öznitelikler çoğunlukla alan uzmanları tarafından manuel olarak tanımlanmakta ya da mevcut veri üzerinde gerçekleştirilen sınırlı analizler sonucunda seçilmektedir. Ancak bu tür manuel süreçler, hem zaman açısından maliyetli olmakta hem de bazı durumlarda yetersiz veya eksik özniteliklerin seçilmesine yol açabilmektedir.

Bu sınırlamaların üstesinden gelebilmek amacıyla geliştirilen otomatik öznitelik oluşturma (automated feature generation) yöntemleri, veri seti üzerinde yeni ve daha bilgilendirici özniteliklerin sistematik bir biçimde türetilmesini hedeflemektedir. Bu yaklaşımlarda, makine öğrenmesi ve istatistiksel teknikler kullanılarak veri kümesindeki gizli örüntüler, ilişkiler ve yapılar analiz edilmekte; bu analizler sonucunda yeni öznitelikler otomatik olarak üretilmektedir. Böylece, insan müdahalesine olan bağımlılık azaltılırken, daha zengin ve açıklayıcı bir öznitelik uzayı elde edilmesi amaçlanmaktadır.

Bu çalışma kapsamında da manuel öznitelik tasarımı yerine otomatik bir öznitelik oluşturma stratejisi (Piramuthu ve Sikora, 2009) temel alınarak, üzerine farklı işlemler ve mekanizmalar eklenerek geliştirilmiştir. Temel yaklaşım olarak, başlangıçta mevcut ana öznitelik uzayı ele alınmakta ve bu uzayda yer alan öznitelikler çeşitli matematiksel operatörler aracılığıyla ikili kombinasyonlar hâlinde işleme tabi tutulmaktadır. Bu işlem sonucunda elde edilen yeni türetilmiş öznitelikler ile birlikte, mevcut klasik öznitelikler bir arada değerlendirilmekte ve ki-kare (chi-square) istatistiği kullanılarak hedef değişken ile olan ilişkilerine göre önem sırasına konulmaktadır. Ardından, en yüksek önem derecesine sahip ilk k öznitelik seçilerek öznitelik kümesi güncellenmektedir.

Bu öznitelik oluşturma ve seçme süreci, önceden tanımlanmış bir durdurma kriteri (stopping criteria) sağlanana kadar iteratif olarak tekrarlanmaktadır. Böylece, her yinelemede daha bilgilendirici özniteliklerin öne çıkarıldığı, aşamalı ve kontrollü bir öznitelik uzayı inşa edilmektedir. Söz konusu otomatik öznitelik oluşturma algoritmasının işleyişini özetleyen sözde kod, Algoritma 3'te sunulmaktadır.

Bu algoritma, başlangıçtaki öznitelik çiftlerini temel alarak, bu öznitelikler üzerinde

çeşitli matematiksel işlemler uygulanması yoluyla yeni özniteliklerin oluşturulmasını amaçlamaktadır.

Algoritma 3: İteratif Öznitelik Oluşturma Algoritması

Gerekli:

Δ: Başlangıç öznitelik kümesi (n adet temel öznitelik)
π: Matematiksel operatörler kümesi (m adet operatör)
$FC_\pi(\Delta)$: Δ kümesi üzerinde π operatörleri uygulanarak elde edilen yeni öznitelikler kümesi

Ensure:

Δ: Nihai seçilmiş öznitelik kümesi

Başlatma:
$\Delta \leftarrow$ başlangıç öznitelik kümesi
while durdurma kriteri sağlanana kadar **do**
    Genişletilmiş öznitelik uzayını oluştur:
    $\Delta' \leftarrow \Delta \cup \log(\Delta) \cup FC_\pi(\Delta)$
    Korelasyon analizini gerçekleştir:
    $\Sigma \leftarrow find_correlated_features(\Delta')$
    $feature_list \leftarrow [\,]$
    Her korelasyon grubu için öznitelik seçimi yap:
    for each grup $\sigma \in \Sigma$ **do**
        $\delta \leftarrow$ ilgili gruptaki korelasyonlu özniteliklerin listesi
        $\delta' \leftarrow \chi^2(\delta)$ ▷ Ki-kare önem skorlarını hesapla
        AddItem($feature_list$, $\delta'[1]$) ▷ En önemli özniteliği ekle
    end for
    Model tabanlı öznitelik eleme:
    $RF \leftarrow feature_list$ kullanılarak eğitilen Rastgele Orman modeli
    $feature_list \leftarrow$ RF önem skoru medyanın üzerinde olan öznitelikler
    Öznitelik kümesini güncelle:
    $\Delta \leftarrow feature_list$
end while

Algoritmanın ikinci adımında, özgün uygulamada tanımlandığı şekilde matematiksel operatör kümesi $\pi = \{+, -, \times, /\}$ olarak belirlenmiş ve bu operatörler mevcut öznitelik kümesi üzerine uygulanmıştır. Buna ek olarak, her bir öznitelik için logaritmik dönüşüm ayrı ayrı hesaplanarak öznitelik uzayına dâhil edilmiştir.

Tanımlanan durdurma kriteri sağlanana kadar, bir önceki iterasyondan elde edilen öznitelik kümesi ile yeni oluşturulan öznitelikler birleştirilerek daha geniş bir öznitelik

uzayı elde edilmiştir. Bu genişletilmiş öznitelik uzayı üzerinde, algoritmanın altıncı adımında tüm öznitelik çiftleri arasındaki korelasyon değerleri hesaplanmış ve yüksek korelasyon sergileyen öznitelikler aynı gruplar altında toplanmıştır. Bu adım, çoklu doğrusal bağlantı problemine yol açabilecek potansiyel öznitelik fazlalıklarının belirlenmesini amaçlamaktadır.

Oluşturulan her bir korelasyon grubu için, Ki-kare (χ^2) öznitelik seçme yöntemi ((Forman, 2003)) uygulanmış ve her grup içerisinden hedef değişken ile en güçlü ilişkiye sahip olan öznitelik seçilmiştir. Seçilen bu öznitelikler dışındaki, aynı grup içerisinde yer alan diğer öznitelikler ise bilgi tekrarına yol açtıkları gerekçesiyle veri setinden çıkarılmıştır. Bu aşamanın ardından, seçilen öznitelik listesi kullanılarak bir Rastgele Orman (Random Forest) modeli eğitilmiş ve model tarafından hesaplanan öznitelik önem skorları değerlendirilmiştir. Önem skoru medyan değerin altında kalan öznitelikler elenmiş; geriye kalan öznitelikler, modelleme sürecinde kullanılacak en faydalı öznitelikler olarak belirlenmiştir.

Söz konusu algoritma, bu çalışmada iki iterasyon boyunca uygulanmıştır; başka bir ifadeyle durdurma kriteri iterasyon sayısı iki olarak belirlenmiştir. İterasyon sayısının bu şekilde sınırlandırılmasının temel nedeni, özniteliklerin giderek daha karmaşık hâle gelmesi, ve bunun sonucunda aşırı uyum (overfitting) probleminin ortaya çıkmasına sebep olmasıdır. Ayrıca, öznitelik uzayının kontrolsüz biçimde büyümesinin beraberinde getireceği yüksek hesaplama maliyetlerinin önüne geçilmesi de bu sınırlamanın bir diğer önemli gerekçesini oluşturmaktadır. Bunun sonucunda klasik öznitelikler ile birlikte toplamda 56 öznitelik oluşturulmuştur.

4.8. Model Eğitimi

Model eğitimi aşamasında, eğitim algoritması olarak Rastgele Orman algoritması tercih edilmiştir. RF modeli; görece basit yapısı, yüksek yorumlanabilirliği, hızlı eğitim süresi ve aşırı uyuma (overfitting) karşı dayanıklılığı nedeniyle bu çalışma için uygun bir seçim olarak değerlendirilmiştir. Modelin en uygun hiperparametrelerinin belirlenmesi amacıyla, parametre ayarlama sürecinde ızgara arama (grid search) stratejisi uygulanmıştır. Izgara arama yöntemi, tüm olası parametre kombinasyonlarını denemesi

nedeniyle yüksek hesaplama maliyetine sahip olabilmektedir. Ancak bu çalışmada kullanılan veri seti ve model yapılandırmaları göz önüne alındığında, eğitim sürelerinin kabul edilebilir düzeylerde olduğu görülmüş ve bu nedenle grid search stratejisinin uygulanabilir olduğu değerlendirilmiştir.

Eğitim sürecinde, ızgara arama stratejisi ile birlikte 10-katlı tabakalı grup-temelli çapraz geçerleme (10-stratified group cross validation) yaklaşımı kullanılmıştır. Bölüm 4.2.'de detaylandırıldığı üzere, grup-temelli katlama yaklaşımı sayesinde iki makine arasındaki herhangi bir veri iletimine ait kayıtların yalnızca tek bir kat (fold) içerisinde yer alması garanti altına alınmaktadır. Böylece, aynı iletim kanalına ait kayıtların hem eğitim hem de doğrulama aşamalarında eş zamanlı olarak yer almasının önüne geçilmekte ve veri sızıntısı (data leakage) engellenmektedir. Buna ek olarak, tabakalama (stratification) işlemi ile her bir kat içerisinde sınıf dağılımlarının korunması sağlanarak, eğitim ve doğrulama süreçlerinin daha dengeli ve temsili bir yapıda gerçekleştirilmesi amaçlanmıştır. Bu bütüncül strateji ile, modelin daha güvenilir, tarafsız ve genellenebilir bir şekilde eğitilmesi hedeflenmiştir.

Önceki bölümlerde de tartışıldığı üzere, ToN_IoT veri seti on farklı sınıf içermekte olup sınıflar arasında belirgin bir dengesizlik bulunmaktadır. Model eğitiminde kullanılan eğitim veri setine ait sınıf dağılımları Tablo 4.13'te gösterilmiştir.

Tablo 4.13. Eğitim Veri Seti İçin Sınıf Başına Düşen Örnek Sayıları

Sınıf Adı	**Örnek Sayısı**
Normal	226.531
Ransomware	13.662
DoS	13.658
Scanning	13.528
XSS	13.461
Password	13.424
Backdoor	13.365
Injection	13.352
DDoS	13.236
MITM	700

Makine öğrenmesi literatüründe yaygın olarak kullanılan birçok algoritma, hedef sınıfların veri seti içerisinde yaklaşık olarak dengeli (uniform) bir dağılım sergilediği varsayımı üzerine inşa edilmektedir (Ahmad ve Alsmadi, 2021; Moustafa ve Slay, 2015). Bu nedenle, sınıflar arası dengesizliğin yüksek olduğu veri setlerinde eğitilen modellerin, çoğunluk sınıfa aşırı uyum gösterirken azınlık sınıfları yeterince öğrenemediği sıklıkla gözlemlenmektedir. Oysa başarılı bir öğrenme süreci için, modelin tüm sınıfları benzer düzeyde temsil edebilecek biçimde eğitilmesi kritik öneme sahiptir.

Eğitim veri setine ait sınıf dağılımları incelendiğinde, Normal trafik örneklerinin 226.531 adet ile veri setinin büyük bir bölümünü oluşturduğu görülmektedir. Buna karşılık, MITM (Man-In-The-Middle) saldırı türüne ait yalnızca 700 adet örnek bulunmaktadır. Diğer saldırı sınıflarının örnek sayılarının ise genel olarak birbirine yakın ve görece dengeli olduğu söylenebilir. Bu durum, özellikle MITM gibi nadir görülen saldırı türlerinin öğrenilmesini zorlaştırmakta; modelin bu sınıflar üzerinde düşük performans sergilemesine yol açabilmektedir. Dolayısıyla, veri setindeki aşırı temsil edilen normal trafik örneklerinin azaltılması ve azınlık sınıflara ait örneklerin artırılması, daha dengeli ve etkili bir model eğitimi açısından faydalı olacaktır.

Bu amaç doğrultusunda, model eğitim sürecinde veri yeniden örnekleme (resampling) yöntemlerinin kullanılması tercih edilmiştir. Yeniden örnekleme yaklaşımı, çoğunluk sınıfa ait örneklerin azaltılması ve azınlık sınıflara ait örneklerin artırılması yoluyla sınıflar arasındaki örnek sayısı farkının minimize edilmesini hedeflemektedir. Bu çalışmada, yeniden örnekleme işlemi için SCUT (SMOTE and Cluster-based Undersampling) algoritması kullanılmıştır (Agrawal ve diğ., 2015). SCUT algoritmasının sözde kodu, Algoritma 4'te gösterilmiştir.

SCUT algoritması, öncelikle her bir sınıfa ait örnek sayısını hesaplamakta ve bu değerlerin ortalamasını hedef örnek sayısı olarak belirlemektedir. Ardından, çoğunluk sınıflar üzerinde aşağı-örnekleme (undersampling) uygulanarak örnek sayıları azaltılırken, azınlık sınıflar üzerinde yukarı-örnekleme (oversampling) gerçekleştirilerek bu ortalama değere yaklaşılması sağlanmaktadır. Böylece, sınıflar arasında daha dengeli bir dağılım elde edilmektedir. Çoğunluk sınıflar için, Beklenti-Maksimizasyon

(Expectation Maximization, EM) algoritması kullanılarak veriler kümelenmekte ve bu kümeler, birden fazla Gauss dağılımının birleşimi olarak modellenmektedir. Daha sonra, her bir Gauss bileşeninden rastgele örnekler seçilerek, sınıf dağılımının genel yapısı korunacak şekilde örnekleme yapılmaktadır. Bu yaklaşım, yalnızca örnek sayısını azaltmayı değil, aynı zamanda veri dağılımının karakteristik özelliklerini korumayı amaçlamaktadır.

Algoritma 4: SCUT Yeniden Örnekleme Algoritması (Agrawal ve diğ., 2015)

Gerekli:
D: n sınıfa sahip veri seti
m: Tüm sınıflar için hedef örnek sayısı (tüm sınıfların ortalama örnek sayısı)
Ensure:
D': Tüm sınıfların m örnek içerdiği dengelenmiş veri seti
Başlatma:
$D \leftarrow \{D_1, D_2, \ldots, D_n\}$ ▷ Her D_i tek bir sınıfa aittir
Alt Örnekleme (Undersampling):
for each sınıf D_i, $i = 1, 2, \ldots, n$ **ve** $|D_i| > m$ **do**
    D_i üzerinde k küme oluştur ▷ EM algoritması kullanılarak
    for each küme C_j, $j = 1, 2, \ldots, k$ **do**
        C_j içerisinden rastgele örnekler seç
        Seçilen örnekleri C'_j kümesine ekle
    end for
    $D'_i \leftarrow \bigcup_{j=1}^{k} C'_j$
end for
Üst Örnekleme (Oversampling):
for each sınıf D_i, $i = 1, 2, \ldots, n$ **ve** $|D_i| < m$ **do**
    D_i üzerine SMOTE uygula ve D'_i elde et
end for
Dengelenmiş Veri Setini Oluştur:
for each sınıf D'_i, $i = 1, 2, \ldots, n$ **do**
    $D' \leftarrow D' \cup D'_i$
end for
return D'

Azınlık sınıflar için ise SMOTE (Synthetic Minority Over-sampling Technique) yöntemi uygulanmaktadır (Chawla ve diğ., 2002). SMOTE, mevcut azınlık sınıf örnekleri arasında doğrusal interpolasyon gerçekleştirerek yeni sentetik örnekler üretmektedir. Bu yöntemde, rastgele seçilen bir örnek için k en yakın komşu belirlenmekte ve bu komşular ile seçilen örnek arasında oluşturulan doğrusal uzay üzerinde yeni veri

noktaları türetilmektedir. Bu sayede, azınlık sınıf örnek sayısı artırılırken, veri dağılımının sürekliliği korumak amaçlanmıştır.

Model eğitimi sürecinde, veri yeniden örnekleme işlemi çapraz geçerleme mekanizmasının her bir adımında ayrı ayrı uygulanmıştır. 10-katlı çapraz geçerleme yaklaşımı dikkate alındığında, her bir iterasyonda öncelikle eğitim amacıyla kullanılan 9-kat üzerinde yeniden örnekleme gerçekleştirilmiş; elde edilen dengelenmiş veri seti ile ilgili iterasyona ait model eğitimi tamamlanmıştır. Ardından, geriye kalan tek kat doğrulama amacıyla kullanılarak modelin performansı değerlendirilmiştir. Bu süreç, her bir kat için tekrarlanarak toplamda on iterasyon tamamlanmış ve böylece ilgili model için eğitim süreci sonlandırılmıştır. Aynı prosedür, tüm model–hiperparametre kombinasyonları için uygulanarak deneysel değerlendirme ortamı oluşturulmuştur.

Modelleme aşamasında sınıflandırıcı olarak Rastgele Orman (Random Forest) algoritması tercih edilmiştir. En uygun model yapılandırmasının belirlenebilmesi amacıyla, hiperparametre optimizasyonunda açgözlü bir arama yöntemi olan ızgara arama (grid search) stratejisi kullanılmıştır. Bu yöntemde, önceden tanımlanan parametre uzayı içerisindeki tüm olası parametre kombinasyonları sistematik olarak denenmekte ve her bir kombinasyonun performansı ölçülmektedir. Parametre uzayının teorik olarak sınırsız olabileceği göz önünde bulundurularak, bu çalışmada yalnızca belirlenen parametre aralığı içerisindeki kombinasyonlar değerlendirilmiştir. Kullanılan hiperparametre uzayı Tablo 4.14'te gösterilmiştir.

Tablo 4.14. Rastgele Orman Modeli için Hiperparametre Uzayı

Parametre	**Açıklama**	**Değerler**
n_estimators	Rastgele ormanda yer alan toplam karar ağacı sayısı	8, 16, 32, 64, 128, 256
criterion	Karar ağaçlarının oluşturulmasında kullanılan bölme kalite metriği	gini, entropy
max_depth	Bir karar ağacının alabileceği maksimum derinlik	None, 5, 10
min_samples_split	Bir düğümün bölünebilmesi için gerekli minimum örnek sayısı	2, 5, 10
min_samples_leaf	Bir yaprak düğümde bulunması gereken minimum örnek sayısı	1, 2, 4

Hem orijinal öznitelikler hem de üretilmiş öznitelikler için iki ayrı eğitim aşaması gerçekleştirilmiştir. Üretilmiş öznitelikler, iteratif öznitelik üretim algoritması kullanılarak elde edilmiştir. Eğitim sürecinde amacımız, modeli eğitirken yalnızca duyarlılığı (recall) en üst düzeye çıkarmakla kalmayıp aynı zamanda MITM (Man-in-the-Middle) saldırılarının kesinliğini (precision) de en üst düzeye çıkarmaktır. Bu bağlamda, bir sınıf için yapılan tahminlerin gerçekten o sınıfa ait olmasını sağlayacak modeli seçerken, MITM saldırılarının diğer saldırı türleri ile karıştırılmaması hedeflenmiştir. Bir model sınıflandırma yaptığında, tahmin edilen sınıfın gerçekten o sınıfa ait olma sayısını en üst düzeye çıkarmak amaçlanmaktadır. Bu doğrultuda, bir saldırı tespit edildiğinde, bu saldırının gerçek bir saldırı olduğunun garanti altına alınması hedeflenmektedir. Kullanılan maliyet fonksiyonu Denklem 4.2'de gösterilmektedir:

$$Loss = precision_{MITM} * recall_{model} \tag{4.2}$$

Buradaki temel amaç modelin toplam duyarlılık (recall) değerini arttırırken mitm saldırıları için kesinliğin (precision) arttırılmasıdır. Burada modeller arasından bir sınıf için yapılan tahminlerin gerçekten o sınıf olduğu modeli seçerken aynı zamanda mitm saldırıların diğer saldırılarla karıştırılmamasını sağlamamak amaçlanmıştır.

İki farklı öznitelik uzayındaki en iyi modellerin tespiti için her bir öznitelik uzayında 324 model eğitilmiş ve test edilmiştir. Burada en iyi modelin seçilmesi konusu önemlidir. İlk önce model değerlendirmesinde 'Eğri Altında Kalan Alan' (Receive Operator Characteristic – Area Under Curve, ROC-AUC) kullanılsa da bu algoritma kullanılarak oluşturulan modeller için özellikle MITM tipi saldırıların tespit başarısının çok düşük kaldığı gözlemlenmiştir. Bu sebeple Denklem (4.2)'de gösterilen bir kayıp fonksiyonu (loss function) maksimize edilmiştir.

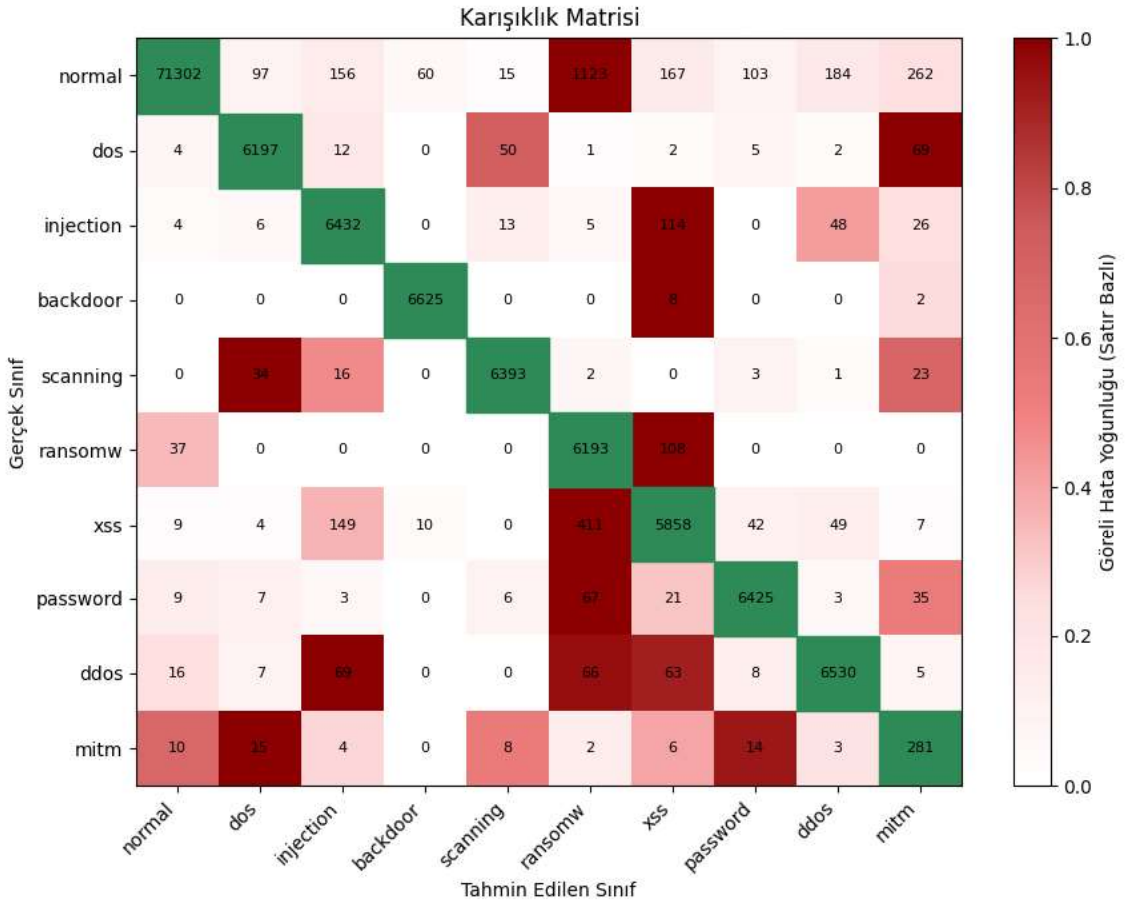
4.9. Sonuçlar

4.9.1. Orjinal ve Üretilmiş Özniteliklerin Karşılaştırılması

Tablo 4.15'te, orijinal öznitelikler ile üretilmiş öznitelikler kullanılarak eğitilen modellerin test verisi üzerindeki sınıf bazlı performansları karşılaştırılmıştır. Tabloda

yer alan metrikler; özgüllük (specificity), duyarlılık, kesinlik (precision) ve F-skoru (F-score) şeklindedir. Ayrıca üretilmiş öznitelikler kullanılan modele ait karışıklık matrisi (confusion matrix)'de Şekil 4.2'dedir.

Elde edilen sonuçlar incelendiğinde, üretilmiş öznitelikler kullanılarak eğitilen modelin birçok saldırı sınıfında özellikle duyarlılık ve F-skoru metriklerinde belirgin iyileşmeler sağladığı görülmektedir.



Şekil 4.2. Üretilmiş Öznitelikler için Karışıklık Matrisi

Örneğin, DoS ve Injection saldırı türlerinde üretilmiş öznitelikler kullanıldığında duyarlılık değerleri sırasıyla 0,999 ve 0,992 seviyelerine ulaşmış, bu da modelin saldırıları yakalama başarısının anlamlı biçimde arttığını göstermiştir. Benzer şekilde, Scanning ve Backdoor sınıflarında her iki model de yüksek performans sergilese de, üretilmiş öznitelikler kullanılan modelin kesinlik ve F-skoru açısından daha dengeli sonuçlar verdiği gözlemlenmiştir.

Tablo 4.15. Test Verisi Üzerinde Alınan Sonuçlar

Sınıf	Orijinal Öznitelikler				Üretilmiş Öznitelikler			
	Özgüllük (Specificity)	**Duyarlılık** (Recall)	**Kesinlik** (Precision)	**F-Skoru** (F-score)	**Özgüllük** (Specificity)	**Duyarlılık** (Recall)	**Kesinlik** (Precision)	**F-Skoru** (F-score)
Normal	**0,999**	**0,984**	**0,999**	0,992	**0,999**	0,982	**0,999**	0,991
DoS	0,998	0,976	0,980	0,978	**0,999**	**0,999**	**0,988**	**0,993**
Injection	0,993	0,969	0,893	0,929	**0,998**	**0,992**	**0,973**	**0,982**
Backdoor	**0,999**	**0,999**	**0,999**	**0,999**	**0,999**	**0,999**	0,997	0,998
Scanning	**0,999**	**0,999**	0,983	0,987	**0,999**	**0,999**	**0,997**	**0,998**
Ransomware	**0,999**	**0,999**	**0,990**	**0,994**	0,989	0,978	0,792	0,875
XSS	**0,998**	**0,977**	**0,964**	**0,970**	**0,998**	0,931	0,952	0,942
Password	**0,999**	**0,989**	**0,995**	**0,992**	**0,999**	**0,989**	0,987	0,988
DDoS	**0,998**	0,981	0,970	0,976	**0,998**	**0,986**	**0,972**	**0,979**
MITM	0,997	0,886	0,466	0,611	**0,998**	**1,000**	**0,638**	**0,779**
Ortalama	**0,9979**	0,9759	0,9239	0,9428	0,9976	**0,9855**	**0,9295**	**0,9525**

Özellikle dikkat çekici sonuçlardan biri MITM (Ortadaki Adam) saldırıları için elde edilmiştir. Orijinal özniteliklerle eğitilen modelde MITM sınıfı için kesinlik değeri 0,466 seviyesinde kalırken, üretilmiş öznitelikler kullanıldığında bu değer 0,638'e yükselmiştir. Duyarlılık değeri 1,000 seviyesine ulaşması, modelin MITM saldırılarını kaçırmadan tespit edebildiği görülmüştür. Bu durum, üretilmiş özniteliklerin karmaşık ve nadir saldırı türlerinin ayırt edilmesinde önemli bir katkı sunduğunu göstermektedir.

Genel performans açısından değerlendirildiğinde, Tablo 4.15'in alt kısmında sunulan ortalama değerler üretilmiş özniteliklerin üstünlüğünü açık biçimde desteklemektedir. Orijinal öznitelikler ile elde edilen ortalama F-skoru 0,9428 iken, üretilmiş öznitelikler kullanıldığında bu değer 0,9525'e yükselmiştir. Benzer şekilde, ortalama duyarlılık değeri 0,9759'dan 0,9855'e çıkmıştır. Bu artışlar, üretilmiş özniteliklerin modelin genelleme yeteneğini artırdığını ve daha dengeli bir sınıflandırma performansı sağladığını göstermektedir.

Sonuç olarak, üretilmiş öznitelikler kullanılarak eğitilen modelin, hem sınıf bazlı hem de genel performans açısından orijinal özniteliklere kıyasla daha başarılı olduğu söylenebilir. Özellikle saldırı tespit sistemleri açısından kritik öneme sahip olan nadir ve zor ayırt edilen saldırı türlerinde sağlanan iyileşmeler, önerilen öznitelik üretim yaklaşımının etkinliğini ortaya koymaktadır.

4.9.2. Literatürdeki Diğer Yöntemlerle Karşılaştırma

Bu bölümde, önerilen saldırı tespit modeli ile literatürde ToN-IoT veri seti üzerinde eğitilmiş farklı saldırı tespit sistemlerinin karşılaştırmalı sonuçları sunulmaktadır. Karşılaştırma, sınıf bazında kesinlik, duyarlılık ve F1-skoru metrikleri üzerinden gerçekleştirilmiştir. Literatür ile adil bir kıyaslama yapabilmek için, eğitilen iki modelden daha iyi olanı, yani, üretilmiş özniteliklerden oluşan model karşılaştırma için esas alınmıştır.

Karşılaştırma yapılacak olan ilk model olan EvolCostDeep (Telikani ve diğ., 2022), Endüstriyel Nesnelerin İnterneti (IIoT) ortamlarında saldırı tespiti amacıyla geliştirilmiş özel bir DL çerçevesidir. Bu model, IIoT sistemlerinde nadir görülen (düşük frekanslı)

saldırı türlerini daha etkin biçimde tespit edebilmek için evrimsel optimizasyon tabanlı bir yaklaşım kullanmaktadır. Model; ön işleme adımlarını, sezgisel başlatma stratejilerini ve JADE algoritması ile optimize edilen bir maliyet matrisi yapısını bir araya getirmektedir. Eğitim sürecinde yığılmış oto-kodlayıcılar ve CNN ile birlikte kullanılmakta ve sınıf dengesizliğini gözeten maliyet-duyarlı bir kayıp fonksiyonu ile öğrenme gerçekleştirilmektedir. Ayrıca EvolCostDeep, IIoT saldırı tespit problemlerinde sıklıkla karşılaşılan veri dengesizliği sorununu doğrudan ele almakta ve ölçeklenebilirlik ile algılama verimliliğini artırmak amacıyla sis bilişim (fog computing) ortamında çalışmaya uygun bir yapı sunmaktadır. Modelin etkinliği, UNSW-NB15 ve ToN-IoT gibi kıyaslama veri setleri üzerinde gerçekleştirilen deneylerle doğrulanmış ve diğer yöntemlere kıyasla daha başarılı sonuçlar elde edildiği gösterilmiştir.

İkinci model olan CMTSNN (Zhu ve diğ., 2023) ise, IoT ağlarında anormal ve şifrelenmiş trafiğin tespiti için önerilmiş çok sınıflı bir sınıflandırma yaklaşımıdır. CMTSNN modeli, öznitelik çıkarımı aşamasında BiLSTM ve 1D-CNN tabanlı hibrit bir mimari kullanmaktadır. Veri dengesizliği problemini azaltmak amacıyla maliyet ceza matrisi entegre edilmiş ve sınıflandırma aşamasında geliştirilmiş bir çapraz-entropy kayıp fonksiyonu tercih edilmiştir. Deneysel sonuçlar, CMTSNN modelinin doğruluk, kesinlik, duyarlılık ve F1 skoru metrikleri açısından diğer derin öğrenme tabanlı modellere kıyasla daha üstün performans sergilediğini göstermektedir. Ayrıca, özellikle şifreli trafik içerisinde yer alan azınlık sınıf örneklerinin tespitinde etkili sonuçlar elde edilmesi, modelin gerçek dünya uygulamaları açısından potansiyelini ortaya koymaktadır.

Son olarak LSTM-SPIP modeli (Keshk ve diğ., 2023), saldırı tespitinde açıklanabilirliği ön planda tutan derin öğrenme tabanlı bir çerçevedir. Model, saldırı tespiti için Uzun Kısa Süreli Bellek (LSTM) ağlarını kullanmakta ve karar verme sürecinin anlaşılabilirliğini artırmak amacıyla kendi kendini açıklayabilen (self-explainable) bir mekanizma ile desteklenmektedir. Yerel ve küresel açıklama yöntemlerini birlikte ele alan bu yaklaşım, saldırı tespit sistemlerinde yorumlanabilirlik problemine önemli bir çözüm sunmaktadır. Model kapsamında SHAP ve Permütasyon Öznitelik Önemi (Permutation Feature Importance – PFI) gibi açıklanabilir yapay zekâ (XAI) teknikleri kullanılarak, belirli saldırı türlerinin tespitinde etkili olan kritik öznitelikler belirlenmektedir.

Modellere ilişkin karşılatırma sonuçları, Tablo 4.16'da sunulmuştur. Tablo 4.16 incelendiğinde, önerilen modelin özellikle DoS, Scanning ve Backdoor saldırı türlerinde tüm değerlendirme metrikleri açısından literatürde yer alan yöntemlerden daha yüksek performans sergilediği görülmektedir.

Bu saldırı türlerinde hem duyarlılık hem de kesinlik değerlerinin yüksek olması, modelin ilgili saldırıları hem doğru şekilde tespit edebildiğini hem de yanlış alarm üretme oranının düşük olduğunu göstermektedir.

Injection, Password ve DDoS saldırı türleri için ise önerilen model, duyarlılık ve F1-skoru bakımından EvolCostDeep modeline kıyasla daha başarılı sonuçlar üretirken, kesinlik metriğinde görece daha düşük değerler elde etmiştir. Bu durum, önerilen modelin bu saldırı türlerinde daha fazla saldırıyı yakalayabildiğini ancak bazı durumlarda yanlış pozitif üretme eğiliminin bulunduğunu göstermektedir. Buna karşın, özellikle saldırı tespit sistemleri açısından duyarlılığın yüksek olması, güvenlik bağlamında tercih edilen bir özellik olarak değerlendirilmektedir.

Sadece XSS ve MITM saldırı türlerinde EvolCostDeep modelinin önerilen modeli geride bıraktığı gözlemlenmiştir. Ancak MITM saldırılarının veri seti içerisindeki örnek sayısının oldukça sınırlı olduğu dikkate alındığında, bu saldırı türü için elde edilen sonuçların genellenebilirliği sınırlıdır. Dolayısıyla, MITM saldırılarına ilişkin daha sağlıklı bir karşılaştırma yapılabilmesi için daha büyük ve dengeli veri kümelerine ihtiyaç bulunmaktadır. Genel değerlendirme yapıldığında, önerilen modelin EvolCostDeep yaklaşımına kıyasla genel performans açısından daha başarılı olduğu söylenebilir. Bu sonuçların, daha karmaşık ve hesaplama maliyeti yüksek derin öğrenme mimarileri yerine, daha basit bir model kullanılarak elde edilmiş olması özellikle dikkat çekicidir. Bu durum, güçlü bir öznitelik mühendisliği ve uygun bir eğitim şeması tasarlandığında, geleneksel makine öğrenmesi modellerinin hâlen rekabetçi olabileceğini ortaya koymaktadır.

Önerilen model ayrıca, sinir ağları tabanlı bir saldırı tespit sistemi olan CMTSNN ile de karşılaştırılmıştır. CMTSNN çalışmasında model performansı yalnızca F1-skoru üzerinden raporlandığı için, bu karşılaştırma F1-skoru bazında gerçekleştirilmiştir.

Tablo 4.16. Diğer Yöntemler ile Karşılaştırmalı Sonuçlar

Etiket	**Önerilen Model**			**EvolCostDeep**			**CMTSNN**	**LSTM-SPIP**		
	Kesinlik	**Duyarlılık**	**F1 Skoru**	**Kesinlik**	**Duyarlılık**	**F1 Skoru**	**F1 Skoru**	**Kesinlik**	**Duyarlılık**	**F1 Skoru**
Normal (Normal)	**0,999**	0,982	0,991	0,988	**0,99**	0,989	**0,997**	Veri Yok (No Data)		
Hizmet Reddi (DoS)	**0,988**	**0,999**	**0,993**	0,981	0,98	0,98	0,984	Veri Yok (No Data)		
Enjeksiyon (Injection)	0,973	**0,992**	0,982	**0,998**	0,96	0,979	**0,987**	Veri Yok (No Data)		
Arka Kapı (Backdoor)	**0,997**	**0,999**	**0,998**	0,994	0,989	0,991	0,976	0,870	0,901	0,885
Tarama (Scanning)	**0,997**	**0,999**	**0,998**	0,86	0,993	0,921	0,988	0,796	0,894	0,842
Fidye Yazılımı (Ransomware)	0,792	**0,978**	0,875	**1,000**	0,957	**0,978**	0,878	Veri Yok (No Data)		
Siteler Arası Betik (XSS)	**0,952**	0,931	0,942	0,938	**0,988**	0,962	**0,996**	0,629	0,736	0,678
Parola Saldırısı (Password)	0,987	**0,989**	**0,988**	0,998	0,971	0,984	**0,993**	Veri Yok (No Data)		
Dağıtık Hizmet Reddi (DDoS)	0,972	**0,986**	0,979	**0,976**	0,974	0,974	**0,993**	0,822	0,902	0,859
Ortadaki Adam (MITM)	0,638	**1,000**	0,779	1,000	0,802	0,89	**0,939**	Veri Yok (No Data)		

Tablo 4.16'ya göre, dokuz saldırı sınıfı arasında önerilen model en yüksek F1-skorunu birden fazla sınıf için elde ederken, EvolCostDeep yalnızca tek bir sınıf için en yüksek F1-skorunu sunabilmiştir. CMTSNN modeli ise beş farklı sınıf için en yüksek F1-skoruna ulaşmıştır. Bu sonuçlar, bazı saldırı türlerinde önerilen modelin daha başarılı olduğunu, bazı durumlarda ise CMTSNN'in üstünlük sağladığını göstermektedir.

Bu noktada dikkat edilmesi gereken önemli bir husus, modellerin test edildiği veri bölme stratejileridir. CMTSNN modelinde veri setinin %90'ı eğitim, yalnızca %10'u test amacıyla kullanılmıştır. Buna karşılık, bu çalışmada %70 eğitim – %30 test oranı tercih edilmiştir. Test aşamasında daha fazla örneğin kullanılması, elde edilen performans metriklerinin daha güvenilir ve genellenebilir olmasını sağlamaktadır. Bu açıdan değerlendirildiğinde, önerilen modelin daha zorlu bir test senaryosunda rekabetçi sonuçlar ürettiği söylenebilir.

Son olarak, LSTM-SPIP modeli ile yapılan karşılaştırmada, bu yöntemin özellikle ikili sınıflandırma problemleri için tasarlandığı, çok sınıflı saldırı tespitinin ise ikincil bir analiz olarak ele alındığı görülmektedir. LSTM-SPIP yaklaşımı, açıklanabilir yapay zekâ teknikleri ile öznitelik çıkarımına odaklanmakta olup, çok sınıflı saldırı tespitinde önerilen modelin gerisinde kalmıştır. Tablo 4.16'da görüldüğü üzere, LSTM-SPIP modeli önerilen yaklaşım tarafından tüm saldırı sınıflarında geride bırakılmıştır.

Genel olarak elde edilen bulgular, bu çalışmada sunulan modelin; karmaşık derin öğrenme mimarileri kullanmadan, uygun veri ön işleme, dengeli örnekleme stratejileri ve etkili öznitelik mühendisliği ile literatürdeki ileri seviye yöntemlerle rekabet edebildiğini, bazı saldırı türlerinde ise daha üstün performans sunduğunu açıkça göstermektedir.

Bu durum, saldırı tespit sistemlerinde model karmaşıklığından ziyade, veri temsili ve öğrenme şemasının belirleyici bir rol oynadığını ortaya koymaktadır.

4.10. Sonuçlar

Bu bölümde, Nesnelerin İnterneti (IoT) cihazları için çok sınıflı saldırı tespitine yönelik olarak Rastgele Orman (Random Forest) tabanlı bir saldırı tespit sistemi önerilmiştir. Model eğitim sürecinde, eğitim ve test veri setlerinde yer alan örneklerin farklı gruplara

ait olacak şekilde ayrıştırıldığı bir eğitim şeması tasarlanmış ve bu sayede olası yanlılık (bias) durumlarının azaltılması hedeflenmiştir. Buna ek olarak, katlar (fold) arasındaki sınıf dağılımlarının oransal olarak korunmasını sağlayan tabakalama (stratification) adımı uygulanarak, modelin tüm saldırı türlerini yeterli düzeyde öğrenmesi amaçlanmıştır.

Önerilen modelde, farklı ağ saldırı türleri arasındaki ayırt edilebilirliği artırmak amacıyla iteratif bir öznitelik oluşturma algoritması kullanılarak üretilmiş öznitelikler değerlendirilmiştir. Bu yaklaşım sayesinde, ham verilerden doğrudan elde edilmesi zor olan anlamlı ve ayrıştırıcı özniteliklerin modele kazandırılması sağlanmıştır. Ayrıca, çalışmada standart eğitim yaklaşımlarının çoğu zaman yetersiz kalabileceği gösterilmiş ve bir saldırı tespit sisteminin daha güvenilir şekilde eğitilebilmesi için uygun bir eğitim metodolojisi sunulmuştur.

Uygulanan yöntemler sonucunda, basit ancak etkili bir Rastgele Orman modeli tasarlanmış ve bu modelin, daha karmaşık derin öğrenme tabanlı yöntemlerle rekabet edebilir, hatta birçok açıdan daha başarılı sonuçlar üretebildiği gösterilmiştir. ToN-IoT veri seti üzerinde gerçekleştirilen deneyler sonucunda, model hem eğitim hem de test veri setleri için %99,99 ROC-AUC başarımına ulaşmıştır. Ayrıca, dokuz saldırı sınıfının sekizi için en yüksek duyarlılık değerleri elde edilerek, modelin özellikle yanlış negatif alarmlara karşı oldukça dayanıklı olduğu ortaya konmuştur.

Bunun yanı sıra, DoS, Backdoor, Scanning ve Password saldırı türlerinde F1-skorlarında belirgin iyileşmeler sağlanmıştır. CMTSNN modeli ile yapılan karşılaştırmalarda, bazı saldırı türleri için daha düşük F1-skorları elde edilmiş olsa da, bu sınıfların beşinden dördü için 0,97–0,98 aralığında oldukça yüksek F1-skorları yakalanmıştır. Ayrıca, Ransomware ve MITM saldırıları haricindeki tüm sınıflar için 0,95–0,99 aralığında yüksek kesinlik değerleri elde edilmiştir.

Önerilen yaklaşımın basit bir model yapısına sahip olması; bakım kolaylığı, hesaplama maliyeti ve uygulanabilirlik açısından önemli avantajlar sunmaktadır. Bu sonuçlar, uygun veri ön işleme adımları, dengeli bir eğitim şeması ve etkili öznitelik mühendisliği ile desteklenen geleneksel makine öğrenmesi modellerinin, karmaşık DL modellerine kıyasla hâlen güçlü ve güvenilir sonuçlar üretebileceğini açıkça göstermektedir.

5. AĞAÇ TABANLI BİR ÖZNİTELİK ÜRETME ALGORİTMASI

Bölüm 4.7.'de, Algoritma 3 içerisinde, (Piramuthu ve Sikora, 2009) tarafından sunulan öznitelik üretme algoritması üzerinde çeşitli değişiklikler yapılarak tasarlanan, genişletilmiş bir iteratif öznitelik üretme algoritması ToN-IoT veri setinde uygulanarak, kompleks öznitelikler üretilmiştir. Daha sonrasıda, bu üretilen özniteliklerin etkisi orjinal öznitelikler ile kıyaslanmış ve Bölüm 4.9.'da da gösterildiği üzere model tahminlerinde olumlu tahmin katkısı yaparak, IoT saldırı tespit sistemlerinde, öznitelik üretim algoritmalarının önemli olabileceğini göstermiştir.

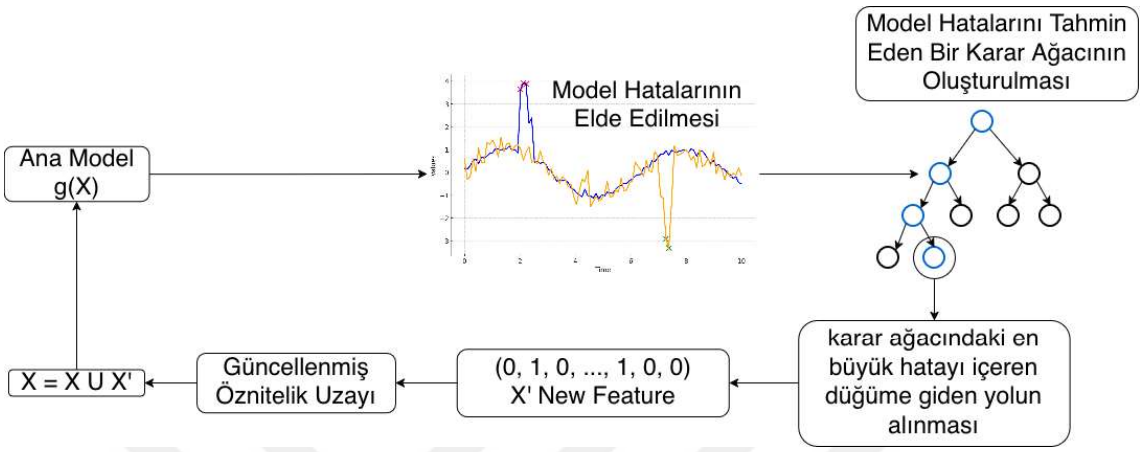
Bu algoritmada, öznitelikler matematiksel operatörler aracılığıyla yinelemeli biçimde birleştirilerek yeni öznitelikler üretilmiş ve bu öznitelikler modelin eğitim sürecinde kullanılmıştır. Ancak literatürde daha önce önerilen bazı öznitelik çıkarma yaklaşımlarının temel sınırlılığı, veri kümesinin yapısal ve istatistiksel karakteristiklerini dikkate almaksızın, doğrudan önceden tanımlı matematiksel operatör kombinasyonları üzerinden öznitelik üretimine odaklanmalarıdır. Bu durum, gereksiz derecede fazla öznitelik kombinasyonunun denenmesine yol açmakta; ayrıca model tarafından oluşturulan karmaşık özniteliklerin hem yorumlanabilirliğini azaltmakta hem de analitik açıdan anlaşılması güç yapılar ortaya çıkarmaktadır.

5.1. Ağaç Tabanlı Yorumlanabilir Bir Öznitelik Üretme Algoritması

Bu bölümde sunulan çalışmada ise amaç, yalnızca matematiksel kombinasyonlara dayalı bir üretim sürecinden ziyade, modelin yaptığı hatalardan öğrenen bir öznitelik üretim mekanizması uygulamaktır. Önerilen yöntem, veri kümesinin farklı karakteristiklerini temsil edebilen, daha yorumlanabilir ve aynı zamanda sınıflandırma başarımını artıran özniteliklerin seçilmesini hedeflemektedir. Bu sayede, hem öznitelik uzayının kontrolsüz biçimde büyümesi engellenmiş hem de model performansı ile açıklanabilirlik arasında daha dengeli bir yapı kurulmuştur.

Buna istinaden Açıklanabilir Güçlendirilmiş Doğrusal Regresyon (Explainable Boosted Linear Regression, EBLR) algoritması modifiye edilerek, bu algoritma üzerine geliştirilmiş bir algoritma tasarlanmıştır (Ilic ve diğ., 2021). Bu algoritma temel olarak

eğitilen bir ana modelin hataları üzerinden tekrar eğitilen bir ağaç tabanlı model ile yenilikçi öznitelikler elde etmeyi amaçlar. Temel olarak regresyon ağaçlarını kullanır (Breiman ve diğ., 2017). Bu algoritmanın temel akış diyagramı Şekil 5.1'de gösterilmiştir.



Şekil 5.1. EBLR Algortimasının İş Akışı

EBLR algoritması, zaman serisi tahmini amacıyla doğrusal modeller ile regresyon ağaçlarını bir araya getirerek, artık (residual) hataların analizi üzerinden öznitelik üretimine dayalı yinelemeli bir iyileştirme süreci sunmaktadır. İlk aşamada, Temel Öğrenici (Base Learner - $g(x)$) olarak adlandırılan genel bir tahmin modeli eğitilerek başlangıç tahminleri elde edilir.

İkinci aşamada ise, mevcut modelin artıklarını (hatalarını) analiz etmek amacıyla, mevcut tüm öznitelikler kullanılarak, hesaplanan artık değerlerin etkilet olduğu bir regresyon ağacı eğitilir. Bu ağaç yapısı, öznitelik uzayında hatanın yüksek olduğu bölgeleri işaret eden kuralları öğrenerek, hatanın hangi öznitelik bölgelerinde yoğunlaştığını temsil etmeyi amaçlar. Hata miktarına en fazla katkı sağlayan kural belirlenerek -ki bu en yüksek hatanın olduğu düğümdür-, bu kuraldan yeni bir öznitelik türetilir.

Oluşturulan bu yeni öznitelik, ilk aşamadaki tahmin modeline dahil edilirerk, modelin bu hata bilgisini de hesaba katacak şekilde, yeniden eğitilmesi sağlanır. Bu süreç, önceden tanımlanmış bir durdurma kriteri sağlanana kadar yinelemeli olarak devam eder.

Bu yöntem esas olarak zaman serisi tahmini problemleri için geliştirilmiş bir algoritmadır. Bu bağlamda, öznitelik üretiminde kullanılan ağaç tabanlı model, ana tahmin modelinden

(Şekil 5.1'de $g(x)$ ile gösterilen) elde edilen ve tahmin edilen değerler ile gerçek değerler arasındaki farkı nicel olarak ölçen sürekli hata metrikleri (Ortalama Karekök Hata – RMSE, Ortalama Mutlak Hata – MAE vb.) kullanılarak eğitilmektedir.

Bu çalışmada ise temel amaç, on farklı sınıfı yüksek doğruluk ve kesinlik ile tahmin edebilen bir sınıflandırma modeli geliştirmektir. Dolayısıyla, EBLR algoritmasının öznitelik çıkarım mekanizmasının sınıflandırma problemlerine uyarlanabilmesi için bazı yapısal değişikliklerin yapılması gerekmektedir. Bu amaçla, artık (residual) hata metriği olarak log-kaybı (log loss) fonksiyonu kullanılmıştır (Cox, 1958).

Log-kaybı, tahmin edilen sınıf olasılıklarının gerçek etiketlerle ne derece uyumlu olduğunu ölçen bir hata metriğidir. Düşük log-kayıp değerleri, modelin veri kümesine iyi uyum sağladığını gösterirken, ideal durumda log-kayıp değerinin sıfıra yaklaşması beklenir. Buna karşılık, yüksek log-kayıp değerleri, tahmin edilen sınıf olasılıklarının gerçek etiketlerden önemli ölçüde saptığını ifade etmektedir. Log-kaybı metriği Denklem (5.1)'de gösterilmiştir:

$$L(y,p) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{5.1}$$

Burada:

- N, örnek sayısını,
- y_i, i. örneğin gerçek etiketini (0 veya 1),
- p_i, i. örneğin sınıf 1'e ait olma olasılığını ifade etmektedir.

EBLR algoritmasının sınıflandırma problemine uyarlanmasında çözülmesi gereken bir diğer önemli husus, her bir sınıf için ayrı kuralların çıkarılması gerekliliğidir. Orijinal EBLR yaklaşımı sürekli hedef değişkenler üzerinde çalışmakta ve yalnızca tek bir hedef değişken için artık hataları kullanarak kural üretmektedir. Ancak bu çalışmada ele alınan IoT saldırı tespit probleminde, her bir sınıf kendi hata profiline(bu çalışmada log-kayıp değerleri) sahiptir ve bu profillerin ayrı ayrı ele alınması gerekmektedir.

Bu nedenle, geliştirilen yinelemeli algoritmanın her bir sınıf için en yüksek hatayı

üreten düğümü (node) belirlemeye odaklanması gerekmektedir. Bu durum, algoritmanın tasarımında özellikle dikkate alınmıştır. Bu kapsamda, log-kayıp metriğini kullanarak her sınıf için en yüksek hata üreten düğümü belirleyen algoritmanın sözde kodu Algoritma 5'te sunulmaktadır.

Algoritma 5: Yinelemeli Öznitelik Üretimi için Sözde Kod

Gerekli:
D: Eğitim veri kümesi
C: Sınıf kümesi ($|C| = K$)
N: Veri kümesindeki toplam kayıt sayısı
max_iter: Maksimum yineleme sayısı veya durdurma kriteri
Ensure:
D: Genişletilmiş öznitelik kümesini içeren veri kümesi
Yinelemeli Öznitelik Üretimi Başlat:
while durdurma kriteri sağlanana kadar **do**
    $rule_list \leftarrow \emptyset$
    Temel Model Eğitimi:
    Rastgele Orman modeli eğit:
        $RF \leftarrow TrainRandomForest(D)$
    Log-kayıp değerlerini hesapla:
        $L \leftarrow LogLoss(RF, D)$
    Sınıf Bazlı Kural Çıkarımı:
    for her bir sınıf $c_i \in C$ **do**
        $D_{c_i} \leftarrow$ yalnızca c_i sınıfına ait örnekler
        Karar ağacı eğit:
            $DT_{c_i} \leftarrow TrainDecisionTree(D_{c_i}, L)$
        En yüksek hataya sahip düğümden kural çıkar:
            $rule_{c_i} \leftarrow ExtractRule(DT_{c_i})$
        Tespit edilen kuralı kural listesine ekle:
        $rule_list \leftarrow rule_{c_i}$
    end for
    Öznitelik Uzayını Genişlet:
    $D \leftarrow D \cup rule_list$
end while

Bu algoritmada, öznitelik uzayının kontrolsüz biçimde büyümesini ve buna bağlı olarak aşırı uyum (overfitting) riskinin ortaya çıkmasını önlemek amacıyla ölçüt tabanlı bir yineleme yapısı benimsenmiştir. Algoritmanın her bir yinelemesinde, her sınıf için ayrı öznitelik üretimi gerçekleştirilmesi, öznitelik uzayının hızlı bir şekilde genişlemesine neden olabileceğinden, bu çalışma kapsamında yineleme sayısı sınırlandırılmıştır.

Deneysel sonuçlar, iki yinelemeden sonra elde edilen ek özniteliklerin model başarımına anlamlı bir katkı sağlamadığını, beş yinelemeden sonra ise aşırı uyum eğiliminde olduğu özlemlendiğinden, en fazla iki yineleme ile sınırlandırılmıştır.

Algoritmanın başlangıcında, durdurma kriteri sağlanana kadar devam eden döngü içerisinde, yeni üretilecek kuralları saklamak amacıyla boş bir kural listesi oluşturulmaktadır. Ardından, mevcut öznitelik kümesi kullanılarak bir Rastgele Orman (Random Forest) modeli eğitilmekte ve bu model üzerinden her bir örnek için log-kayıp (log loss) değerleri hesaplanmaktadır. Bu log-kayıp değerleri, modelin hangi örneklerde ve hangi sınıflar için hata yaptığını nicel olarak temsil etmektedir.

Bu aşamayı takiben, sınıf bazlı kural çıkarımı aşamasına geçilmektedir. Bu aşamada, her bir sınıf için veri kümesi yalnızca ilgili sınıfa ait örnekleri içerecek şekilde alt kümelere ayrılmaktadır. Daha sonra, bu alt kümeler üzerinde log-kayıp değerleri hedef değişken olarak kullanılarak bir karar ağacı modeli eğitilmektedir. Eğitilen karar ağacı üzerinde, en yüksek hata değerine sahip düğüme karşılık gelen yol bir mantıksal ifade olarak tanımlanmakta ve bu ifade yeni bir öznitelik olarak çıkarılmaktadır. Elde edilen bu kurallar, yeni oluşturulan öznitelik listesine eklenmektedir.

Son aşamada, oluşturulan tüm yeni öznitelikler mevcut öznitelik kümesi ile birleştirilerek öznitelik uzayı genişletilmektedir. Bu süreç, belirlenen durdurma kriteri sağlanana kadar yinelemeli olarak devam etmektedir. Yinelemeler tamamlandığında, model hatalarından öğrenerek oluşturulmuş, yorumlanabilir ve veri karakteristiğinin farklı yönlerini temsil eden bir öznitelik uzayı elde edilmektedir.

Bu yaklaşım uygulanmadan önce, ToN-IoT veri setinin ham hâlinde bulunan 46 öznitelik, Bölüm 4'te bahsedilen, Bölüm 4.2. - 4.6. arasındaki tüm ön işleme adımları aynen uygulanarak, ve Algoritma 5 uygulanarak 20 özniteliğe indirgenmiştir. Bu öznitelik listesi Tablo 5.1'de gösterilmiştir.

Burada f_1, f_2, f_3 ve f_4 öznitelikleri, EBLR tabanlı öznitelik üretim algoritması kullanılarak elde edilen türetilmiş öznitelikleri temsil etmektedir. Bu özniteliklerin tam mantıksal ifadeleri aşağıda sunulmaktadır:

Tablo 5.1. Önerilen Sistem İçin Kullanılan Öznitelik Listesi

Öznitelik	**Açıklama**
src_port	Kaynak port numarası. Ağ bağlantısının başlatıldığı portu ifade eder.
dst_port	Hedef port numarası. Bağlantının yönlendirildiği hedef portu belirtir.
duration	Bağlantı süresi. Bir ağ bağlantısının aktif kaldığı zaman aralığını ölçer.
src_bytes	Kaynaktan gönderilen bayt sayısı. Kaynak makineden hedefe iletilen veri miktarını gösterir.
missed_bytes	İletim sırasında kaybedilen bayt sayısı. Bağlantı sürecinde iletilemeyen veya kaybolan veriyi ifade eder.
src_ip_bytes	Kaynak IP adresine gönderilen toplam bayt sayısı.
dst_ip_bytes	Hedef IP adresine gönderilen toplam bayt sayısı.
dns_rejected	Reddedilen DNS sorgularının sayısı.
dns_qtype_group_2	DNS sorgu tipi grup 2. Farklı DNS sorgu türlerinin belirli bir grup altında toplanmış hâlidir.
dns_qtype_group_3	DNS sorgu tipi grup 3. DNS sorgu türlerini temsil eden bir diğer gruplandırma.
proto_tcp	TCP protokolünün kullanılıp kullanılmadığını gösteren ikili (binary) öznitelik.
proto_udp	UDP protokolünün kullanılıp kullanılmadığını gösteren ikili (binary) öznitelik.
conn_state_cat_OTH	"OTHER" bağlantı durumu kategorisi. Diğer bağlantı durumlarını temsil eder.
conn_state_cat_PART	"PARTIAL" bağlantı durumu kategorisi. Tamamlanmamış veya kısmi bağlantıları ifade eder.
conn_state_cat_S0	"S0" bağlantı durumu kategorisi. Başlatılmış ancak tamamlanmamış bağlantıları gösterir.
conn_state_cat_SUC	"SUCCESS" bağlantı durumu kategorisi. Başarıyla tamamlanmış bağlantıları ifade eder.
f1	Hedefe daha fazla veri gönderilen, DNS grup 2 sorgularının düşük olduğu ve diğer bağlantı durumlarının baskın olmadığı bağlantıları temsil eden türetilmiş öznitelik.
f2	Hedefe gönderilen verinin az olduğu, UDP protokolünün kullanılmadığı, bağlantı süresinin kısa olduğu ve kaynağın yüksek port kullandığı bağlantıları temsil eden türetilmiş öznitelik.
f3	f2 özniteliğine benzer yapıda olup, daha düşük kaynak veri miktarı ve daha düşük kaynak port numaraları içeren bağlantıları temsil eden türetilmiş öznitelik.
f4	UDP protokolü kullanılarak gerçekleştirilen, yüksek veri transferi ve kaynaktan önemli miktarda veri içeren bağlantıları temsil eden türetilmiş öznitelik.

f_1 : dst_ip_bytes $> 14.0 \wedge$ dns_qtype_group_2 $\leq 0.5 \wedge$

dst_port $\leq 9101.5 \wedge$ proto_udp $\leq 0.5 \wedge$

conn_state_cat_OTHER ≤ 0.5

f_2 : dst_ip_bytes $\leq 14.0 \wedge$ proto_udp $\leq 0.5 \wedge$

duration $\leq 0.266 \wedge$ src_ip_bytes $> 42.0 \wedge$

src_port > 1011.5

f_3 : dst_ip_bytes $\leq 14.0 \wedge$ proto_udp $\leq 0.5 \wedge$

duration $\leq 0.266 \wedge$ src_ip_bytes $\leq 42.0 \wedge$

src_port > 88.0

f_4 : dst_ip_bytes $> 14.0 \wedge$ dns_qtype_group_2 $\leq 0.5 \wedge$

dst_port $\leq 9101.5 \wedge$ proto_udp $> 0.5 \wedge$

src_ip_bytes > 81.5

f_1 özniteliği, proto_udp ≤ 0.5 koşulu ile tanımlandığı üzere ağırlıklı olarak TCP tabanlı trafiği temsil etmektedir. TCP trafiğinin UDP'ye kıyasla daha güvenilir olması nedeniyle, bu öznitelik özellikle normal ağ trafiği ile anormal davranışların ayrıştırılmasında önemli rol oynamaktadır. Ayrıca belirli DNS sorgu gruplarının (dns_qtype_group_2) ve bilinmeyen bağlantı durumlarının (conn_state_cat_OTHER) dışlanması, bu özniteliğin sınıflandırma başarımına katkısını artırmaktadır.

f_2 özniteliği, kısa süreli bağlantıları (duration ≤ 0.266) ve kaynak IP adreslerinden yüksek miktarda veri aktarımını (src_ip_bytes > 42.0) temsil etmektedir. Aynı zamanda bu bağlantıların belirli port aralıklarında gerçekleştiğini göstermektedir (src_port > 1011.5). Bu öznitelik, özellikle kısa ömürlü, TCP tabanlı ve kaynak ağırlıklı anormal veri akışlarının tespitinde etkilidir.

f_3 özniteliği ise, kaynak IP adreslerinden daha düşük veri transferlerinin gerçekleştiği (src_ip_bytes ≤ 42.0) ve belirli portlar üzerinden çalışan bağlantıları tanımlamaktadır

(src_port $>$ 88.0). Bu öznitelik, düşük bant genişliğine sahip ancak yüksek frekanslı saldırı davranışlarını modellemekte olup, küçük veri hacimleriyle gerçekleştirilen tekrarlayan anormalliklerin tespitinde öne çıkmaktadır.

f_4 özniteliği, UDP tabanlı trafiği (proto_udp $>$ 0.5) ve yüksek veri transferlerini (src_ip_bytes $>$ 81.5) temsil etmektedir. Bu öznitelik aynı zamanda belirli DNS sorgu grupları ve hedef port kısıtlarını da içermektedir. Bu nedenle, özellikle yüksek hacimli UDP trafiği ile ilişkili Dağıtık Hizmet Engelleme (DDoS) saldırılarının modellenmesinde kritik bir rol oynamaktadır.

Bu dört öznitelik birlikte değerlendirildiğinde, kaynak ve hedef IP adresleri arasındaki veri transfer hacimleri (src_ip_bytes, dst_ip_bytes), DNS sorgu türleri ve bağlantı durumları gibi faktörleri bütüncül bir biçimde temsil etmektedir. Bu entegrasyon, modelin normal ve anormal ağ trafiği davranışlarını ayırt etme yeteneğini önemli ölçüde artırmaktadır. Özellikle f_1 ve f_4 DNS sorguları ve bilinmeyen bağlantı durumlarını hedef alırken, f_2 ve f_3 veri akışı hacimleri ve bağlantı süreleri üzerinden saldırı tespitini desteklemektedir. Bu özellikler birlikte ele alındığında, DoS, DDoS ve MITM gibi saldırı türlerinin tespitinde kritik bir rol üstlenmektedir.

Basit modellerin düşük hesaplama maliyeti göz önüne alındığında, önerilen bu yinelemeli öznitelik üretim ve seçim süreci, gerçekçi zaman kısıtları içinde yerel bir bilgisayar ortamında uygulanabilir niteliktedir. Bu durum, yöntemin pratik uygulanabilirliğini ve ölçeklenebilirliğini artırmaktadır.

5.2. Model Eğitimi

Eğitim ve test veri setlerinin ayrışımı, tabakalandırma işlemleri, kayıp veri analizi, çoklu korelasyona bağlı öznitelik filtrelenmesi gibi hususlar, tamamen Bölüm 4'te açıklandığı gibi olup, tek farklılık kullanılan öznitelik üretme algoritmasından dolayı, bu bölümde doğrudan sonuçlar üzerine odaklanılacaktır.

Model eğitimleri için, 10-katlı çapraz geçerleme aşamasında, yine SCUT yeniden örnekleme algoritması kullanılmış, ve eğitim aşamasındaki her iterasyon için, veri setindeki dengeleme işlemleri gerçekleştirilmiştir.

Model değerlendirmesinde, yine Bölüm 4 altında yer alan değerlendirme aşamaları gibi, öncelikli olarak Orjinal öznitelikler ile Üretilmiş özniteliklerin olduğu modeller karşılaştırılmıştır. Buna ek olarak elde edilen model, literatürdeki diğer derin öğrenme tabanlı popüler araçlarla karşılaştırılarak, modele ait avantaj ve dezavantaj oluşturan durumlar incelenmiştir.

5.2.1. Orjinal ve Üretilmiş Özniteliklerin Kıyaslanması

Bu bölümde, tamamen ToN-IoT'den alınmış ve önişleme aşamalarından geçirilmiş ham veri seti üzerinde bir eğitim şeması gerçekleştirilip, ToN-IoT üzerine Algoritma 5'te gerçekleştirilen ağaç tabanlı iteratif öznitelik üretme algoritmasının uygulandığı ve 27 öznitelikten (bkz. Tablo 5.1) oluşan modellerin başarım kıyaslamaları gerçekleştirilecektir. Model eğitiminde bu bölümde, farklı olarak, daha önce eğitim aşamasında kullanılan kayıp fonksiyonu (bkz. Denklem (4.2)) yerine doğrudan AUC-ROC metriği kullanılmaktadır. Denklem (4.2)'de gösterilen kayıp fonksiyonu, MITM saldırılarını iyi yakalamak için, MITM saldırılarının kesinlik değerlerini hesaba katmaktadır. Ancak bu durum, sonuçların özellik fidye yazılımı (ransomware) saldırı türünde, kesinlik değerinde düşüşlerin olmuşmasına sebebiyet vermiştir. Dolayısıyla, model başarısını sadece 700'e yakın örnek başarısına sahip olan MITM saldırılarını arttırmak yerine ve 13.000 üzerindeki Fidye Yazılımı saldırılarındaki performans düşüşüne değmeyeceğine karar verilmiştir. Sonuç olarak model eğitimlerinin, klasik AUC-ROC skoru üzerinden elde edilmesine karar verilmiştir.

Orjinal öznitelikler ile Orjinal ve Üretilmiş özniteliklerin kombinasyonundan oluşan bir öznitelik uzayından eğitilmiş modellere ait karşılaştırma, Tablo 5.2'de gösterilmiştir.

Tablo 5.2 incelendiğinde, hem orijinal hem de yeni üretilmiş öznitelikleri kullanan modelin, yalnızca orijinal özniteliklerle eğitilen modele kıyasla; performans metriğinde genel olarak daha başarılı olduğu görülmektedir. Tüm sınıflar değerlendirildiğinde, üretilmiş özniteliklerin modele eklenmesi özellikle daha karmaşık yapıya sahip veya veri setinde görece az temsil edilen saldırı türlerinde anlamlı iyileştirmeler sağlamıştır. Bu durum, geleneksel özniteliklerin bu tür saldırılara ait ayırt edici örüntüleri yakalamakta yetersiz kalabildiğini göstermektedir.

Tablo 5.2. Elde Edilen Sonuçlar

Sınıf	Orijinal Öznitelikler				Orijinal + Yeni Öznitelikler			
	Özgüllük (Specificity)	**Duyarlılık** (Recall)	**Kesinlik** (Precision)	**F-Skoru** (F-score)	**Özgüllük** (Specificity)	**Duyarlılık** (Recall)	**Kesinlik** (Precision)	**F-Skoru** (F-score)
Normal	0,9993	0,9881	**0,9995**	0,9938	**0,9994**	**0,9884**	**0,9996**	**0,9940**
DoS	**0,9989**	0,9765	0,9808	0,9786	**0,9989**	**0,9768**	**0,9810**	**0,9789**
Injection	**0,9955**	**0,9696**	0,9233	0,9459	**0,9955**	0,9693	**0,9238**	**0,9460**
Backdoor	**0,9999**	**1,0000**	**0,9993**	**0,9996**	**0,9999**	0,9998	**0,9993**	**0,9996**
Scanning	**0,9991**	**0,9922**	0,9836	**0,9879**	**0,9991**	0,9918	**0,9837**	0,9877
Ransomware	**0,9994**	**0,9995**	0,9888	**0,9946**	**0,9994**	**0,9995**	**0,9896**	0,9945
XSS	**0,9980**	**0,9759**	**0,9639**	**0,9699**	**0,9980**	**0,9759**	**0,9636**	0,9697
Password	0,9996	**0,9892**	**0,9943**	**0,9917**	**0,9997**	**0,9892**	**0,9943**	**0,9917**
DDoS	**0,9983**	**0,9803**	**0,9712**	**0,9757**	**0,9983**	0,9801	0,9708	0,9755
MITM	**0,9972**	0,8629	0,4633	0,6028	**0,9972**	**0,8804**	**0,4808**	**0,6220**
Ortalama	0,9976	0,9736	0,9209	0,9433	**0,9977**	**0,9751**	**0,9289**	**0,9482**

Her iki modelin de özgüllük değerlerinin yüzde 99'un üzerinde olması, pratik uygulamalarda yanlış alarm oranlarının düşük tutulması açısından kritik öneme sahiptir. Duyarlılık metriği açısından bakıldığında ise, orijinal ve yeni öznitelikleri birlikte kullanan modelin; enjeksiyon (Injection), XSS, MITM ve tarama (Scanning) saldırı sınıflarında yalnızca orijinal özniteliklere dayalı modele kıyasla daha yüksek başarım sergilediği görülmektedir. Bu bulgu, üretilmiş özniteliklerin özellikle gelişmiş ve karmaşık saldırı türlerinin tespitinde önemli katkılar sunduğunu ortaya koymaktadır.

Tüm değerlendirme metrikleri birlikte ele alındığında, yalnızca orijinal özniteliklerle eğitilen modelin arka kapı (Backdoor) ve DDoS saldırı türlerinde nispeten daha iyi sonuçlar verdiği; buna karşın orijinal ve yeni özniteliklerin birlikte kullanıldığı modelin normal trafik, DoS, enjeksiyon, MITM, tarama, parola saldırıları, fidye yazılımı (Ransomware) ve XSS sınıflarında daha üstün performans sergilediği görülmektedir. Bu durum, üretilmiş özniteliklerin özellikle farklı saldırı vektörlerini ele alırken daha dengeli bir kesinlik–duyarlılık dengesi sağladığını göstermektedir.

MITM saldırıları için kesinlik değerlerinin hâlen görece düşük kalması, bu sınıfın veri setinde oldukça sınırlı sayıda örnekle temsil edilmesine bağlanabilir. Buna rağmen, genel sonuçlar değerlendirildiğinde, üretilmiş özniteliklerin kullanıldığı modelin performansında belirgin bir iyileşme sağlandığı görülmektedir. Önerilen model; yüzde 99,94 özgüllük, yüzde 99,19 duyarlılık, yüzde 99,48 kesinlik ve yüzde 99,32 F-skoru değerlerine ulaşarak saldırı tespit kabiliyetinin önemli ölçüde geliştirildiğini ortaya koymaktadır.

Her iki yaklaşım da eğitim ve test veri setlerinde yüzde 99,99 ROC-AUC (Receiver Operating Characteristic – Area Under Curve) değerine ulaşmış olup, bu durum modelin genel saldırı tespit başarımının oldukça yüksek olduğunu göstermektedir. Ayrıca, ağ akışlarının ayrı ayrı ele alınması ve eğitim–test aşamalarında tabakalı grup katlamalı doğrulama stratejisinin kullanılması, aşırı uyum (overfitting) riskini azaltmış ve önerilen sistemin sağlamlığını desteklemiştir.

Özetle, üretilmiş özniteliklerin modele dâhil edilmesi, özellikle karmaşık ve veri setinde az temsil edilen saldırı türleri için saldırıların tespit ve sınıflandırma başarımını belirgin

biçimde artırmış; böylece daha güvenilir ve kapsamlı bir saldırı tespit sistemi çözümü sunulmasını sağlamıştır.

Buna ek olarak, farklı değerlendirme kriterleri açısından sonuçlar aşağıdaki şekilde özetlenebilir:

- Özgüllük: Genel olarak sabit kalmış veya "Orijinal + Yeni Öznitelikler" kümesinde ihmal edilebilir düzeyde artış göstermiştir. Özellikle DoS ve Parola (Password) sınıflarında küçük iyileşmeler gözlemlenmiştir.
- Duyarlılık: "Orijinal + Yeni Öznitelikler" yaklaşımı; Normal, DoS ve MITM sınıflarında daha yüksek duyarlılık sağlamıştır. Bu durum, gerçek pozitiflerin yakalanmasının kritik olduğu senaryolar için önemlidir.
- Kesinlik: DDoS sınıfı haricinde neredeyse tüm sınıflarda iyileşme gözlemlenmiştir. Özellikle DoS ve Parola saldırılarında artan kesinlik değerleri, yanlış pozitif oranlarının azaldığını göstermektedir.
- F-Skoru (F1-Score): Normal, DoS, Enjeksiyon, Arka Kapı, Fidye Yazılımı, Parola ve MITM sınıflarında belirgin iyileşmeler elde edilmiştir.

Sınıf bazında değerlendirildiğinde ise mevcut durum şu şekilde özetlenebilir:

- Normal: Özgüllük, duyarlılık, kesinlik ve F-skoru metriklerinin tamamında iyileşme gözlemlenmiştir.
- DoS: Duyarlılık, kesinlik ve F-skoru değerlerinde artış sağlanmıştır.
- Injection: Kesinlik ve F-skoru metriklerinde iyileşme elde edilmiştir.
- Backdoor: Belirgin bir iyileşme gözlemlenmemiştir.
- Scanning: F-skorunda sınırlı düzeyde artış sağlanmıştır.
- Ransomware, XSS, DDoS ve Password: Küçük ölçekli veya ihmal edilebilir düzeyde iyileşmeler gözlemlenmiş olup, orijinal özniteliklerin bu sınıflar için yeterli olduğu söylenebilir.
- MITM: Duyarlılık, kesinlik ve F-skoru metriklerinde belirgin iyileşmeler sağlanmış olsa da, genel performans diğer sınıflara kıyasla görece daha düşük kalmıştır.

Bu sonuçlar doğrultusunda, önerilen öznitelik üretim algoritmasının model başarımını

çok boyutlu olarak iyileştirdiği, herhangi bir sınıf veya metrik bazında anlamlı bir performans düşüşüne yol açmadığı ve genel başarımı artırdığı sonucuna varılabilir.

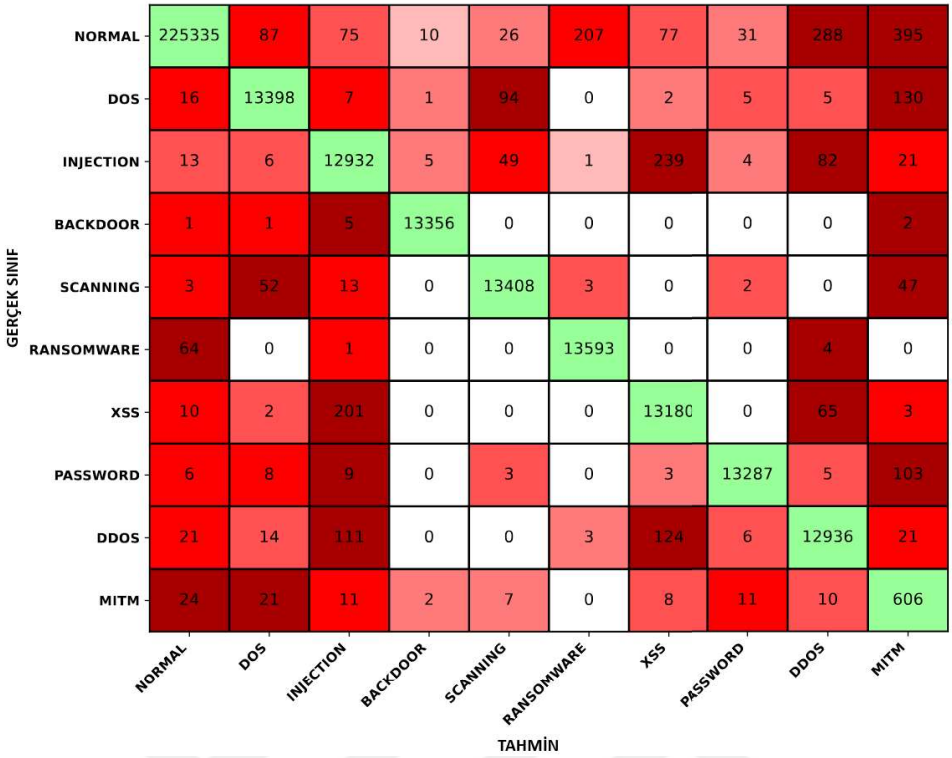
5.2.2. Karışıklık Matrisi Analizi

Karışıklık matrisleri, modelin ayırt etmekte zorlandığı saldırı türlerine ilişkin önemli ipuçları sunmakta olup, bu durum gerçek dünya uygulamaları açısından kritik sonuçlar doğurabilmektedir. Eğitim ve test veri kümelerine ait karışıklık matrisleri sırasıyla Şekil 5.2'de gösterilmektedir. Bu görsellerde yeşil tonlar, model tarafından doğru sınıflandırılan örnekleri temsil ederken; kırmızı tonlar yanlış sınıflandırmaları ifade etmektedir. Renk koyuluğunun artması, modelin ilgili gerçek ve tahmin edilen sınıflar arasında daha yüksek düzeyde karışıklık yaşadığını göstermektedir.

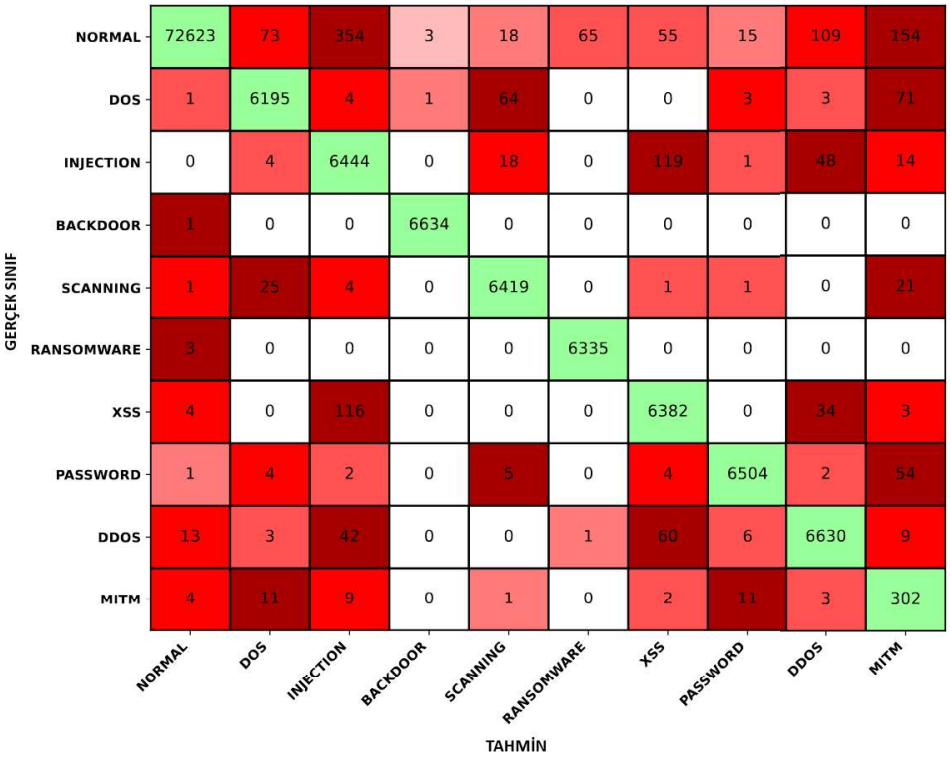
Her iki veri kümesinde de özellikle Hizmet Engelleme (DoS) ile Tarama (Scanning) saldırıları arasında ve Enjeksiyon (Injection) ile Siteler Arası Betik (XSS) saldırıları arasında belirli düzeyde karışıklık gözlemlenmiştir. Bu yanlış sınıflandırmaların temel nedeni, söz konusu saldırı türlerinin ağ davranışı açısından benzer örüntüler sergilemesi olabilir. Özellikle yüksek trafik hacmi, kısa süreli fakat yoğun istekler veya tekrar eden bağlantılar, modelin bu saldırıların birbirinden ayırt etmesini zorlaştırmaktadır.

DoS ve Scanning saldırıları arasındaki karışıklık, saldırının şiddetinin olduğundan düşük değerlendirilmesine yol açabilecek önemli bir risktir. Scanning saldırıları çoğu zaman daha büyük ölçekli saldırıların öncüsü olmakla birlikte, tek başına değerlendirildiğinde genellikle tam ölçekli bir DoS saldırısına kıyasla daha az yıkıcıdır. Buna karşın, bir DoS saldırısının Scanning olarak sınıflandırılması durumunda sistem yeterli alarm seviyesini üretmeyebilir ve bu durum saldırının daha uzun süre devam ederek ağ kesintilerine veya hizmet kalitesinde ciddi bozulmalara neden olmasına yol açabilir.

Benzer şekilde, Enjeksiyon ve XSS saldırıları arasındaki karışıklık da sistem güvenliği açısından kritik sonuçlar doğurabilir. SQL enjeksiyonu gibi enjeksiyon saldırıları çoğunlukla veritabanlarını hedef alarak ciddi veri ihlallerine neden olurken; XSS saldırıları istemci taraflı kodları hedef almakta ve kullanıcı verilerinin ele geçirilmesine veya yetkisiz erişimlere yol açabilmektedir. Bu iki saldırı türünün yanlış sınıflandırılması,



(a) Eğitim veri kümesi



(b) Test veri kümesi

Şekil 5.2. Karışıklık Matrisleri (Orijinal + Üretilmiş Öznitelikler)

sistem tarafından uygulanacak karşı önlemlerin yetersiz veya yanlış olmasına ve dolayısıyla gerçek tehdidin tam olarak bertaraf edilememesine neden olabilir.

MITM saldırı sınıfında da özellikle test veri kümesinde DoS ve DDoS saldırılarıyla sıkça karışıklık yaşandığı gözlemlenmiştir. Bu durum özellikle endişe vericidir; zira MITM saldırıları yüksek derecede sofistike olup hassas verilerin ele geçirilmesi veya değiştirilmesiyle sonuçlanabilir. Bu tür saldırıların doğru şekilde tespit edilememesi, özellikle finansal sistemler veya kritik altyapılar gibi şifreli iletişimin hayati olduğu ortamlarda uzun süreli veri sızıntılarına yol açabilir.

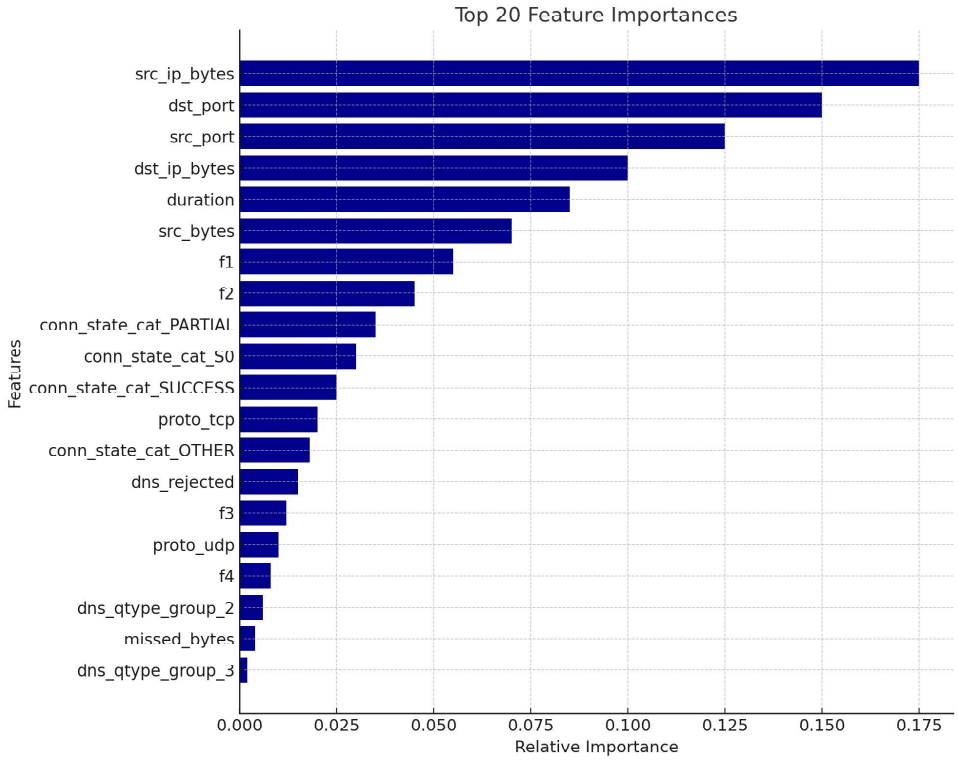
Bu yanlış sınıflandırmaların giderilmesi, sistemin güvenilirliğini artırmak açısından büyük önem taşımaktadır. Gelecek çalışmalarda, daha gelişmiş öznitelik mühendisliği yaklaşımlarının kullanılması veya bağlamsal ek bilgilerin modele dâhil edilmesi, birbirine yakın saldırı türlerinin daha etkin biçimde ayrıştırılmasına katkı sağlayabilir. Bu kapsamda, önceki bölümlerde sunulan sonuçlarla da uyumlu biçimde, Orijinal + Üretilmiş Öznitelikler kullanan modelin genel olarak daha yüksek başarım sergilemesi nedeniyle bu model üzerinde daha ayrıntılı bir analiz yapılması tercih edilmiştir.

Tüm bu zorluklara rağmen, elde edilen sonuçlar modelin saldırı türlerinin büyük çoğunluğunu yüksek doğrulukla tespit edebildiğini göstermektedir. Aynı zamanda karışıklık matrisleri, sistemin güçlü yönlerinin yanı sıra hangi saldırı türlerinde ek iyileştirmelere ihtiyaç duyulduğunu açık biçimde ortaya koymakta ve gelecekte yapılacak geliştirmeler için yol gösterici nitelik taşımaktadır.

5.2.3. Üretilmiş Özniteliklerin Model Perfomansına Etkisi

Ayrıca, geliştirilen özniteliklerin önemi permütasyon önemi (permutation importance) analizi kullanılarak değerlendirilmiştir. En önemli 20 özniteliği vurgulayan sonuçlar Şekil 5.3'te sunulmaktadır.

Bu kapsamda, $f1$, $f2$, $f3$ ve $f4$, önerilen öznitelik üretim algoritması ile elde edilen öznitelikleri ifade etmektedir. Üretilen özniteliklerin tam tanımları aşağıda verilmiştir:



Şekil 5.3. En Önemli 20 Öznitelik

$$f_1 : \text{dst_ip_bytes} > 14,0 \wedge \text{dns_qtype_group_2} \leq 0,5 \wedge$$
$$\text{dst_port} \leq 9101,5 \wedge \text{proto_udp} \leq 0,5 \wedge$$
$$\text{conn_state_cat_OTHER} \leq 0,5$$

$$f_2 : \text{dst_ip_bytes} \leq 14,0 \wedge \text{proto_udp} \leq 0,5 \wedge$$
$$\text{duration} \leq 0,266 \wedge \text{src_ip_bytes} > 42,0 \wedge$$
$$\text{src_port} > 1011,5$$

$$f_3 : \text{dst_ip_bytes} \leq 14,0 \wedge \text{proto_udp} \leq 0,5 \wedge$$
$$\text{duration} \leq 0,266 \wedge \text{src_ip_bytes} \leq 42,0 \wedge$$
$$\text{src_port} > 88,0$$

$$f_4 : \text{dst_ip_bytes} > 14,0 \wedge \text{dns_qtype_group_2} \leq 0,5 \wedge$$
$$\text{dst_port} \leq 9101,5 \wedge \text{proto_udp} > 0,5 \wedge$$
$$\text{src_ip_bytes} > 81,5$$

Öznitelik önem grafiği, makine öğrenmesi tabanlı IoT saldırı tespit sistemi içerisinde kullanılan en önemli 20 özniteliğin göreli katkılarını göstermektedir. Grafikten de görülebileceği üzere, src_ip_bytes, dst_port ve src_port gibi geleneksel ağ trafiği öznitelikleri en yüksek öneme sahiptir. Bu durum, kaynak IP adreslerinden gönderilen veri hacminin, potansiyel saldırıların tespitinde kritik bir gösterge olduğunu ortaya koymaktadır. Benzer şekilde, hedef ve kaynak portlara ilişkin öznitelikler de, farklı saldırı vektörleriyle ilişkilendirilen port kullanım kalıplarını yansıttıkları için büyük önem taşımaktadır.

En önemli 20 öznitelik arasında, önerilen otomatik öznitelik üretim algoritması ile elde edilen dört öznitelik ($f1$, $f2$, $f3$ ve $f4$) yer almaktadır. Bu durum, algoritmanın geleneksel ağ özniteliklerini tamamlayıcı nitelikte ve modele anlamlı katkılar sağlayan yeni öznitelikler üretebildiğini göstermektedir. Özellikle $f1$ ve $f2$ öznitelikleri ilk 10 içerisinde yer almakta olup, nispeten yüksek önem değerlerine sahiptir. Bu durum, söz konusu özniteliklerin ağ trafiğindeki karmaşık ve örtük örüntüleri başarılı bir şekilde temsil ettiğini göstermektedir.

Üretilen özniteliklerin içerikleri incelendiğinde, IP byte değerleri, port numaraları ve DNS sorgu türleri gibi, geleneksel metrikler tarafından da önemli olduğu tespit edilen bilgilerin bütünleşik bir şekilde kullanıldığı görülmektedir. Bu durum, üretilmiş özniteliklerin verideki kritik örüntüleri güçlendirdiğini ve modelin saldırı sınıflarını daha doğru ayırt edebilmesine katkı sağladığını göstermektedir. Örneğin, $f1$ ve $f4$ özniteliklerinde yer alan dns_qtype_group_2 $\leq$ 0,5 koşulu, belirli DNS sorgu türlerinin ağ akışı içerisinde bulunmadığını ifade etmektedir. Bu durum, normal ağ trafiği ile saldırı trafiğinin ayrıştırılmasında önemli bir ipucu sunmaktadır.

dns_qtype_group_2 grubu; NS (2), PTR (12), TXT (16), NAPTR (32), SRV (33), DS (43), DNSKEY (48) ve ANY (255) gibi sorgu türlerini içermektedir. DNS sorgu türleri, IoT tabanlı saldırı tespit sistemlerinde potansiyel tehditlerin belirlenmesinde kritik rol oynamaktadır. Örneğin, NS kayıtları ağın idari yapısını açığa çıkarabilirken, PTR kayıtları ağ içerisindeki cihazların haritalanmasına olanak tanımaktadır. TXT kayıtları sıklıkla güvenlik politikalarını içerirken, SRV ve NAPTR kayıtları servis tabanlı

saldırıların (örneğin DDoS) hedeflerini belirleyebilir. DS ve DNSKEY kayıtları ise DNSSEC mekanizmalarıyla ilişkili olup, manipülasyonları MITM saldırılarına işaret edebilir. ANY sorguları ise aşırı trafik üretimi yoluyla gerçekleştirilen DDoS saldırılarıyla ilişkilendirilmektedir.

Bu bağlamda, dns_qtype_group_2 $\leq$ 0,5 koşulu, söz konusu kritik sorgu türlerinin bulunmadığını ve bunun normal trafik davranışına işaret edebileceğini göstermektedir. Buna karşın, bu türlerin varlığı bir anomali göstergesi olarak değerlendirilebilir. Benzer şekilde, protokol türlerine ilişkin koşullar da dikkat çekicidir. $f1$, $f2$ ve $f3$ özniteliklerinde yer alan proto_udp $\leq$ 0,5 koşulu, ağ akışlarının ağırlıklı olarak TCP tabanlı olduğunu ifade etmektedir. Buna karşılık, f4 özniteliğindeki proto_udp $>$ 0.5 koşulu, UDP tabanlı saldırıların ayırt edilmesine yönelik önemli bir ayrım sunmaktadır.

Bağlantı durumlarına ilişkin bilgiler de üretilmiş özniteliklerde önemli bir yer tutmaktadır. Örneğin, $f1$ özniteliğinde yer alan conn_state_cat_OTHER $\leq$ 0,5 koşulu, bağlantının bilinmeyen veya tanımsız bir durumda olmadığını göstermektedir. Bu ayrım, potansiyel olarak zararlı bağlantıların tespit edilmesinde kritik bir rol oynamaktadır. Ayrıca, conn_state_cat_PARTIAL, conn_state_cat_S0 ve conn_state_cat_SUCCESS gibi bağlantı durumu özniteliklerinin de ilk 20 içerisinde yer alması, bağlantı durumlarının normal ve zararlı trafik davranışlarının ayrıştırılmasında ne kadar önemli olduğunu ortaya koymaktadır.

Sonuç olarak, geleneksel ağ öznitelikleri ile algoritmik olarak üretilmiş yeni özniteliklerin birlikte kullanılması, IoT ortamları için çok boyutlu ve güçlü bir saldırı tespit yaklaşımı sunmaktadır. f-önekli üretilmiş öznitelikler tekil olarak her zaman en üst sıralarda yer almasa da, birlikte ele alındıklarında modelin tahmin gücünü önemli ölçüde artırmaktadır. Bu kapsamlı öznitelik mühendisliği yaklaşımı, önerilen IDS modelinin genel başarımını ve dayanıklılığını anlamlı biçimde güçlendirmektedir.

5.2.4. Literatürdeki Diğer Yöntemlerle Karşılaştırma

Literatürdeki diğer araçlarla karşılaştırma yaparak, Bölüm 4.9.2.'de belirtilen modellerle benzer kıyaslamalar gerçekleştirilmiştir. Burada öncelikli olarak, diğer araçlar tekrardan

hatırlatılacak, sonrasında ise karşılaştırmalı analizler gerçekleştirilecektir.

İlk model olan EvolCostDeep (Telikani ve diğ., 2022), Endüstriyel Nesnelerin İnterneti (IIoT) ortamlarında saldırı tespiti amacıyla geliştirilmiş özel bir derin öğrenme (DL) çerçevesidir. Bu model, IIoT sistemlerinde nadir görülen (düşük frekanslı) saldırı türlerini daha etkin biçimde tespit edebilmek için evrimsel optimizasyon tabanlı bir yaklaşım kullanmaktadır. Model; ön işleme adımlarını, sezgisel başlatma stratejilerini ve JADE algoritması ile optimize edilen bir maliyet matrisi yapısını bir araya getirmektedir. Eğitim sürecinde yığılmış oto-kodlayıcılar (stacked autoencoders) ve evrişimli sinir ağları (CNN) birlikte kullanılmakta ve sınıf dengesizliğini gözeten maliyet-duyarlı bir kayıp fonksiyonu ile öğrenme gerçekleştirilmektedir. Ayrıca EvolCostDeep, IIoT saldırı tespit problemlerinde sıklıkla karşılaşılan veri dengesizliği sorununu doğrudan ele almakta ve ölçeklenebilirlik ile algılama verimliliğini artırmak amacıyla sis bilişim (fog computing) ortamında çalışmaya uygun bir yapı sunmaktadır. Modelin etkinliği, UNSW-NB15 ve ToN-IoT gibi kıyaslama veri setleri üzerinde gerçekleştirilen deneylerle doğrulanmış ve diğer yöntemlere kıyasla daha başarılı sonuçlar elde edildiği gösterilmiştir.

İkinci model olan CMTSNN (Zhu ve diğ., 2023), IoT ağlarında anormal ve şifrelenmiş trafiğin tespiti için önerilmiş çok sınıflı bir sınıflandırma yaklaşımıdır. CMTSNN modeli, öznitelik çıkarımı aşamasında BiLSTM ve 1D-CNN tabanlı hibrit bir mimari kullanmaktadır. Veri dengesizliği problemini azaltmak amacıyla maliyet ceza matrisi entegre edilmiş ve sınıflandırma aşamasında geliştirilmiş bir çapraz-entropy kayıp fonksiyonu tercih edilmiştir. Deneysel sonuçlar, CMTSNN modelinin doğruluk, kesinlik, duyarlılık ve F1 skoru metrikleri açısından diğer derin öğrenme tabanlı modellere kıyasla daha üstün performans sergilediğini göstermektedir. Ayrıca, özellikle şifreli trafik içerisinde yer alan azınlık sınıf örneklerinin tespitinde etkili sonuçlar elde edilmesi, modelin gerçek dünya uygulamaları açısından potansiyelini ortaya koymaktadır.

Son olarak LSTM-SPIP modeli (Keshk ve diğ., 2023), saldırı tespitinde açıklanabilirliği ön planda tutan derin öğrenme tabanlı bir çerçevedir. Model, saldırı tespiti için Uzun Kısa Süreli Bellek (LSTM) ağlarını kullanmakta ve karar verme sürecinin anlaşılabilirliğini artırmak amacıyla kendi kendini açıklayabilen (self-explainable) bir mekanizma ile

desteklenmektedir. Yerel ve küresel açıklama yöntemlerini birlikte ele alan bu yaklaşım, saldırı tespit sistemlerinde yorumlanabilirlik problemine önemli bir çözüm sunmaktadır. Model kapsamında SHAP ve Permütasyon Öznitelik Önemi (Permutation Feature Importance – PFI) gibi açıklanabilir yapay zekâ (XAI) teknikleri kullanılarak, belirli saldırı türlerinin tespitinde etkili olan kritik öznitelikler belirlenmektedir.

Bu çalışmada gerçekleştirilen karşılaştırmalarda kesinlik, duyarlılık ve F1 skoru temel değerlendirme metrikleri olarak ele alınmıştır. Ancak CMTSNN modelinin yalnızca F1 skoru değerlerini raporlaması, tüm metrikler açısından kapsamlı bir karşılaştırma yapılmasını sınırlandırmaktadır. Buna ek olarak, LSTM-SPIP modeli normal trafik, DoS, enjeksiyon, fidye yazılımı ve MITM saldırı türleri için sonuç sunmamaktadır. Karşılaştırmaya ilişkin ayrıntılı sonuçlar Tablo 5.3 içerisinde sunulmaktadır.

Modellerin karşılaştırılmasında kesinlik, duyarlılık ve F1 skoru metrikleri esas alınmıştır. Ancak CMTSNN modelinin yalnızca F1 skoru değerlerini raporlaması, tüm metrikler açısından kapsamlı bir karşılaştırma yapılmasını sınırlandırmaktadır. Buna ek olarak, LSTM-SPIP modeli normal, DoS, enjeksiyon, fidye yazılımı ve MITM olmak üzere beş saldırı türü için sonuç sunmamaktadır. Bu karşılaştırmaya ait ayrıntılı sonuçlar Tablo 5.3'te verilmiştir.

Normal Trafik Tespiti: Önerilen model, 0,999 kesinlik değeri ile karşılaştırılan tüm modeller arasında en yüksek sonucu elde etmiştir. Bu bulgu, modelin normal (zararsız) ağ trafiğini doğru şekilde tanımlama ve yanlış alarm üretimini en aza indirme konusundaki başarısını açıkça göstermektedir. 0,988 duyarlılık ve 0,994 F1 skoru değerleri de bu sonucu desteklemekte olup, önerilen modelin gerçek dünya uygulamalarında ağ kararlılığını koruma ve gereksiz uyarıları azaltma açısından oldukça uygun olduğunu ortaya koymaktadır.

DoS Saldırılarının Tespiti: DoS saldırılarında önerilen model, 0,981 kesinlik ve 0,977 duyarlılık değerleri ile 0,978 F1 skoru elde etmiştir. EvolCostDeep modeli de 0,981 kesinlik değerine ulaşmış olsa da, daha düşük duyarlılık ve F1 skoru değerleri nedeniyle önerilen modelin daha dengeli bir performans sunduğu görülmektedir. Bu denge, özellikle yüksek öneme sahip saldırıların gözden kaçırılmaması açısından kritik öneme sahiptir.

Tablo 5.3. Diğer Yöntemler ile Karşılaştırmalı Sonuçlar

Etiket	**Önerilen Model**			**EvolCostDeep**			**CMTSNN**	**LSTM-SPIP**		
	Kesinlik	**Duyarlılık**	**F1 Skoru**	**Kesinlik**	**Duyarlılık**	**F1 Skoru**	**F1 Skoru**	**Kesinlik**	**Duyarlılık**	**F1 Skoru**
Normal (Normal)	**0,999**	0,988	0,994	0,988	**0,990**	0,989	**0,997**	Veri Yok (No Data)		
Hizmet Reddi (DoS)	**0,981**	0,977	0,978	**0,981**	0,980	0,980	**0,984**	Veri Yok (No Data)		
Enjeksiyon (Injection)	0,924	**0,969**	0,946	**0,998**	0,960	0,979	**0,987**	Veri Yok (No Data)		
Arka Kapı (Backdoor)	**0,999**	**0,999**	**0,999**	0,994	0,989	0,991	0,976	0,870	0,901	0,885
Tarama (Scanning)	**0,984**	0,992	**0,988**	0,860	**0,993**	0,921	0,988	0,796	0,894	0,842
Fidye Yazılımı (Ransomware)	0,989	**0,999**	**0,995**	**1,000**	0,957	0,978	0,878	Veri Yok (No Data)		
Siteler Arası Betik (XSS)	**0,964**	0,976	0,970	0,938	**0,988**	0,962	**0,996**	0,629	0,736	0,678
Parola Saldırısı (Password)	0,994	**0,989**	0,992	0,998	0,971	0,984	**0,993**	Veri Yok (No Data)		
Dağıtık Hizmet Reddi (DDoS)	0,970	**0,979**	0,975	**0,976**	0,974	0,974	**0,993**	0,822	0,902	0,859
Ortadaki Adam (MITM)	0,481	**0,880**	0,620	1,000	0,802	**0,890**	**0,939**	Veri Yok (No Data)		

Enjeksiyon Saldırılarının Tespiti: Enjeksiyon saldırılarında dikkat çekici bir karşıtlık gözlemlenmektedir. EvolCostDeep modeli 0,998 ile en yüksek kesinlik değerine ulaşırken, önerilen model 0,969 duyarlılık değeri ile daha fazla enjeksiyon saldırısını yakalamayı başarmış ve 0,946 F1 skoru elde etmiştir. Bu durum, önerilen modelin daha geniş bir enjeksiyon saldırı kümesini tespit edebildiğini, ancak bunun nispeten daha yüksek yanlış pozitif oranı pahasına gerçekleştiğini göstermektedir. Gelecek çalışmalarda maliyet-duyarlı öğrenme yaklaşımlarının eklenmesiyle bu dengenin daha da iyileştirilebileceği öngörülmektedir.

Arka Kapı (Backdoor) Saldırılarının Tespiti: Önerilen model, arka kapı saldırılarının tespitinde kesinlik, duyarlılık ve F1 skoru metriklerinin tamamında 0,999 değerine ulaşarak tüm karşılaştırılan yöntemleri geride bırakmıştır. Bu olağanüstü sonuç, özellikle kullanılan öznitelik mühendisliği stratejilerinin arka kapı saldırılarına özgü ince ve gizli örüntüleri başarıyla yakalayabildiğini göstermektedir.

Tarama (Scanning) Saldırılarının Tespiti: Tarama saldırıları için önerilen model 0,984 kesinlik, 0,992 duyarlılık ve 0,988 F1 skoru elde etmiştir. Bu değerler, literatürdeki modellerle karşılaştırıldığında rekabetçi hatta üstün bir performansa işaret etmektedir. Elde edilen sonuçlar, modelin yüksek frekanslı saldırı türlerini etkin biçimde ele alabildiğini göstermektedir.

Fidye Yazılımı (Ransomware) Tespiti: Fidye yazılımı saldırılarında EvolCostDeep modeli 1,000 kesinlik değeri ile en yüksek sonucu elde ederken, önerilen model 0,989 kesinlik, 0,999 duyarlılık ve 0,995 F1 skoru ile çok yakın bir performans sergilemiştir. Özellikle neredeyse mükemmel duyarlılık değeri, önerilen modelin fidye yazılımı saldırılarını kaçırma riskinin son derece düşük olduğunu göstermektedir.

XSS Saldırılarının Tespiti: XSS saldırılarında önerilen model 0,964 kesinlik, 0,976 duyarlılık ve 0,970 F1 skoru elde etmiştir. CMTSNN modeli 0,996 ile daha yüksek bir F1 skoruna ulaşmış olsa da, önerilen modelin sunduğu dengeli kesinlik–duyarlılık profili, pratik uygulamalar açısından güvenilir bir seçenek olduğunu ortaya koymaktadır.

Parola Saldırılarının Tespiti: Parola saldırılarında önerilen model 0,994 kesinlik, 0,989

duyarlılık ve 0,992 F1 skoru değerleri ile literatürdeki yöntemlerle karşılaştırılabilir bir performans sunmuştur. Yüksek kesinlik değeri, meşru parola denemelerinin saldırı olarak işaretlenme riskini azaltarak kullanıcı güvenini korumaktadır.

DDoS Saldırılarının Tespiti: DDoS saldırılarında önerilen model 0,970 kesinlik, 0,979 duyarlılık ve 0,975 F1 skoru elde etmiştir. EvolCostDeep modelinin kesinlik değeri 0,976 ile biraz daha yüksek olsa da, aradaki fark ihmal edilebilir düzeydedir. Bu sonuçlar, önerilen modelin büyük ölçekli DDoS saldırılarını tespit etmede yeterince sağlam olduğunu göstermektedir.

MITM Saldırılarının Tespiti: MITM saldırıları, önerilen model açısından en zorlu sınıfı oluşturmaktadır. Model bu saldırı türünde 0,481 kesinlik, 0,880 duyarlılık ve 0,620 F1 skoru elde etmiştir. Buna karşılık EvolCostDeep modeli daha yüksek kesinlik ve F1 skoru değerleri sunmaktadır. Bu fark, MITM saldırılarının nadir ve karmaşık yapısından kaynaklanmakta olup, bu sınıf için topluluk öğrenmesi (ensemble learning) ya da sentetik veri üretimi gibi gelişmiş yöntemlerin entegrasyonunun faydalı olabileceğini göstermektedir.

Tüm saldırı türleri birlikte değerlendirildiğinde, önerilen modelin çoğu sınıfta kesinlik ve duyarlılık açısından güncel derin öğrenme tabanlı yöntemlerle rekabet edebilir hatta bazı durumlarda daha iyi performans sunduğu görülmektedir. Modelin öne çıkan yönleri; sadeliği, hesaplama verimliliği ve farklı saldırı türleri arasında güçlü genelleme kabiliyetidir. MITM saldırılarındaki görece zayıf performans ise, gelecekte hibrit yaklaşımlarla (klasik ML + DL) daha da iyileştirilebilecek bir alan olarak öne çıkmaktadır.

Sonuç olarak, önerilen model IoT tabanlı saldırı tespit sistemleri için derin öğrenme modellerine güçlü, verimli ve yorumlanabilir bir alternatif sunmaktadır. Titizlikle tasarlanan ön işleme ve öznitelik mühendisliği adımları sayesinde yüksek başarı elde edilirken, modelin hesaplama maliyeti düşük tutulmuştur. Bu özellikler, özellikle kaynak kısıtlı IoT ortamlarında modelin uygulanabilirliğini ve güvenilirliğini artırmaktadır.

5.3. Hesaplama Karmaşıklığı

Bu çalışmada sunulan tüm hesaplamalı adımlar; 16 GB 3200 MHz RAM, AMD Ryzen 5 1600 işlemci, 1 TB HDD, 256 GB SSD ve GTX 1060 6 GB ekran kartına sahip bir bilgisayar üzerinde gerçekleştirilmiştir. Tüm analizler, veri ön işleme, öznitelik üretimi ve model eğitim süreçleri Python programlama dili kullanılarak yürütülmüş; NumPy Walt ve diğ., 2011, Pandas (McKinney, 2011), Scikit-Learn (Kramer, 2016), Skopt (Markov, 2017) kullanılmıştır. Bölüm 4'te tanımlanan ön işleme adımları, hesaplama karmaşıklığı açısından ayrı ayrı incelenmiş ve aşağıda özetlenmiştir:

- Eksik Değerlerin Filtrelenmesi: Süreç, pandas veri çerçevesi üzerinde isna() fonksiyonu kullanılarak eksik değerlerin tespiti ile başlamakta ve her bir hücre için Boole türünde bir çıktı üretmektedir. Bu adımın hesaplama karmaşıklığı, n satır ve m sütun olmak üzere $O(n \times m)$'dir. Daha sonra, eksik değer oranı %95'in üzerinde olan sütunlar belirlenmekte ve kaldırılmaktadır. Bu işlemin karmaşıklığı $O(m)$ olup, toplam karmaşıklık yine $O(n \times m)$ seviyesindedir.
- Sabit ve Yarı-Sabit Özniteliklerin Elenmesi: Her bir sütun, içerdiği benzersiz değerler açısından değerlendirilmekte olup, bu işlem her sütun için tüm satırların kontrol edilmesini gerektirmektedir. m adet sütun ve n adet gözlem için bu adımın hesaplama karmaşıklığı $O(m \times n)$'dir.
- Korelasyonlu Öznitelik Gruplarının Belirlenmesi ve Elenmesi: Algoritma 2'de tanımlandığı üzere bu süreç; korelasyon matrisinin hesaplanması, korelasyon gruplarının oluşturulması ve her grup içerisinden en anlamlı özniteliğin seçilmesi adımlarını içermektedir. Bu adımların karmaşıklıkları sırasıyla $O(m^2)$, $O(m \times n)$ ve $O(g \times k \times n_{\text{train}})$'dir. Burada g grup sayısını, k her bir gruptaki öznitelik sayısını ve n_{train} eğitim örneklerinin sayısını ifade etmektedir.

Bu ön işleme adımlarının tamamı, deneysel olarak ortalama 1–1,5 dakika içerisinde tamamlanmıştır. Hesaplama maliyeti açısından dikkate alınması gereken bir diğer önemli bileşen ise Algoritma 5'te tanımlanan ve EBLR yaklaşımı temel alınarak geliştirilen öznitelik üretim algoritmasıdır.

- Algoritmanın temel hesaplama yükü, 11. adımda yer alan RF eğitimine

dayanmaktadır. Bu adımın karmaşıklığı $O(t \times m \times n \log n)$ olarak ifade edilebilir; burada t ağaç sayısını temsil etmektedir.

- 13. adımda gerçekleştirilen log-kayıp hesaplamasının karmaşıklığı $O(n)$'dir.
- 18. adımda, her bir sınıf c_i için karar ağacı eğitimi gerçekleştirilmektedir. Bu işlemin karmaşıklığı $O(m \times n_i \log n_i)$ olup, tüm sınıflar için tekrarlandığında toplam karmaşıklık $\sum_{i=1}^{C} O(m \times n_i \log n_i)$ şeklindedir. Tüm veri kümesi dikkate alındığında bu ifade yaklaşık olarak $O(m \times n \log n)$ olarak değerlendirilmektedir.
- 22. ve 25. adımlarda yeni özniteliklerin eklenmesiyle birlikte hesaplama yükü her iterasyonda artmakta olup, bu adımların karmaşıklığı genel olarak $O(n)$ seviyesindedir.

Bu bölümde tanımlanan öznitelik üretim işlemleri, kullanılan donanım üzerinde ortalama 10–12 dakika içerisinde tamamlanmıştır.

Son olarak, en yüksek hesaplama maliyeti model eğitim sürecinde ortaya çıkmaktadır. Bu aşama, yeniden örnekleme (resampling) stratejisi ile birlikte kullanılan çapraz doğrulama (cross-validation) şemasını içermektedir. Her bir çapraz doğrulama iterasyonunda, dokuz kat (fold) üzerinde yeniden örnekleme uygulanmakta ve ardından belirli bir hiperparametre uzayı için model eğitilmektedir. On katlı çapraz doğrulama senaryosunda, ilk olarak SCUT algoritması çalıştırılmakta, ardından ilgili parametre kombinasyonu için eğitim gerçekleştirilmektedir.

SCUT algoritması kapsamında, aşağı örnekleme işlemleri Gauss Karışım Modeli (GMM) tabanlı kümeleme yöntemiyle (Dempster ve diğ., 1977); yukarı örnekleme işlemleri ise SMOTE algoritması (Chawla ve diğ., 2002) kullanılarak gerçekleştirilmektedir. Ancak GMM tabanlı yaklaşımın yüksek hesaplama maliyeti nedeniyle, deneysel analizlerde GMM ile Rastgele Aşağı Örnekleme (RUS) arasında anlamlı bir fark gözlemlenmemiştir. Bu nedenle SCUT uygulamasında GMM yerine RUS tercih edilmiştir. Bu değişiklik sayesinde yeniden örnekleme süreci birkaç saniye içerisinde tamamlanabilmiştir.

Eğitim sürecindeki asıl hesaplama yükü, Rastgele Orman algoritmasından kaynaklanmaktadır. Ağaç sayısı T, her düğümde değerlendirilen rastgele öznitelik sayısı k (burada $\sqrt{d}$ seçilmiştir), toplam örnek sayısı n ve öznitelik sayısı d olmak üzere,

algoritmanın karmaşıklığı $O(T \times k \times n \times \log n \times d)$ şeklinde ifade edilebilir.

Bu çalışmada, on katlı çapraz doğrulama altında tek bir ağaç için eğitim süresi yaklaşık 35 saniye ile 1,5 dakika arasında değişmektedir. Tablo 4.14'te gösterildiği üzere toplamda 324 farklı model-parametre kombinasyonu eğitilmiştir. Bu nedenle, tüm model kombinasyonlarının eğitimi yaklaşık 3,5 saat sürmüştür.

Sonuç olarak, 324 modelden oluşan bir ızgara arama (grid search) süreci yaklaşık 4–5,5 saatlik bir zaman dilimi içerisinde tamamlanabilmekte ve bu sayede en iyi performansı gösteren model etkin bir şekilde belirlenebilmektedir.

5.4. Sonuçlar

Bu bölümde, özellikle RF başta olmak üzere klasik makine öğrenmesi modellerinin, etkili bir IoT-IDS geliştirme potansiyeli incelenmiştir. Eksik veri analizi ve uygun doldurma (imputation) yöntemleri, doğru öznitelik kodlama stratejileri ve çoklu doğrusal bağlantının azaltılmasını içeren dikkatle tasarlanmış ön işleme adımları sayesinde, klasik ML modellerinin güncel DL yaklaşımları ile rekabet edebilecek düzeyde performans sergileyebildiği gösterilmiştir. Ayrıca, yinelemeli ağaç tabanlı artık (residual) analizi üzerine kurulu ve EBLR yöntemini genişleten özgün bir öznitelik üretim algoritması geliştirilmiş; bu yaklaşımın, IoT veri setlerinde karmaşık saldırı örüntülerinin yakalanmasında modelin ayırt edici gücünü artırdığı ortaya konmuştur.

Bu bölümde elde edilen temel bulgulardan biri, EvolCostDeep ve CMTSNN gibi derin öğrenme modellerinin geleneksel olarak kesinlik ve duyarlılık açısından klasik yöntemlerden daha üstün kabul edilmesine rağmen, veri odaklı ön işleme ve gelişmiş öznitelik mühendisliği ile desteklenen önerilen modelin bu ileri düzey yaklaşımlarla rekabet edebilen sonuçlar üretebildiğidir. Önerilen model; normal ağ trafiği, DoS ve fidye yazılımı (ransomware) saldırıları gibi birçok saldırı türünde yüksek kesinlik ve duyarlılık değerleri elde etmiş, performans farkları çoğunlukla %1–5 aralığında kalmıştır. Veri setinde yeterince temsil edilmeyen MITM gibi saldırı türlerinde, derin öğrenme modellerine kıyasla daha düşük kesinlik değerleri gözlemlenmiş olsa da, modelin diğer saldırı türlerindeki başarımı, klasik ML yaklaşımlarının IoT ortamlarına özgü biçimde

optimize edildiğinde halen güçlü bir alternatif sunduğunu göstermektedir.

Sonuç olarak bu bölümde, iyi tasarlanmış ön işleme akışları ve öznitelik üretim teknikleri ile desteklenen klasik makine öğrenmesi modellerinin, IoT saldırı tespiti problemlerinde açıklanabilir, hesaplama açısından verimli ve yüksek performanslı bir alternatif sunabildiği ortaya konmuştur. Bulgular, özellikle kaynak kısıtlı IoT ortamlarında, daha karmaşık derin öğrenme mimarileri yerine klasik modellerin göz ardı edilmemesi gerektiğine işaret etmektedir.

6. IOT VERİ SETLERİNDE DENGESİZ DAĞILIM PROBLEMİ

Makine öğrenmesinde dengesiz sınıf dağılım problemi, gerçek dünya problemlerinde sıklıkla karşılaşılan ve bir makine öğrenmesi problemi için başarımı olumsuz etkileyebilecek bir fenomen olarak karşımıza çıkmaktadır. Finansal sahtekarlık tespiti, hastalık tanıma ve siber güvenlik alanı gibi çok farklı senaryolarda sıkça karşılaşılan bir zorluktur (Abdulhammed ve diğ., 2018; Khalilia ve diğ., 2011; Wei ve diğ., 2013). Dengesiz dağılım problemi, bir sınıflandırma probleminde, model eğitiminde kullanılacak veri setindeki farklı sınıfların oransal dağılımındaki uç farklılıklar olarak tanımlanabilir. Bu tür bir doğaya sahip bir veri seti ile çalışıldığında, eğitilecek model, özellikle çoğunluk sınıfın baskınlığı sebebiyle, orantısız bir şekilde bu baskın sınıfı öğrenmeye çalışarak, azınlık sınıfının gözardı edilmesine yol açabilir. Bu durum ise, çoğunluk sınıfında aşırı uyum sağlama (overfitting), azınlık sınıflarda ise düşük uyum sağlama (underfitting) durumlarına yol açarak, doğruluk (accuracy) gibi yanıltıcı değerlendirme metrikleri de dahil olacak şekilde, pek çok probleme yol açabilir (Manikandan ve diğ., 2024).

Sınıf dengesizliği problemini çözebilmek için, 3 farklı yaklaşımdan birisi genel olarak tercih edilmektedir. Bu yöntemlerden ilki, aşağı örnekleme (undersampling) olarak adlandırılan yöntem ile, çoğunluk sınıfındaki (majority class) örneklerin daha küçük bir alt kümesini seçerek, bu sınıfa ait yüksek oransal baskınlığın daha ılımlı seviyelere düşürülmesi işlemidir. Bu yaklaşım ile çoğnuluk sınıfına ait örneklem sayısı azalacağından, bir model eğitimi için kullanılacak olan çoğunluk sınıfı daha az örnekle temsil edilecektir. Bu ise çoğunluk sınıfının baskınlığını azaltarak, ilgili modelin diğer azınlık sınıf (minority class) örüntülerini baskılamasını engeller, ve modelin daha iyi bir genelleştirme yapabilmesine olanak tanır (Andriani ve diğ., 2024).

Çoğunluk sınıfın örneklem sayısının azaltılmasına karşın, azınlık sınıfın örneklem sayısının arttırılması yoluna da gidilebilir. Yukarı örnekleme (oversampling) olarak adlandırılan bu yöntem, azınlık sınıf örneklerinin yapay olarak arttırılması olarak ifade edilebilir. Bu yöntem ile, azınlık sınıfının örnekleri çeşitli şekillerde arttırılarak, girdi olarak verilecek modele daha fazla sayıda örnek verilmesi, yanlılığın azaltılarak modelin genelleştirilebilirliğinin arttırılması hedeflenir (Elmobark ve Elhishi, 2025).

Üçüncü ve son olarak da, yukarıda bahsi geçen yukarı ve aşağı örnekleme metotları birlikte kullanılarak, her iki dengesizlik problemi de giderilmeye çalışılır.

Literatürde, çok çeşitli aşağı örnekleme yaklaşımları önerilmiştir. Bunlardan en temel olanı Rastgele Aşağı Örnekleme (Random Undersampling), çoğunluk sınıfından örnekleri rastgele bir şekilde seçerek sayısal denge sağlamayı amaçlamaktadır. Ancak bu yöntem, örnek seçiminde dağılımsal özellikleri göz ardı ettiğinden, sınıfın anlamlı örüntülerini dışlayabilir ve bu da modelin genelleme yeteneğini olumsuz etkileyebilir (Batista ve diğ., 2004).

Bu sınırlamaları aşmak amacıyla geliştirilen NearMiss algoritmaları, karar sınırlarını daha keskin hale getirmek ve anlamsız verileri ortadan kaldırmak için çoğunluk sınıfındaki örneklerin seçimini çeşitli kurallara göre yapmaktadır. NearMiss-1, azınlık sınıfına en yakın çoğunluk örneklerini seçerken; NearMiss-2, azınlık örneklerinden daha uzakta bulunanları dikkate alır. NearMiss-3 ise her azınlık sınıfı örneğinin çevresinden sabit sayıda çoğunluk örneği seçerek yerel veri yapısını korumayı hedefler (Mani ve Zhang, 2003).

Tomek Link yöntemi, sınıf sınırlarında gürültü olabileceğini varsayarak, farklı sınıflardan birbirine en yakın örnek çiftlerini belirler. Bu çiftlerde yer alan çoğunluk sınıfı örnekleri veri kümesinden çıkarılarak karar sınırının daha net hale gelmesi sağlanır (Altalhan ve diğ., 2025).

Yoğunlaştırılmış En Yakın Komşu (Condensed Nearest Neighbour – cNN) tekniği, küçük bir çekirdek örnek kümesi ile başlayarak, yanlış sınıflandırılan verileri k-En Yakın Komşu algoritmasıyla aşamalı şekilde dahil etmeyi amaçlar (Garcia, 2012). Benzer şekilde, Düzenlenmiş En Yakın Komşu (Edited Nearest Neighbor – ENN), veri noktalarının çevresindeki komşularıyla uyumluluğunu kontrol ederek, sınır bölgelerdeki belirsiz ya da gürültülü örnekleri temizlemektedir (Guzmán-Ponce ve diğ., 2020).

Tek Taraflı Seçim (One-Sided Selection – OSS) yöntemi, Tomek Link ve cNN prensiplerini birleştirerek, sınıflandırmaya katkısı sınırlı ya da zararlı olabilecek çoğunluk sınıfı örneklerini elimine etmeye çalışmaktadır (Kubat, 1997).

Bu yöntemlerin ortak amacı karar sınırlarını iyileştirmek ve gürültüyü azaltmak olsa da, çoğunluk sınıfının genel yapısal özelliklerini göz ardı edebilmeleri, modelin bazı önemli örüntüleri öğrenememesine neden olabilir. Bu da, yetersiz genelleme veya aşırı öğrenme riskini beraberinde getirebilir. Bu nedenle bu tekniklerin uygulanması, dikkatli analiz ve değerlendirme gerektirir. Buna istinaden, bu bölümde, özellikle sınıf dengesizliği problemlerinin oldukça sık olduğu IoT saldırı verisetlerinde de kullanılabilecek, ve aşağı örneklenecek sınıfın dağılımsal özelliklerinin korunmasını maksimize etmeyi amaçlayan, yinelemeli bir Kullback-Leibler (KL) Ayrışımı (Kullback-Leibler Divergence) tabanlı bir aşağı örnekleme algoritması tasarlanmıştır (Kullback ve Leibler, 1951). KL-Ayrışımı, iki olasılık dağılımı arasındaki farkı ölçmektedir (Sugiyama ve diğ., 2013). Bu ayrışım ölçüm tenkiği, alt örnekleme yapılmış bir sınıf örneklem kümesinin, orjinal dağılıma ne kadar benzediğinin bir ölçütü olarak kullanılmıştır. KLSampler olarak da adlandırdığımız bu yöntem, öncelikle çoğunluk sınıfı üzerinde K-Ortalamalar (K-Means) algoritmasından faydalanarak (McQueen, 1967), çoğunluk sınıfın kümeler ve bu kümelerden örneklemler seçerek, bir başlangıç alt kümesi oluştur. Bu aşamayı takiben, orjinal veri dağılımından özyinelemeli bir biçimde örnekler seçerek başlangıç alt kümesini genişletir. Bu işlemde ise bir durma kriteri (stopping criteria) olarak da KL-Ayrışımını kullanarak, alt örneklem kümesinin istatistiksel olarak orjinal veri kümesiyle uyumlu olmasını sağlar. Bu sayede, IoT saldırı veri setleri başta olmak üzere, çok çeşitli veri setlerinde uygulanabilecek bir aşağı örnekleme algoritması elde edilmiş olunur.

6.1. Ağ Saldırı Veri Setlerinin İncelemesi

Bu bölümde, literatürde, hem model eğitmi hem de model sonuçlarının karşılaştırılması (bencmark) için sıklıkla kullanılan BoT-IoT, ToN-IoT, ve UNSW-NB15 ağ saldırısı veri setleri incelenerek, aşağı örneklemenin öneminin anlatılması amaçlanmaktadır. Bu veri setlerine ait detaylı açıklamalar, Bölüm 4.1.'de açıklanmıştır. İlgili veri setlerine ait, farklı saldırı türlerinin dağılımı, Tablo 6.1'de gösterilmiştir.

Tablo 6.1'de sunulan Bot-IoT, ToN-IoT ve UNSW-NB15 veri setleri incelendiğinde, saldırı türlerinin dağılımlarının hem veri setleri arasında hem de veri setlerinin kendi içlerinde oldukça heterojen olduğu açıkça görülmektedir. Bu durum, makine öğrenmesi

Tablo 6.1. Veri Setlerinde Saldırı Türlerinin Dağılımı

Saldırı Tipi	Bot-IoT	ToN-IoT	UNSW-NB15
Normal (Benign)	135037 (0.36%)	6099469 (36.0%)	2295222 (96.02%)
Hizmet Engelleme (DoS)	16673183 (44.14%)	712609 (4.21%)	5794 (0.24%)
Dağıtık Hizmet Engelleme (DDoS)	18331847 (48.56%)	2026234 (11.96%)	-
Keşif / Bilgi Toplama (Reconnaissance)	2620999 (6.94%)	-	12779 (0.53%)
Arka Kapı (Backdoor)	-	16809 (0.10%)	2169 (0.09%)
Kod Enjeksiyonu (Injection)	-	684465 (4.04%)	-
Tarama (Scanning)	-	3781419 (22.32%)	-
Parola Saldırıları (Password)	-	1153323 (6.81%)	-
Fidye Yazılımı (Ransomware)	-	3425 (0.02%)	-
Ortadaki Adam (MITM)	-	7723 (0.05%)	-
Siteler Arası Betik (XSS)	-	2455020 (14.49%)	-
Açık İstismarı (Exploits)	-	-	31551 (1.32%)
Bulanıklaştırma Saldırıları (Fuzzers)	-	-	22310 (0.93%)
Genel Amaçlı Saldırılar (Generic)	-	-	16560 (0.69%)
Analiz Tabanlı Saldırılar (Analysis)	-	-	2299 (0.10%)
Kabuk Kodu (Shellcode)	-	-	1427 (0.06%)
Solucanlar (Worms)	-	-	164 (0.01%)
Hırsızlık (Theft)	2431 (0.01%)	-	-
Toplam	**37763497**	**16940496**	**2390275**

tabanlı modellerin güvenilir ve genellenebilir bir şekilde eğitilmesini zorlaştırmaktadır.

Bot-IoT veri seti ele alındığında, örneklerin yaklaşık %93'ünden fazlasının yalnızca DoS ve DDoS saldırılarından oluştuğu görülmektedir. Buna karşın, Normal trafik yalnızca %0.36 oranında temsil edilmektedir. Bu derece dengesiz bir dağılımda eğitilen bir model, çoğunluk sınıflarını ayırt etmede yüksek doğruluk elde edebilse

bile, azınlık sınıflarını —özellikle benign trafiği— doğru şekilde tanımlamakta ciddi başarısızlıklar gösterebilir. Bu durum, pratikte yanlış alarm oranlarının artmasına ve sistemin operasyonel kullanılabilirliğinin azalmasına yol açmaktadır.

ToN-IoT veri seti, Bot-IoT'a kıyasla daha çeşitli saldırı türleri içermesine rağmen, yine belirgin bir dengesizlik sergilemektedir. Özellikle Benign (%36.0), Scanning (%22.32) ve XSS (%14.49) saldırıları veri setinin büyük bir bölümünü oluştururken, Ransomware (%0.02), MITM (%0.05) ve Backdoor (%0.10) gibi kritik saldırı türleri oldukça sınırlı sayıda örnekle temsil edilmektedir. Bu tür nadir saldırılar genellikle yüksek etkiye sahip olmalarına rağmen, eğitim sürecinde baskın sınıflar tarafından gölgelenmekte ve modelin karar sınırlarında yeterince temsil edilememektedir.

UNSW-NB15 veri seti ise farklı bir problemli yapı sergilemektedir. Veri setinin yaklaşık %96'sı Benign trafikten oluşmakta, saldırı sınıflarının tamamı ise oldukça düşük oranlarla temsil edilmektedir. Bu tür aşırı dengesiz bir senaryoda, doğruluk (accuracy) gibi klasik performans metrikleri yanıltıcı hale gelmekte; model, saldırıları neredeyse tamamen göz ardı etmesine rağmen yüksek doğruluk değerleri raporlayabilmektedir. Bu durum, özellikle güvenlik odaklı sistemlerde kabul edilemez sonuçlar doğurmaktadır.

Makine öğrenmesi modelleri, doğaları gereği veri dağılımına duyarlı (data-driven) yapılardır ve eğitim sürecinde kayıp fonksiyonlarını minimize ederken çoğunluk sınıflarına daha fazla ağırlık verme eğilimindedir. Sınıf dengesizliği bulunan veri setlerinde bu durum, karar sınırlarının çoğunluk sınıflar lehine kaymasına, azınlık sınıfların ise hatalı veya belirsiz bölgelerde temsil edilmesine neden olmaktadır. Özellikle ağaç tabanlı yöntemler, boosting algoritmaları ve derin öğrenme modelleri, yeterli dengeleme yapılmadığı sürece nadir saldırı türlerini öğrenmekte zorlanmaktadır.

Bu nedenlerle, Tablo 6.1'de özetlenen veri setleri üzerinde doğrudan model eğitimi gerçekleştirmek, gerçekçi ve güvenilir bir saldırı tespit performansı elde etmek için yeterli değildir. Sınıf dengesizliğini ele alan örnekleme stratejileri, maliyet duyarlı öğrenme yaklaşımları ve uygun değerlendirme metrikleri kullanılmadan elde edilen sonuçlar, yanıltıcı ve pratikte uygulanabilir olmayan çıkarımlara yol açabilmektedir. Buna istinaden, bu tez kapsamında, KL-Ayrışımı tabanlı bir yeniden örnekleme

algoritması tasarlanmıştır. Bu algoritma çoğunluk sınıflarını yeniden örneklerken, veri dağılımını da koruyarak, çoğunluk sınıfının genel dağılım profilini korumayı ve eğitim sırasında aşağı örneklenmiş modelin veri kaybına rağmen genel örüntülerinin tutulması amaçlanmaktadır.

6.2. KLSampler Algoritması

Bu tez kapsamında önerilen algoritmik tasarımın temel motivasyonu, çoğunluk sınıfına ait örneklerin aşağı örneklenmesini sağlarken, bu sınıfın temel dağılımsal özelliklerini korumaktır. Daha önceki tartışmalarda da vurgulandığı üzere, geleneksel aşağı örnekleme yöntemleri, çoğunluk sınıfına ait örneklerin sayısını azaltırken veri setinin orijinal dağılımını koruma konusunda yetersiz kalmaktadır. Ancak, bir makine öğrenmesi modelinin genelleme performansının etkili olabilmesi için, veri kümesinin varyansını ve içsel dağılım özelliklerini mümkün olduğunca eksiksiz biçimde korumak büyük önem taşımaktadır.

Bu doğrultuda, çalışmanın amacı; dengesiz veri kümelerinde, çoğunluk sınıfının dağılımsal karakteristiklerini mümkün olan en yüksek doğrulukla koruyacak şekilde aşağı örneklenmesini sağlayan bir yaklaşım geliştirmektir. Bu süreçte, çoğunluk sınıfı yalnızca gerekli olan en az sayıda örnekle temsil edilmekte ve böylece veri bütünlüğüyle model başarımı arasında denge kurulmaktadır.

Bu kritik probleme çözüm üretmek amacıyla, aşağı örnekleme sürecinde sıklıkla karşılaşılan dağılımsal bütünlük kaybını minimize etmeye yönelik özel mekanizmalar içeren KLSampler algoritması, bu tez kapsamında önerilmektedir.

6.2.1. Algoritmanın Betimleyici Açıklaması

Bu bölümde, algoritma açıklamasına geçmeden önce, KLSampler algoritmasının temel işleyişini netleştirmek amacıyla genel bir bakış sunulması amaçlanmıştır. Buna istinaden, algoritmanın görsel akışına ait detaylar temsil Şekil 6.1'de verilmiştir.

KLSampler algoritması, özellikle ikili sınıflandırma problemleriyle (IoT problemi çerçevesinde Normal (0): Saldırı (1)) ilişkili veri kümeleri üzerinde çalışmak üzere

tasarlanmıştır. Bu nedenle, veri kümesinde iki temel sınıf etiketinin bulunduğu varsayılmaktadır: çoğunluk ve azınlık sınıfları. Bu varsayıma dayanarak, algoritma ilk olarak veri kümesini çoğunluk ve azınlık sınıflarına ayırmaktadır. KLSampler bir aşağı örnekleme algoritması olduğundan, esas olarak çoğunluk sınıfı üzerinde işlem yapmakta ve yeterli temsil gücüne sahip bir alt küme elde edilene kadar çoğunluk sınıfını yinelemeli biçimde örneklemektedir.

Algoritmanın temel amacı, orijinal çoğunluk sınıfı ile aşağı örneklenmiş versiyonu arasındaki dağılımsal profil benzerliğini ölçmek olduğundan, orijinal verinin dağılım profilinin doğru şekilde karakterize edilmesi büyük önem taşımaktadır. Çoğunluk sınıfı X_{maj} notasyonu ile temsil edilmiştir. Bu bağlamda, X_{maj}, m örnek sayısı, n ise öznitelik sayısı olmak üzere, mxn boyutlarında bir matris şekilde tanımlanır.

Bu çoğunluk sınıfına ait veri kümesinin dağılımını etkili bir şekilde temsil edebilmek amacıyla, Çekirdek Yoğunluk Tahmini (Kernel Density Estimation, KDE) yöntemi kullanılmıştır. KDE yöntemi, Denklem (6.1)'de gösterilmiştir:

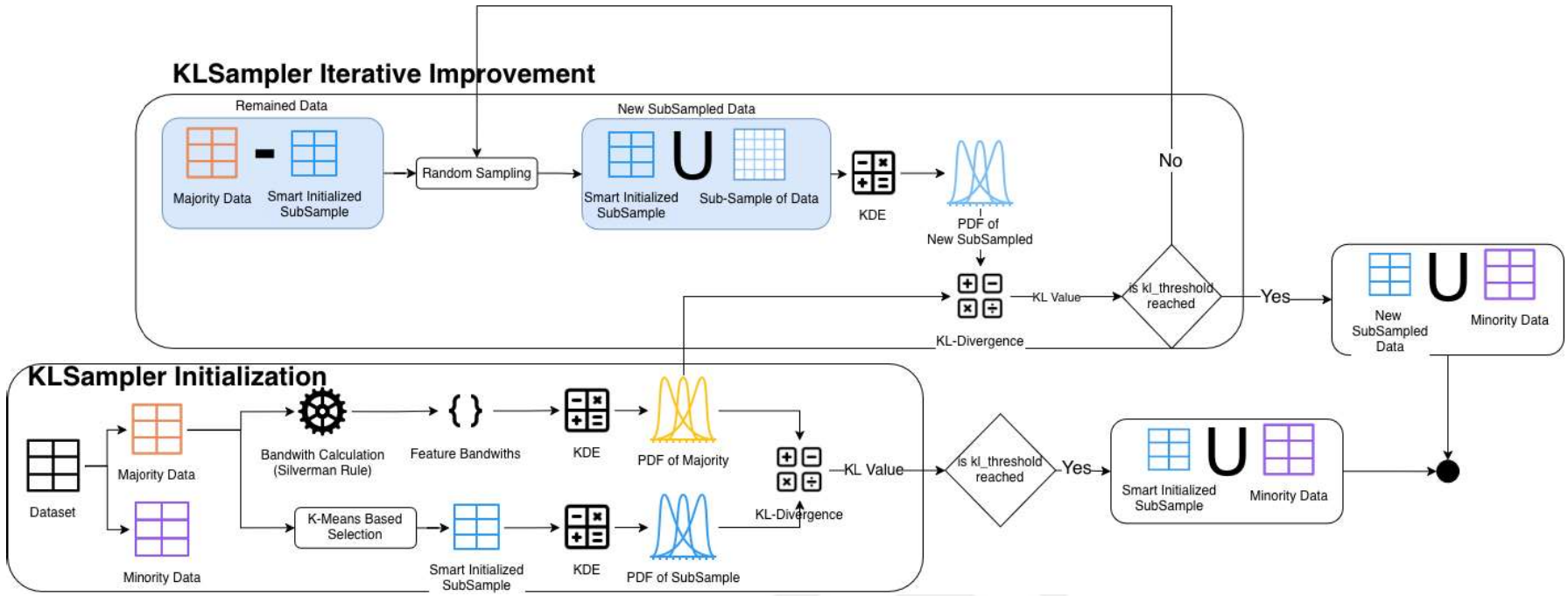
$$P(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{6.1}$$

Burada, $P(x)$ ifadesi çoğunluk sınıfı X_{maj}'a ait Olasılık Yoğunluk Fonksiyonu'nu (Probability Density Function, PDF) temsil etmektedir. K, çekirdek fonksiyonunu (örneğin, Gauss çekirdeği) belirtirken; h, bir özelliğe ait bant genişliğini göstermektedir. Bant genişliği parametresi sabit bir değer olarak belirlenebileceği gibi, Silverman Kuralı (Silverman, 2018) gibi sezgisel (heuristic) bir yöntemle de hesaplanabilmektedir.

Silverman Kuralı, Denklem (6.2)'de ifade edilmiştir:

$$bandwith_f = 1,06 * \sigma * n^{-1/3} \tag{6.2}$$

Burada, f belirli bir özelliği (feature) ifade ederken, σ ilgili özelliğin standart sapmasını, n ise X_{maj} veri kümesindeki örnek sayısını temsil etmektedir. Bu adımın tamamlanmasının ardından, orijinal çoğunluk sınıfına ait olasılık yoğunluk fonksiyonu (PDF) elde edilmiş olunur.



Şekil 6.1. KL-Sampler Algoritmasına Ait İş Akışı

Bu işleme paralel olarak, algoritma başlangıçta X_{maj} veri kümesinden aşağı örneklenmiş bir alt küme ($X_{sampled}$) gerektirmektedir. Ardından, algoritma bu alt kümeyi yinelemeli olarak yeni örnekler ekleyerek genişletmekte ve böylece alt kümenin, orijinal veri kümesinin dağılımını etkili bir biçimde temsil ettiği noktaya ulaşmayı hedeflemektedir. Orijinal verinin olasılık yoğunluk fonksiyonu (PDF) belirlendikten sonra, benzer bir yaklaşım kullanılarak, KDE ve Silverman Kuralı aracılığıyla, aşağı örneklenmiş alt kümenin PDF'i hesaplanmaktadır. Eğer orijinal çoğunluk sınıfının olasılık yoğunluk fonksiyonu $P(x)$ ile, aşağı örneklenmiş veri kümesinin ($X_{sampled}$) PDF'i ise $Q(x)$ ile gösterilirse, bu iki dağılım arasındaki benzerlik, her bir öznitelik için ayrı ayrı hesaplanan KL-ayrışımı yöntemiyle ölçülebilmektedir. Bu işlem, Denklem (6.3)'te gösterildiği şekilde ifade edilmektedir:

$$KL(f) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \tag{6.3}$$

Her bir öznitelik (feature) için KL-ayrışımı (KL divergence) ayrı ayrı hesaplandıktan sonra, bu ayrışım değerlerinin toplamı alınarak tüm öznitelikler arasındaki kümülatif fark derecesi Denklem (6.4)'te ifade edildiği şekilde elde edilmektedir:

$$total_kl = \sum_{f \in F} KL(f) \tag{6.4}$$

Bu denklemde, F tüm öznitelik uzayını (feature space) temsil etmektedir.

Bunun ardından KLSampler algoritması, belirlenen KL-ayrışımı eşiğinin sağlanıp sağlanmadığını değerlendirmektedir. Eğer bu eşik sağlanmışsa, model aşağı örneklenmiş veri kümesini azınlık sınıfı ile birleştirerek $X_{sampled} \cup X_{min}$ çıktısını üretmektedir. Ancak, belirlenen KL-ayrışımı için eşiği henüz sağlanmamışsa, algoritma, kalan çoğunluk sınıfı örneklerinden ($X_{maj}/X_{sampled}$) rastgele örnekler seçerek bunları yinelemeli olarak $X_{sampled}$ kümesine eklemektedir.

Her yinelemede, algoritma KDE yöntemiyle güncellenmiş $X_{sampled}$ kümesine ait olasılık yoğunluk fonksiyonunu (PDF) yeniden hesaplamakta, bu dağılımın orijinal çoğunluk sınıfı dağılımından olan KL-ayrışımını ölçmekte ve bu işlemi belirlenen $total_kl$

eşiği sağlanana dek sürdürmektedir. Sonuç olarak, hedeflenen KL-ayrışımı seviyesine ulaşıldığında, algoritma $X_{sampled} \cup X_{min}$ birleşim kümesini çıktı olarak vermektedir.

Bu süreçte en kritik hususlardan biri, $total_kl$ değerinin uygun biçimde belirlenmesidir. Yukarıda açıklandığı şekilde, özniteliklerin kümülatif dağılımına dayalı olarak KL ayrıklığının hesaplanması ve karşılaştırılması, özniteliklerin doğrusal bağımsızlığını gerektirmektedir. Bu çalışma kapsamında, bu koşulun sağlandığı açıkça varsayılmakta ve ilerleyen uygulama aşamalarında gösterileceği üzere bu konuda bir çalışma gerçekleştirilmiştir. Dolayısıyla, örnekleme algoritması çalıştırılmadan önce, veri kümesindeki özniteliklerin doğrusal olarak bağımsız olduğundan emin olunması gerekmektedir. Zira bu koşul, hem KL-ayrışımı hesaplamalarının geçerliliği hem de algoritmanın etkinliği açısından kritik bir öneme sahiptir.

6.2.2. KLSampler Sözde Kodu

Bu bölümde, önerilen KLSampler algoritmasına ait sözde kod (pseudo-code) sunulmakta ve ayrıntılı biçimde açıklanmaktadır. Özellikle, önceki bölümlerde kavramsal olarak tanıtılan yaklaşımın algoritmik uygulaması adım adım gösterilerek, önerilen aşağı örnekleme sürecinin nasıl çalıştığı ortaya konulmaktadır.

Algoritma, temel olarak giriş veri kümesini (X) azınlık (X_{min}) ve çoğunluk (X_{maj}) sınıflarına ayırmakta, ardından çoğunluk sınıfından başlangıçta akıllı bir şekilde seçilen bir alt küme ($X_{sampled}$) oluşturarak bu küme üzerine iteratif olarak veri noktaları eklemektedir. Algoritmanın başlatma adımları, Algoritma 6'da gösterilmiştir.

Satır 1–9 arasında, algoritmanın çalıştırılması için gerekli parametreler açıklanmaktadır. Burada X, üzerinde aşağı örnekleme uygulanacak olan orijinal veri kümesini ifade eder. Bu küme, $n \times m$ boyutlarında bir matristir; burada n veri noktası sayısını, m ise öznitelik (feature) sayısını göstermektedir.

Sınıf etiketlerini içeren y vektörü, veri kümesinin her satırına karşılık gelen ikili sınıf etiketlerinden (0 veya 1) oluşmaktadır. $target_kl$ parametresi ise modelin durma kriterini temsil eden kümülatif KL-ayrışım değeridir.

Eğer bu eşik sağlanmazsa, satır 9'da tanımlanan *increment* parametresi, her yinelemede

Algoritma 6: KLSampler Algoritması - Başlatma Aşaması (Initialization)

Gerekli:

X: Özniteliklerden oluşan veri çerçevesi ($n \times m$ boyutunda)
y: Sınıf etiketlerini içeren seri (n)
$target_kl$: Hedef KL-ayrışım eşiği
$increment$: Her yinelemede eklenecek örnek sayısı
$initial_samples$: Başlangıç örneklem sayısı (s_0)
$bandwidth$: KDE için bant genişliği parametresi
max_iters: Maksimum yineleme sayısı
$patience$: Erken durdurma toleransı
$random_state$: Rastgelelik için başlangıç değeri

Ensure:

$X_sampled$: Orijinal dağılımı koruyan alt örneklem kümesi

Başlatma:

$best_kl \leftarrow \infty$
$no_improvement_count \leftarrow 0$
$X_maj \leftarrow X[y = 0]$
$X_min \leftarrow X[y = 1]$

Her bir öznitelik için KDE bant genişliğini başlat:

if $bandwidth = \emptyset$ **then**
    for each feature f **do**
        $bandwidth_f \leftarrow 1.06 * \sigma_f * n^{-1/3}$
    end for
else
    for each feature f **do**
        $bandwidth_f \leftarrow bandwidth$
    end for
end if

Çoğunluk sınıfı için KDE hesapla:

for each feature f **do**
    Olasılık yoğunluk fonksiyonu:

$$P_f(x) = \frac{1}{n * h_f} \sum_{i=1}^{n} K\left(\frac{x - X_maj_i^f}{h}\right)$$

end for
burada K: çekirdek fonksiyonu (örneğin Gauss), h: bant genişliğidir

Başlangıç örneklem kümesini oluştur:

$X_sampled \leftarrow SmartInitialSamples(X_maj, initial_samples, random_state)$
$n_{\text{samples}} \leftarrow |X_sampled|$

çoğunluk sınıfı içinden $X_{sampled}$ kümesine eklenecek yeni örnek sayısını belirler. *initial_samples* ise başlangıçta seçilecek olan örnek sayısını tanımlar.

Satır 12'de, henüz herhangi bir ayrışım hesaplaması yapılmadığı için *best_kl* değişkeni ∞ olarak başlatılmaktadır. Benzer şekilde, satır 13'te *no_improvement_count* değişkeni 0 olarak baştlatılarak ve başta herhangi bir iyileşme durumu olmadığını ifade etmektedir.

Satır 14-16 arasında, veri kümesi azınlık (X_{min}) ve çoğunluk (X_{maj}) sınıflarına ayrılmaktadır. Satır 17-26 arasında ise KDE işlemleri için gerekli olan bant genişliği parametreleri hesaplanmaktadır. Bu parametreler, kullanıcı tarafından belirtilmediği takdirde, her bir öznitelik için Silverman kuralı kullanılarak otomatik olarak belirlenmektedir.

Satır 27-31'da, çoğunluk sınıfına (X_{maj}) ait orijinal veri dağılımına ait olasılık yoğunluk fonksiyonları (PDF) öznitelik bazında hesaplanmakta ve bu hesaplama sırasında daha önce belirlenen bant genişlikleri kullanılmaktadır. Son olarak, satır 32-34'de ise, çoğunluk sınıfı içerisinden başlangıç örnekleri başka bir algoritmaya gönderilerek (SmartInitialSamples) akıllıca seçecek şekilde $X_{sampled}$ kümesi oluşturulmakta ve bu küme, algoritmanın yinelemeli örnekleme sürecindeki ilk adımı olarak kullanılmaktadır.

Algoritma 7, önerilen algoritmada kullanılan başlangıç örnekleme stratejisini oluşturmak amacıyla özel olarak tasarlanmış akıllı aşağı örnekleme mekanizmasını göstermektedir.

Algoritma 7: SmartInitialSamples Fonksiyonu

1: Silhouette Skoru kullanarak optimal küme sayısı k'yı hesapla:

$$\text{Silhouette}(k) = \frac{b-a}{\max(a,b)}$$

burada a: Küme içi ortalama uzaklık, b: En yakın komşu küme ile olan ortalama uzaklıktır.

2: X üzerine k-Ortalama (k-means) kümeleme uygula:

$$\text{Küme Merkezleri} = \{C_1, C_2, \ldots, C_k\}$$

3: n_{samples} sayısını kümeler arasında orantılı biçimde dağıt

4: Her kümeden, belirlenen orana göre rastgele örnek seç

5: **Return** Başlangıç örneklem alt kümesi

Bu algoritmadaki temel amaç, çoğunluk sınıfına ait veri kümesinden (X_{maj}) orijinal dağılım profilini en doğru şekilde temsil edecek bir başlangıç alt kümesi ($X_{sampled}$) belirlemektir. Bu hedef doğrultusunda, dağılımsal öznitelikleri en iyi şekilde koruyan bir başlangıç örnekleme stratejisi seçmeye odaklanılmıştır.

İlk adımda, çoğunluk sınıfına ait veri kümesine farklı k değerleri (küme sayıları) için K-Ortalama kümeleme algoritması uygulanmaktadır. Ardından, elde edilen kümeleme yapılarına göre küme sayısı k için en uygun değer, kümeleme algoritmalarında sıkça kullanılan ve küme kalitesini ölçen köklü bir metrik olan Silhouette Skoru (Silhouette Score) yardımıyla belirlenmektedir. Bu kümeleme işlemi ile, veri kümesindeki yoğunluk bölgeleri tespit edilmeye çalışılmaktadır.

Optimal küme sayısı belirlendikten sonra, her bir kümeden, ilgili kümenin büyüklüğüne orantılı olacak şekilde rastgele örnekler seçilmektedir. Bu orantılı örnekleme yaklaşımı, aşağı örneklenmiş veri kümesinin, orijinal verideki yoğun veri bölgelerini (yani kümeleri) yeterli düzeyde temsil etmesini sağlamaktadır.

Sonuç olarak, bu yöntemle oluşturulan aşağı örneklenmiş veri kümesi, Algoritma 6'ya giriş olarak sağlanmakta ve algoritmanın yinelemeli işlemleri için sağlam ve temsil gücü yüksek bir başlangıç alt kümesi elde edilmiş olmaktadır.

Algoritmanın son aşaması, Algoritma 8'de gösterilen yinelemeli bir iyileştirme sürecinden oluşmaktadır.

Algoritma 8'de gösterildiği üzere, satır 2'de yinelemeli bir döngü başlatılmakta ve $X_{sampled}$ kümesine, önceden tanımlanmış maksimum yineleme sayısına (max_iters) kadar örnekler yinelemeli olarak eklenmektedir. Satır 3-4'te, mevcut $X_{sampled}$ kümesi için KDE yöntemiyle PDF hesaplanmakta ve ardından bu dağılım, orijinal dağılımla karşılaştırılarak her bir öznitelik bazında KL-ayrışımı hesaplanmaktadır.

Satır 5'te, öznitelikler için KL-ayrışım değerlerinin toplamı alınarak toplam KL-ayrışımı ($total_kl$) hesaplanmaktadır. Satır 6–10'da ise, KL-ayrışımda iyileşme tespit edilirse ilgili değerler güncellenmekte ve $no_improvement_count$ sıfırlanmaktadır.

Satır 11–13'te, ilk erken durdurma kriterini test edilmektedir: Eğer hesaplanan toplam KL-ayrışımı, hedeflenen eşik değerinden ($target_kl$) küçük veya eşitse, algoritma mevcut en iyi örneklem kümesini ($best_sampled_data$) azınlık sınıfı (X_{min}) ile birleştirerek döndürmektedir.

Algoritma 8: KLSampler Algoritması - Yinelemeli İyileştirme Aşaması

Yinelemeli İyileştirme:
for $iter = 1$ **to** max_iters **do**
    Her bir öznitelik f için aşağı örneklenmiş küme üzerinde KDE kullanarak olasılık yoğunluk fonksiyonu hesapla:

$$Q_f(x) = \frac{1}{n * h_f} \sum_{i=1}^{n} K\left(\frac{x - X_sampled_i^f}{h}\right)$$

    burada K: çekirdek fonksiyonu (örneğin Gauss), h: bant genişliği
    Her bir öznitelik f için KL-Ayrışımını hesapla:

$$KL(f) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

    Toplam KL-Ayrışımını hesapla:

$$total_kl = \sum_{f \in \text{features}} KL(f)$$

    if $total_kl < best_kl$ **then**
        $best_kl \leftarrow total_kl$
        $best_sampled_data \leftarrow sampled_data$
        $no_improvement_count \leftarrow 0$
    end if
    if $total_kl \leq target_kl$ **then**
        return $best_sampled_data \cup X_{\text{min}}$
    end if
    $no_improvement_count \leftarrow no_improvement_count + 1$
    if $no_improvement_count \geq patience$ **then**
        return $best_sampled_data \cup X_min$
    end if
    Yeni örnekleri seç:

$$new_samples \leftarrow \text{random}(X_{\text{majority}} \setminus X_sampled,\ increment)$$

    $X_sampled \leftarrow X_sampled \cup new_samples$
    $n_{\text{samples}} \leftarrow |X_sampled|$
end for
return $X_sampled \cup X_min$

Satır 14'te, beklenen iyileşme gözlenemediğinden yineleme sayısını takip eden değişken bir artırılmaktadır. Satır 15-17'de, ikinci bir erken durdurma koşulunu içermektedir: Eğer belirli sayıda ardışık yineleme boyunca iyileşme sağlanamamışsa ($no_improvement_count \geq patience$), algoritma çalışması durdurularak ve eldeki en iyi örneklem kümesi azınlık sınıfı ile birleştirilerek döndürülmektedir.

Eğer bu durma koşullarının hiçbiri sağlanmazsa, KLSampler algoritması, satır 18-21'de rastgele seçilen yeni örnekleri $X_{sampled}$ kümesine ekleyerek alt kümenin boyutunu artırmakta ve süreci yinelemeye devam etmektedir. Son olarak, satır 22'de, algoritma mevcut $X_{sampled}$ kümesini azınlık sınıfı (X_{min}) ile birleştirerek nihai çıktı olarak döndürmektedir.

6.3. KLSampler Algoritmasının Karmaşıklık Analizi

Önerilen KLSampler algoritmasının zaman karmaşıklığı, (i) Başlatma (Initialization) aşaması ve (ii) Yinelemeli İyileştirme (Iterative Refinement) aşaması olarak iki temel bileşenin çerçevesinde değerlendirilmesi mümkündür. Algoritm kapsamında tanımlanan bu iki bölüm, farklı hesaplama maliyetleri içeren alt işlemlerden oluşmaktadır. Buna istinaden, algoritmaya ait toplam karmaşıklık bu işlemlerin birleşiminden oluşmaktadır.

Başlatma aşamasında dikkate alınması gereken ilk işlem, bant genişliği (bandwidth) değerlerinin hesaplanmasıdır. Özellikle Silverman'ın kuralı temel alınarak yapılan bu tahmin, her bir özelliğe karşılık gelen standart sapma veya interkuartil aralık (Interquantile Aralık - IQR) hesaplamasını içerir. n örnek ve m öznitelik içeren bir veri kümesi için bu işlemin karmaşıklığı Denklem (6.5)'te gösterilmiştir:

$$O(n * m) \tag{6.5}$$

Bir sonraki önemli adım ise, çoğunluk sınıfına (X_{maj}) ait veriler üzerinde KDE yöntemi ile olasılık yoğunluk fonksiyonunun tahmin edilmesidir. KDE hesaplamaları, her bir değerlendirme noktası (p adet) için tüm örneklerle (n adet) olan uzaklıkların m boyutlu öznitelikler üzerinden hesaplanmasını gerektirir. Bu işlemin toplam hesaplama maliyeti ise Denklem (6.6)'da gösterilmiştir:

$$O(m * n * p) \tag{6.6}$$

Aynı zamanda, başlangıç alt kümesinin seçiminde kullanılan k-Ortalama algoritmasının karmaşıklığı da dikkate alınmalıdır. k küme sayısı, n örnek ve m öznitelik üzerinden t iterasyon sonunda yakınsama sağlandığı varsayıldığı durumda, bu işlemin toplam zaman karmaşıklığı Denklem (6.7)'deki gibidir:

$$O(t * k * n * m) \tag{6.7}$$

Bu işlem adımlarına ek olarak, KDE tahmini sonucunda elde edilen dağılımların doğruluğunu ölçmek amacıyla KL-ayrışımı kullanılmaktadır. Her yinelemede, örneklenmiş veri kümesi ($X_{sampled}$) ile orijinal dağılım ($P(x)$) arasındaki fark ölçülür. m adet öznitelik, her iterasyonda seçilen s örnek, p değerlendirme noktası ve $i_{\max}$ maksimum iterasyon varsayımı altında, bu işlemin hesaplama yükü Denklem (6.8)'deki gibidir:

$$O(i_{\max} * m * s * p) \tag{6.8}$$

Benzer şekilde, yineleme sürecinde her adımda KDE işlemi tekrarlandığı için, örneklenmiş alt küme üzerindeki KDE işlemi de yinelemeli olarak Denklem (6.9)'daki gibi hesaplanmaktadır:

$$O(i_{\max} * m * n * p) \tag{6.9}$$

Tüm bu adımlar bir arada değerlendirildiğinde, önerilen algoritmanın toplam zaman karmaşıklığı Denklem (6.10)'da gösterilmiştir:

$$O(t * k * n * m) + O(n * m) + O(i_{\max} * m * n * p) + O(i_{\max} * m * s * p) \tag{6.10}$$

Sonuç olarak, KLSampler algoritması, polinomsal zaman karmaşıklığına sahip olmakla birlikte, her bir bileşenin optimize edilebilir ve paralelleştirilebilir yapısı sayesinde uygulanabilirliği yüksek bir çözüm sunmaktadır. Özellikle orta ölçekli veri kümelerinde (örneğin binlerce örneklem ile buna ek olarak onlarca/yüzlerce öznitelik), hem dağılımsal

bütünlüğü koruyan örnekleme yapısı hem de hesaplama verimliliği açısından dengeli bir yapı sergilemektedir. Bu bağlamda, dengesiz veri kümeleriyle çalışılan senaryolarda, önerilen algoritmanın etkili bir örnekleme çözümü sunduğu değerlendirilmektedir.

6.4. Deneysel Sonuçlar

6.4.1. Çeşitli Veri Setlerinde Alınan Sonuçlar

Önerilen KL-ayrışımı temelli aşağı örnekleme algoritmasının etkinliğini ve sağlamlığını değerlendirmek amacıyla, hem gerçek dünya hem de sentetik senaryoları kapsayan beş farklı veri kümesi üzerinde bir dizi deneysel çalışma gerçekleştirilmiştir. Bu veri kümeleri, sınıf dengesizliğinin farklı derecelerini, veri kümesi boyutlarını ve alan özniteliklerini yansıtacak şekilde seçilmiştir. Bu değerlendirmenin temel amacı, önerilen yöntemin yalnızca sınıf dengesizliğini azaltmakla kalmayıp, aynı zamanda çoğunluk sınıfının istatistiksel dağılımını da koruyarak Değişken Kaymasını (Covariate Shift) en aza indirdiğini göstermektir.

Tüm deneyler, 32 GB RAM, Intel Core i7-1255U (1.70 GHz, 12. nesil) işlemciye sahip ve Windows 11 Pro 64-bit işletim sistemi çalıştıran kişisel bir bilgisayar üzerinde gerçekleştirilmiştir. Donanımın sınırlı kaynaklarına rağmen, algoritma tüm yeniden örnekleme işlemlerini makul süreler içerisinde tamamlamış; bu durum, yöntemin hem verimli hem de ölçeklenebilir olduğunu ortaya koymuştur. Hesaplama verimliliğini daha da artırmak amacıyla, Algoritma 8'de detaylandırılan yinelemeli iyileştirme süreci paralel işlemeye uygun şekilde yapılandırılmıştır. Özellikle, X_{maj} ve $X_{sampled}$ kümeleri arasındaki KL-ayrışımı hesaplamaları paralel olarak yürütülerek, öznitelik bazlı ayrıklık tahminine ilişkin zaman karmaşıklığı azaltılmıştır. Bu iyileştirme sayesinde, algoritmanın daha büyük veri kümelerine ölçeklenebilirliği etkin biçimde sağlanmıştır.

Farklı veri setlerine ait KLSampler algoritmasının uygulanmasına dair sonuçlar Tablo 6.2'de gösterilmiştir.

Dağılımsal korumanın etkinliğini değerlendirmek amacıyla, deneylerde iki farklı ölçüt kullanılmıştır: KL-Ayrışımı, çoğunluk sınıfı ile alt örneklenmiş alt küme arasındaki dağılımsal farklılıkları ölçerken, değişken kaymasını ise, orijinal ve yeniden örneklenmiş

veri kümeleri arasında ayrım yapmaya çalışan bir alan sınıflandırıcısının Alıcı İşletim Karakteristiği (Receiver Operating Characteristic, ROC) eğrisi altındaki alanını (Area Under Curve, AUC) temsil etmektedir. Yeniden örnekleme işlemi tamamlandıktan sonra, her bir örneğe ait olup olmadığına göre ikili etiket atanmakta; yani örneğin orijinal çoğunluk sınıfına mı yoksa alt örneklenmiş alt kümeye mi ait olduğu belirtilmektedir. Bu birleştirilmiş veri kümesi üzerinde, dağılımsal kayma derecesini tahmin etmek amacıyla bir Rasgele Orman (Random Forest) sınıflandırıcısı eğitilmiştir. Elde edilen AUC değerinin 0.5'e yakın olması, sınıflandırıcının orijinal ve yeniden örneklenmiş veriyi ayırt edemediğini ve dolayısıyla dağılımın yüksek düzeyde korunduğunu göstermektedir. Kullanılan tüm veri kümelerinde orijinal sınıf dağılımı belirgin şekilde dengesiz olup, azınlık sınıf oranı %11,69 ile %26,53 arasında değişmektedir.

Telekom Churn veri kümesinde (Blastchar, 2018), orijinal dengesizlik oranı (%73,46 : %26,53) yalnızca 4,64 saniyede tam olarak dengelenmiş (%50 : %50) ve örnekleme sonrası çoğunluk sınıfının dağılımında gözlemlenen değişiklik oldukça düşük kalmıştır (KL-ayrışımı = 0,005). Ayrıca, değişken kayma (covariate shift) AUC değeri 0,541 olarak ölçülmüş olup bu değer, modelin orijinal ve yeniden örneklenmiş verileri ayırt edemediğini ve böylece dağılımsal bütünlüğün korunduğunu göstermektedir.

Banka Pazarlama (Bank Marketing) veri kümesinde ise (Moro ve diğ., 2014), (%88,30 : %11,69), algoritma daha yoğun bir şekilde çoğunluk örneklerini elemeye ihtiyaç duymuştur. 271,16 saniyede tamamlanan yeniden örnekleme işlemi sonucunda, sınıf oranı %65,82 : %34,18 seviyesine gelmiş ve KL-ayrışım değeri 5,565 olarak ölçülmüştür. Bu artış, örnekleme yoğunluğu dikkate alındığında kabul edilebilir düzeydedir. Elde edilen AUC değeri 0,580 olup, özellik uzayının yapısal bütünlüğünün büyük ölçüde korunduğunu göstermektedir.

Yetişkin (Adult) veri kümesinde (Becker ve Kohavi, 1996), başlangıçta %75,91 : %24,08 olan sınıf oranı, yeniden örnekleme sonucunda %60,56 : %39,44 olarak güncellenmiş; KL-ayrışımı 0,082 ve AUC değeri 0,546 olarak hesaplanmıştır. Bu durum, dağılımsal koruma ile sınıf dengelemesi arasında etkili bir denge sağlandığını ortaya koymaktadır.

Tablo 6.2. Veri Kümeleri İçin Yeniden Örnekleme İstatistikleri

Veri Kümesi Özellikleri					
Veri Kümesi	**Örnek Sayısı**	**Öznitelik Sayısı**	**Orijinal Sınıf Dağılımı**	**Sınıf Oranları**	
Telekom Churn	7043	21	5174:1869	73,46%:26,53%	
Çevrimiçi Ziyaretçi	12330	17	10422:1908	84,53%:15,47%	
Yetişkin Verisi	32561	14	24720:7841	75,91%:24,08%	
Banka Pazarlama	45211	16	39922:5289	88,30%:11,69%	
ToN_IoT (Örneklenmiş)	70347	18	53681:16666	76,30%:23,70%	
Sentetik 100K	100000	15	79700:20300	79,70%:20,30%	
Yeniden Örnekleme Sonuçları					
Veri Kümesi	**Süre (sn)**	**Dengeleme Sonrası**	**Yeni Sınıf Oranı**	**KL-Ayrışımı**	**AUC**
Telekom Churn	4,64	1869:1869	50:50	0,005	0,541
Çevrimiçi Ziyaretçi	25,20	3808:1908	66,62:33,38	0,141	0,534
Yetişkin Verisi	111,74	12041:7841	60,56:39,44	0,082	0,546
Banka Pazarlama	271,16	10189:7841	65,82:34,18	5,565	0,580
ToN_IoT (Örneklenmiş)	401	51166:16666	75:25	0,00006	0,473
Sentetik 100K	1582,05	20900:20300	50,72:49,27	0,497	0,6313

Benzer şekilde, en yüksek dengesizlik oranına sahip veri kümesi olan Çevrimiçi Ziyaretçi (Online Shoppers)(Sakar ve Kastro, 2018) veri kümesinde (%84,53 : %15,47), yeniden örnekleme sonrası %66,62 : %33,38 oranına ulaşmış; KL-ayrışımı 0,141 ve AUC değeri 0,534 olarak ölçülmüştür. Bu değerler, yeniden örnekleme sürecinin sınıf dağılımını önemli ölçüde değiştirmeden gerçekleştirilmiş olduğunu göstermektedir.

ToN_IoT veri kümesi (Moustafa, 2021), ağ saldırı tespiti alanında yaygın şekilde kullanılan bir benchmark olup bu çalışmada ikili sınıflandırma problemi olarak ele alınmıştır. Veri kümesi, normal trafik ve saldırı örneklerini içermektedir. Eksik veri oranlarının yüksek olduğu özellikler dışlanarak, sadece 18 özellik kullanılmıştır. Başlangıçta %76,30 : %23,70 olan sınıf oranı, KLSampler algoritması tarafından %75 : %25 düzeyine getirilmiştir. KL-ayrışım değeri 0.0006 ve ve AUC değeri 0.473 olarak gözlemlenmiş, bu da orijinal dağılımın yüksek yapısal karmaşıklığının korunduğunu göstermektedir.

Sentetik 100K veri kümesi, her iki sınıfın Gauss dağılımı ile üretildiği sentetik bir veri kümesidir ve 79.700 çoğunluk, 20.300 azınlık örneği içermektedir. KLSampler algoritması bu veri kümesini %50,72 : %49,27 oranına dengelemiş, işlem 1582,05 saniyede tamamlanmıştır. KL-ayrışımı 0,497 ve AUC değeri 0,6313 olarak ölçülmüştür. Bu değerler nispeten yüksek olsa da, sentetik verilerin yüksek boyutlu yapısal seyrekliğe sahip olmaması nedeniyle anlamlı kabul edilmektedir.

Tüm veri kümeleri incelendiğinde, AUC değerlerinin 0,534 ile 0,6313 arasında değiştiği gözlemlenmiş ve bu durum algoritmanın orijinal veri ile yeniden örneklenmiş veri arasındaki ortak özellik dağılımını başarılı şekilde koruduğunu göstermiştir. Özellikle, çoğu gerçek dünya veri kümesinde KL-ayrışımı değerinin 0,1'in altında olması, algoritmanın dağılımsal benzerlik ilkesine dayalı etkinliğini desteklemektedir. Ayrıca, yeniden örnekleme süreleri veri büyüklüğüyle doğru orantılı şekilde artmasına rağmen, uygulama açısından kabul edilebilir düzeyde kalmıştır.

Sonuç olarak, KL-ayrışım temelli alt örnekleme stratejisi, yüksek dengesizlik içeren veri kümelerinde etkili bir yeniden örnekleme yöntemi sunmaktadır. Klasik rastgele alt örnekleme yaklaşımlarının, çoğunluk sınıfındaki önemli örüntüleri rastlantısal biçimde

ortadan kaldırma riski taşırken, önerilen yöntem orijinal veri dağılımının istatistiksel yapısını korumaya odaklanmaktadır. Bu, özellikle üretim ortamına benzer dağılım gerektiren kritik sınıflandırma görevlerinde modelin genelleme başarısı açısından büyük önem taşımaktadır. Bu bölümde ele alınan veri setlerinde yer alan özniteliklerin, KDE kullanılarak iki boyutlu uzayda gösterimi ve buna bağlı olarak da öznitelik çiftlerinin nasıl dağıldığı ve KLSampler'ın dağılımı nasıl koruduğu görsel örneklerle Şekil 6.2 ile Şekil 6.5 arasında gösterilmiştir.

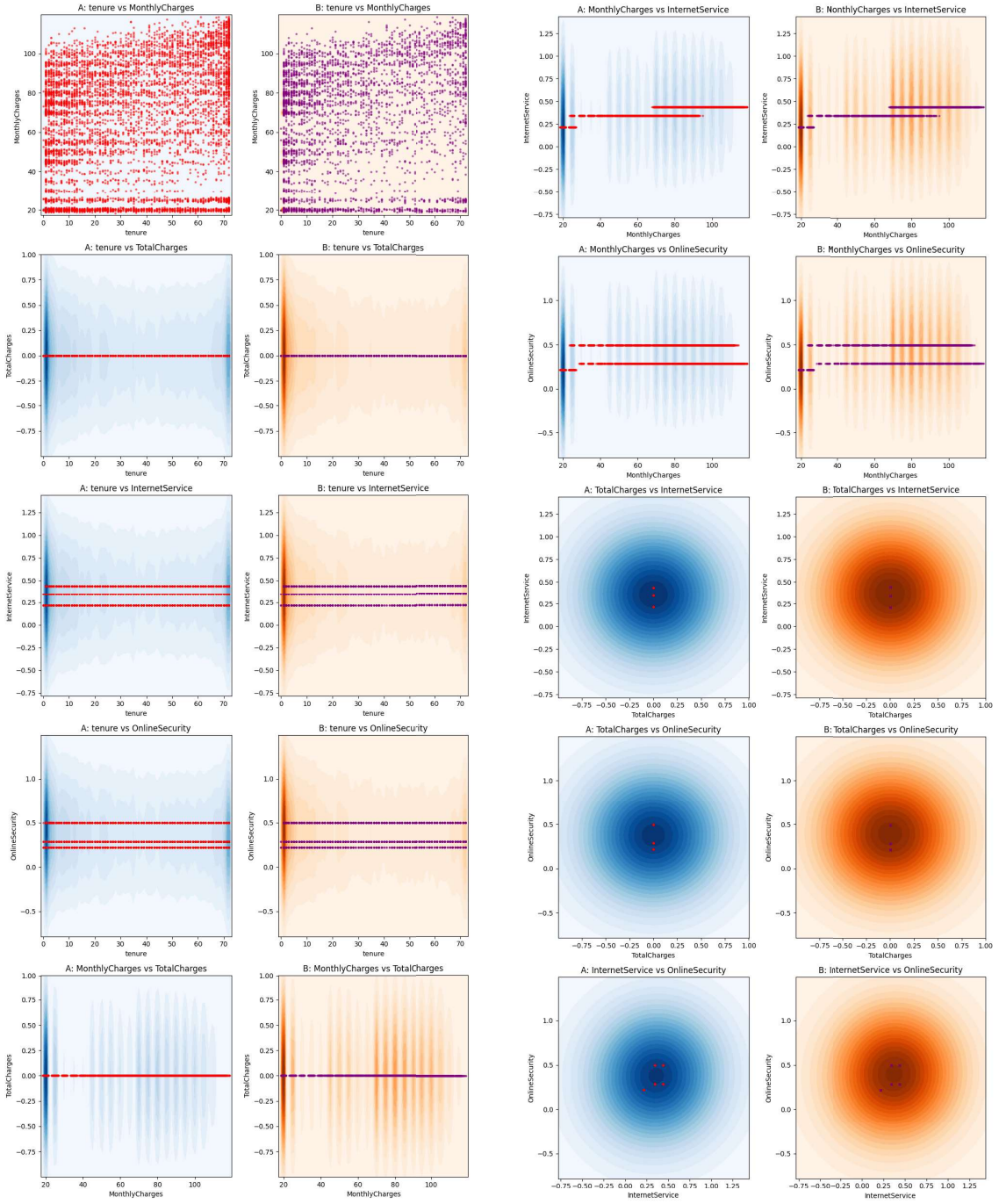
İlgili grafiklerin sol tarafı, herhangi bir örnekleme işlemi uygulanmamış orijinal veri kümesine ait ortak (joint) dağılımı temsil ederken; sağ tarafta yer alan benzer yapıdaki grafikler ise Tablo 6.2'de özetlenen sonuçlar doğrultusunda KLSampler uygulanarak elde edilen aşağı örneklenmiş veri setine ait dağılımları göstermektedir.

Şekil 6.2'de Churn veri setine ait öznitelik çiftlerinin iki boyutlu ortak dağılımları KDE tabanlı görselleştirme ile sunulmuştur. Şeklin sol tarafında herhangi bir yeniden örnekleme uygulanmamış orijinal veri kümesinin dağılımı yer almakta; sağ tarafta ise KLSampler algoritması uygulanarak elde edilen aşağı örneklenmiş veri setinin dağılımı gösterilmektedir.

Orijinal veri kümesinde, özellikle tenure-MonthlyCharges ve MonthlyCharges-TotalCharges öznitelik çiftlerinde belirgin yoğunluk bölgeleri ve doğrusal ilişki yapıları gözlemlenmektedir. Çoğunluk sınıfının baskın olduğu bölgelerde yoğunluk kümelenmeleri belirgin şekilde artmaktadır. Rastgele alt örnekleme yöntemlerinde bu yoğun bölgelerin parçalanması ve yerel örüntülerin kaybolması sık karşılaşılan bir durumdur. Buna karşılık sağ tarafta görülen KLSampler çıktısında, yoğunluk merkezlerinin konumunun ve kontur yapılarının büyük ölçüde korunduğu görülmektedir.

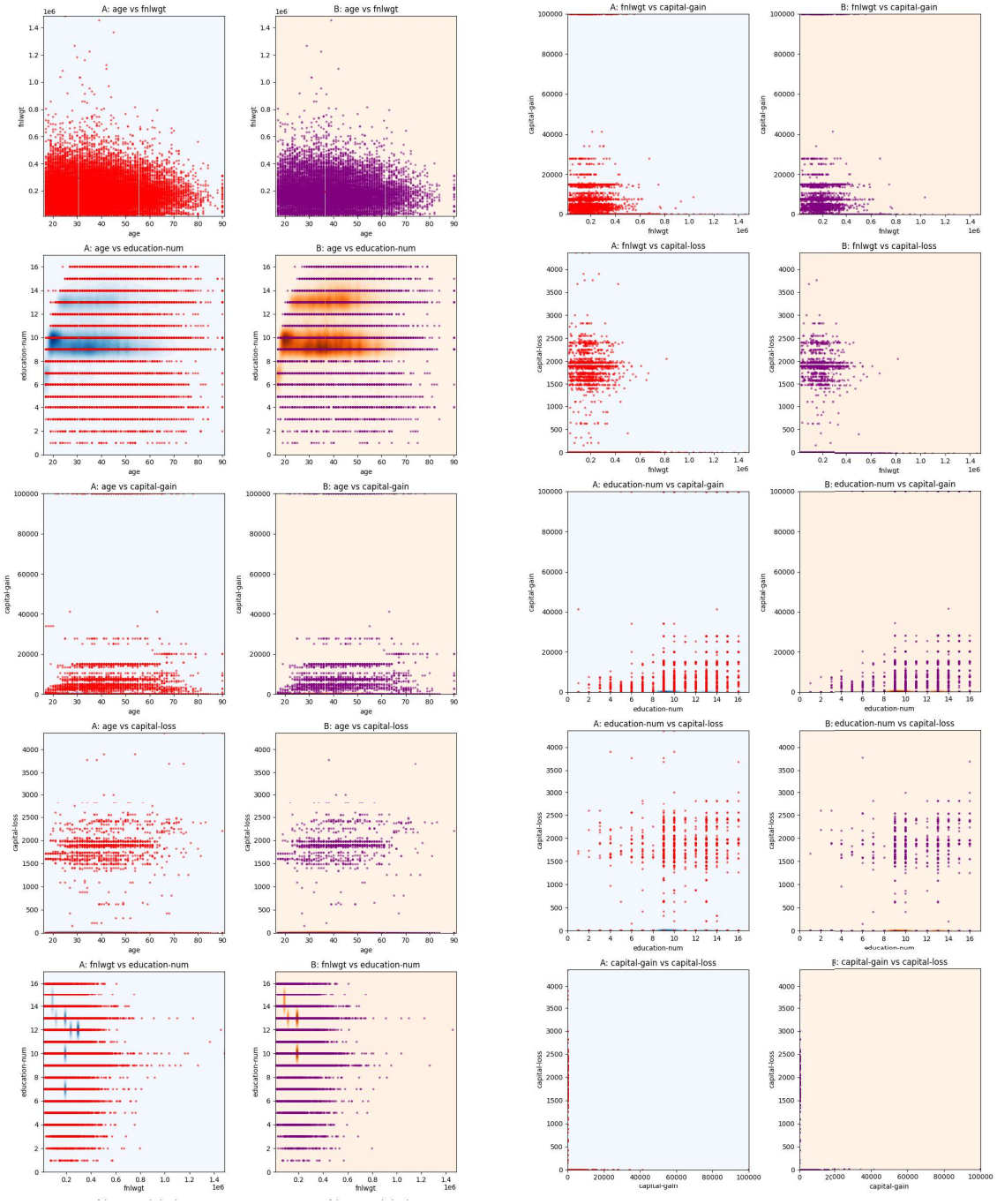
Özellikle InternetService–OnlineSecurity ve TotalCharges–InternetService öznitelik çiftlerinde gözlenen iki boyutlu dağılımın geometrik formu büyük ölçüde korunmuş, yalnızca veri noktalarının yoğunluğunda sınırlı bir azalma meydana gelmiştir. Başka bir ifadeyle, örnek sayısındaki düşüşe rağmen karar sınırlarını etkileyebilecek yapısal bir bozulma ya da dağılımın temel eğilimlerinde belirgin bir sapma gözlemlenmemiştir. Bu durum, önerilen yöntemin verinin içsel topolojisini ve sınıflar arası ilişkileri istatistiksel

açıdan muhafaza ettiğini, müdahalenin yalnızca örnek hacmini azaltmakla sınırlı kaldığını ortaya koymaktadır. Dolayısıyla, yeniden örnekleme sürecinin dağılımın şekline zarar vermeden çalıştığı ve modelleme aşamasında gerekli olan temsil gücünü koruduğu söylenebilir.



Şekil 6.2. Churn Veri Seti için KDE Kıyaslamaları

Şekil 6.3, Adult veri setindeki sayısal özniteliklerin ortak dağılımlarını göstermektedir. Sol tarafta orijinal veri kümesinin; sağ tarafta ise KLSampler uygulanmış veri setinin KDE yoğunluk haritaları yer almaktadır.

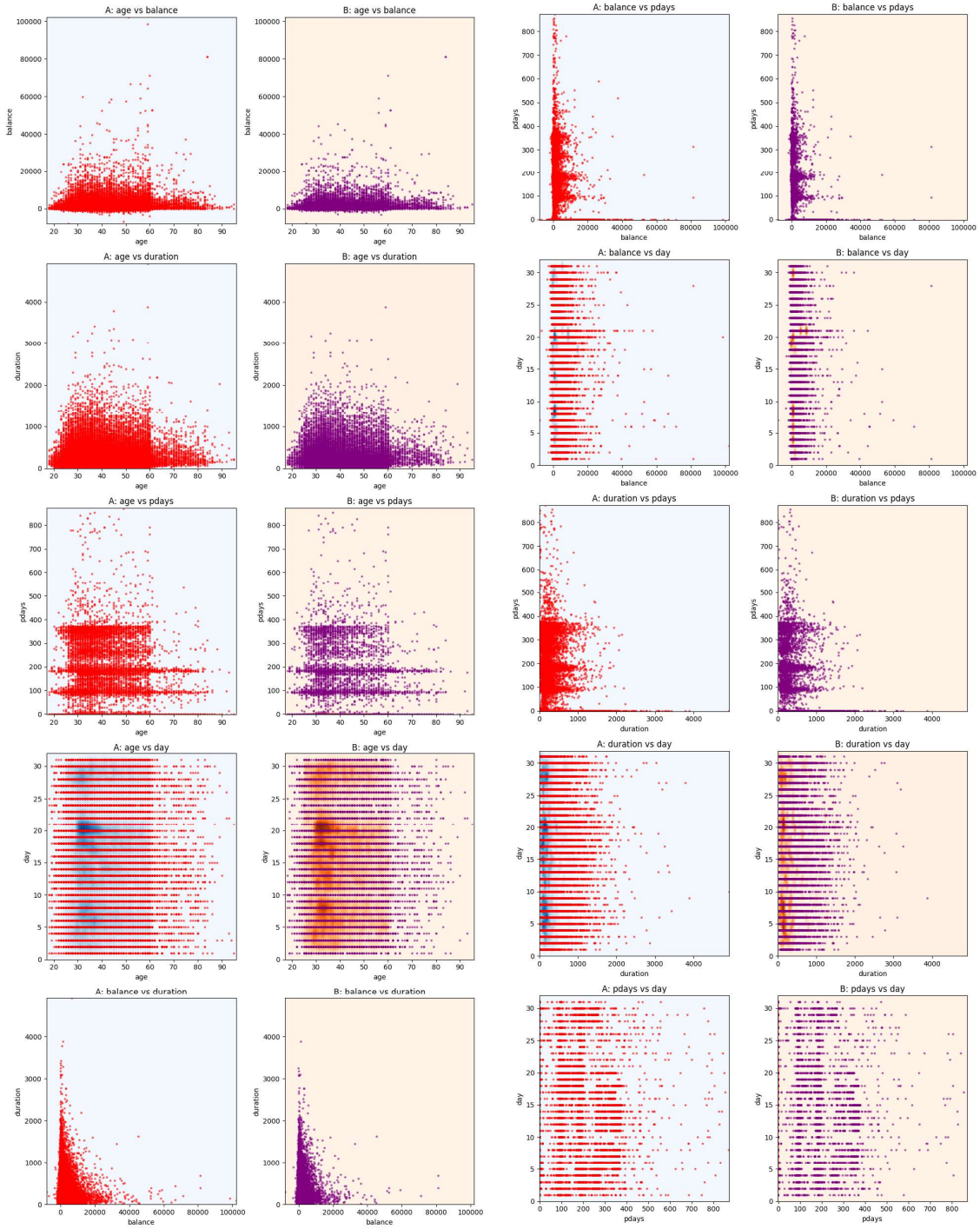


Şekil 6.3. Adult Veri Seti için KDE Kıyaslamaları

Özellikle age-fnlwgt, education_num-capital_gain ve fnlwgt-capital_loss çiftlerinde veri yoğunluğunun belirli bantlarda toplandığı ve bazı değişkenlerde yüksek derecede

çarpıklık (skewness) bulunduğu gözlemlenmektedir.

Capital-gain ve capital-loss değişkenlerinde yoğunluğun sıfır çevresinde toplanması, veri setindeki seyrek ama yüksek değerli örneklerin varlığına işaret etmektedir.



Şekil 6.4. Bank Marketing Veri Seti için KDE Kıyaslamaları

KLSampler sonrası elde edilen dağılımda, bu çarpık yapı korunmakta ve yoğunluk çizgilerinin orijinal veri ile uyumlu kaldığı görülmektedir. Özellikle uç değerlerin tamamen ortadan kaldırılmadığı, yalnızca oransal olarak azaltıldığı dikkat çekmektedir. Bu durum, yöntemin dağılımın kuyruk yapısını da dikkate aldığını ve yalnızca merkez yoğunluk bölgelerini değil tüm istatistiksel formu korumaya çalıştığını göstermektedir.

Şekil 6.4'te Bank Marketing veri setine ait öznitelik çiftlerinin yoğunluk dağılımları sunulmaktadır. Orijinal veri kümesinde özellikle age-balance, duration-pdays ve balance-duration çiftlerinde belirgin kümelenmeler ve asimetrik yoğunluk yapıları görülmektedir.

Veri setinde özellikle balance değişkeninin uzun kuyruklu bir dağılıma sahip olduğu ve yüksek değerli az sayıda örneğin bulunduğu gözlemlenmektedir. Rastgele alt örnekleme yöntemleri bu uzun kuyruk yapısını bozma riski taşırken, KLSampler uygulanmış veri setinde yoğunluk konturlarının benzer biçimde devam ettiği görülmektedir.

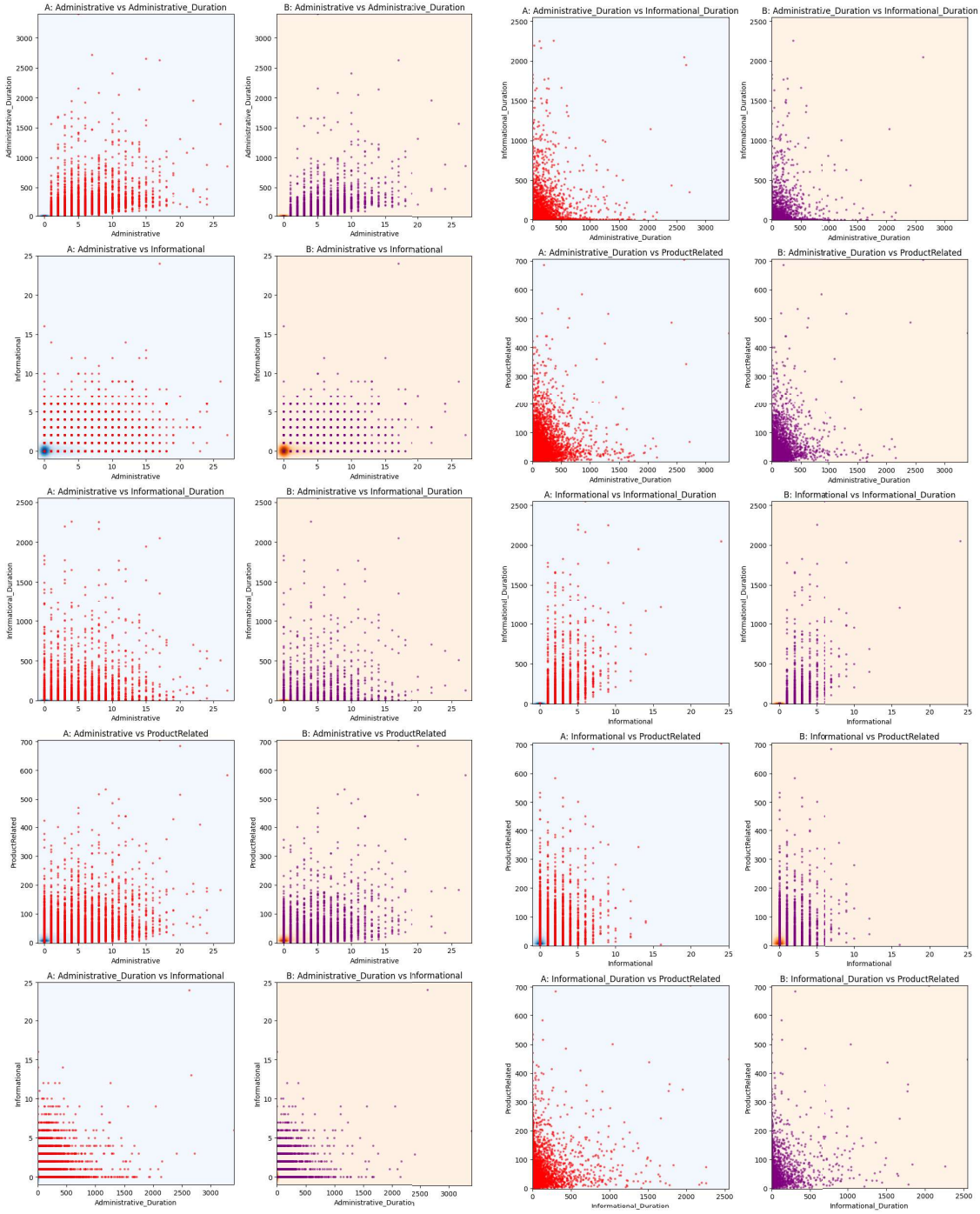
Sağ taraftaki görsellerde dağılımın makro geometrik formu korunmuş; yoğunluk merkezlerinin konumu değişmemiştir. Özellikle duration-pdays düzleminde görülen seyrek yüksek değer bölgeleri korunarak modelin karar sınırlarını etkileyebilecek kritik örüntülerin kaybolmasının önüne geçilmiştir.

Şekil 6.5'te Online Shoppers veri setine ait davranışsal değişken çiftlerinin dağılımları gösterilmektedir. Özellikle Administrative-Administrative_Duration, Informational-Informational_Duration ve ProductRelated-ProductRelated_Duration değişken çiftlerinde belirgin doğrusal ve pozitif korelasyon yapıları gözlemlenmektedir.

Orijinal veri kümesinde yoğunluk, düşük ziyaret sayısı ve düşük süre bölgelerinde toplanmakta; ancak belirli segmentlerde uzun süreli ve yüksek etkileşimli kullanıcı kümeleri de dikkat çekmektedir. Bu yapı, satın alma davranışının modellenmesi açısından kritik örüntüler içermektedir.

KLSampler sonrası elde edilen dağılımda, bu korelasyon yapılarının korunmuş olduğu açıkça görülmektedir. Özellikle yoğunluk merkezlerinin konumu ve kontur şekillerinin benzerliği, yöntemin iki boyutlu ortak dağılımı muhafaza ettiğini göstermektedir. Seyrek

fakat davranışsal olarak önemli olabilecek yüksek süreli oturumların tamamen elimine edilmemesi, modelin genelleme kabiliyeti açısından önem arz etmektedir.



Şekil 6.5. Online Shoppers için KDE Kıyaslamaları

Sonuç olarak, örnekleme sonrası veri hacmi önemli ölçüde azaltılırken, öznitelikler arasındaki yoğunluk yapısının ve dağılımın büyük ölçüde korunduğu gözlemlenmektedir.

Bu durum, KLSampler'ın KL-ayrışımı temelli örnekleme stratejisi sayesinde çoğunluk sınıfının istatistiksel özelliklerini bozmadan örnekleme gerçekleştirdiğini ve yüksek düzeyde dağılımsal sadakat (distributional fidelity) sağladığını göstermektedir. Dolayısıyla, elde edilen bulgular KLSampler'ın sınıf dengesizliğini giderirken veri setinin çok değişkenli (multivariate) dağılım yapısını koruma açısından etkili ve güvenilir bir örnekleme yöntemi sunduğunu doğrulamaktadır.

6.4.2. KLSampler'ın Ağ Saldırı Verisetlerinde Test Edilmesi

Bu bölümde, tasarlanan KLSampler algoritmasının, farklı ağ saldırı tespit sistemlerinde uygulanabilirliğinin denetlenmesi amaçlanmıştır. Bu amaca yönelik olarak, daha önce Bölüm 6.1.'de bahsi geçen 3 farklı veri seti (BoT-IoT, ToN-IoT, ve UNSW-NB15) üzerinde KLSampler çalıştırılmış ve veri setindeki dağılımı koruyacak şekilde bir aşağı örnekleme yapılıp yapılamadığı araştırılmıştır.

İlgili çalışmanın gerçekleştirilmesi amacıyla, bahsi geçen veri setlerinin ilk sürümlerinden farklı bir versiyonu kullanılmıştır. (Sarhan ve diğ., 2022) çalışmasında, üç farklı saldırı veri seti arasında karşılaştırılabilir ve tutarlı bir analiz yapılabilmesi için ortak bir öznitelik uzayı oluşturulması hedeflenmiştir. Bu kapsamda, söz konusu veri setlerinin ham hâlleri, Paket Yakalama (PCAP; Packet Capture) formatında elde edilmiş ve ağ trafiğini akış-temelli olarak temsil eden NetFlow adlı araç kullanılarak CSV formatına dönüştürülmüştür (Cisco, 2025). Bu veri setleri, kamuya açık bir şekilde sunulmuştur (M Sarhan and S Layeghy and M Portmann, 2022).

Bu dönüşüm sayesinde, farklı ağ ortamlarından ve farklı saldırı senaryolarından toplanmış veri setlerinin, aynı öznitelik uzayı içerisinde temsil edilmesi sağlanmıştır. Böylece, veri setleri arasındaki yapısal farklılıkların ve öznitelik uyumsuzluklarının, model performansını veya örnekleme algoritmalarının davranışını yanıltıcı biçimde etkilemesinin önüne geçilmiştir. Özellikle, bu çalışmada kullanılan KLSampler algoritması, sınıflar arası dağılım farklarını KL ayrışımı üzerinden ölçerek örnekleme gerçekleştirdiğinden, karşılaştırılan veri setlerinin aynı öznitelik uzayında tanımlanması değerlendirmenin doğrulu açısından ayrı bir önem arz etmektedir.

Sonuç olarak, farklı ağ ortamlarından elde edilen veri setlerinin ortak bir öznitelik uzayında birleştirilmesi, hem dağılım tabanlı örnekleme yöntemlerinin sağlıklı biçimde uygulanabilmesini hem de elde edilen deneysel sonuçların doğrudan karşılaştırılabilir ve genellenebilir olmasını mümkün kılmıştır.

Netflow kullanılarak üretilmiş olan veri setlerine ait çeşitli istatistikler Tablo 6.1'de daha önce paylaşılmıştır. Bu veri setlerinde herhangi bir çalışma yapmadan önce, çeşitli filtreleme işlemleri uygulanmıştır. Öncelikli olarak veri setlerinden BoT-IoT ve ToN-IoT, sırasıyla 37.763.497 ve 16.940.496 kayıt içermektedir. Disk içerisinde BoT-IoT 6GB yer kaplarken, ToN-IoT veri seti ise yaklaşık olarak 2.86GB'lık bir yer işgal eder. Bu veri setlerinin doğrudan kullanılması 2 farklı açıdan pratik kullanımda problemlere neden olabilir. İlk olarak, veri setleri oldukça büyük boyutlara sahip olduklarından dolayı, özellikle klasik makine öğrenmesi modellerinde eğitim sürelerinin uzamasına hatta çeşitli bellek (memory) hatalarına yol açarak model eğitiminin gerçekleştirilememsine yol açabilir (Sun ve diğ., 2009). İkinci bir değerlendirme noktası ise, halihazırda bu veri setlerini kullanan literatür çalışmalarında da genellikle, verisetlerini oluşturan çalışmalar tarafından yayınlanan ve verisetlerinin %5'ini içeren bir alt küme ile çalışmaktadırlar. Dolayısıyla buradaki en büyük veri uzayına gerek duyulmadığı için, Netflow'dan alınan büyük veri, daha birincil makalelerde paylaşılan alt kümeye indirgenmiştir. Bunun sonucunda da, BoT-IoT bellek boyutu, diskte 2.13 GB'a düşürülürken, bu değer ToN-IoT'de 84.2 MB'a düşürülmüştür. Bu sayede veri setleri hem diğer çalışmalarda kullanılan kayıtlara indirgenirken, hem de daha yönetilebilir boyutlara getirilmiştir.

Son olarak ise burada ortak saldırı tipleri tespit edilerek, 3 veri setinden en az ikisinde yer alan saldırı tiplerinin ele alınmasına karar verilmiştir. Bunun temel sebebi, Bölüm 7'de detaylı olarak paylaşılmış olan, farklı veri setlerinin birleştirilerek daha global bir saldırı tespit sisteminin oluşturulmasına ilişkin gerçekleştirlen çalışmalar için temel oluşturmasıdır. Burada aynı tipte saldırı türlerini, tüm veri setlerinde alarak, bu veri setlerinin çeşitli kombinasyonları ile eğitim ve test aşamaları gerçekleştirildiği için, uygun bir şekilde test süreçlerinin çalıştırılması ve model değerlendirmesinin mümkün kılınması bu şekilde mümkündür.

Buna istinaden, Normal/Zararsız, Dos, DDoS ve Reconnaissance saldırı tipleri seçilmiştir. Bu saldırı türleri, en az 3 veri setinin ikisinde yer aldığından, her türlü eğitim-test kombinasyonlarında, bu saldırı türleri test edilebilmektedir. Sonuç olarak, filtrelenmiş veri setlerine ait temel saldırı dağılımları Tablo 6.3'te gösterilmiştir.

Tablo 6.3. Seçili saldırı sınıflarının veri setlerine göre dağılımı

Saldırı Sınıfı	**BoT-IoT**	**ToN-IoT**	**UNSW-NB15**
Normal/Zararsız	135037 (%1,00)	57614 (%18,30)	2295222 (%99,20)
DoS	4514065 (%33,37)	21384 (%6,79)	5794 (%0,25)
DDoS	8815828 (%65,16)	235740 (%74,90)	-
Keşif (Reconnaissance)	62504 (%0,46)	-	12779 (%0,5)
Toplam Örnek	**13527434**	**314738**	**2313795**

Tablo 6.3 incelendiğinde, KLSampler uygulanmadan önce veri setlerinin saldırı sınıfları açısından ciddi bir dengesizlik içerdiği açıkça görülmektedir. Özellikle BoT-IoT veri setinde DDoS saldırıları toplam örneklerin %65,16'sını (8.815.828 örnek) oluştururken, DoS sınıfı %33,37 oranında temsil edilmekte ve Keşif saldırıları yalnızca %0,46 seviyesinde kalmaktadır. Benzer şekilde ToN-IoT veri setinde DDoS saldırıları %74,90 gibi son derece baskın bir oranla veri setine hâkimken, DoS sınıfı %6,79 ve Keşif saldırıları ise ihmal edilebilir düzeydedir. UNSW-NB15 veri seti ise bu durumun tersine, %99,20 oranında Normal/Zararsız trafik içermekte ve saldırı sınıfları veri seti içerisinde son derece sınırlı bir yer tutmaktadır. Bu tür aşırı dengesiz dağılımlar, makine öğrenmesi tabanlı saldırı tespit modellerinin çoğunluk sınıflarına aşırı uyum sağlamasına ve azınlık saldırı sınıflarını yeterince öğrenememesine yol açabilir.

Buna istinaden, verilen veri setleri üzerinde KLSampler algoritması uygulanmıştır. KLSampler için Algoritma 6'da tanımlanan parametreler aşağıdaki şekilde algoritmaya girdi olarak verilmiştir:

- $target_kl = 0,02$
- $incremet = 5000$
- $max_iters = 20$
- $intial_samples(BoT_IoT) = 135037$

- $intial_samples(ToN_IoT) = 57614$
- $intial_samples(UNSW - NB15) = 12779$

Burada *initial_samples* parametresi özellikle, azınlık sınıfların içerisinde maksimum örneklem sayısına eşitlenmiştir. Bunun sebebi, halihazırda çok fazla olan bir çoğunluk sınıfının, çok az örneklem seçerek, gereksiz yere hedef KL-ayrışım değerine olan uzaklığını arttırarak, daha fazla iterasyon yapılmasını önlemektir.

Algoritma 7'de kullanılan ve başlangıç örneklemelerini oluşturmak için yapılan k-Ortalamalar algoritması yerine bu deney çalışmasında, Mini-Yığın K-Ortalamalar (Mini-Batch K-Means) algoritması kullanılmıştır (Sculley, 2010). K-Ortalamalar, tüm veri noktalarını kullanarak kümeleme gerçekleştirdiğinden, hesaplama maliyeti oldukça yüksektir. Burada, mini-yığın k-ortalamalar kullanıldığında, veri setinin tümü seçilmek yerine, belirli bir kısmı üzerinden bir kümeleme çalıştırılır ve bu da maliyeti düşürür (Scikit-Learn, 2025). Dolayısyıla burada 10000 örneklem seçilerek bir kümeleme yapılarak, başlangıç örneklem kümesinin daha hızlı hesaplanması sağlanmıştır.

Ek olarak, KDE hesaplaması, tüm veri noktalarını kullanarak yapılması maliyetli olduğu için *grid_size* isimli ek bir parametre eklenmiştir. Burada, KDE hesaplamasını tüm veri noktalarına bakarak yapmak yerine, belirli sayıda noktaya bakarak hesaplamaya yaramaktadır. Bu ise, algoritmanın KDE hesaplama maliyetini düşürerek, daha hızlı bir yakınsamaya olanak sağlar.

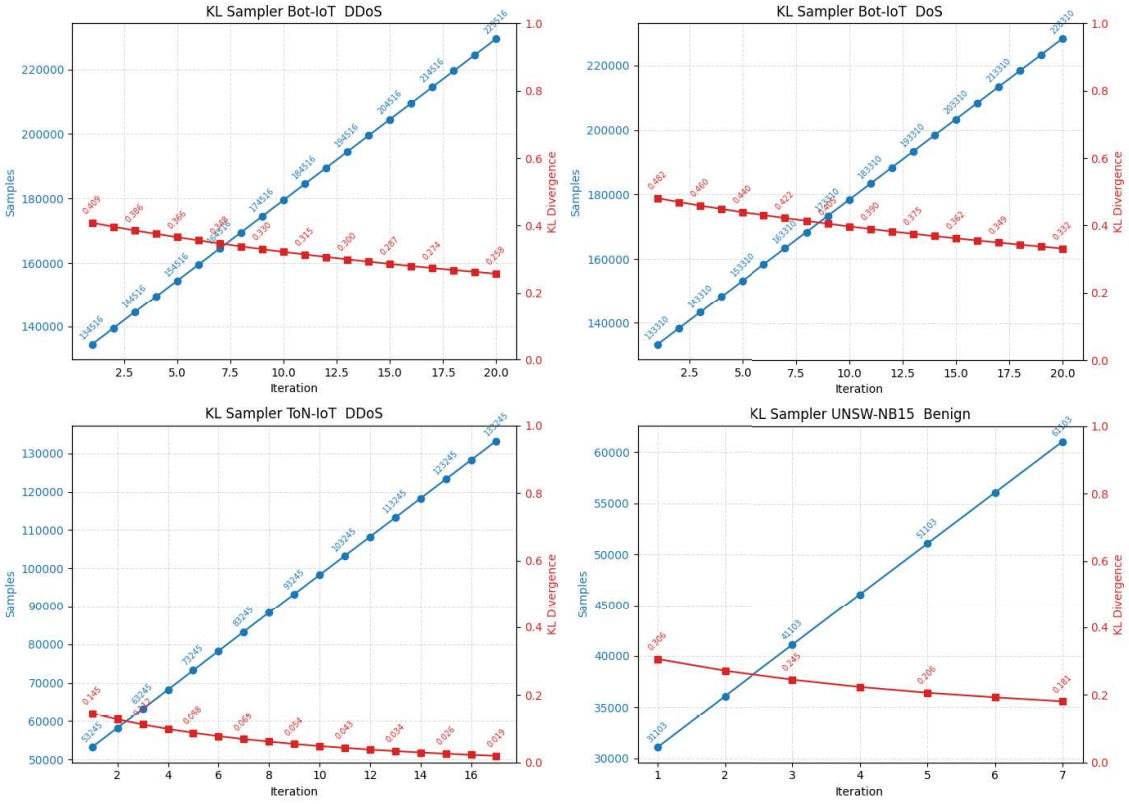
Yapılan güncellemeler sonucunda, KLSampler algoritmasının uygulanması sonucunda elde edilen sınıf dağılımları Tablo 6.4'te gösterilmiştir.

Tablo 6.4. KLSampler Sonrası Sınıf Dağılımları

Saldırı Sınıfı	**BoT-IoT**	**ToN-IoT**	**UNSW-NB15**
Normal/Zararsız	135037 (%20,61)	57614 (%27,14)	61103 (%76,68)
DoS	228310 (%34,83)	21384 (%10,07)	5794 (%7,20)
DDoS	229516 (%35,03)	133245 (%62,77)	–
Keşif (Reconnaissance)	62504 (%9,54)	-	12779 (%16,03)
Toplam Örnek	**655367**	**212243**	**79676**

KLSampler sonrası elde edilen dağılımlar incelendiğinde ise, yöntemin sınıf baskınlığını etkili bir biçimde azalttığı ve veri setlerini daha dengeli hale getirdiği net olarak görülmektedir. BoT-IoT veri setinde DDoS ve DoS sınıflarının oranları sırasıyla %35,03 ve %34,83 seviyelerine çekilirken, Normal/Zararsız trafik %20,61 ve Keşif saldırıları %9,54 oranına yükselmiştir. ToN-IoT veri setinde DDoS oranı %74,90'dan %62,77'ye düşürülmüş, DoS sınıfı %10,07 seviyesine taşınmış ve Normal/Zararsız trafik %27,14 oranında temsil edilir hale gelmiştir. UNSW-NB15 veri setinde ise Normal/Zararsız sınıfın oranı %99,20'den %76,68'e düşürülerek saldırı sınıflarının göreli temsili belirgin biçimde artırılmış, özellikle Keşif saldırılarının oranı %16,03'e ulaşmıştır. Buradaki sonuçlar KLSampler'ın örnek sayısını daha dengeli bir hale getirdiğini göstermektedir.

Bu sonuçlar, KLSampler'ın yalnızca örnek sayısını azaltmakla kalmayıp, sınıflar arası dağılımı istatistiksel olarak daha dengeli hale getirdiğini kontrol etmek için, KLSampler algoritmasına ilişkin, veri setin-saldırı çiftlerine ait, zaman ve veri boyutunu gösteren grafik, Şekil 6.6'da gösterilmiştir.



Şekil 6.6. KLSampler İterasyon Sonuçları

Şekil 6.6'da sunulan sonuçlar, KLSampler algoritmasının farklı veri seti–saldırı türü çiftlerinde iteratif olarak nasıl kararlı ve kontrollü bir biçimde yakınsadığını açık biçimde göstermektedir. BoT-IoT veri setinde DDoS saldırıları için başlangıçta 135.037 olan çoğunluk sınıfı örnek sayısı, 20 iterasyon sonunda 229.516 seviyesine çıkarılırken, KL ayrışımı değeri 0,40875'ten 0,25763'e düşürülmüştür. Bu süreç toplamda yaklaşık 5937 saniye (yaklaşık 99 dakika) sürmüş olup, her iterasyonda KL değerinin monoton biçimde azalması, dağılımın hedef referans dağılıma istikrarlı şekilde yaklaştığını göstermektedir. Benzer bir davranış BoT-IoT DoS sınıfı için de gözlemlenmiş; 20 iterasyon sonunda KL ayrışımı 0,48153'ten 0,33167'ye düşerken, toplam örnekleme süresi 4922 saniye (yaklaşık 82 dakika) olarak ölçülmüştür.

ToN-IoT veri setinde DDoS saldırıları için KLSampler çok daha hızlı bir yakınsama sergilemiştir. Başlangıçta 57.614 örnek ile başlatılan süreç, 17 iterasyon sonunda 133.245 örnek seviyesine ulaşmış ve KL ayrışımı 0,14505'ten 0,01945'e kadar düşürülmüştür. Hedef KL eşiğine erken ulaşılması nedeniyle süreç otomatik olarak durdurulmuş ve toplam örnekleme süresi yalnızca 453,57 saniye (yaklaşık 7,5 dakika) olarak kaydedilmiştir. Bu durum, ToN-IoT veri setinde DDoS sınıfının dağılımsal olarak daha homojen olduğunu ve KLSampler'ın bu tür yapılarda çok daha hızlı ve etkili çalıştığını göstermektedir. UNSW-NB15 veri setinde Benign sınıfı için yürütülen örneklemede ise 7 iterasyon sonunda KL ayrışımı 0,30604'ten 0,18082'ye düşürülmüş ve hedef eşik sağlanarak işlem 1356 saniye (yaklaşık 22,6 dakika) içinde tamamlanmıştır. Bu sonuç, aşırı Normal/Zarasız dominant veri setlerinde dahi KLSampler'ın sınıf baskınlığını kontrollü biçimde azaltabildiğini ortaya koymaktadır.

6.5. Sonuçlar

KLSampler'ın iteratif yapısı, her adımda örnek sayısını ayarlarken dağılımsal bozulmayı sürekli ölçüp KL ayrışımını sistematik olarak düşürmekte; yüksek hacimli saldırı sınıflarında daha uzun süreler gerekse de dağılım bütünlüğünü koruyan, kontrollü bir yeniden dengeleme sağlayarak heterojen ve ciddi sınıf dengesizliği içeren IoT saldırı tespit veri setleri için güvenilir bir örnekleme yaklaşımı sunmaktadır.

7. GLOBAL BİR IOT-IDS SİSTEMİ OLUŞTURMA ÇALIŞMALARI

Bölüm 4 ve Bölüm 4.7.'de, ToN-IoT veri setinde IoT-IDS sistemleri oluşturulmuş ve başarımları incelenmiştir. Bu bölümlerde, ToN-IoT veri seti üzerinde çeşitli önişleme aşamaları gerçekleştirilmiş ve tasarlanan öznitelik çıkarma algoritmaları ile modeller kurularak, uygun veri önişleme teknikleri ve öznitelik çıkarma algoritmaları ile etkin, basit ve derin öğrenme modelleri ile yarışabilecek sonuçlar elde edilmiştir. Ancak hem bu bölümlerde oluşturulan modeller hem de Bölüm 2.4.1. ve Bölüm 3.1.'de belirtilen modeller, bir veri seti seçerek, hem eğitim hem de test aşamalarını aynı veri setinde yürütmüşlerdir. Bu bölümde farklı veri setleri kullanılarak, farklı ortamlarda kullanılabilecek daha global bir IoT-IDS üretilebilir mi sorusuna cevap aranacaktır. Temel amacımız, farklı veri setlerini bir arada kullanarak, modellerin farklı ağ desenlerini öğrenebilmesi ve tamamen bağımsız farklı bir ağ içerisinde ne şekilde sonuçlar vereceğini incelemektir. Bu bölümde, literatürde oldukça az olan bu türden bir deney dizaynının ne ölçüde işe yarayabileceği tartışılmıştır.

7.1. Farklı Veri Setlerinin Standardizasyonu

İlgili çalışmanın gerçekleştirilebilmesi için, doğal olarak farklı veri setlerinin bir arada kullanımına ihtiyaç duyulmaktadır. Genel olarak, farklı veri setleri, farklı test ortamları ve farklı ağ yakalama araçları ile dizayn edildiğinden dolayı, bütünleşik bir öznitelik seti bu veri setlerinin ham hallerinden elde edilememektedir. Burada, Sarhan ve diğ. (2022) tarafından yapılan çalışmada, 4 farklı veri setinin aynı öznitelik seti ile temsil edilebilmesine dair bir çalışma gerçekleştirilmiştir. (aynı yer), paket yakalama (PCAP; Packet Capture) formatında aldıkları farklı veri setlerinden, NetFlow (Cisco, 2025) adlı ağ analiz aracını kullanarak, aynı öznitelik uzayına sahip olacak şekilde, CSV dosyalarına dönüştürmüşlerdir (M Sarhan and S Layeghy and M Portmann, 2022). Bu veri setlerinden, ToN-IoT, BoT-IoT ve UNSW-NB15 veri setleri indirilerek, orjinal veri sağlayıcılarının sundukları, tüm verinin %5'ini içeren alt kümeye eşlenerek, veri boyutunun azaltılması sağlanmıştır. Bu aşamayı takiben, Bölüm 6.2.'da belirtilen KL-Ayrışımına dayalı yeniden örnekleme algoritması kullanılarak veri dengesizliği giderilerek, bu bölümde ileriye taşınacak tartışmanın yapılacağı ana veri setleri oluşturulmuştur. Veri setlerinin

indirilmesi, orjinal veri seti sağlayıcılarının sunduğu %5'lik veriye göre filtrelenmesi, ve analize hazır hale getirmek amacıyla uygulanan KLSampler algoritmasının adım adım gerçekleştirilme detayları, Bölüm 6.4.2.'de sunulmuştur.

Sonraki bölümlerde, aynı öznitelik uzayına sahip, veri dengesizlik problemi giderilmiş olan, BoT-IoT, ToN-IoT ve UNSW-NB15 üzerinden, ileri analizler ve çeşitli deney kombinasyonları oluşturularak, global bir saldırı tespit sistemi inşa edilebilir mi, araştırılacaktır.

7.2. Veri Ön İşleme Aşamaları

Veri ön işleme aşamalarına başlamadan önce, kullanılacak olan veri setlerine ait, Netflow aracı tarafından çıkarılmış, özniteliklerin listesi, Tablo 7.1'de sunulmuştur.

Bu çalışmada kullanılan öznitelikler içerisinde yer alan bazı ağ ve protokol terimleri, kısaltmalarıyla birlikte belirli teknik kavramlara karşılık gelmektedir. Bu kavramların doğru yorumlanabilmesi açısından aşağıda kısaca açıklanmıştır.

Alan Adı Sistemi (Domain Name System, DNS), alan adlarının IP adreslerine çevrilmesini sağlayan temel bir ağ servisidir. DNS'e ait öznitelikler; sorgu türleri, zaman aşımı değerleri ve yanıt kodları üzerinden ağ keşfi (scanning), bilgi toplama ve yönlendirme temelli saldırıların tespitinde önemli rol oynamaktadır.

Yaşam Süresi (Time To Live, TTL), bir IP paketinin ağ üzerinde yönlendirilebileceği maksimum atlama (hop) sayısını ifade eder. TTL değerlerindeki anormal değişimler, paket manipülasyonu, yönlendirme anomalileri veya saldırı faaliyetlerinin göstergesi olabilmektedir.

İnternet Denetim Mesaj Protokolü (Internet Control Message Protocol, ICMP), ağ cihazları arasında hata bildirimleri ve kontrol mesajlarının iletilmesi amacıyla kullanılan bir protokoldür. ICMP türü ve kod bilgileri, özellikle ağ keşfi, erişilebilirlik testleri ve servis dışı bırakma girişimlerinin ayırt edilmesinde kullanılmaktadır.

İletim Kontrol Protokolü (Transmission Control Protocol, TCP), bağlantı tabanlı ve

Tablo 7.1. NetFlow v2 Öznitelikleri

Öznitelik(ler)	**Açıklama**
IPV4_SRC_ADDR IPV4_DST_ADDR	IPv4 kaynak ve hedef adresleri. Akışı başlatan ve hedeflenen düğümlerin IP adreslerini ifade eder.
L4_SRC_PORT L4_DST_PORT	Kaynak ve hedef port numaraları (TCP/UDP).
PROTOCOL L7_PROTO	IP protokol tanımlayıcı baytı ve uygulama katmanı protokol kimliği.
IN_BYTES OUT_BYTES	Kaynak–hedef ve hedef–kaynak yönünde iletilen toplam bayt miktarı.
IN_PKTS OUT_PKTS	Kaynak–hedef ve hedef–kaynak yönünde iletilen toplam paket sayısı.
FLOW_DURATION_MILLISECONDS	Akışın toplam süresi (milisaniye).
DURATION_IN DURATION_OUT	İstemci–sunucu ve sunucu–istemci yönündeki akış süreleri (ms).
TCP_FLAGS CLIENT_TCP_FLAGS SERVER_TCP_FLAGS	Akış boyunca gözlemlenen TCP bayrakları (genel, istemci ve sunucu tarafı).
MIN_TTL MAX_TTL	Akışta gözlemlenen minimum ve maksimum TTL değerleri.
LONGEST_FLOW_PKT SHORTEST_FLOW_PKT	Akış içindeki en büyük ve en küçük paket boyutları (bayt).
MIN_IP_PKT_LEN MAX_IP_PKT_LEN	Gözlemlenen minimum ve maksimum IP paket uzunlukları.
SRC_TO_DST_SECOND_BYTES DST_TO_SRC_SECOND_BYTES	Yön bazlı saniye başına iletilen bayt miktarı.
SRC_TO_DST_AVG_THROUGHPUT DST_TO_SRC_AVG_THROUGHPUT	Yön bazlı ortalama bant genişliği (bps).
NUM_PKTS_UP_TO_128_BYTES NUM_PKTS_128_TO_256_BYTES NUM_PKTS_256_TO_512_BYTES NUM_PKTS_512_TO_1024_BYTES NUM_PKTS_1024_TO_1514_BYTES	Farklı boyut aralıklarındaki paket sayıları. Trafiğin paket boyutu dağılımını karakterize eder.

güvenilir veri iletimini sağlayan bir taşıma katmanı protokolüdür. TCP bayrakları (flags), pencere boyutları ve yeniden iletim bilgileri; bağlantı durumu, ağ tıkanıklığı ve anormal iletişim davranışlarının tespit edilmesinde kritik öneme sahiptir.

Kullanıcı Datagram Protokolü (User Datagram Protocol, UDP) ise bağlantısız ve düşük gecikmeli veri iletimi sunan bir taşıma katmanı protokolüdür. UDP tabanlı akışlar özellikle yüksek hacimli ve hızlı saldırılar olan DoS ve DDoS senaryolarında yaygın

olarak kullanılmaktadır.

Uygulama Katmanı Protokolleri (Layer 7 Protocols, L7), HTTP, FTP ve DNS gibi doğrudan uygulama seviyesinde çalışan protokolleri ifade etmektedir. Bu protokollere ait öznitelikler, uygulama tabanlı saldırıların ve servis kötüye kullanımının tespitinde önemli ayırt edici bilgiler sunmaktadır.

Bu kavramların birlikte değerlendirilmesi, ağ akışlarının hem yapısal hem de davranışsal açıdan analiz edilmesine olanak tanımakta ve saldırı türlerinin daha yüksek doğrulukla sınıflandırılmasını destekleyecek bilgiler barındırmaktadır.

7.2.1. Sabit ve Kısmi-Sabit Öznitelikler

Veri ön işleminin ilk adımı olarak, sabit ve kısmi-sabit özniteliklerin öznitelik listesinden çıkarılması ile ön işleme adımına başlanmıştır. Bu türden öznitelikler, tamamen 0 varyansa sahip olup hiç bilgi taşımayacakları gibi, çok düşük varyansa sahip olup ayırt edici bilgi taşımazlar, ve hem modelin daha fazla özniteliği öğrenmesine neden olarak maliyetinin artmasına yol açarken, aynı zamanda herhangi bir bilgi taşımadıklarından dolayı model başarımına pozitif bir etki yapmamaktadırlar. Buna istinaden, ToN_IoT, BoT_IoT ve UNSW-NB15 veri setlerin üzerinden, %95 eşik değeri seçilerek, bu eşik değerinin üzerinde aynı değere sahip olan öznitelikler sistemden çıkartılmıştır.

Bu öznitelikler her bir veri seti için şu şekildedir:

- **ToN_IoT**: ICMP_IPV4_TYPE, ICMP_TYPE, RETRANSMITTED_IN_BYTES, RETRANSMITTED_IN_PKTS
- **BoT_IoT**: RETRANSMITTED_OUT_BYTES, RETRANSMITTED_OUT_PKTS, DNS_QUERY_ID, DNS_QUERY_TYPE, DNS_TTL_ANSWER, NUM_PKTS_256_ TO_512_BYTES, NUM_PKTS_512_TO_1024_BYTES, NUM_PKTS_1024_ TO_1514_BYTES
- **UNSW_NB15**: DNS_TTL_ANSWER

Yukarıdaki listede geçen tüm öznitelikler, belirtilen üç veri setinden de çıkartılmıştır.

7.2.2. Uygun Saldırı Grubunun Seçilmesi

Oluşturulacak olan IoT-IDS sisteminde, farklı veri setleri kullanılacağı için, her bir veri seti test ortamına göre farklı saldırı gruplarına sahip olacaktır. Kullanılacak üç veri setine ait saldırı türleri ve sayıları Tablo 7.2'de verilmiştir.

Tablo 7.2. Veri Setlerinde Saldırı Türlerinin Dağılımı

Saldırı Tipi	**Bot-IoT**	**ToN-IoT**	**UNSW-NB15**
Normal (Benign)	135.037 (0,36%)	6.099.469 (36,0%)	2.295.222 (96,02%)
Hizmet Engelleme (DoS)	16.673.183 (44,14%)	712.609 (4,21%)	5794 (0,24%)
Dağıtık Hizmet Engelleme (DDoS)	18.331.847 (48,56%)	2.026.234 (11,96%)	–
Keşif / Bilgi Toplama (Recon.)	2.620.999 (6,94%)	–	12.779 (0,53%)
Arka Kapı (Backdoor)	–	16.809 (0,10%)	2169 (0,09%)
Kod Enjeksiyonu (Injection)	–	684.465 (4,04%)	–
Tarama (Scanning)	–	3.781.419 (22,32%)	–
Parola Saldırıları (Password)	–	1.153.323 (6,81%)	–
Fidye Yazılımı (Ransomware)	–	3425 (0,02%)	–
Ortadaki Adam (MITM)	–	7723 (0,05%)	–
Siteler Arası Betik (XSS)	–	2.455.020 (14,49%)	–
Açık İstismarı (Exploits)	–	–	31.551 (1,32%)
Bulanıklaştırma (Fuzzers)	–	–	22.310 (0,93%)
Genel Amaçlı (Generic)	–	–	16.560 (0,69%)
Analiz Tabanlı (Analysis)	–	–	2299 (0,10%)
Kabuk Kodu (Shellcode)	–	–	1427 (0,06%)
Solucanlar (Worms)	–	–	164 (0,01%)
Hırsızlık (Theft)	2431 (0,01%)	–	–
Toplam	**37.763.497**	**16.940.496**	**2.390.275**

Bütünleşik bir IoT saldırı tespit sistemi tasarlanırken, eğitim ve test aşamalarında kullanılan veri setlerinde yer alan saldırı türlerinin tutarlı olması oldukça önemlidir. Farklı veri setlerinde saldırı sınıflarının örtüşmemesi durumunda, model eğitim aşamasında

öğrenilen sınıf-temelli karar sınırları test aşamasında geçerliliğini yitirmekte ve sistem tahmin edeceği uzayda farklı bir problem tanımı altında değerlendirilmektedir. Bu durum, özellikle birden fazla IDS modelinin birleştirildiği bütünleşik yapılarda karar uzaylarının uyumsuzluğuna yol açar ve oylama (voting)/birleştirme (stacking) mekanizmalarının anlamsızlaşmasına neden olur. Ayrıca, sınıf bazlı performans metriklerinin (örneğin doğruluk, F1-skoru gibi) güvenilirliğine zarar vererek, elde edilen sonuçların genelleme başarısını yansıtmasını engeller. Dolayısıyla, saldırı türlerinin tüm eğitim ve test veri setlerinde aynı şekilde tanımlanması, bütünleşik IDS mimarisinin tutarlı, karşılaştırılabilir ve bilimsel olarak geçerli bir şekilde değerlendirilmesi açısından zorunludur.

Bu kapsamda incelendiğinde 3 veri seti için ortak ve adil bir değerlendirmeyi sağlayacak saldırı türleri şu şekilde seçilmiştir: Normal (Benign), DoS, DDoS, Keşif/Bilgi Toplama (Reconnaissance).

7.2.3. Veri Setleri Arası Öznitelik Yayılımı

Nesnelerin İnterneti (IoT) tabanlı saldırı tespit sistemlerinde geliştirilen makine öğrenmesi modellerinin başarımı, büyük ölçüde kullanılan veri setlerinin istatistiksel özelliklerine ve bu veri setlerinde yer alan özniteliklerin dağılım yapısına bağlıdır. Literatürde yaygın olarak kullanılan BoT-IoT, ToN-IoT ve UNSW-NB15 gibi veri setleri, farklı ağ ortamları, saldırı senaryoları ve veri toplama mekanizmaları kullanılarak oluşturulmaktadır. Bu durum, her bir veri setinin kendine özgü öznitelik dağılımlarına ve baskın değer yapılarına sahip olmasına yol açabilir. Ancak, pek çok çalışmada bu veri setleri birbirinden bağımsız olarak ele alınmakta ve öznitelik düzeyinde veri setleri arası tutarlılık yeterince incelenmemektedir.

Makine öğrenmesi tabanlı saldırı tespit modelleri, eğitim sürecinde veri setlerinde en sık gözlemlenen öznitelik değerlerine karşı yüksek duyarlılık geliştirebilmekte ve bu baskın değerleri karar mekanizmalarının temel belirleyicileri haline getirebilmektedir. Bu durum, modelin gerçek saldırı davranışlarını öğrenmesinden ziyade, belirli bir veri setine özgü örüntüleri veya yanlılıkları öğrenmesine neden olabilmektedir. Sonuç olarak, eğitim veri seti üzerinde yüksek doğruluk elde eden modeller, farklı veri setleri veya gerçek dünya ağ trafiği üzerinde ciddi genelleme problemleri yaşayabilmektedir.

Bu çalışmada ele alınan veri setleri arası öznitelik değeri yayılımı analizi, söz konusu genelleme problemlerinin temel nedenlerini öznitelik ve öznitelik değeri düzeyinde incelemeyi amaçlamaktadır. Bu kapsamda, her bir veri setinde en sık gözlemlenen (baskın) öznitelik değerleri belirlenmekte ve bu değerlerin diğer veri setlerinde hangi sıklıkta ve hangi sınıf dağılımlarıyla ortaya çıktığı analiz edilmektedir. Böylece, belirli bir veri setinde saldırı veya normal trafik ile güçlü biçimde ilişkilendirilen bir öznitelik değerinin, farklı veri setlerinde benzer bir anlamsal karşılığa sahip olup olmadığı nicel olarak değerlendirilebilmektedir. Önerilen analiz yaklaşımı, veri setine özgü öznitelik değerlerinin tespit edilmesine, potansiyel yanlılık kaynaklarının ortaya çıkarılmasına ve modellerin hangi özniteliklere aşırı bağımlılık geliştirdiğinin anlaşılmasına olanak sağlamaktadır. Ayrıca, bu analiz sayesinde veri setleri arasında tutarlı biçimde yayılım gösteren, diğer bir deyişle veri setinden bağımsız olarak benzer davranış sergileyen öznitelik değerleri belirlenebilmekte ve bu öznitelikler daha güvenilir modelleme bileşenleri olarak öne çıkarılabilmektedir. Bu yönüyle, veri setleri arası öznitelik değeri yayılım analizi, hem model genelleme kabiliyetinin artırılması hem de daha sağlam ve gerçekçi saldırı tespit sistemlerinin geliştirilmesi açısından kritik bir ön işleme ve değerlendirme adımı sunmaktadır. Veri setleri arasındaki öznitelik değeri yayılım analizinin nasıl yapıldığını gösteren sözde-kod Algoritma 9'da gösterilmiştir.

Bu çalışmada sunulan algoritma, veri setleri arası öznitelik değeri yayılımını incelemek amacıyla tasarlanmıştır. Algoritma, her bir veri setini sırasıyla kaynak veri seti olarak ele almakta ve bu veri setindeki her bir öznitelik için en sık gözlemlenen, diğer bir ifadeyle baskın olan değeri belirlemektedir. Bu baskın değer, ilgili özniteliğin kaynak veri setindeki dağılımını en iyi temsil eden karakteristik bir örüntü olarak kabul edilmektedir. Analiz sürecinde sınıf etiketi sütunu ve önceden tanımlanmış hariç tutulacak öznitelikler dikkate alınmayarak, yalnızca anlamlı öznitelikler üzerinden değerlendirme yapılmaktadır.

Belirlenen baskın öznitelik değeri için, algoritma bu değerin tüm diğer veri setlerinde hangi sıklıkta gözlemlendiğini ve toplam örnekler içerisindeki oranını hesaplamaktadır.

Bu aşama, aynı öznitelik değerinin farklı veri setlerinde ne ölçüde tekrarlandığını ve veri

setleri arasında bir yayılım gösterip göstermediğini ortaya koymaktadır.

Algoritma 9: Veri Setleri Arası Öznitelik Değeri Yayılım Analizi

Gerekli:
$\mathscr{D} = \{D_1, D_2, \ldots, D_M\}$: Veri setleri kümesi
$\mathscr{N} = \{n_1, n_2, \ldots, n_M\}$: Veri seti adları
y: Sınıf etiketi sütunu ($y \in \{0,1\}$)
E: Hariç tutulacak öznitelikler kümesi
Ensure:
R: Veri setleri arası öznitelik değeri yayılım sonuçları
Başlatma:
$R \leftarrow \emptyset$
Her veri setini kaynak olarak ele al:
for each $(D_s, n_s) \in (\mathscr{D}, \mathscr{N})$ **do**
    for each feature $f \in \text{columns}(D_s)$ **do**
        if $f \in E$ **or** $f = y$ **then**
            continue
        end if
        Kaynak veri setinde baskın değeri belirle:
        $v_f \leftarrow \arg\max_v \text{count}_{D_s}(f = v)$
        Baskın değerin diğer veri setlerindeki yayılımını ölç:
        for each $(D_t, n_t) \in (\mathscr{D}, \mathscr{N})$ **do**
            if $f \notin \text{columns}(D_t)$ **then**
                $c \leftarrow 0$, $\phi \leftarrow \text{None}$
                $\rho_0 \leftarrow 0$, $\rho_1 \leftarrow 0$
            else
                $I \leftarrow \{i \mid D_t^{(i)}[f] = v_f\}$
                $c \leftarrow |I|$
                $\phi \leftarrow \frac{c}{|D_t|}$
                if $c > 0$ **then**
                $\rho_0 \leftarrow \frac{|\{i \in I | y_i = 0\}|}{c}$
                $\rho_1 \leftarrow \frac{|\{i \in I | y_i = 1\}|}{c}$
                else
                $\rho_0 \leftarrow 0$, $\rho_1 \leftarrow 0$
                end if
            end if
            $R \leftarrow R \cup \{(n_s, n_t, f, v_f, \phi, c, \rho_0, \rho_1)\}$
        end for
    end for
end for

Eğer ilgili öznitelik hedef veri setinde mevcut değilse, bu durum açıkça işaretlenmekte ve

ilgili metrikler tanımsız olarak ele alınmaktadır. Öznitelik mevcut olduğunda ise, baskın değere sahip örnekler alt küme olarak seçilmekte ve bu alt kümenin büyüklüğü ile göreli sıklığı nicel olarak ifade edilmektedir.

Algoritmanın en kritik aşamalarından biri, baskın öznitelik değerine sahip örneklerin sınıf dağılımının analiz edilmesidir. Bu kapsamda, seçilen alt küme içerisindeki normal ve saldırı örneklerinin oranları ayrı ayrı hesaplanmaktadır.

Böylece, bir veri setinde saldırı veya normal trafik ile güçlü biçimde ilişkilendirilen bir öznitelik değerinin, farklı veri setlerinde benzer bir sınıfsal anlam taşıyıp taşımadığı değerlendirilebilmektedir. Elde edilen bu oranlar, öznitelik değerlerinin veri setine özgü yanlılıklar içerip içermediğinin ve potansiyel olarak kısa-yol öğrenmeye neden olup olmadığının anlaşılmasına doğrudan katkı sağlamaktadır.

Sonuç olarak algoritma, her bir kaynak veri seti, öznitelik ve hedef veri seti üçlüsü için; baskın değer, bu değerin hedef veri setindeki görülme sıklığı ve ilgili sınıf dağılımlarını içeren kapsamlı bir sonuç kümesi üretmektedir. Bu çıktı, veri setleri arasında tutarlı biçimde yayılım gösteren öznitelik değerlerinin yanı sıra, yalnızca belirli veri setlerine özgü kalan ve genelleme açısından risk oluşturan özniteliklerin sistematik olarak tespit edilmesine olanak tanımaktadır. Bu yönüyle önerilen yaklaşım, veri seti yanlılığının analizi ve daha sağlam, genellenebilir saldırı tespit modellerinin geliştirilmesi için önemli bir ön değerlendirme adımı sunmaktadır.

Buna göre öznitelikler ve ilgili analiz sonucunda elde edilen öznitelik bazlı çıkarımlar Tablo 7.3'te sunulmuştur.

Veri setleri arası öznitelik değeri yayılım analizi sonucunda her bir öznitelik için alınan kararlar özetlenmiştir. Bu analiz kapsamında toplam 19 öznitelik, ya etiket ile güçlü ve veri setine özgü bir ilişki göstermesi etiket ile ilişkili ya da hedef veri setlerinde son derece düşük frekansla gözlemlenmesi nedeniyle modelleme sürecinden çıkarılmıştır. Etiketle doğrudan veya dolaylı biçimde ilişkili olan bu özniteliklerin modele dâhil edilmesi, veri setine özgü yanlılıkların öğrenilmesine yol açabileceğinden, bu tür özniteliklerin elenmesi modelin genelleme kabiliyetini artırmak açısından kritik görülmüştür.

Tablo 7.3. Öznitelik Seçim Kararları

Öznitelik	**Karar**
CLIENT_TCP_FLAGS	Güvenli (birleşik)
DNS_QUERY_ID	Etiket ile ilişkili
DNS_QUERY_TYPE	Etiket ile ilişkili
DNS_TTL_ANSWER	Etiket ile ilişkili
DST_TO_SRC_AVG_THROUGHPUT	Düşük frekans
DST_TO_SRC_SECOND_BYTES	Güvenli (birleşik)
DURATION_IN	Güvenli (birleşik)
DURATION_OUT	Etiket ile ilişkili
FLOW_DURATION_MILLISECONDS	Güvenli (birleşik)
ICMP_IPV4_TYPE	Düşük frekans
ICMP_TYPE	Düşük frekans
(IN_ & OUT_) BYTES	Güvenli (birleşik)
(IN_ & OUT_) PKTS	Güvenli (birleşik)
(MIN_ & MAX_) TTL	Etiket ile ilişkili
L7_PROTO	Güvenli (birleşik)
LONGEST_ SHORTEST_ FLOW_PKT	Güvenli (birleşik)
MAX_IP_PKT_LEN	Güvenli (birleşik)
MIN_IP_PKT_LEN	Güvenli (birleşik)
NUM_PKTS_1024_TO_1514_BYTES	Korundu
NUM_PKTS_128_TO_256_BYTES	Etiket ile ilişkili
NUM_PKTS_256_TO_512_BYTES	Etiket ile ilişkili
NUM_PKTS_512_TO_1024_BYTES	Etiket ile ilişkili
NUM_PKTS_UP_TO_128_BYTES	Güvenli (birleşik)
PROTOCOL	Etiket ile ilişkili
RETRANSMITTED_ (IN_ & OUT_) BYTES	Düşük frekans
RETRANSMITTED_(IN_ & OUT_) PKTS	Düşük frekans
SERVER_TCP_FLAGS	Etiket ile ilişkili
SRC_TO_DST_AVG_THROUGHPUT	Güvenli (birleşik)
SRC_TO_DST_SECOND_BYTES	Güvenli (birleşik)
TCP_FLAGS	Güvenli (birleşik)
TCP_WIN_MAX_IN	Güvenli (birleşik)
TCP_WIN_MAX_OUT	Etiket ile ilişkili

Buna ek olarak, Attack, Label, IPV4_SRC_ADDR, IPV4_DST_ADDR, L4_SRC_PORT ve L4_DST_PORT gibi öznitelikler; sınıf etiketi, doğrudan tanımlayıcı bilgiler veya akışa özgü benzersiz kimlikler içermeleri nedeniyle analiz kapsamı dışında bırakılmıştır. Bu öznitelikler hem öznitelik değeri yayılım analizine uygun olmamaları hem de gerçekçi bir saldırı tespit modelinde genellenebilir bilgi taşımamaları sebebiyle modelleme aşamasına

da dahil edilmemiştir. Sonuç olarak, elde edilen nihai öznitelik kümesi, veri setleri arasında daha tutarlı davranış sergileyen, etiket sızıntısı riski göreceli olarak az ve genelleme açısından daha güvenilir olduğu düşünülen özniteliklerden oluşturulmuştur.

7.2.4. Eğitim-Test Kombinasyonlarının Oluşturulması

Bu aşamadan sonra artık farklı deney setlerinin oluşturulması amacıyla veri setleri çeşitli kombinasyonlarla birleştirilmiştir. Önceki bölümlerde, veri setlerinin kendi karakteristiklerinden meydana ve birbirleri arasında birleştirme esnasında meydana gelebilecek problemlerden kaçınmak amacıyla veri setleri birbirlerinden ayrı olarak analiz edilmiştir. Bu aşamadan sonra ise, modelleme aşamasında kullanılacak bütünleşik eğitim ve test setleri oluşturularak gelecek analizler bu veri setleri üzerinden yapılacaktır. Buna istinaden Tablo 7.4'te sunulan deney grupları oluşturulmuştur.

Tablo 7.4. Veri Setleri Arası Eğitim ve Test Senaryoları

Deney Adı	**Eğitim Seti**	**Test Seti**
BoT-ToN Train	BoT_IoT + ToN_IoT	UNSW-NB15
BoT-UNSW Train	BoT_IoT + UNSW-NB15	ToN_IoT
ToN-UNSW Train	ToN_IoT + UNSW-NB15	BoT_IoT

Bu bölümden sonraki her analiz bu veri setleri kombinasyonlarına ayrı ayrı uygulanacaktır.

7.2.5. Çoklu Korelasyon Analizi

Makine öğrenmesi tabanlı modellerin güvenilir, kararlı ve genellenebilir sonuçlar üretebilmesi için, modelleme sürecinin farklı aşamalarında öznitelikler arasındaki ilişkilerin sistematik olarak yeniden değerlendirilmesi gerekmektedir (Liu ve diğ., 2025). Özellikle veri ön işleme ve öznitelik seçimi adımlarında uygulanan korelasyon temelli analizlerin, farklı veri alt kümeleri veya dönüşümler sonrasında tekrarlanması, modelin yapısal tutarlılığının korunması açısından önem taşımaktadır. Bu bağlamda, önceki bölümlerde sunulan korelasyon analizi yaklaşımı, bu aşamada da referans alınarak yeniden uygulanmıştır.

İlgili tez çalışmasının önceki bölümlerinde, Algoritma 2 kullanılarak öznitelikler arasındaki ikili korelasyon ilişkileri analiz edilmiş ve yüksek derecede çoklu korelasyon içeren öznitelik grupları belirlenmiştir. Tezin bu bölümünde ise, ilgili üç veri seti üzerinde aynı metodolojik çerçeve korunarak, öznitelikler arasındaki korelasyon yapısı yeniden incelenmiştir. Böylece, uygulanan veri temizleme ve öznitelik eleme adımlarının, farklı aşamalarda benzer yanlılıklara veya tekrarlı bilgi yapılarına yol açıp açmadığı denetlenmiştir.

Bu kapsamda, öznitelikler arasındaki mutlak korelasyon katsayıları hesaplanmış ve mutlak korelasyon değeri %80'in üzerinde olan öznitelikler, önceki analizle tutarlı biçimde aynı grup içerisinde değerlendirilmiştir. Bu eşik değer, yüksek derecede benzer bilgi taşıyan özniteliklerin belirlenmesi ve modelde gereksiz bilgi tekrarının önüne geçilmesi amacıyla tercih edilmiştir. Böylece, önceki aşamada olduğu gibi, çoklu korelasyonun modele olası olumsuz etkileri kontrol altına alınmıştır.

Oluşturulan her bir öznitelik grubu için, hedef değişken ile olan ilişkilerin göreli önemini değerlendirmek amacıyla yine bir RF modeli kullanılmıştır. RF modeli tarafından üretilen öznitelik önem skorları doğrultusunda, her bir korelasyon grubu içerisinden hedef değişken ile en güçlü ilişkiye sahip öznitelik korunmuş; aynı grup içerisinde yer alan ve bilgi tekrarına neden olabilecek diğer öznitelikler veri setinden çıkarılmıştır. Bu yaklaşım, önceki bölümde sunulan öznitelik eleme stratejisi ile metodolojik olarak birebir örtüşmektedir.

Sonuç olarak, Algoritma 2'de tanımlanan korelasyon temelli öznitelik eleme süreci, bu aşamada da farklı veri setleri üzerinde tekrar uygulanmış; böylece hem öznitelik uzayı sadeleştirilmiş hem de modelin kararlılığı ve genelleme performansının korunması hedeflenmiştir.

Yapılan çoklu korelasyon analizi sonucunda farklı deney kombinasyonlarından çıkarılacak öznitelikler şu şekilde özetlenebilir:

BoT-ToN Eğitimi (BoT_IoT + ToN_IoT):

- ICMP_TYPE

- RETRANSMITTED_IN_PKTS
- MAX_TTL
- IN_BYTES
- IN_PKTS
- OUT_PKTS
- NUM_PKTS_512_TO_1024_BYTES
- NUM_PKTS_1024_TO_1514_BYTES
- SERVER_TCP_FLAGS
- CLIENT_TCP_FLAGS
- RETRANSMITTED_OUT_PKTS

BoT-UNSW Eğitimi (BoT_IoT + UNSW-NB15):

- ICMP_TYPE
- MAX_TTL
- OUT_PKTS
- IN_PKTS
- NUM_PKTS_512_TO_1024_BYTES
- NUM_PKTS_1024_TO_1514_BYTES
- SERVER_TCP_FLAGS
- RETRANSMITTED_IN_PKTS
- RETRANSMITTED_OUT_PKTS
- TCP_WIN_MAX_IN

ToN-UNSW Eğitimi (ToN_IoT + UNSW-NB15):

- ICMP_TYPE
- MAX_TTL
- TCP_FLAGS
- SERVER_TCP_FLAGS
- IN_PKTS
- OUT_PKTS
- NUM_PKTS_256_TO_512_BYTES

- NUM_PKTS_512_TO_1024_BYTES
- NUM_PKTS_1024_TO_1514_BYTES
- DURATION_IN
- TCP_WIN_MAX_IN
- RETRANSMITTED_IN_PKTS
- RETRANSMITTED_OUT_PKTS

Bu aşamayı takiben özniteliklerin tekil analizleri gerçekleştirilecektir.

7.2.6. Özniteliklerin Tekil Analizi

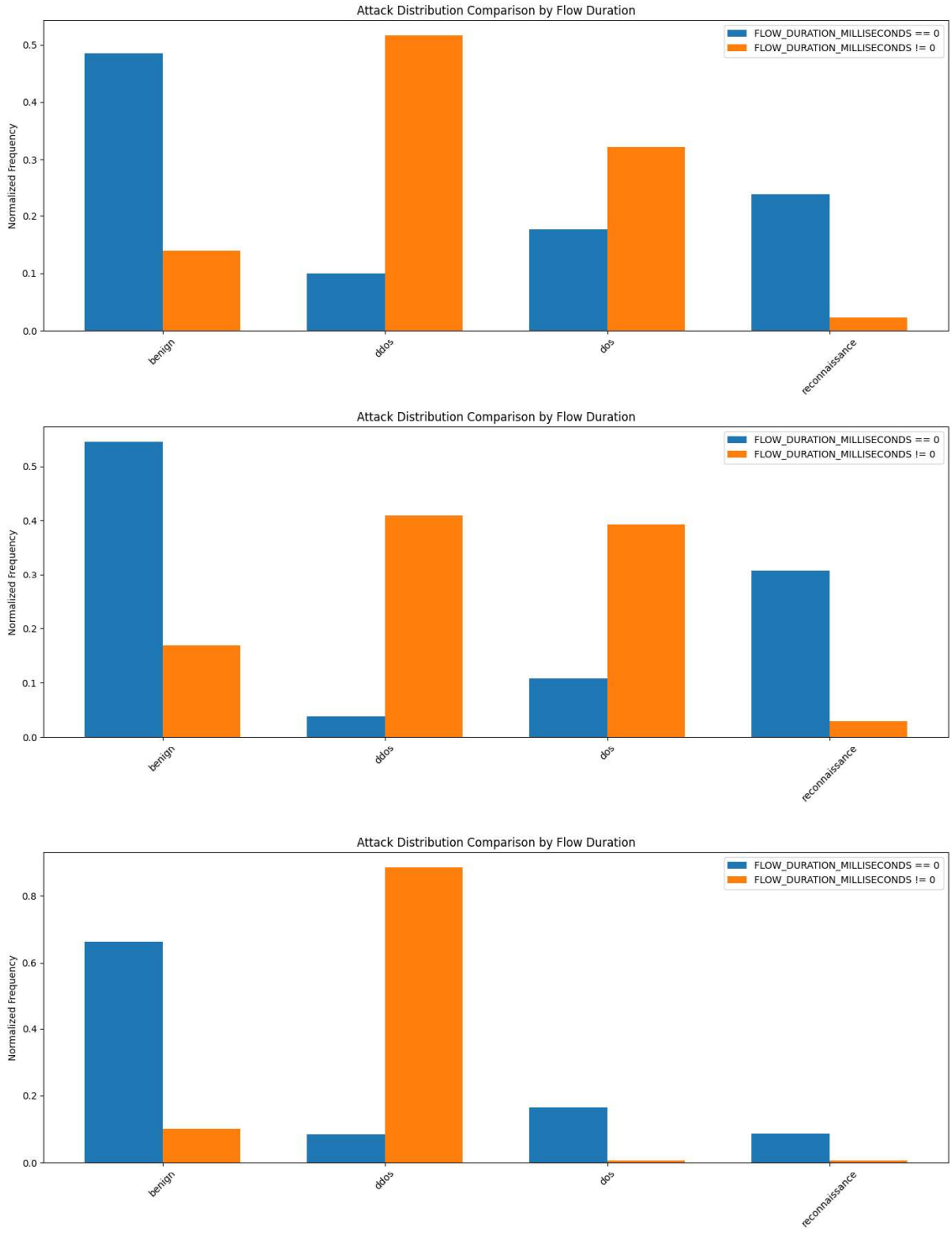
Bu bölümde, önceki bölümler sonucunda elde edilen öznitelikler tekil olarak detaylıca incelenerek, uygun ön işleme aşamalarına tabii tutulmuşlardır. Aşama aşama her bir öznitelik kendi içinde incelenerek, gerekli görülen ön işleme adımları detaylandırılarak sunulmuştur.

FLOW_DURATION: Bu öznitelik incelendiğinde, bu özniteliğin sıfır (zero) veya sıfırdan farklı (non-zero) değerler almasına bağlı olarak normal ağ akışı ile saldırı sınıflarının oransal dağılımlarında belirgin değişimler gözlemlenmektedir. Bu durum Şekil 7.1'de, sırasıyla, BoT-ToN, BoT-UNSW ve ToN-UNSW için net bir biçimde gözlemlenebilir.

Özellikle FLOW_DURATION değerinin sıfır olduğu durumlarda benign örneklerin oranının anlamlı biçimde arttığı, sıfırdan farklı olduğu durumlarda ise saldırı örneklerinin daha baskın hâle geldiği görülmektedir. Bu gözlem, ilgili özniteliğin ikili bir gösterim (sıfır / sıfır değil) şeklinde kodlanmasının ayırt edici bilgi taşıyabileceğini göstermektedir. Dolayısıyla FLOW_DURATION özniteliği, özellikle normal (benign) trafiğin tespitinde etkili bir gösterge olarak değerlendirilebilir.

Buna istinaden bu özniteliğin gerçek değeri yerine 0 veya 1 olduğunu temsil eden bir ikili indikatör (binary indicator) değişkeni tanımlanarak (FLOW_DURATION_ZERO) özniteliğin kendisi veri setlerinden çıkartılmıştır.

DURATION_IN: DURATION_IN özniteliği, dağılımsal özellikleri itibarıyla doğrudan bir makine öğrenmesi modelinde kullanılmaya elverişli değildir. Bu özniteliğin ham



Şekil 7.1. Eğitim Kombinasyonları için Flow Duration Dağılımları

hâli, sınıflar arasında yeterli ayırt ediciliği sağlayamamakta ve model performansını olumsuz yönde etkileyebilecek düzensiz bir yapı sergilemektedir. Bu nedenle, söz konusu özniteliğin temsil gücünü artırmak ve sınıflar arasındaki ayrımı daha belirgin hâle

getirmek amacıyla denetimli (karar ağacı tabanlı) bir bölütleme (binning) yaklaşımı uygulanmıştır. Bu yaklaşım sayesinde, DURATION_IN özniteliği hedef değişken ile olan ilişkisi gözetilerek daha anlamlı aralıklara ayrılmış ve modelin ayırt edici kapasitesinin artırılması hedeflenmiştir. Bu tercihin arkasındaki gerekçeler aşağıda ayrıntılı olarak sunulmaktadır.

Dağılımsal Zorluklar: DURATION_IN değişkeni incelendiğinde, birden fazla istatistiksel sorun göze çarpmaktadır. Özniteliğe ait dağılımlar farklı veri setleri için Tablo 7.5'te gösterimiştir.

Tablo 7.5. DURATION_IN Özniteliğine ait İstatistikler

Eğitim Seti	**Örnek**	**Ort.**	**Std.**	**Min**	**%25**	**Med.**	**%75**	**Maks**
BoT-ToN	869 607	654.57	680.49	0.0	0.0	467.0	1216.0	20795.0
BoT-UNSW	735 043	735.09	704.46	0.0	0.0	707.0	1305.0	3500.0

Burada DURATION_IN, ToN-UNSW'de, önceki bölümlerde açıklanan yöntemlerde elendiğinden, bu öznitelik ilgili veri setinde yer almamaktadır. Tablo 7.5'da Ort. ortalamayı, Std. Standart Sapmayı (standard deviation), Min. minimum değeri, %25, Med. ve %75 ise çeyrekler arası aralıkları (interquantile range, IQR), Maks. ise veri setindeki maksimum değeri göstermektedir.

Öncelikle, gözlemlerin yaklaşık üçte birinin 0 ms değerine sahip olması, öznitelikte belirgin bir sıfır-yoğunluk (zero-inflation) problemi olduğunu göstermektedir. Buna ek olarak, değerlerin büyük bir kısmının 1–30 ms aralığında yoğunlaşması, dağılımın aşırı derecede sağa çarpık (skewed) bir yapı sergilemesine neden olmaktadır. Özellikle BoT–ToN eğitim senaryosunda, 20.000 ms'yi aşan uç değerlerin varlığı, ağır kuyruklu (heavy-tailed) bir dağılım yapısını ortaya koymaktadır. Bu uç değerler ve uzun sağ kuyruk, logaritmik ölçekleme, çeyrekler arası dönüşümü veya standartlaştırma gibi yaygın ön işleme yöntemlerinin etkinliğini önemli ölçüde azaltmaktadır. Dolayısıyla, bu özniteliğin ham hâliyle kullanılması, model kararsızlığına ve sınıflandırma performansında düşüşe yol açabilmektedir.

Süre Tabanlı Doğal Sınıf Ayrışımı: IoT trafik veri setleri üzerinde gerçekleştirilen ayrıntılı

analizler, saldırı türlerinin süre örüntüleri açısından doğal kümelenmeler sergilediğini göstermektedir. Örneğin, DDoS saldırıları genellikle çok kısa ve patlama şeklinde gerçekleşen akışlar üretmekte ve çoğunlukla 0–1 ms aralığında yoğunlaşmaktadır. Keşif (reconnaissance) saldırıları daha kısa süreli olmakla birlikte daha geniş bir varyansa sahiptir. Buna karşılık, benign trafik daha orta ve kararlı süre değerleri sergilerken, arka kapı (backdoor) aktiviteleri daha uzun ve süreklilik gösteren akışlarla karakterize edilmektedir. Bu ampirik gözlemler, süre bilgisinin saldırı sınıflarını ayırt etmede güçlü bir belirleyici olduğunu ortaya koymaktadır. Dolayısıyla, DURATION_IN değişkeninin uygun biçimde bölütlenmesi, modelin sınıflar arasındaki ayrımı daha etkin bir şekilde öğrenmesini sağlamaktadır.

Karar Ağacı Tabanlı Denetimli Bölütlemenin Avantajları: Denetimli bölütleme yaklaşımı, özellikle karar ağacı tabanlı yöntemler kullanıldığında, hedef sınıflar arasındaki bilgi kazancını maksimize eden eşik değerlerini otomatik olarak belirleyebilmektedir. Bu yöntemin temel avantajları şu şekilde özetlenebilir: Karar ağacı yapısı, saldırı sınıflarını en iyi ayıran bölünme noktalarını veri odaklı olarak tespit eder. Elde edilen bölütler, ağ trafiğindeki anlamlı davranış aralıklarına karşılık gelir. Ayrıca, bu dönüşüm; sıfır-yoğunluk problemini ve ağır kuyruk etkilerini etkin bir biçimde ele alarak gürültüyü azaltır ve sınıf ayrılabilirliğini artırır. Denetimsiz yaklaşımlar (kuantil bölütleme, eşit aralıklı bölütleme, Gauss Karışım Modeli (Gaussian Mixture Model, GMM) veya çok sınıflı optimal bölütleme) ile karşılaştırıldığında, denetimli bölütlemenin daha net ve kararlı sınıf ayrımları sağladığı gözlemlenmiştir. Uygulamada, alternatif bölütleme yöntemlerinin bu öznitelik için ya yetersiz performans sunduğu ya da hesaplama açısından verimsiz olduğu belirlenmiştir.

Büyük Ölçekli IoT Veri Setleri Açısından Uygulanabilirlik: Bu çalışmada kullanılan BoT-IoT, ToN-IoT ve UNSW-NB15 veri setleri milyonlarca örnek içeren büyük ölçekli yapılara sahiptir. Bu ölçekte, GMM, MDLP veya çok sınıflı optimal bölütleme gibi hesaplama maliyeti yüksek yöntemler pratikte kararsız veya uygulanamaz hâle gelebilmektedir. Buna karşılık, karar ağacı tabanlı denetimli bölütleme yaklaşımı hesaplama açısından verimli ve ölçeklenebilir bir çözüm sunmaktadır. Bu nedenle, DURATION_IN özniteliği üzerinde Algoritma 10'da gösterildiği üzere ağaç tabanlı bir

bölütleme stratejisi uygulanmıştır.

Algoritma 10, DURATION_IN özniteliğini ham (sürekli) hâliyle modele vermek yerine, verinin istatistiksel yapısını ve sınıf ayrımını dikkate alan denetimli bir bölütleme (binning) dönüşümü üretmek amacıyla tasarlanmıştır.

Algoritma 10: DURATION_IN için Denetimli (Karar Ağacı Tabanlı) Bölütleme

Gerekli:

x: Süre özniteliği (DURATION_IN)
y: Sınıf etiketleri ($y \in \{0,1\}$)
$d_{\max}$: Maksimum ağaç derinliği
$n_{\min}$: Yaprak düğüm için minimum örnek sayısı
p_{tail}: Kuyruk yüzdelik değeri

Ensure:

B: Bölütleme eşikleri listesi

Başlatma:

$x \leftarrow \text{array}(x)$
$y \leftarrow \text{array}(y)$

Adım 1: Sıfır-değer bölütü tanımla

$b_0 \leftarrow 0$

Adım 2: Pozitif süreleri ayır

$x^+ \leftarrow \{x_i \mid x_i > 0\}$
$y^+ \leftarrow \{y_i \mid x_i > 0\}$

Adım 3: Kuyruk kesim noktasını belirle

$t_{\text{cut}} \leftarrow \text{percentile}(x^+, p_{\text{tail}})$
$x^c \leftarrow \{x_i \in x^+ \mid x_i \leq t_{\text{cut}}\}$
$y^c \leftarrow \{y_i \in y^+ \mid x_i \leq t_{\text{cut}}\}$

Adım 4: Denetimli karar ağacı ile bölütleme

Eğitilebilir karar ağacı oluştur:

$$\text{DT}(d_{\max}, n_{\min})$$

Karar ağacını (x^c, y^c) üzerinde eğit

Adım 5: Bölütleme eşiklerini çıkar

$T \leftarrow$ karar ağacındaki pozitif bölme eşikleri
$T \leftarrow \text{sorted}(T)$

Adım 6: Nihai bölüt listesini oluştur

$B \leftarrow [b_0]$
$B \leftarrow B \cup T$
$B \leftarrow B \cup \{t_{\text{cut}}, +\infty\}$

return B

Bu kapsamda algoritma, ilk olarak girdiler olan x (süre değerleri) ve y (etiketler) dizilerini standart bir dizi biçimine dönüştürerek işleme hazırlar. Ardından, DURATION_IN değişkeninde sıfır değerlerin belirgin bir davranış sergilemesi nedeniyle, $b_0 = 0$ olacak şekilde "sıfır-değer bölütü" (zero-bin) ayrı bir kategori olarak tanımlanır; böylece $x = 0$ gözlemleri diğer pozitif sürelerden ayrıştırılarak, modelin sıfır-enflasyonlu yapıyı daha net yakalaması hedeflenir.

Sonraki adımda algoritma yalnızca pozitif süreleri ele alır; yani $x_i > 0$ koşulunu sağlayan örnekler seçilerek x^+ ve bu örneklere karşılık gelen etiketler y^+ oluşturulur. Bu ayrımın temel motivasyonu, sıfırların hem dağılımı baskılaması hem de karar ağacının bölme eşiklerini gereksiz biçimde "0 çevresine" yığmasına yol açabilmesidir. Pozitif süreler ayrıldıktan sonra, ağır kuyruk etkisini ve uç değerlerin bölütleme kararlarını domine etmesini engellemek için bir kuyruk kesim noktası hesaplanır. Bu amaçla x^+ üzerinde p_{tail} yüzdelik değerine karşılık gelen t_{cut} belirlenir ve yalnızca $x_i \leq t_{\text{cut}}$ koşulunu sağlayan çekirdek bölge (core) alınarak x^c ve y^c oluşturulur. Böylece karar ağacı, çok büyük sürelerin oluşturduğu uç noktaların etkisinden arındırılmış daha temsil edici bir aralık üzerinde eğitilir.

Denetimli bölütleme aşamasında, maksimum derinliği $d_{\max}$ ve yaprak başına minimum örnek sayısı $n_{\min}$ ile sınırlandırılmış bir karar ağacı sınıflandırıcısı tanımlanır ve bu model (x^c, y^c) üzerinde eğitilir. Bu noktada karar ağacı, sınıfları ayırma gücünü maksimize edecek şekilde (bilgi kazancı/Gini azalımı gibi ölçütlere dayalı olarak) en uygun bölme eşiklerini otomatik biçimde öğrenir. Eğitim tamamlandıktan sonra, karar ağacında kullanılan pozitif bölme eşikleri çıkarılarak T kümesi elde edilir ve bu eşikler artan sırada sıralanır. Elde edilen T, DURATION_IN değişkeninin sınıf ayrımına katkı sağlayan anlamlı aralıklarına karşılık gelen denetimli bölüt sınırlarını temsil eder.

Son adımda algoritma, nihai eşik listesini birleştirerek B listesini üretir. Bu liste; sıfır-değer bölütünü temsil eden $b_0 = 0$ başlangıcını, karar ağacından türetilen denetimli eşikleri (T), kuyruk başlangıcını temsil eden t_{cut} değerini ve tüm uç değerleri kapsamak üzere $+\infty$ sınırını içerir. Böylece dönüşüm sonucunda, (i) sıfır değerler ayrı bir bin olarak korunmuş, (ii) çekirdek bölgede sınıf ayrımını güçlendiren denetimli aralıklar

üretilmiş ve (iii) ağır kuyruk bölgesi tek bir üst bin altında toplanarak modelleme sürecinde hem kararlılık hem de ayrıştırıcılık artırılmış olur. Algoritma çıktısı olan B eşikleri, DURATION_IN özniteliğinin ham değerleri yerine, bu eşiklere göre bölütlenmiş kategorik/ordinal bir temsil oluşturmak için DURATION_IN_BIN kullanılmaktadır.

MAX_IP_PKT_LEN: Genel olarak BoT-IoT, ToN-IoT ve UNSW-NB15 veri setlerinin tamamı, Wi-Fi ve Ethernet gibi standart İiternet protokollerine dayanmaktadır. Bu nedenle, 1500 baytın üzerindeki paket boyutu değerleri, bu ağ katmanlarının teknik sınırlarıyla uyumlu değildir ve gerçek IP paketlerini temsil etmemektedir. Ethernet ve Wi-Fi tabanlı ağlarda fiziksel iletim birimi, Maksimum İletim Birimi (Maximum Transmission Unit, MTU) tarafından sınırlandırılmakta olup, bu değer yaklaşık olarak 1500 bayttır. tcpdump, CICFlowMeter ve Argus gibi yaygın ağ trafiği yakalama ve akış çıkarım araçları da paketleri bu fiziksel kısıta uygun biçimde kaydetmektedir.

1500 baytın üzerindeki paket boyutu değerleri, çoğunlukla ölçüm veya ön işleme aşamasında ortaya çıkan yapay gürültülerden (noise) kaynaklanmaktadır. Bu gürültüler; TCP yeniden birleştirme (packet reassembly), akış düzeyinde segment hesaplamaları veya yakalama aracı tarafından gerçekleştirilen tampon (buffer) işlemleri gibi nedenlerle oluşabilmektedir. Bu tür yapay aykırı değerler, makine öğrenmesi iş akışlarını olumsuz yönde etkilemektedir. Özellikle normalizasyon süreçlerini bozmakta, gradyan veya bölme (split) kararlarını gerçekçi olmayan biçimde yönlendirmekte ve gerçek ağ davranışlarını yansıtmayan gürültünün modele dâhil edilmesine neden olmaktadır.

Bu sebeple, 1500 baytın üzerindeki paket boyutu değerlerinin tamamen veri setinden çıkarılması yerine, bu değerlerin 1500 bayt ile sınırlandırılması (clipping) daha uygun bir yaklaşım sunacaktır. Bu yöntem, fiziksel ağ kısıtlarıyla tutarlı bir üst sınır uygularken gereksiz veri kaybını önlemekte ve modelin sahte aykırı değerlerin etkisinden arındırılmış, daha kararlı ve genellenebilir biçimde öğrenmesini sağlamaktadır. Ayrıca, sınırlandırma (clipping) işlemi potansiyel olarak anlamlı olabilecek büyük segment davranışlarını tamamen ortadan kaldırmadığından, 1500 bayt üzerindeki gözlemler ihtiyaç duyulması hâlinde ayrı bir aykırı değer bayrağı (outlier flag) özniteliği olarak da kodlanabilmektedir.

Paket boyutu değerleri 1500 bayt ile sınırlandırıldıktan sonra, bu özniteliğe logaritmik dönüşüm uygulanmaktadır. Bu adımın temel gerekçeleri şu şekilde özetlenebilir: Paket boyutu öznitelikleri genellikle yüksek derecede çarpık (skewed) dağılımlara sahiptir ve ham değerler geniş bir sayısal aralığa yayılmaktadır. Derin öğrenme modellerinde bu tür büyük ve çarpık değerler, gradyan güncellemelerini kararsız hâle getirerek eğitim sürecini yavaşlatmaktadır. Logaritmik dönüşüm, değer aralığını sıkıştırarak gradyanların daha dengeli olmasını sağlamakta ve modelin daha anlamlı örüntüler öğrenmesine katkı sunmaktadır. Ağaç tabanlı modellerde ise log dönüşüm, dağılımı daha dengeli hâle getirerek aşırı uç değerlere zorunlu bölmeler yerine daha bilgilendirici eşiklerin bulunmasına imkân tanımaktadır.

Özetle, logaritmik dönüşüm çarpıklığı azaltmakta, öğrenme sürecini kararlı hâle getirmekte, modelin sağlamlığını artırmakta ve hem derin öğrenme hem de ağaç tabanlı algoritmalarda daha iyi genelleme performansı elde edilmesine katkı sağlamaktadır. Yapılan sınırlandırma ve log dönüşümü sonrası MAX_IP_PKT_LEN özniteliğinin çarpıklık değeri, 3,97 mertebelerinden, 1,14 seviyelerine inmiştir.

LONGEST_FLOW_PKT_LEN: Bu öznitelikte MAX_IP_PKT_LEN ile çok benzer karakteristikler gösterdiğinden dolayı, aynı işlemler uygulanmıştır. Önce sınırlandırma daha sonrasında log transformasyonu uygulanarak daha az çarpık bir veri elde edilmiştir.

MIN_IP_PKT_LEN: IoT ve genel ağ trafiği literatüründe yaygın olarak kabul edildiği üzere, bir akış içerisinde gözlemlenen minimum IP paket uzunluğu (MIN_IP_PKT_LEN), doğrudan protokol tasarımı ve ağ trafiğinin işlevsel karakteristikleri tarafından şekillendirilmektedir. IoT ve uygulama katmanı protokollerini inceleyen çalışmalar, IoT cihazlarının bant genişliği ve enerji tüketimini minimize etmek amacıyla sıklıkla UDP tabanlı telemetri ve CoAP ile MQTT gibi hafif protokoller kullandığını vurgulamaktadır. Bu bağlamda, mesaj boyutları bilinçli olarak küçük tutulmakta ve 0–20 bayt ile 20–40 bayt aralıkları, kontrol mesajları ve kısa telemetri iletimleri için doğal bölgeler olarak ortaya çıkmaktadır (Azab ve diğ., 2024).

Paket boyutu dağılımlarını modelleyen akademik çalışmalar, ağ trafiğinin önemli bir bölümünün 40–44 bayt civarında yoğunlaştığını göstermektedir. Bu durum, büyük ölçüde

TCP SYN/ACK/RST gibi kontrol segmentleri ve çok küçük yük taşıyan mesajlardan kaynaklanmaktadır. Buna ek olarak, yaklaşık 1400–1500 bayt aralığında ikinci bir belirgin tepe gözlemlenmekte olup, bu tepe MTU sınırına yakın büyük veri paketlerini temsil etmektedir. Bu bulgular, 40–60 bayt aralığını hem doğal TCP kontrol davranışları hem de tarama, DoS ve DDoS saldırılarında gözlemlenen küçük paket örüntüleri açısından kritik bir bölge hâline getirmektedir (Castro ve diğ., 2010; Prajapati ve diğ., 2025).

IoT zararlı yazılım trafiği ve saldırı tespitine odaklanan çalışmalar, SYN/ACK tabanlı tarama ve flood saldırılarının çoğunlukla yük taşımayan veya son derece küçük yük içeren TCP segmentlerine dayandığını rapor etmektedir. Bu tür saldırılar, saniyede yüksek paket sayısına (pps) sahip kötü amaçlı örüntüler üretmekte ve küçük paketlerin baskın olduğu trafik profilleri oluşturmaktadır. Bu durum, MIN_IP_PKT_LEN değerinin 40–60 bayt aralığında olmasının hem bağlantı kurma/sonlandırma davranışlarını hem de anormal saldırı örüntülerini işaret edebileceğini desteklemektedir (Huston, 2024).

Uygulama trafiği ve protokol sınıflandırmasına ilişkin literatür, DNS ve benzeri protokollerin genellikle birkaç yüz bayt seviyesinde paketler ürettiğini; kısa HTTP istekleri ve hafif uygulama mesajlarının ise sıklıkla 60–100 bayt aralığında yer aldığını göstermektedir. Bu gözlem, düzenli uygulama akışlarında MIN_IP_PKT_LEN değerinin çoğunlukla bu aralıkta ortaya çıktığını, daha büyük minimum paket boyutlarının ise kararlı ve iyi biçimlendirilmiş uygulama davranışlarını yansıttığını göstermektedir (Azab ve diğ., 2024).

TLS el sıkışmaları ve HTTP/TLS trafiğine ilişkin teknik ve akademik analizler, ClientHello/ServerHello mesajları ile kısa HTTP isteklerinin genellikle 100–300 bayt aralığında olduğunu ortaya koymaktadır. Bu paketler, web ve uygulama protokollerinde veri taşıyan en küçük paketleri temsil etmektedir. Dolayısıyla, minimum paket uzunluğunun bu aralıkta olması, akışın yalnızca kontrol trafiği değil, yapılandırılmış uygulama katmanı verisi taşıdığına işaret etmektedir (Davies, 2025).

Son olarak, paket boyutu dağılımı çalışmaları ile MTU/MSS analizleri, İnternet trafiğindeki ikinci büyük yoğunluğun MTU sınırına yakın büyük paketlerden oluştuğunu

göstermektedir. Bu tür paketler ağırlıklı olarak web içerik dağıtımı, dosya transferleri ve video akışları tarafından üretilmektedir. Bu nedenle, MIN_IP_PKT_LEN değerinin 300 bayt ve üzerinde olması genellikle normal, yük taşıyan veri akışlarını temsil ederken; birçok saldırı türü çok daha küçük minimum paket boyutları üretmektedir (Imperva Group, 2025; Prajapati ve diğ., 2025).

Bu bağlamda, 0–20, 20–40, 40–60, 60–100, 100–300 ve 300+ bayt sınırları, literatürde yaygın olarak gözlemlenen ağ trafiğinin doğal iki modlu yapısını — küçük kontrol paketleri ile büyük veri paketleri — yansıtmaktadır. MIN_IP_PKT_LEN özniteliğinin bu aralıklar kullanılarak bölütlenmesi MIN_IP_PKT_LEN_BIN, kötü amaçlı ve benign trafiğin ayrıştırılmasını güçlendiren, protokol temelli ve sezgisel bir ön işleme stratejisi sunmakta; aynı zamanda gerçek ağ akışlarının davranışsal özellikleriyle doğrudan uyum sağlamaktadır.

SHORTEST_FLOW_PKT: Bu öznitelikte MIN_IP_PKT_LEN ile çok benzer karakteristikler gösterdiğinden dolayı, aynı bölütleme işlemi uygulanmıştır.

NUM_PKTS_UP_TO_128_BYTES: NUM_PKTS_UP_TO_128_BYTES özniteliği, son derece çarpık (skewed) bir dağılım sergilemektedir. Bu durum Tablo 7.6'da gösterilmiştir.

Tablo 7.6. NUM_PKTS_UP_TO_128_BYTES İstatistikleri

Eğitim Seti	**Örnek**	**Ort.**	**Std.**	**Min**	**Q1**	**Medyan**	**Q3**	**Maks**
BoT-UNSW	735 043	10.99	178.61	0	0	2	3	20 771
ToN-UNSW	293 916	30.58	407.33	0	2	6	14	84 359
BoT-ToN	869 607	7.59	282.88	0	1	2	4	84 359

İncelenen veri setlerinde, gözlemlerin yaklaşık %75'inin çok küçük değerlere yoğunlaştığı, buna karşılık az sayıda akışın son derece yüksek paket sayılarına sahip olduğu görülmektedir. Bu durum, ham dağılımın ciddi biçimde dengesiz ve sayısal olarak aşırı gerilmiş (numerically stretched) olmasına neden olmaktadır. Böyle bir dağılım yapısında doğal aralık sınırları bulunmadığından, klasik bölütleme (binning) yaklaşımları anlamlı sonuçlar üretmemekte; oluşturulan bölütlerin büyük bir kısmı aynı değer aralığında yığılmaktadır.

Bu dengesizliği ele almak amacıyla, öznitelik üzerinde iki aşamalı bir ön işleme stratejisi uygulanmıştır. İlk olarak, aşırı uç değerlerin etkisini sınırlamak için üst sınır kırpma (clipping) işlemi gerçekleştirilmiştir. NUM_PKTS_UP_TO_128_BYTES değişkeninde 100'ün üzerinde kalan değerler son derece nadir olup, sınıflar arasındaki ayrımı anlamlı biçimde güçlendirmemektedir. Buna karşın bu uç değerler, normalizasyon işlemlerini bozmakta, istatistiksel momentleri (ortalama ve varyans gibi) gerçekçi olmayan aralıklara taşımakta ve hem ağaç tabanlı modellerde ilgisiz bölünmelere hem de derin öğrenme modellerinde kararsız gradyan güncellemelerine yol açabilmektedir. Bu nedenle, 100 değeri üst sınır olarak belirlenmiş ve bu eşik üzerindeki değerler kırpılarak dağılımın genel yapısı korunurken gürültü azaltılmıştır.

İkinci aşamada, kırpma işleminin ardından logaritmik dönüşüm uygulanmıştır. Bu dönüşüm, dağılımın uzun sağ kuyruğunu sıkıştırarak uç değerlerin etkisini azaltmakta, sıfıra yakın bölgede yoğunlaşan gözlemleri daha dengeli bir şekilde yaymakta ve özniteliğin daha simetrik, model dostu bir yapıya kavuşmasını sağlamaktadır. Logaritmik dönüşüm, özellikle derin öğrenme modellerinde gradyan güncellemelerini stabilize ederken, ağaç tabanlı modellerin daha anlamlı ve bilgilendirici bölme noktaları bulmasına da olanak tanımaktadır.

Uygulanan bu iki aşamalı ön işleme stratejisinin etkisi, dağılım istatistikleri üzerinden açıkça gözlemlenmiştir. Örneğin, BoT–ToN Train veri setinde NUM_PKTS_UP_TO_128_BYTES özniteliğinin çarpıklık (skewness) değeri logaritmik dönüşüm öncesinde 8.45 iken, dönüşüm sonrasında 0.88 seviyesine düşmüştür. Bu sonuç, özniteliğin çok daha sağlıklı, öğrenilebilir ve modelleme açısından elverişli bir dağılıma kavuştuğunu göstermektedir.

IN_BYTES: IN_BYTES özniteliği, yüksek derecede çarpık bir dağılıma sahip olup belirgin bir sağ kuyruk (long right tail) davranışı sergilemektedir. Bu dağılım yapısı, IoT cihazları tarafından üretilen ağ trafiği için tipik değildir. IoT tabanlı haberleşmelerde genellikle orta seviyede ve görece kararlı bayt miktarları gözlemlenirken, son derece yüksek bayt değerleri çoğunlukla ölçüm artefaktları, hatalı yapılandırılmış cihazlar veya olağandışı akış davranışlarından kaynaklanmaktadır.

Uygun bir kesme (cutoff) noktası belirlemek amacıyla, benign örnek oranının dağılımın üst yüzdelik dilimleri boyunca nasıl değiştiği analiz edilmiştir. Elde edilen bulgular, yaklaşık 2000 bayt sonrasında benign trafiğin oranının baskın hâle geldiğini ve bu oranın istikrarlı biçimde devam ettiğini göstermektedir. Bu durum, söz konusu eşik değerinin üzerindeki bayt büyüklüklerinin saldırı türlerinin ayırt edilmesi açısından bilgilendirici olmadığını ve büyük ölçüde gürültü (noise) niteliği taşıdığını ortaya koymaktadır.

Bu nedenle, sınıf ayrımına katkı sağlamayan ve çoğunlukla gürültü içeren aşırı yüksek bayt değerlerini elemek amacıyla IN_BYTES özniteliğine üstten kırpma (clipping) işlemi uygulanmıştır. Ardından gerçekleştirilen logaritmik dönüşüm, ağır sağ kuyruğun etkisini azaltarak dağılımın daha dengeli hâle gelmesini sağlamıştır. Bu iki adımın birlikte uygulanması sonucunda, özniteliğin çarpıklık (skewness) değeri 3.77'den 0.66'ya düşürülmüş; böylece modelin daha kararlı, anlamlı ve genellenebilir örüntüler öğrenmesi mümkün hâle getirilmiştir.

OUT_BYTES: OUT_BYTES özniteliği incelendiğinde, değerin sıfır olup olmamasının sınıflar arasında güçlü bir ayırt edici sinyal sunduğu gözlemlenmiştir. Sıfır bayt içeren akışlarda benign oranı yalnızca %7.5 seviyesinde kalırken, sıfırdan farklı değerlere sahip akışlarda bu oran yaklaşık %38'e yükselmektedir. Bu durum, sıfır–sıfır olmayan ayrımının başlı başına anlamlı bir öznitelik olduğunu göstermektedir.

Sıfırdan farklı OUT_BYTES değerleri için benign oranının dağılımı incelendiğinde, bu oranın tekdüze (monotonik) bir artış veya azalış sergilemediği; belirli aralıklarda yükselip düşerek tekrar yükseldiği görülmektedir. Bu gözlem, farklı bayt büyüklüklerinin farklı ağ davranışlarını temsil ettiğini ve bu davranışların doğrudan modele yansıtılmasının faydalı olacağını ortaya koymaktadır. Bu doğrultuda iki yardımcı öznitelik tanımlanmıştır: OUT_BYTES_IS_ZERO, sıfır ve sıfır olmayan akışları ayırt etmek amacıyla kullanılırken; OUT_BYTES_HIGH_BURST ise çok yüksek bayt hacmine sahip ve güçlü biçimde benign trafikle ilişkili akışları işaretlemektedir.

Devamında, OUT_BYTES özniteliği yüksek patlama (high-burst) eşiğinde kırpılmış ve dağılımın çarpıklığını azaltmak amacıyla logaritmik dönüşüm uygulanmıştır. Her ne kadar kırpma işlemi aşırı değerler hakkında bilgi kaybı riski taşısa da, bu bilgi

OUT_BYTES_HIGH_BURST özniteliği aracılığıyla korunmaktadır. Böylece model, eğitim sürecinde hem dengelenmiş bir dağılımdan faydalanmakta hem de kırpılan aşırı davranışlara ilişkin sinyali dolaylı olarak geri kazanabilmektedir.

SRC_TO_DST_BYTES ve DST_TO_SRC_BYTES : Bu öznitelikler de OUT_BYTES özniteliğine benzer istatistiksel özellikler sergilemektedir. Bu öznitelikler de belirgin sağ kuyruklara, yüksek çarpıklığa ve sınıf oranlarının belirli değer aralıklarında anlamlı biçimde değiştiği davranışsal bölgelere sahiptir.

Bu nedenle, söz konusu öznitelikler için de OUT_BYTES özniteliğinde uygulanan ön işleme stratejisine benzer bir yaklaşım benimsenmiştir. Öncelikle, benign trafiğin baskın hâle geldiği dinamik eşik değerleri belirlenmiş; ardından bu eşiklerin üzerindeki değerler kırpılmıştır. Son aşamada ise logaritmik dönüşüm uygulanarak dağılımlar daha dengeli ve öğrenilebilir hâle getirilmiştir. Bu süreç, hem gürültü etkisini azaltmayı hem de modelin daha kararlı ve genellenebilir örüntüler öğrenmesini sağlamayı amaçlamaktadır.

TCP_WIN_MAX_IN: Sürekli sayısal özniteliklerin aksine (örneğin IN_BYTES, OUT_BYTES, paket sayıları veya süre değişkenleri), TCP_WIN_MAX_IN özniteliği hedef değişken ile tek yönlü veya monoton bir ilişki sergilememektedir. İncelenen veri setlerinde, hem küçük hem orta hem de büyük pencere boyutlarının normal ve saldırı trafiği ile eşlenebildiği, dolayısıyla sınıf olasılıklarının pencere boyutuna bağlı olarak doğrusal veya düzgün bir biçimde değişmediği gözlemlenmiştir. Bu durum, pencere boyutunun büyümesiyle birlikte benign olasılığının kademeli olarak artması ya da azalması gibi bir varsayımı geçersiz kılmaktadır.

Bu bağlamda, TCP_WIN_MAX_IN için logaritmik dönüşüm uygulanması, özniteliğin doğasına aykırı sonuçlar doğurmaktadır. Log dxönüşüm, pencere boyutunun sürekli ve monoton bir ölçek üzerinde sınıf olasılığını belirlediği varsayımını dayatmakta; oysa gerçek ilişkiler keskin eşiklere sahip, basamaklı ve kategorik bir yapı sergilemektedir. Bu tür bir dönüşüm, farklı davranışsal desenlerin tek bir yumuşatılmış dağılım altında toplanmasına neden olarak sınıflar arasındaki ayırt edici sınırları bulanıklaştırmakta ve modelin protokol seviyesindeki anlamsal farkları öğrenme yeteneğini zayıflatmaktadır.

Benzer şekilde, kırpma (clipping) işlemi de bu öznitelik için uygun değildir. Yüksek pencere boyutları, bu bağlamda istatistiksel uç değerler (outlier) değil; aksine yüksek bant genişliğine sahip, sağlıklı ve benign akışların doğal bir göstergesidir. Dolayısıyla büyük pencere değerlerini kırpmak, anlamlı ve ayırt edici protokol bilgisinin kaybına yol açmaktadır.

Analizler, TCP_WIN_MAX_IN değişkeninin güçlü protokol temelli semantik anlamlar taşıdığını ve üç belirgin davranışsal rejime ayrıldığını göstermektedir: (i) düşük pencere boyutları (0–1024 bayt), genellikle sınırlı tamponlara sahip, bozuk veya yoğun saldırı trafiğini temsil etmekte ve büyük oranda saldırı sınıfı ile ilişkilendirilmektedir; (ii) orta düzey pencere boyutları (1024–32768 bayt), tipik istemci ve IoT cihaz davranışlarını yansıtmakta ve çoğunlukla benign akışlardan oluşmaktadır; (iii) yüksek pencere boyutları (32768–65535 bayt) ise yüksek verimli, sunucu sınıfı ve kararlı benign iletişimi temsil etmektedir.

Bu açık ve monoton olmayan sınıf geçişleri dikkate alındığında, TCP_WIN_MAX_IN özniteliği için sürekli dönüşümler yerine kategorik bölütleme tercih edilmiştir. Bu kapsamda öznitelik üç ayrık bölüte ayrılmıştır: BIN_0 ($\leq$1024), BIN_1 (1024–32768) ve BIN_2 ($\geq$32768). Bu bölütleme, TCP protokolünün gerçek davranışlarını doğrudan yansıtmakta ve modelin, pencere boyutuna bağlı olarak ortaya çıkan keskin sınıfsal ayrımları daha etkin biçimde öğrenmesine olanak sağlayacağı düşünülmektedir.

7.3. Yayılım Kayması Analizi

Makine öğrenmesinde kayma (drift) kavramı, bir modelin eğitildiği verilerle, gerçek zamanlı karşılaştığı (test verisi de olabilir) verilerin arasındaki istatistiksel dağılım değişikliği olarak ifade edilebilir (Ackerman ve diğ., 2020). Kayma terimi genellikle, bir bağlamda eğitilen bir makine öğrenmesi modelinin başka bir bağlamda uygulanması durumunda ortaya çıkan değişiklikleri ifade eder (Goldenberg ve Webb, 2019). Kaymaların en kritik sonucu, modelin tahmin gücünü yitirmesi ve güvenilirliğinin azalmasıdır (Goldenberg ve Webb, 2019; Mannapur, 2025). Bunun temel sebebi olarak, modein eğitim ve test edildiği veri seti arasında istatistiki farklılıkların olmasıdır. Bu da aslında modelin eğitildiği veri ile test edildiği veya gerçek dünyada çalıştırıldığı verilerin

aslında nitelik ve nicelik açısından farklı olmasıdır. Dolayısıyla modele öğretilen dağılım ve istatistiki desenler, modele sorulanlardan farklı olacaktır. Bu da modelin başarımında ciddi düşüşler meydana gelmesine sebep vermektedir.

Yayılım Kayması temel olarak üç ana kategoride incelenebilir:

Kavram Kayması (Concept Shfit): Girdi öznitelikleri X ile hedef değişken Y arasındaki ilişkinin, diğer bir deyişle, $P(Y|X)$ olasılığının zamanla değişmesidir (Ackerman ve diğ., 2020; Rahmani ve diğ., 2023). Bu durum, modelin öğrendiği temel mantığın artık geçerli olmayacağı anlamına gelecektir (Sahiner ve diğ., 2023). Örneğin, bir hastalığın tanımın veya çeşitli teşhis kriterlerinin değişmesi, bu türeden bir kaymaya örnek olarak verilebilir (Rahmani ve diğ., 2023).

Veri Kayması (Data Drift/Covariate Shift): edef değişkenle olan ilişki değişmese bile, sadece girdi verilerinin kendi dağılımının ($P(X)$) değişmesidir (Ackerman ve diğ., 2020; Goldenberg ve Webb, 2019).

Öncel Sınıf Olasılığının Kayması (Prior Class Probability Shift): Sınıf etiketleri dağılımındaki ($P(Y)$) değişimdir (Ackerman ve diğ., 2020; Goldenberg ve Webb, 2019).

Kaymalar farklı şekillerde meyadana gelerek model başarımını tehdit eden, dikkatle incelenmesi gereken ana modelleme konularının başında gelir. Bu çalışmada global bir IoT-IDS modeli kurarken, ToN_IoT, BoT_IoT ve UNSW-NB15 veri setleri kullanılacaktır. Bu modeller için elimizde ortak öznitelik havuzu bulunsa da, bu veri setleri farklı zamanlarda, farklı amaçlara yönelik ve farklı test ortamlarında elde edilmişlerdir. Dolayısıyla benzer özniteliklere şuan için sahip olsalarda, veri dağılımları ve karakteristikleri birbirlerinden oldukça farklı olması beklenen bir durumdur. Bu yüzden de ileri bir modelleme aşamasına geçmeden önce, yayılım kayması analizlerinin farklı şekillerde gerçekleştirilmesi, olası bir başarısızlık durumunda en önemli faktörlerden biri olması beklenmiştir. Dolayısıyla bu veri setleri üzerinden, çeşitli analizlerle, yayılım kaymasının derecesi hakkında bilgi sahibi olmaya çalışılmıştır.

7.3.1. Yayılım Kayması Analizi İçin Kullanılan Yöntemler

Bu bölümde, veri setlerine uygulanacak ve yayılım kayması için kullanılacak olan çeşitli metodolojilere yer verilecektir.

Kolmogrov-Smirnov Testi (Kolmogrov-Smirnov Test, KS): Kolmogrov-Smirnov testi, iki dağılımın birbirine benzerliğini kümülatif dağılım fonksiyonları (cumılative distrubiton functions, CDF) kullanarak ölçen, parametrik olmayan (dağılım varsayımı yapmayan) bir istatistiksel testtir (Kolmogorov, 1933). Temel olarak, "iki dağılımın kümülatif dağılımları arasında herhangi bir noktaki en büyük fark, ne kadardır?" sorusuna cevap vermeye çalışır. Bu en büyük fark literatürde KS-istatistiği (D) olarak adlandırılır.

Matematiksel olarak KS-İstatistiği (D), Denklem (7.1)'de gösterilmiştir:

$$D = sup_x |F_1(x) - F_2(x)| \tag{7.1}$$

Burada $F_1(x)$ ve $F_2(x)$ iki dağılıma ait kümlatif dağılım fonksiyonlarını (CDF) verirken, D ise maksimum mesafeyi ifade eder. Burada ilk CDF eğitim setine aitken, ikinci CDF'in test setine ait olduğu düşünülürse, KL-İstatistiği aslında dağılımsal olarak bu iki değerin ne kadar farklı olduğunu ölçer. Bu tez çalışmasında, KL-İstatistiği için CDF olarak, Ampirik Kümlatif Dağılım Fonksiyonu (Empirical Cumulative Distribution Function, ECDF) kullanılmıştır. ECDF, bir veri örneklemi için CDF'in örneklem temelli, yani verinin kendisinden alınan bir versiyonudur.

Bir $x_1, x_2, x_3, ..., x_n$ örneklemi için ECDF, Denklem (7.2)'deki gibi ifade edilebilir:

$$F'_n(x) = \frac{1}{n} \sum_{i=1}^{n} 1(x_i \leq x) \tag{7.2}$$

Burada n toplam örnek sayısı, $(1(.)$ gösterge fonksiyonu, $F'_n(x)$ ise x'ten küçük veya eşit gözlemlerin oranını belirtir. Buradaki her bir x_i değeri ilgili veri setindeki bir özniteliğe ait değerler olarak düşünülebilir. Bu sayede her bir öznitelik için, eğitim ve test veri seti arasında KL-istatistiği uygulanarak, CDF'ler açısından ne kadarlık bir farklılık olduğu gözlemlenebilmektedir.

Kullback-Leibler Ayrışımı: KL-Ayrışımı, iki olasılık dağılımı arasındaki farkı bilgi kaybına (information loss) göre ölçen, asimetrik bir nitelik taşıyan bir olasılık tabanlı uzaklık ölçüsüdür (Kullback ve Leibler, 1951). Genel olarak, "P gerçek dağılımı, Q adında ikinci bir dağılım ile modellenirse, bilgi kaybı ne olur?", sorusuna cevap verir.

KL-Ayrışımı ayrık dağılımlar (discrete distributions) için, Denklem (7.3)'te gösterilmiştir:

$$D_{KL}(P||Q) = \sum_{x} P(X) log \frac{P(x)}{Q(X)} \tag{7.3}$$

Sürekli zamanlı dağılımlar (continious distributions) için ise, Denklem (7.4)'te gösterilmiştir:

$$D_{KL}(P||Q) = \int P(X) log \frac{P(x)}{Q(X)} \tag{7.4}$$

Burada $P(x)$ referans alınan gerçek dağılımı verirken, $Q(x)$ ise $P(x)$'i temsil etmek için referans alınan dağılımı göstermektedir. Eğer $Q(x)$, $P(x)$'in yüksek olduğu yerlerde küçükse burada KL-Ayrışımı toplamına katkı büyük olurken, birbirlerine daha yakın değerler alıyorlarsa, KL-Ayrışımına katkı daha küçük olacaktır. KL-Ayrışımı $[0 - \infty)$ arası değer alırken, 0 değeri dağılımların birbiriyle tamamen aynı olduğunu, ∞'a doğru gittikçe dağılımların birbirinden giderek uzak olduklarını ifade eder.

KL-Ayrışımı asimetrik değildir, yani, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ şeklindedir.

Jensen-Shannon Ayrışımı (Jensen-Shannon Divergence, JS): JS-Ayrışımı, KL-Ayrışımının, simetrik ve sayısal olarak daha kararlı bir versiyonu olarak düşünülebilir. Burada, "iki olasılık dağılımı, ortak bir otalama dağılımdan ne kadar sapmaktadır?", sorusuna cevap aranmaktadır (Lin, 1991).

JS-Ayrışımı için teorik formülasyon, Denklem (7.5) ve Denklem (7.6)'da gösterilmiştir:

$$M(x) = \frac{1}{2}(P(x) + Q(x)) \tag{7.5}$$

$$D_{JS}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M) \tag{7.6}$$

JS-Ayrışımı $[0,1]$ arasında değer üretir, bu değer 0'a yakın olduğunda iki dağılımda, hesaplanan ortak dağılımla (M) birebir yakın olduğunu, 1 değeri ise tamamen farklı olduklarını ifade etmektedir. JS-Ayrışımı, iki yönlü KL-ayrışımını ölçtüğü için simetriktir, yani, $D_{JS}(P||Q) = D_{JS}(Q||P)$ olacaktır.

Wasserstein Uzaklığı (Wasserstein Distance, WD): Wasserstein uzaklığı, iki olasılık dağılımı arasındaki farkı, "bir dağılımı diğerine dönüştürmek için olsılık dağılımını büyüklük (magnitude) ve yön (direction) olarak ne kadar değiştirilmesi gerekir?", sorusuna cevap vermeye çalışırak ölçer (Villani, 2009). İki dağılım arasındaki, geometrik farkı ölçerek, sadece olasılık değerlerini değil, olasılıkların pozisyonel konumlarının da nerede olduklarını ölçerek, ortalama, varyans kuyruk (tail) kaymalarını doğal olarak yakalamaktadır. KS gibi sadece değerlerin farklarına bakmaz; KL/JS gibi sadece yoğunluk oranlarına da bakılmamaktadır. Burada öznitelikler teker teker incelendiğinden, 1-boyutlu WD kullanılmıştır. 1-boyutlu WD matematiksel olarak Denklem (7.7)'de gösterilmiştir:

$$W_1(P,Q) = \int_0^1 \left| F_P^{-1}(u) - F_Q^{-1}(u) \right| du \tag{7.7}$$

Burada $F_P^{-1}(*)$ ve $F_Q^{-1}(*)$, sırasıyla P ve Q dağılımlarına ait kuantil (ters kümülatif dağılım fonksiyonları, inverse CDF) fonksiyonlarını ifade etmektedir. 1-boyutlu durumda Wasserstein mesafesi, iki dağılımın karşılık gelen kuantilleri arasındaki mutlak farkların $[0,1]$ aralığında ortalaması olarak hesaplanır. Bu tanım, dağılımlar arasındaki farkı yalnızca olasılık yoğunlukları üzerinden değil, değerlerin geometrik konumlarını da dikkate alarak ölçer. Bu nedenle ortalama, varyans ve kuyruk davranışlarındaki kaymaları doğal ve kararlı bir biçimde yakalayabilmektedir.

Tablo 7.7'de yukarıda bahsedilen yayılım kaymasının tespit edilmesinde kullanılacak metrikler özet olarak sunulmuştur.

Pratik olarak, tez kapsamında yayılım kayması analiz edilirken, metriklerin yorumlanma biçimleri aşağıdaki şekilde özetlenebilir:

Tablo 7.7. Yayılım Kayması Ölçütlerinin Karşılaştırılması

Ölçüt	Amaç	Güçlü Yönler	Zayıf Yönler	En Uygun Kullanım Alanı
KS	ECDF şekil farkları	Parametrik olmayan, basit ve hızlı	Yalnızca maksimum sapmayı dikkate alır	İlk aşama dağılım kayması (drift) taraması
KL	Bilgi kaybı (information loss)	Yeni modlara ve sistematik farklara duyarlıdır	Asimetrik, sınırsız değer alabilir	Görülmemiş veya nadir davranışların tespiti
JS	Simetrik bilgi benzerliği	Kararlı, sınırlı ve simetrik	Olasılık yoğunluğu tahmini gerektirir	Isı haritaları, çoklu öznitelik karşılaştırmaları
Wasser.	Olasılık kütlesinin geometrik yer değiştirmesi	Ortalama, varyans ve kuyruk kaymalarını yakalar	Ölçek bağımlıdır	Sürekli özniteliklerde dağılım kayması analizi

- KS metriği, dağılımın genel yapısında meydana gelen belirgin ve yapısal şekil farklılıklarını tespit etmek için kullanılmıştır.
- JS metriği, dağılımlar arasındaki genel benzerliği kararlı ve dengeli bir biçimde ölçmek amacıyla tercih edilmiştir.
- KL metriği, asimetrik farklılıkları ve daha önce gözlemlenmemiş veya nadir davranışları yakalamada etkili olduğu için seçilmiştir.
- Wasserstein mesafesi, sayısal özniteliklerde meydana gelen geometrik kaymaları analiz etmek için kullanılmıştır.
- Tüm ölçütlerin birlikte değerlendirilmesi, dönüştürülmesi veya veri setinden çıkarılması gereken kararsız özniteliklerin belirlenmesini sağlaması amacıyla kullanılmıştır.

Tablo 7.8, farklı dağılım kayması ölçütleri için kullanılan şiddet eşiklerini özetlemektedir. KS istatistiği maksimum yapısal farkları, KL ve JS divergence ölçütleri dağılımlar arasındaki bilgi temelli farklılıkları, Wasserstein mesafesi ise sayısal özniteliklerdeki geometrik kaymaları nicel olarak değerlendirmeye olanak sağlamaktadır. Bu eşikler,

öznitelik bazlı dağılım kararlılığının sistematik bir şekilde yorumlanmasını mümkün hale getirmiştir.

Tablo 7.8. Yayılım Şiddet Düzeyleri için Eşik Değerler

Drift Seviyesi	**KS**	**KL**	**JS**	**Wasserstein**
Düşük	$< 0,1$	$< 0,1$	$< 0,05$	$< 0,1$
Orta	$0,1$–$0,3$	$0,1$–1	$0,05$–$0,2$	$< 0,1 - 0,3$
Yüksek	$0,3$–$0,6$	1–5	$0,2$–$0,4$	$< 0,3 - 0,6$
Şiddetli	$> 0,6$	> 5	$> 0,4$	$> 0,6$

Bu ölçütlerden faydalanılarak, sonraki bölümde detaylıca açıklandığı üzere, yayılım kayması analizi yapılmıştır.

7.3.2. Yayılım Kayması Analizleri

Bölüm 7.3.'te açıklanan metrikler kullanılarak, 3 farklı deney grubu için yayılım kayması analizi yapılmıştır. Bu deney grupları şu şekilde özetlenebilir:

- BoT-UNSW eğitim, ToN_IoT Test
- ToN-UNSW eğitim, BoT_IoT Test
- BoT-ToN eğitim, UNSW-NB15 Test

Bu bölümde, genel akışın bozulmaması, ve grafiksel sunumun gereğinden fazla yer kaplamaması adına, yukarı durumlardan sadece, BoT-UNSW veri setilerinin birleşiminden oluşan eğitim seti ve ToN_IoT veri setinin test seti olarak kullanıldığı durum incelencektir. Bu durum, genel olarak, diğer deney grupları için yapılabilecek analizlerin tamamına uygundur. Diğer veri setleri için elde edilen grafiksel gösterimler, Ek-A'te detaylı olarak paylaşılmıştır.

Yayılım analizi için ilk aşama olarak, farklı yayılım analiz metriklerine göre elde edilen sonuçlardır. Bu sonuçlar, Tablo 7.9 ve Şekil 7.2'de gösterilmiştir.

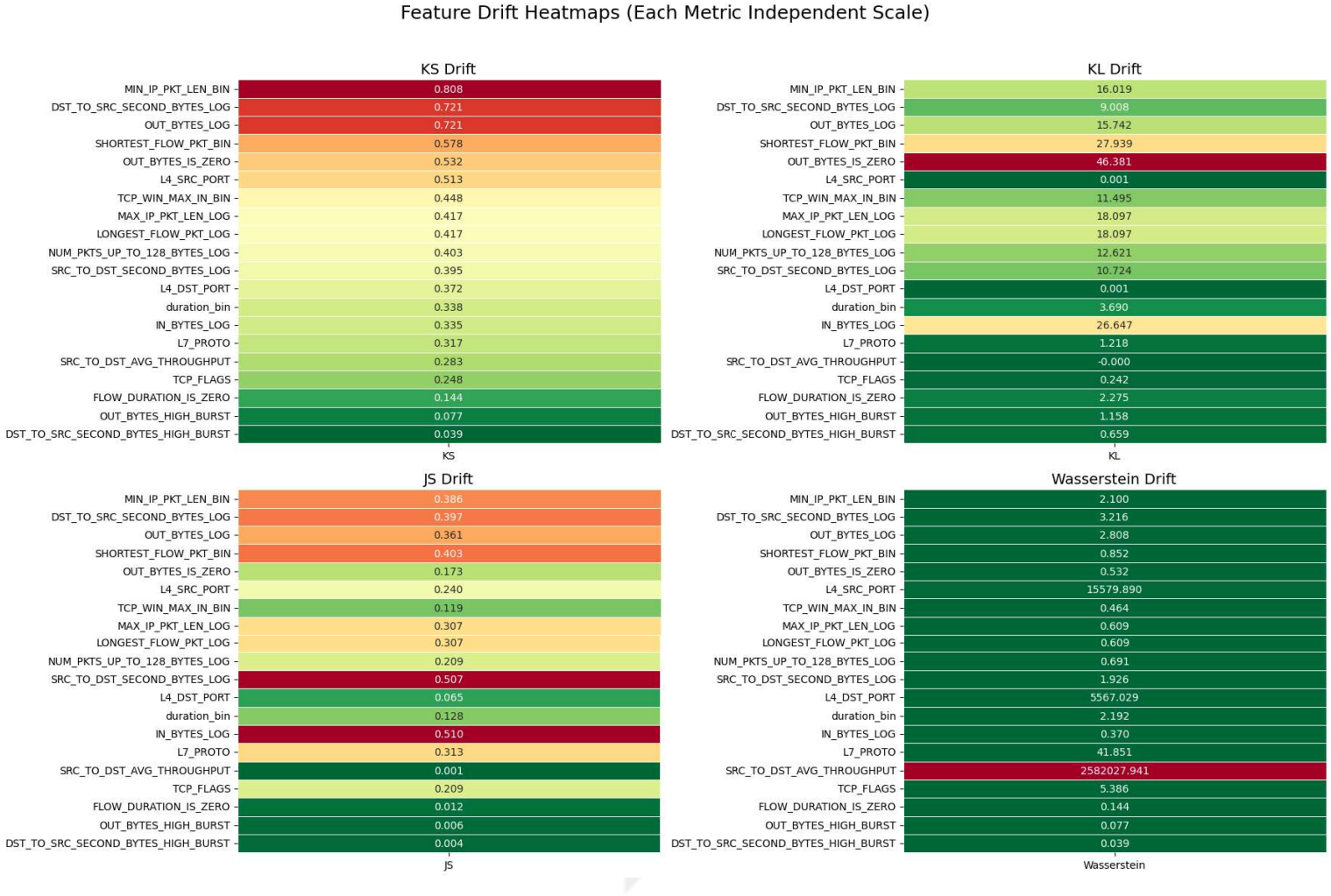
Tablo 7.9, farklı özniteliklerin dağılımsal sapmalarını dört farklı istatistiksel metrik üzerinden (Kolmogorov–Smirnov (KS), Kullback–Leibler (KL), Jensen–Shannon (JS) ve Wasserstein) analiz etmektedir. Burada, kaynak ve hedef veri dağılımları arasında

neredeyse tüm öznitelikler için anlamlı düzeyde bir kayma gözlemlenerek, modelin farklı ortamlarda uygulanabilirliğinin yitirebileceğini göstermektedir.

Tablo 7.9. Öznitelik Düzeyinde Yayılım Metrikleri

Feature	**KS**	**KL**	**JS**	**Wasserstein**
MIN_IP_PKT_LEN_BIN	0,8084	16,0188	0,3857	2,1003
DST_TO_SRC_SECOND_ BYTES_LOG	0,7212	9,0025	0,3602	3,2155
OUT_BYTES LOG	0,7211	15,7417	0,3607	2,0716
SHORTEST FLOW_PKT_BIN	0,5777	27,9394	0,4032	2,9368
OUT_BYTES IS_ZERO	0,5315	36,4804	0,1797	0,5315
L4_SRC PORT	0,5129	0,0014	0,2397	15579,8899
TCP_WIN MAX_IN_BIN	0,4479	11,4953	0,1186	0,4646
MAX_IP PKT_LEN_LOG	0,4172	18,7963	0,1675	1,2089
LONGEST FLOW_PKT_LOG	0,4172	11,0865	0,3070	2,0828
NUM_PKTS_UP_TO 128_BYTES_LOG	0,4029	12,8595	0,1695	0,6641
SRC_TO_DST SECOND_BYTES_LOG	0,3951	10,7240	0,5074	0,9576
L4_DST PORT	0,3727	0,0005	0,0645	5567,6204
DURATION_BIN	0,3378	3,6909	0,1283	2,1921
IN_BYTES LOG	0,3351	26,6470	0,5103	2,3413
L7 PROTO	0,3171	1,1218	0,3127	41,8513
SRC_TO_DST AVG_THROUGHPUT	0,2830	0,0000	0,0008	258202,4978
TCP FLAGS	0,2478	0,2415	0,2805	5,3837
FLOW DURATION_IS_ZERO	0,1441	2,2573	0,1711	0,1711
OUT_BYTES HIGH_BURST	0,0765	1,1577	0,0602	0,0765
DST_TO_SRC_SECOND_ BYTES_HIGH_BURST	0,0394	0,6586	0,0036	0,0394

Özellikle MIN_IP_PKT_LEN_BIN özniteliği; KS (0,8084), KL (16,0188) ve JS (0,3857) değerleriyle dikkat çekici bir sapma sergilerken, benzer şekilde SHORTEST_FLOW_PKT_BIN için KL değeri 27,9394, JS ise 0,4032 olarak hesaplanmıştır. Bu iki öznitelik, ağ trafiğinin paket düzeyindeki davranışlarında belirgin değişimlere işaret etmektedir.



Şekil 7.2. BoT-UNSW ve ToN_IoT için Yayılım Metrikleri

OUT_BYTES_LOG ve DST_TO_SRC_SECOND_BYTES_LOG gibi çok farklı trafik hacmini temsil eden değişkenler, sırasıyla KL: 15,7417 ve 9,0025; JS: 0,3607 ve 0,3602 gibi yüksek değerler göstermiştir. Bu da kaynak ve hedef veri arasında ağ yüklerinin önemli ölçüde farklılaştığını göstermektedir. Öte yandan, OUT_BYTES_IS_ZERO (KL: 36,4804), IN_BYTES_LOG (KL: 26,6470), ve FLOW_DURATION_IS_ZERO gibi binary temelli özniteliklerde de anlamlı düzeyde drift tespit edilmiştir.

Bazı öznitelikler ise özellikle Wasserstein mesafesi üzerinden ciddi sapmalar göstermektedir. Örneğin, SRC_TO_DST_AVG_THROUGHPUT için KL ve JS neredeyse sıfıra yakınken, Wasserstein değeri 258202,4978 olarak hesaplanmıştır. Aynı şekilde L4_SRC_PORT (15579,8899) ve L4_DST_PORT (5567,6204) gibi öznitelikler de entropi tabanlı metriklerle fark edilmesi zor olan, ancak değer düzeyinde ciddi yer değiştirmeler yaşanan özelliklerdir. Bu da yalnızca dağılım benzerliği değil, değerlerin konumsal hareketliliğinin de değerlendirilmesinin gerektiğini ortaya koymaktadır.

L7_PROTO, TCP_FLAGS, DURATION_BIN ve NUM_PKTS_UP_TO_128_BYTES gibi protokol, bayrak ve süre temelli özniteliklerde ise orta düzeyde sapmalar tespit edilmiştir (örneğin L7_PROTO için KL: 1,1218, JS: 0,3127, Wasserstein: 41,8513). Bu öznitelikler, protokol kullanımının ve oturum sürelerinin değiştiğini göstererek model davranışını dolaylı yoldan etkileyebilir.

Daha düşük sapma gösteren sınırlı sayıdaki öznitelikler, düşük metrik değerleri ile göreceli stabilite sergilemektedir. Ancak sistem genelinde bu tür öznitelikler istisna niteliğindedir (örn. OUT_BYTES_HIGH_BURST).

Sonuç olarak, bu analiz veri setinde geniş çaplı ve çok boyutlu yayılım kayması olduğunu göstermektedir.

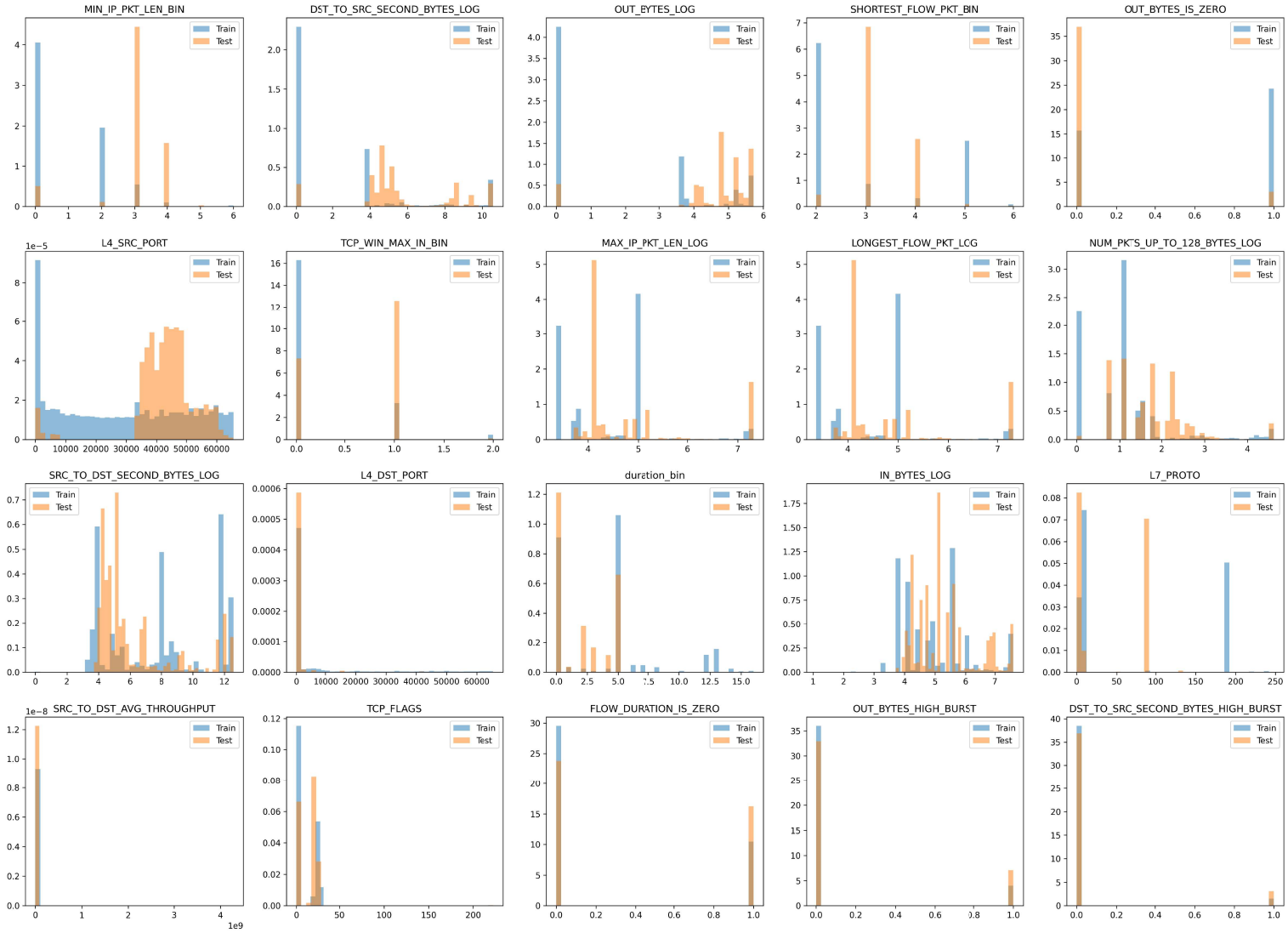
Özniteliklerin hem dağılım yapılarında (KS, JS) hem de değer düzeyinde (Wasserstein) belirgin değişiklikler saptanmış; KL değerleri ise bilgi teorisi açısından anlamlı farklılıkları ortaya koymuştur. Bu bulgular, mevcut veriyle eğitilen bir modelin hedef ortamlarda geçerliliğini ve performansını sürdüremeyebileceğini, özellikle küresel ölçekte dağıtılmış bir IoT-IDS için ciddi bir risk oluşturduğunu göstermektedir.

Modelleme sürecinde özniteliklerin dağılımlarının veri kümeleri arasında tutarlılık gösterip göstermediğini incelemek amacıyla, eğitim verisi olarak kullanılan BoT-IoT/UNSW ve test verisi olan ToN_IoT kümeleri için çeşitli görselleştirme yöntemleriyle analiz gerçekleştirilmiştir. Bu bağlamda, en yüksek yayılım kayması değerine sahip ilk 20 öznitelik, sırasıyla histogram, ECDF, KDE ve KDE farkı grafiklerine tabi tutulmuştur. Her bir görselleştirme, farklı boyuttaki sapmaları daha görünür kılarak dağılımsal değişikliklerin nitelikli bir biçimde incelenmesine olanak sağlamaktadır. BoT-UNSW için görsel analizler, Şekil 7.3 - Şekil 7.6 arasında gösterilmiştir.

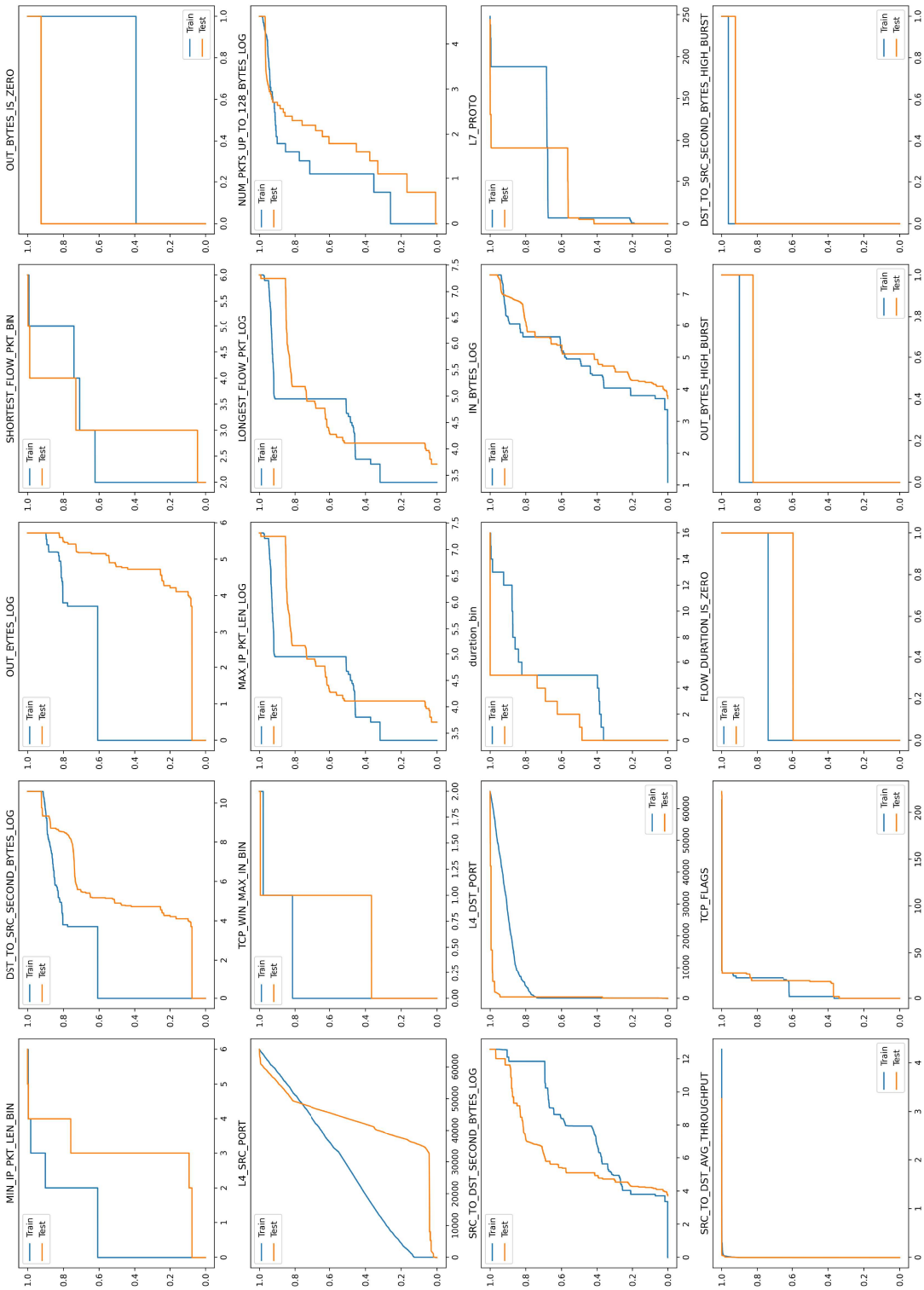
Şekil 7.3'te sunulan histogram grafiklerinde, bazı öznitelikler için dağılımın modlarının ve yoğunluklarının ciddi biçimde değiştiği tespit edilmiştir. Örneğin, OUT_BYTES_LOG, IN_BYTES_LOG ve SHORTEST_FLOW_PKT_BIN gibi öznitelikler, eğitim setinde daha geniş bir aralıkta dağılmışken; test setinde tekil veya yoğunlaşmış aralıklarda toplanmıştır. Bu durum, test verisinin belirli alt protokollerde veya ağ davranışlarında yoğunlaştığını ve modelin karar sınırlarının bu alanlarda zorlanabileceğini göstermektedir. Ayrıca TCP_WIN_MAX_IN_BIN gibi bazı özniteliklerde eğitim setinde neredeyse hiç gözlem bulunmazken, test setinde bu değerler baskın hale gelmiştir. Bu tarz değişiklikler, eğitimde gözlenmeyen yeni trafik tiplerinin testte sıkça görüldüğünü ve kavramsal sapmanın mevcut olduğunu göstermektedir.

Kümülatif dağılım fonksiyonlarının sunulduğu Şekil 7.4'te ise, dağılım farklılıklarının daha global bir perspektiften analiz edilmesini sağlamaktadır. Özellikle DST_TO_SRC_SECOND_BYTES_LOG, MAX_IP_PKT_LEN_LOG, SRC_TO_DST_SECOND _BYTES_LOG gibi özniteliklerde ECDF eğrilerinin birbirinden uzak seyretmesi ve eğrilerin paralel olmaktan uzak şekilde sapmalar göstermesi, bu özniteliklerin yayılım davranışlarının iki küme arasında tutarsız olduğunu açıkça ortaya koymaktadır.

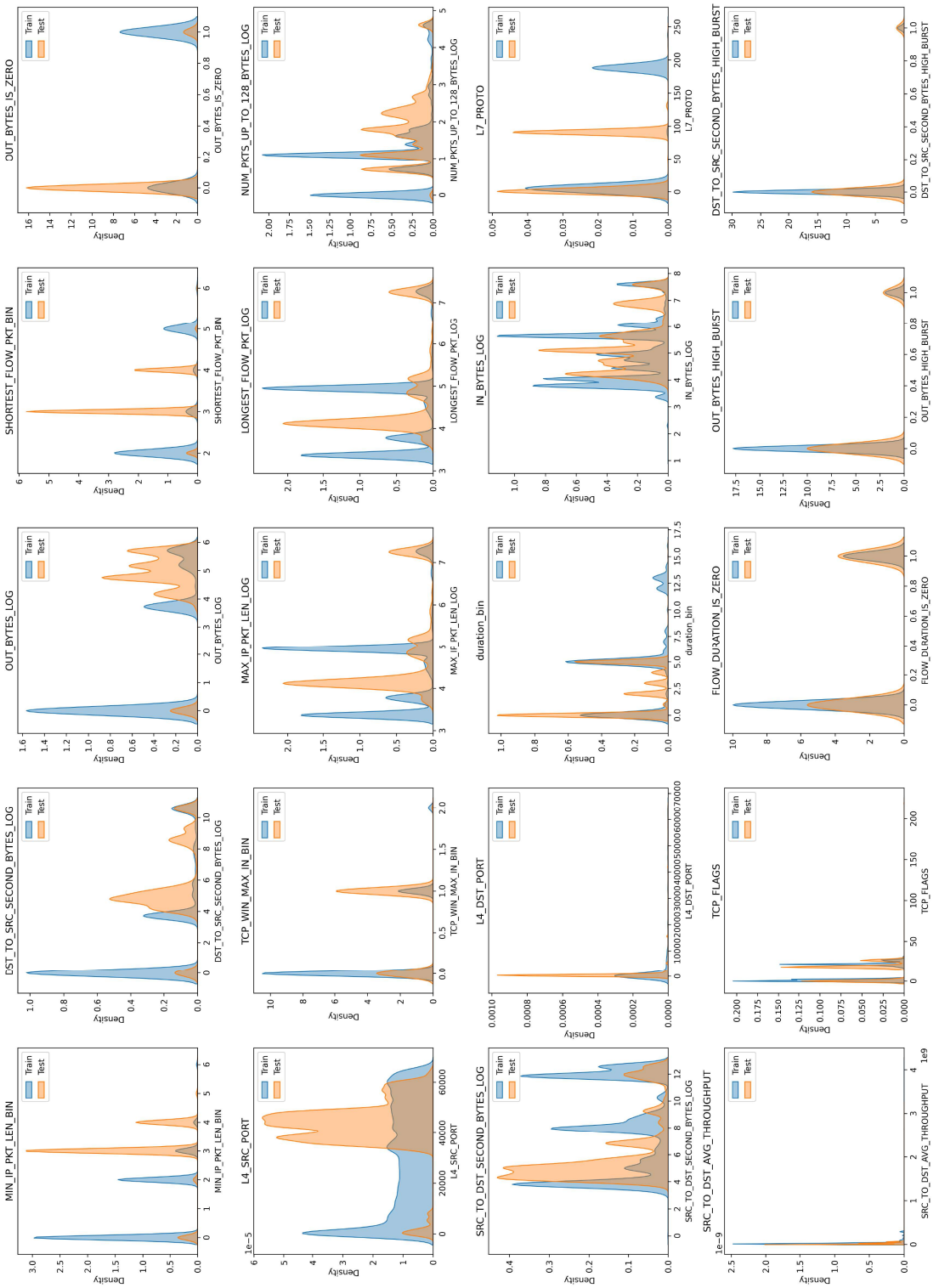
FLOW_DURATION_IS_ZERO gibi ikili özniteliklerde dahi test ve eğitim setleri arasında kümülatif yapı farklılıkları mevcuttur. Bu durum, kavramsal benzerliğin sadece sürekli değişkenlerde değil, kategorik/ikili değişkenlerde de bozulduğunu göstermektedir.



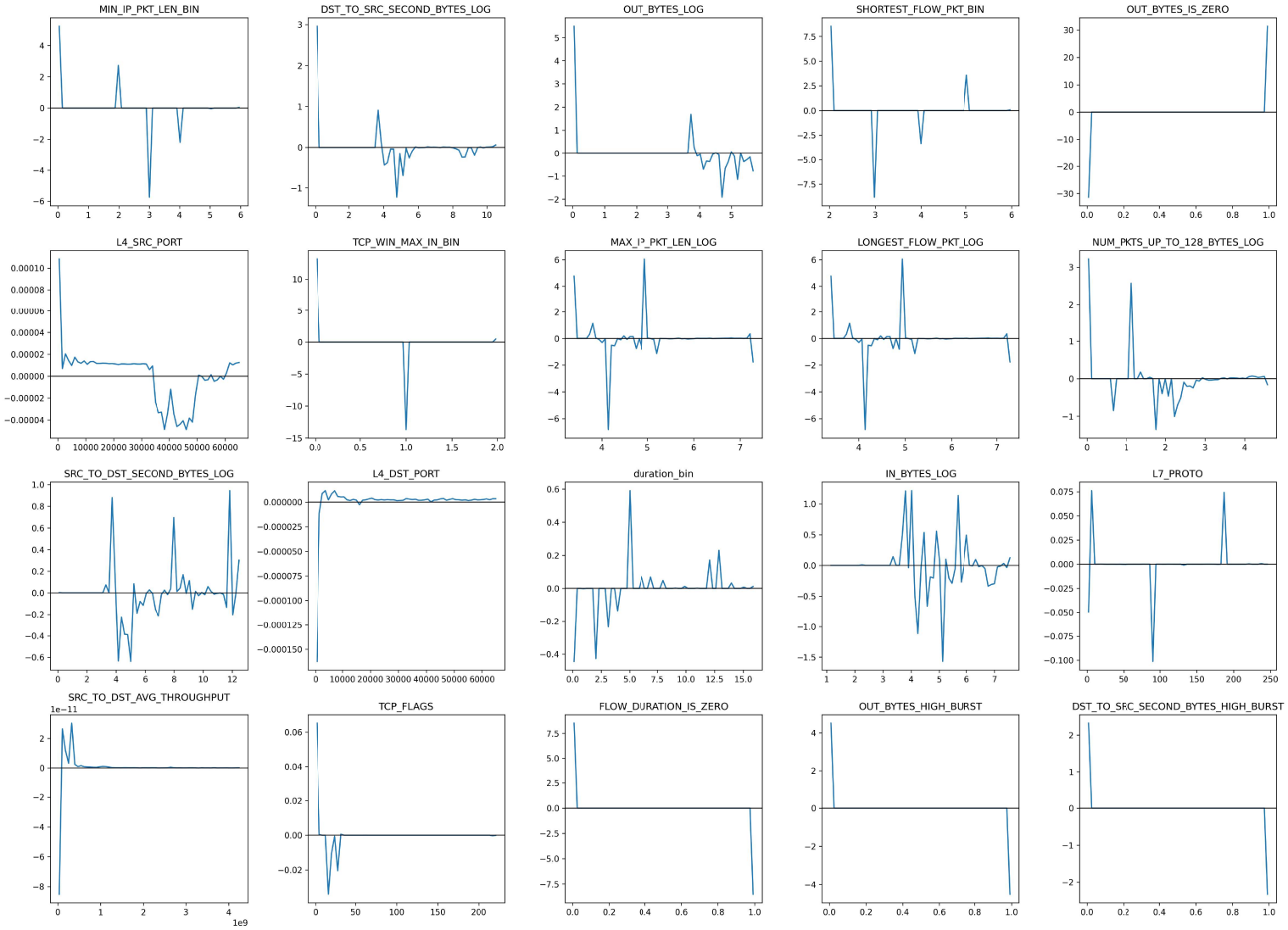
Şekil 7.3. BoT-UNSW ve ToN_IoT için Histogramlar



Şekil 7.4. BoT-UNSW ve ToN_IoT için ECDF Grafikleri



Şekil 7.5. BoT-UNSW ve ToN_IoT için KDE Grafikleri



Şekil 7.6. BoT-UNSW Eğitim ve ToN_IoT için KDE Farkları

Yoğunluk dağılım eğrileri üzerinden gerçekleştirilen Şekil 7.5'deki analizlerde ise çok modlu yapılar dikkat çekmektedir. Özellikle LONGEST_FLOW_PKT_LOG, MAX_IP_PKT_LEN_LOG ve NUM_PKTS_UP_TO_128_BYTES_LOG gibi özniteliklerde eğitim setinde görece daha düz ve yayılmış yoğunluklar gözlenirken, test setinde belirli değerlerde yüksek yoğunluklu pikler oluşmuştur. Bu tarz değişimler, test verisinin çok daha dar ama tekrar eden desenlere sahip olduğunu göstermekte ve bu durum modelin esneklik gerektiren tahmin yetisini kısıtlayabilir. Ayrıca SRC_TO_DST_AVG_THROUGHPUT gibi özniteliklerde eğitim ve test setlerinin yoğunluk eğrilerinin neredeyse hiç kesişmemesi, modelin genel karar sınırlarının bu tür özniteliklerde tamamen işlevsiz kalabileceğine işaret etmektedir.

Son olarak Şekil 7.6'da sunulan KDE fark grafikleri, eğitim ve test seti yoğunlukları arasındaki farkın doğrudan gösterildiği, ayrık sapmaların tespitine uygun grafiklerdir. Bu görsellerde özellikle TCP_FLAGS, L7_PROTO, L4_DST_PORT gibi özniteliklerde yoğunluk farkı eğrileri sıfır çizgisinin çok üstünde ve altında kalmakta; yani iki dağılımın birçok bölgede kesişmediği anlaşılmaktadır. Ayrıca, fark eğrilerinin pozitiften negatife sık geçiş yapması, öznitelik değerlerinin hem konum hem de yayılım açısından tüm aralık boyunca değiştiğini göstermektedir. Özellikle OUT_BYTES_IS_ZERO ve DST_TO_SRC_SECOND_BYTES_HIGH_BURST gibi ikili ya da sınırlı değer aralığına sahip özniteliklerde bile yoğunluk farkının belirgin olması, en temel özniteliklerde bile drift probleminin göz ardı edilemeyecek düzeyde olduğunu göstermektedir.

Sonuç olarak, eğitim ve test veri kümeleri arasında gerçekleştirilen tüm bu görsel analizler, sadece metrik temelli değil, aynı zamanda dağılım biçimi ve yoğunluk karakteri açısından da anlamlı farklılıkların bulunduğunu ortaya koymaktadır. Her bir grafik türü farklı bir perspektiften bu sapmaları desteklemekte ve birbirini tamamlayıcı sonuçlar sunmaktadır. Elde edilen bulgular doğrultusunda, öznitelik düzeyinde gerçekleşen bu yayılım kaymasının (drift shift), modelleme performansı üzerinde olumsuz etkiler oluşturabileceği; özellikle genel amaçlı ve küresel ölçekli çalışan bir IoT tabanlı saldırı tespit sistemi (IoT-IDS) için ciddi bir tehdit teşkil edebileceği açıkça ortaya konmaktadır.

7.4. Geleneksel Yöntemler için Sonuçlar

Bu bölümde, önceki bölümlerde detaylı biçimde analiz edilen yayılım kaymasının (drift/shift), geleneksel makine öğrenmesi tabanlı sınıflandırma modellerinin performansı üzerindeki etkileri nicel olarak incelenmektedir. Bölüm 7.3.2.'de sunulan dağılım kayması analizleri; eğitim ve test veri kümeleri arasında, hem öznitelik dağılımları hem de değer düzeyinde ciddi farklılıklar bulunduğunu ortaya koymuştur. Bu durumun, modellerin genelleme yeteneği üzerinde doğrudan ve olumsuz bir etki oluşturması beklenmektedir.

Bu kapsamda, farklı veri seti kombinasyonları ve saldırı senaryoları için elde edilen sınıflandırma sonuçları, Eğitim ROC-AUC, Test ROC-AUC, kesinlik (Prec.), hassasiyet (Rec.) ve F1 skoru metrikleri üzerinden değerlendirilmiştir. Sunulan sonuç tablolarında, eğitim ve test ROC-AUC değerleri arasındaki farklar, modelin yayılım kaymasına karşı ne ölçüde dayanıklı olduğunu gösteren temel bir gösterge olarak ele alınmıştır. Ardından, sınıf bazlı kesinlik, hassasiyet ve F1 skorları incelenerek, hatalı sınıflandırma davranışlarının hangi senaryolarda yoğunlaştığı analiz edilmiştir. Sınıflandırma için RF algoritmasının, daha güncel ve yayılım kaymaya karşı daha dirençli bir versiyonu olan, Extra Ağaçlar (ExtraTrees) algoritması kullanılmıştır (Geurts ve diğ., 2006). Extra Trees algoritmasının temel farkı, karar ağaçlarında bölünme (split) noktalarını optimize ederek bulmak yerine rastgele seçmesidir. Random Forest'ta her düğümde en iyi split aranırken, Extra Ağaçlar'da hem öznitelikler hem de split noktaları rastgele belirlenir. Bu yaklaşım rastgeleliği artırarak varyansı düşürür, eğitimi hızlandırır ve gürültülü verilerde daha iyi genelleme sağlayabilir. Deneyler sklearn kütüphanesindeki ExtraTrees implementasyonu kullanılarak gerçekleştirilmiştir. Deneylerde, eğitim seti üzerinde 10-kat çapraz geçerleme kullanılırken, ilgili deneyler sırasıyla şu şekildedir:

- Eğitim: BoT-Ton, Test: UNSW-NB15
- Eğitim: BoT-UNSW, Test: ToN_IoT
- Eğitim: ToN-UNSW, Test: BoT_IoT

Model eğitimleri gerçekleştirilirken, en iyi hiperparametre uzayının bulunması için Bayes Arama (Bayesian Search) kullanılmış olup, 20 iterasyon yapılmak üzere,ilgili model eğitimlerinde kullanılan hiperparametre uzayı Tablo 7.10'da sunulmuştur.

Tablo 7.10. ExtraTrees için Hiperparametre Arama Uzayı

Hiperparametre	**Açıklama**	**Aralık / Değerler**
n_estimators	Ormanda oluşturulacak ağaç sayısını ifade etmektedir. Daha fazla ağaç genellikle daha kararlı sonuçlar sağlar.	300 – 700
max_depth	Her bir karar ağacının alabileceği maksimum derinliği belirler. Aşırı öğrenmeyi (overfitting) sınırlandırmak için kullanılmıştır.	3 – 7
min_samples_split	Bir düğümün bölünebilmesi için gereken minimum örnek sayısını tanımlar. Büyük değerler daha sade ağaçlar üretmektedir.	30 – 150
min_samples_leaf	Bir yaprak düğümde bulunması gereken minimum örnek sayısını belirtir ve modelin genelleme yeteneğini artırır.	20 – 80
max_features	Her düğümde bölünme için değerlendirilecek öznitelik sayısının seçilme yöntemini belirler.	log2, sqrt

Deneylere ait sonuçlar, Tablo 7.11 - Tablo 7.14'te sunulurken, farklı veri setleri - sınıf kombinasyonları için karışıklık matrisleri ise, Ek A altında, Şekil A.11 - Şekil A.13 arasında sunulmuştur.

Tablo 7.11'de Benign ve DoS sınıfları için elde edilen sonuçlar sunulmaktadır. BoT+ToN eğitimli modelde, Eğitim ROC-AUC (0,999) ve Test ROC-AUC (0,989) değerlerinin birbirinden oldukça farklı olması, bu senaryoda yayılım kaymasının fazla olduğunu ve modelin genelleme yeteneğini büyük ölçüde kaybettiğini göstermektedir. Bu durum, Benign sınıfı için yüksek kesinlik (0,913), hassasiyet (0,995) ve F1 skoru (0,953) değerlerine sahip olsa da, DoS sınıfı için hassasiyetin son derece düşük olması (0,002), modelin saldırı örneklerinin büyük bir kısmını Benign olarak sınıflandırdığını

göstermektedir. Bu durum, sınıf dengesizliği ve saldırı örneklerinin dağılımının eğitim sürecinde yeterince temsil edilememesi ile ilişkilendirilebilir. BoT+UNSW ve ToN+UNSW senaryolarında da benzer şekilde Test ROC-AUC değerlerinde belirgin düşüşler gözlemlenmektedir (sırasıyla 0,211 ve 0,028).

Tablo 7.11. Benign ve DoS için Test Sonuçları

Eğitim	Sınıf	Prec.	Rec.	F1	Eğitim AUC	Test AUC
BoT+ToN	Benign	0,913	0,995	0,953	0,999	0,596
	DoS	0,034	0,002	0,003		
BoT+UNSW	Benign	0,726	0,982	0,835	0,999	0,211
	DoS	0,022	0,001	0,002		
ToN+UNSW	Benign	0,130	0,253	0,172	0,992	0,028
	DoS	0,000	0,000	0,000		

Özellikle ToN+UNSW senaryosunda, hem Benign hem de DoS sınıfları için kesinlik, hassasiyet ve F1 skorlarının oldukça düşük seviyelerde kalması, yayılım kaymasının model performansını ciddi biçimde olumsuz etkilediğini göstermektedir. Bu sonuçlar, DoS saldırı davranışlarının veri setleri arasında yüksek varyans gösterdiğini ve modelin bu değişimi yakalayamadığını ortaya koymaktadır.

Tablo 7.12. Benign ve DDoS için Test Sonuçları

Eğitim	Sınıf	Prec.	Rec.	F1	Eğitim AUC	Test AUC
BoT+UNSW	Benign	0,302	1,000	0,464	0,999	0,656
	DDoS	0,000	0,000	0,000		
ToN+UNSW	Benign	0,176	0,296	0,221	0,999	0,308
	DDoS	0,307	0,184	0,230		

Tablo 7.12'de, Benign ve DDoS sınıfları için farklı eğitim–test veri seti kombinasyonlarında elde edilen sonuçlar sunulmaktadır. BoT+UNSW eğitimli model incelendiğinde, Eğitim ROC-AUC değerinin oldukça yüksek (0,999) olmasına karşın, Test ROC-AUC değerinin ciddi biçimde düşerek 0,656 seviyesine gerilediği görülmektedir. Bu durum, modelin eğitim verisi üzerinde güçlü bir ayrım yeteneği

kazanmasına rağmen, hedef test ortamında bu yeteneği koruyamadığını açıkça göstermektedir.

Sınıf bazlı metrikler incelendiğinde, Benign sınıfı için BoT+UNSW veri setinde hassasiyet değerinin 1,000 olması, modelin test setindeki neredeyse tüm örnekleri Benign olarak etiketlediğini göstermektedir. Buna karşın, kesinlik değerinin düşük (0,302) olması, Benign tahminlerinin büyük bir kısmının gerçekte DDoS sınıfına ait olduğunu ortaya koymaktadır. DDoS sınıfı için ise kesinlik, hassasiyet ve F1 skoru değerlerinin tamamının sıfıra yakın olması, modelin bu sınıfı test ortamında neredeyse hiç yakalayamadığını göstermektedir. Bu durum, yayılım kayması nedeniyle saldırı davranışlarının model tarafından tanınamadığını açıkça ortaya koymaktadır.

Benzer şekilde, ToN+UNSW eğitimli modelde de Eğitim ROC-AUC değeri yüksek olmasına rağmen (0,999), Test ROC-AUC değeri 0,308 seviyesinde kalmıştır. Bu senaryoda DDoS sınıfı için hassasiyet (0,184) ve F1 skoru (0,230) değerleri bir miktar iyileşme göstermiş olsa da, kesinlik–hassasiyet dengesinin hâlen zayıf olduğu ve modelin kararlı bir ayrım gerçekleştiremediği görülmektedir. Bu bulgular, DDoS saldırı desenlerinin veri setleri arasında ciddi biçimde değiştiğini ve modelin bu değişime uyum sağlayamadığını göstermektedir.

Tablo 7.13. Benign ve DDoS-DoS için Test Sonuçları

Eğitim	**Sınıf**	**Prec.**	**Rec.**	**F1**	**Eğitim AUC**	**Test AUC**
BoT+UNSW	Benign	0,269	0,986	0,423	0,999	0,674
	DDoS-DoS	0,298	0,002	0,004		
ToN+UNSW	Benign	0,091	0,214	0,128	0,997	0,137
	DDoS-DoS	0,613	0,368	0,460		

Tablo 7.13'te, DDoS ve DoS saldırılarının tek bir sınıf altında birleştirildiği senaryoya ait sonuçlar sunulmaktadır. BoT+UNSW eğitimli modelde Eğitim ROC-AUC değerinin yüksek olmasına karşın (0,999), Test ROC-AUC değerinin 0,674 seviyesinde kalması, birleşik saldırı sınıfının hedef ortamda ayırt edilemediğini göstermektedir.

BoT-UNSW veri setinde, Benign sınıfı için hassasiyet oldukça yüksek (0,986) iken,

kesinlik ve F1 skoru değerleri düşüktür. DDoS–DoS sınıfı için ise hassasiyetin neredeyse sıfıra yakın olması, modelin saldırı sınıfını sistematik biçimde göz ardı ettiğini göstermektedir. Bu durum, farklı saldırı türlerinin tek bir sınıf altında toplanmasının, yayılım kayması altında ayrım gücünü artırmak yerine, sınıf içi varyansı artırarak performansı daha da düşürdüğünü göstermektedir.

ToN+UNSW senaryosunda ise, Test ROC-AUC değeri tamamen çökerek 0,137 değerini almıştır. Burada da her iki sınıf içinde hem kesinlik, hassasiyet, ve f1-değerindeki düşüşler yüksek bir dağılım kaymasının olduğunu göstermektedir.

Tablo 7.14. Benign ve Reconnaissance için Test Sonuçları

Eğitim	Sınıf	Prec.	Rec.	F1	Eğitim AUC	Test AUC
BoT+ToN	Benign	0,707	0,249	0,369	0,995	0,369
	Reconnaissance	0,124	0,507	0,199		
BoT+UNSW	Benign	0,998	0,908	0,951	0,990	0,958
	Reconnaissance	0,264	0,990	0,413		
ToN+UNSW	Benign	0,634	0,133	0,220	0,998	0,746
	Reconnaissance	0,308	0,835	0,450		

Tablo 7.14'te Benign ve Reconnaissance sınıfları için elde edilen sonuçlar sunulmaktadır. BoT+ToN senaryosunda Eğitim ROC-AUC (0,995) ile Test ROC-AUC (0,369) arasındaki ciddi fark, Reconnaissance saldırılarının davranışsal olarak veri setleri arasında önemli ölçüde değiştiğini göstermektedir. Bu senaryoda Reconnaissance sınıfı için hassasiyet görece yüksek (0,507) olmasına rağmen, kesinlik ve F1 skoru değerlerinin düşük olması, yanlış pozitif oranlarının yüksekliğine işaret etmektedir.

BoT+UNSW eğitimli modelde ise Test ROC-AUC değerinin 0,958 seviyesinde olması, Reconnaissance saldırılarının bu veri seti kombinasyonunda daha tutarlı desenler sergilediğini göstermektedir. Bu durum, Reconnaissance sınıfı için yüksek hassasiyet (0,990) ve F1 skoru (0,413) değerleri ile de desteklenmektedir.

ToN+UNSW senaryosunda ise Test ROC-AUC değerinin 0,746'ya düşmesiyle birlikte, sınıf bazlı metriklerde belirgin bir dengesizlik gözlemlenmektedir. Reconnaissance sınıfı

için hassasiyet yüksek olmasına rağmen (0,835), kesinliğin düşük kalması (0,308), modelin bu sınıfı aşırı tahmin ettiğini ve yayılım kaymasının karar sınırlarını bozduğunu göstermektedir.

Bu bölümde elde edilen bulgular, klasik makine öğrenmesi temelli yaklaşımların, veri setleri arasında oluşan yayılım kayması (distribution drift) karşısında kararlı ve genellenebilir bir performans sergileyemediğini açıkça ortaya koymaktadır. Eğitim aşamasında yüksek ROC-AUC değerlerine ulaşılmasına rağmen, test ortamında bu başarımın korunamaması, modellerin büyük ölçüde eğitim verisine özgü istatistiksel örüntülere bağımlı kaldığını ve hedef alanın dinamik yapısına uyum sağlayamadığını göstermektedir. Özellikle saldırı sınıflarına ait hassasiyet ve F1 skoru değerlerinde gözlemlenen ciddi düşüşler, klasik yöntemlerin sabit karar sınırları ve sınırlı temsil gücü nedeniyle dağılım farklılıklarını yakalamakta yetersiz kaldığını ortaya koymaktadır.

Bu bulgular doğrultusunda, yayılım kaymasının etkilerini azaltabilen, daha esnek ve hiyerarşik temsiller öğrenebilen yöntemlerin incelenmesi kaçınılmaz hâle gelmektedir. Bu nedenle, bir sonraki bölümde derin öğrenme tabanlı metodolojiler ele alınacak; bu yaklaşımların karmaşık saldırı örüntülerini ve veri setleri arası dağılım farklılıklarını ne ölçüde modelleyebildiği detaylı deneysel analizlerle değerlendirilecektir.

7.5. Derin Öğrenme için Sonuçlar

Geleneksel makine öğrenmesini takiben, değişken kaymasına derin öğrenmenin nasıl tepki verdiğini anlamak için, geleneksel öğrenme deneylerinde kullanılan eğitim-test seti kombinasyonları, bir DL akışı ile de test edilmiş ve sonuçlar elde edilmiştir.

Derin öğrenme tabanlı deneylerde kullanılan model, tabular veri yapısına uygun olacak şekilde tasarlanmış, tam bağlantılı (fully-connected) katmanlardan oluşan ileri beslemeli bir sinir ağı (Feed Forward Neural Network) mimarisidir. Modelin giriş katmanı, veri setindeki öznitelik sayısına bağlı olarak dinamik biçimde yapılandırılmış ve her bir örnek, sayısal öznitelik vektörü olarak modele beslenmiştir.

Gizli katmanlar, doğrusal olmayan ilişkilerin öğrenilebilmesi amacıyla doğrusal olmayan aktivasyon fonksiyonları (activation function) ile donatılmıştır. Bu katmanlar, modelin

karmaşık örüntüleri yakalayabilmesini sağlarken, aşırı öğrenmenin önüne geçmek amacıyla düzenlileştirme mekanizmaları ile desteklenmiştir. Eğitim sürecinde modelin genelleme yeteneğini artırmak ve değişken kayması altında daha kararlı bir davranış sergilemesini sağlamak amacıyla katmanlar arası temsillerin aşırı uyum göstermesi bilinçli olarak sınırlandırılmıştır.

Çıkış katmanı, ikili sınıflandırma problemine uygun olacak şekilde yapılandırılmış olup, model çıktısı bir sınıfa ait olasılık değeri olarak üretilmiştir. Bu yapı sayesinde, yalnızca sınıf etiketi tahmini değil, aynı zamanda eşik bağımlı performans ölçütlerinin (örneğin AUC, precision ve recall) hesaplanmasına da olanak sağlanmıştır.

Model parametreleri, geri yayılım algoritması kullanılarak optimize edilmiş ve eğitim süreci boyunca doğrulama performansı yakından izlenmiştir. Bu mimari tercih, derin ve karmaşık ağ yapılarına kıyasla daha sade bir öğrenme çerçevesi sunarak, derin öğrenme modellerinin değişken kayması karşısındaki davranışlarının daha şeffaf ve karşılaştırılabilir bir biçimde analiz edilmesini hedeflemektedir.

Model performansının değerlendirilmesinde, kayıp (loss) ve doğruluk (accuracy) ölçütlerine ek olarak, sınıf dengesizliğinin etkisini daha iyi yansıtabilen AUC, precision, recall ve F1-skoru metrikleri raporlanmıştır. Ayrıca, saldırı ve normal trafik sınıflarının ayrımındaki başarımı daha ayrıntılı incelemek amacıyla true positive (TP), true negative (TN), false positive (FP) ve false negative (FN) değerleri hesaplanmış; bu değerlerden türetilen özgüllük (specificity / TNR) ve yanlış pozitif oranı (false positive rate / FPR) gibi ikincil performans ölçütleri de analiz edilmiştir.

Tablolar 7.15–7.18'de, farklı veri seti kombinasyonları altında gerçekleştirilen derin öğrenme tabanlı ikili sınıflandırma deneylerinin test sonuçları sunulmaktadır. Elde edilen bulgular, derin öğrenme modellerinin, geleneksel modellere kıyasla, yüksek temsil gücüne rağmen, veri setleri arasındaki dağılımsal farklılıklara karşı her senaryoda tutarlı bir genelleme başarımı sergileyemediğini ve değişken kaymasından yüksek oranda etkilendiğini göstermektedir.

Tablo 7.15'te, Benign ve DoS sınıfları için farklı eğitim veri seti kombinasyonları altında

elde edilen derin öğrenme tabanlı test sonuçları sunulmaktadır.

Tablo 7.15. Benign ve DoS için DL Sonuçları

Eğitim	Sınıf	Prec.	Rec.	F1
BoT+ToN	Benign	0.914	0.982	0.947
	DoS	0.115	0.025	0.041
BoT+UNSW	Benign	0.711	0.874	0.784
	DoS	0.113	0.043	0.063
ToN+UNSW	Benign	0.276	0.645	0.387
	DoS	0.000	0.000	0.000

BoT+ToN eğitim senaryosunda Benign sınıfı için yüksek precision ve recall değerleri elde edilirken, DoS sınıfında hem geri çağırım hem de F1-skorunun oldukça düşük seviyelerde kaldığı görülmektedir. Benzer bir eğilim BoT+UNSW senaryosunda da gözlemlenmiş olup, modelin DoS saldırılarını ayırt etme yeteneğinin sınırlı kaldığı anlaşılmaktadır. ToN+UNSW senaryosunda ise DoS sınıfı için tüm metriklerin sıfıra düşmesi, modelin bu veri seti eşleşmesinde saldırı örüntülerini genelleyemediğini ve eğitim–test dağılımları arasında belirgin bir uyumsuzluk bulunduğunu göstermektedir.

Tablo 7.16. Benign ve DDoS için DL Sonuçları

Eğitim	Sınıf	Prec.	Rec.	F1
BoT+UNSW	Benign	0.302	1.000	0.464
	DDoS	0.400	0.000	0.000
ToN+UNSW	Benign	0.991	0.537	0.696
	DDoS	0.785	0.997	0.879

Tablo 7.16'da Benign ve DDoS sınıfları için elde edilen sonuçlar incelendiğinde, veri seti kombinasyonlarına bağlı olarak model performansında ciddi dalgalanmalar olduğu görülmektedir. BoT+ToN senaryosunda Benign sınıfı için dengeli metrikler elde edilirken, DDoS sınıfında precision, recall ve F1-skorlarının düşük kaldığı gözlemlenmiştir. BoT+UNSW senaryosunda ise modelin neredeyse tüm örnekleri Benign olarak sınıflandırdığı, DDoS sınıfında geri çağırımın sıfıra düştüğü görülmektedir. Buna

karşılık ToN+UNSW senaryosunda DDoS sınıfı için yüksek recall ve F1-skoru değerleri elde edilmesi, bazı veri seti eşleşmelerinde saldırı örüntülerinin daha tutarlı biçimde öğrenilebildiğini, ancak bu başarımın genelleştirilebilir olmadığını ortaya koymaktadır.

Tablo 7.17. Benign ve DoS+DDoS için DL Sonuçları

Eğitim	**Sınıf**	**Prec.**	**Rec.**	**F1**
BoT+UNSW	Benign	0.282	0.923	0.432
	DoS+DDoS	0.814	0.126	0.218
ToN+UNSW	Benign	0.207	0.429	0.279
	DoS+DDoS	0.754	0.515	0.612

Tablo 7.17'de Benign ve DoS+DDoS birleşik sınıfı için elde edilen sonuçlar sunulmaktadır. BoT+UNSW senaryosunda Benign sınıfı için yüksek recall değerleri elde edilirken, birleşik saldırı sınıfında düşük recall ve F1-skorları dikkat çekmektedir. ToN+UNSW senaryosunda ise saldırı sınıfı için daha dengeli metrikler elde edilmiş olsa da Benign sınıfındaki performansın sınırlı kaldığı görülmektedir. Bu sonuçlar, farklı saldırı türlerinin tek bir sınıf altında birleştirilmesinin derin öğrenme modellerinde dahi genelleme başarımını garanti etmediğini ve sınıf içi heterojenliğin model davranışını doğrudan etkilediğini göstermektedir.

Tablo 7.18. Benign ve Reconnaissance için DL Sonuçları

Eğitim	**Sınıf**	**Prec.**	**Rec.**	**F1**
BoT+ToN	Benign	0.793	0.641	0.709
	Reconnaissance	0.105	0.200	0.137
BoT+UNSW	Benign	0.985	0.969	0.977
	Reconnaissance	0.393	0.572	0.466
ToN+UNSW	Benign	0.891	0.695	0.781
	Reconnaissance	0.553	0.816	0.660

Tablo 7.18'de Benign ve Reconnaissance sınıfları için elde edilen test sonuçları verilmiştir. BoT+UNSW eğitim senaryosunda her iki sınıf için de görece dengeli ve yüksek performans değerleri elde edilirken, BoT+ToN ve ToN+UNSW senaryolarında

özellikle sınıflar arasında belirgin precision–recall dengesizlikleri gözlemlenmiştir. Reconnaissance sınıfında bazı kombinasyonlarda yüksek recall değerlerine karşın precision değerlerinin düşük kalması, modelin saldırı örüntülerini aşırı genelleyerek yanlış pozitif üretmeye eğilimli olduğunu göstermektedir. Bu durum, eğitim ve test veri setleri arasındaki öznitelik dağılımlarının yeterince örtüşmediğine işaret etmektedir.

Sonuç olarak, derin öğrenme tabanlı tüm deneyler birlikte değerlendirildiğinde, farklı veri setleri arasında gerçekleştirilen eğitim–test senaryolarında performans metriklerinde gözlemlenen tutarsızlıklar, bu deneylerde de belirgin bir yayılım kaymasının etkisinin güçlü olduğunu ortaya koymaktadır. Elde edilen bulgular, derin öğrenme modellerinin yüksek temsil gücüne rağmen, veri yayılımı uyumsuzluğu karşısında tutarlı bir genelleme başarımı sergileyemediğini göstermekte; bu durumun önceki bölümlerde sunulan geleneksel makine öğrenmesi sonuçları ile tutarlı olduğu görülmektedir.

7.6. Sonuçlar

Bu bölümde, farklı IoT veri setleri arasında gerçekleştirilen eğitim-test senaryoları altında, hem geleneksel makine öğrenmesi hem de derin öğrenme tabanlı modellerin performansları, yayılım kayması açısından, kapsamlı biçimde değerlendirilmiştir. Deneyler, veri setleri arası dağılım farklılıklarının model genelleme yeteneği üzerindeki etkisini incelemek amacıyla, aynı sınıflandırma senaryoları ve benzer değerlendirme ölçütleri kullanılarak yürütülmüştür.

Elde edilen sonuçlar, geleneksel makine öğrenmesi modellerinin bazı veri seti kombinasyonlarında daha kararlı ve öngörülebilir bir davranış sergileyebildiğini, ancak bu başarımın da veri dağılımı uyumuna yüksek derecede bağımlı olduğunu göstermektedir. Derin öğrenme tabanlı modeller ise yüksek temsil gücüne sahip olmalarına rağmen, farklı veri setleri arasında eğitilip test edildiklerinde tutarlı bir genelleme başarımı ortaya koyamamış; bazı senaryolarda saldırı sınıflarını aşırı genelleyerek yanlış pozitif üretmeye, bazı senaryolarda ise saldırı örüntülerini tamamen kaçırmaya eğilimli bir davranış sergilemiştir.

Geleneksel ve derin öğrenme yaklaşımları birlikte değerlendirildiğinde, model

karmaşıklığının tek başına değişken dağılım farklılıklarını telafi etmeye yeterli olmadığı açıkça görülmektedir. Derin öğrenme modelleri, klasik yöntemlere kıyasla daha esnek karar sınırları öğrenebilse de, eğitim ve test veri kümeleri arasındaki öznitelik dağılımları yeterince örtüşmediğinde bu esneklik yer yer dezavantaja dönüşebilmektedir. Benzer şekilde, geleneksel modellerde de belirli özniteliklere aşırı bağımlılık, dağılım kayması altında performansın hızla düşmesine neden olmaktadır.

Bu çalışmanın bulguları, farklı IoT veri setleri arasında yaygın ve belirgin bir yayılım kaymasının bulunduğunu ve hem geleneksel hem de derin öğrenme tabanlı saldırı tespit modellerinde genellemeyi ciddi biçimde sınırladığını göstermektedir. Bu durum, veri setleri arası doğrudan genelleme yapabilen tekil ve evrensel bir IoT-IDS modelinin pratikte zor olduğunu ortaya koymakta; dolayısıyla küresel IoT-IDS sistemlerinde model seçiminden çok, veri dağılımı uyumunu hedefleyen uyarlamalı ve dağılım farklarını ele alan öğrenme stratejilerinin önceliklendirilmesi gerektiğini vurgulamaktadır.

8. SONUÇLAR VE ÖNERİLER

8.1. Tez Çıkarımları

Bu tez kapsamında yapılan literatüre sağlanan katkılar aşağıda özetlenmiştir.

- Kıyaslamalı Literatür Çalışması: Tez kapsamında, özellikle, Bölüm 2 ve Bölüm 3'te literatürdeki saldırı türleri, bu saldırıların taksonomisi, genel olarak ağ ve IoT perspektifinden farklı türde saldırıların kıyaslanması, bu saldırı türlerinin tespiti için geliştirilen IDS sistemlerinin kategorizasyonu ile makine öğrenmesi ve derin öğrenme metodolojilerinin, sistematik olarak incelenmesi, bir literatür taraması olarak sunulmuştur. Ayrıca var olan literatürdeki ML ve DL tabanlı sistemlerin eksiklikleri ile literatürün genel olarak konumlandığı pozisyon ifade edilerek, okuyucuya IoT-IDS konusunda detaylı bir fikir sahibi olması konusunda katkı sağlayacağı düşünülmektedir. Buna ek olarak, literatürdeki eksikliklerin farkedilip, buna bağlı olarak oluşturulan araştırma sorularının da, okuyucuya farklı sorular sorabilmesinin yolunu açması açısından, tez katkısı olarak düşünülmüştür.
- IoT-IDS sistemleri için Sistematik Bir Eğitim Şeması: Tez kapsamında, IoT-IDS modellerinin eğitimi ve yanlılıktan uzak adil bir değerlendirme yapılabilmesi adına, sistematik bir eğitim şeması sunulmuştur. Bu kapsamda, IoT-IDS sistemleri kurulurken, uygun bir şekilde eğitim ve test setinin ayrılması, gruplama, tabakalı olarak bölme, ve dağılım bütünlüğünü koruyacak şekilde eğitim ve test setlerinin belirlenmesine yönelik adımlar detaylandırılarak, gelecekte yapılacak çalışmalar için, uygun bir akış sunulmuştur. Yine bu kapsamda, öznitelikler tek tek detaylandırılarak, yapılacak ön işleme stratejilerinin ne şekilde olması gerektiği detaylandırılmış ve literatürde yapılan oldukça standart yaklaşımların aksine, detaylı ve karşılaştırmalı, nedensellik temel alınarak modelleme aşaması için uygun stratejilerin belirlenmesi amacıyla, okuyucuya kapsamlı bir derlem sunulmuştur.
- İteratif Bir Öznitelik Üretim Algoritması: Bir IoT-IDS sisteminde veya diğer makine öğrenmesi problemlerinde kullanılmak üzere, Bölüm 4'te sunulan iteratif bir öznitelik algoritması genişletilerek, özniteliklerin kendi aralarındaki ilişkileri,

var olandan daha iyi sunacak bir öznitelik üretim yaklaşımı geliştirilmiştir. Bu sayede, elde olan öznitelikler üzerinden, yapılacak çeşitli matematiksel işlemlerle, daha detaylı öznitelikler çıkarılarak, ham özniteliklerden daha başarılı modellerin elde edilebileceği, ve bu sayede literatürdeki genel trend olan, derin öğrenme yaklaşımlarıyla kıyaslanabilecek, ancak IoT doğasına daha uygun geleneksel yöntemlerinde başarılı olabileceği sonucu elde edilmiştir.

- Açıklanabilir ve Ayrılabilirliği Arttıran Öznitelik Üretim Yaklaşımı: Bu tezde, literatürde yaygın olarak kullanılan ham ağ trafiği özniteliklerinin doğrudan modele verilmesi yerine, Bölüm 5'te, sınıflar arası ayrılabilirliği (separability) artırmayı hedefleyen açıklanabilir bir öznitelik üretim stratejileri uygulanmıştır. EBLR adı verilen ağaç tabanlı, ancak sürekli zamanlı verilere uygulanan öznitelik oluşturma süreci, IoT-IDS tabanlı bir sınıflandırma problemine uyarlanarak, model eğitiminin yalnızca istatistiksel dönüşümlerle sınırlı tutulmamasını; özniteliklerin sınıf davranışlarını yansıtma gücü ve dağılım yapıları dikkate alacak şekilde yeniden şekillenmesini sağlamıştır. Bu sayede, IoT-IDS literatüründe standart veri önişleme teknikleri yerine, daha kapsamlı bir öznitelik üretim algoritması sunarak, hem farklı problemlerde uygulanabilecek, hem de bu tezin kapsamındaki çalışmalarda kullanılabilecek, yenilikçi bir öznitelik ön işleme konsepti tanıtılmıştır.
- Dağılımı Koruyan Bir Yeniden Örnekleme Algoritması: Literatürde de görülebileceği üzere, IoT-IDS veri setleri doğası gereği dengesiz bir yapıdadır. Bu ise modelleme süreçlerinde, dengesizlik (imblancity) problemi olarak karşımıza çıkmaktadır. Veri dengesiliği, modelleme esnasında, oluşturulacak olan sistemlerin, çeşitli saldırı desenlerini az öğrenirken, diğer çoğunlukta olan saldırı desenlerini daha çok öğrenmesine yol açarak, modellemenin genelleştirilebilirliğini tehlikeye atmaktadır. Bu yüzden Bölüm 6.2.'de, KLSampler adında yeniden örnekleme algoritması sunulmuştur. Bu algoritma, literatürdeki çoğu aşağı örnekleme algoritmasının aksine, verideki dağılımı ve karakteristiğin korunması üzerine kuruludur. KLSampler ile, çoğunluk verileri aşağı örneklenirken, verinin temel dağılımsal ve istatistiksel özellikleri korunduğundan, aşağı örnekleme aşamasındaki bilgi kaybını minimize etmeyi amaçlamaktadır. Yapılan deneyler, bu tez kapsamında dizayn edilen algoritanın bu bilgi kayıplarını, başta IoT-IDS veri

setlerinde olmak üzere, farklı problem uzaylarında uygulanabilirliğini göstererek, modelleme esnasında dengesiz veri problemini gidermekte etkili olabilecek bir akış sunmaktadır.

- IoT-IDS Setlerinde Dağılım Kaymasının İncelenmesi: Literatürdeki çoğu çalışmada, modelleme ve test aşamaları genellikle aynı veri seti üzerinden yürümektedir. Ancak gerçek dünya problemlerinde, ağ yapısı, cihazlar, saldırıların çeşitliliği oldukça farklılık gösterebilir. Bu yüzden, IoT-IDS literatüründe nadir bir şekilde karşılaşılan, global bir IoT-IDS oluşturulabilirliği üzerine, bu tez detaylı olarak sistematik bir inceleme yapmıştır. Global bir IDS sistemi oluşturulmasındaki temel problemlerden biris olan, yayılım kayması (drift shift) farklı açılardan incelenmiş ve oluşturulan çeşitli modeller ile global bir model oluşturulmasının pratikteki karşılığı araştırılmıştır. Bu açıdan incelendiğinde, gerçekleştirilen tez çalışması, global bir IDS oluşturma çabasını sistematik bir şekilde dile getirerek, literatürdeki gelecek trendler hakkında da çeşitli çıkarımlarda bulunmuştur.

8.2. Araştırma Sorularının Cevapları

Araştırma Sorusu 1: İyi bir önişleme, eğitim-şeması ve öznitelik mühendisliği ile basit modeller ile hala derin öğrenme modellerine eş veya yakın performanslar alınabilir mi?

IoT-IDS literatürünün detaylandırıldığı ve araştırma sorusunun ortaya çıkmasına sebep olan Bölüm 3'da da görülebileceği üzere, son trendler genellikle derin öğrenme modelleri üzerinde yoğunlaşarak, geleneksel modellerden daha iyi performans elde edildiği şeklindedir. Ancak genel olarak literatür incelendiğinde, temel problemin, geleneksel makine öğrenmesi yaklaşımı açısından incelendiğinde, veri önişleme ve uygun modelleme açısından eksik kaldığı farkedilmiştir. Buna istinaden ortaya çıkan araştırma sorusuna dair ilgili çalışmalar Bölüm 4 ve Bölüm 5'te sunulmuştur. Bu bölümlerde, ilgili araştırma sorusunu yanıtlamak üzere, ToN_IoT veri seti üzerinden, uygun bir eğitim şeması, ön işleme stratejileri ve iki farklı öznitelik çıkarma algoritması sunulmuştur. Elde edilen sonuçlara göre ToN_IoT veri setinde hem uygun şekilde işlenen ham öznitelikler hem de öznitelik üretim algoritmalarından çıkarılan öznitelikler kullanılarak oluşturulmuş modeller, %99 seviyesinde AUC-ROC skor değerlerine ulaşmıştır. Bu

sonuçlar literatürdeki diğer derin öğrenme tabanlı benzer çalışmalarla kıyaslandığında, yer yer bu tezde sunulan yaklaşımların, yer yer ise derin öğrenme yaklaşımlarının iyi sonuçlar verdiği sonucuna ulaşılmıştır. Genel değerlendirme yapıldığında, basit modellerle oldukça iyi ve derin öğrenme literatüründeki çalışmalarla kıyaslanabilecek hatta çoğunlukla daha iyi modeller üretilmiştir. Bu ise, bizlere uygun bir şekilde eğitim, ön işleme, öznitelik üretme süreçleri gerçekleştirildiğinde, karmaşık ve kompleks modellere gerek duyulmadan, iyi IoT-IDS'ler üretilebileceğini göstermektedir.

Araştırma Sorusu 2: IoT-IDS sistemlerinde kullanılabilecek, veri dengesizliği problemini iyi bir şekilde çözebilen, bir yeniden örnekleme (resampling) algoritması dizayn edilebilir mi?

Literatürde de ifade edildiği biçimde, pek çok IoT-IDS veri seti, veri dengesizliği problemlerine sahiptir, ve eğitim aşamasında bu problemlerin çözülmesi gerekmektedir. Buna istinaden, Bölüm 6.2.'de, KLSampler adında, KL-Ayrışımı tabanlı dağılımı koruyan yeniden bir örnekleme algoritması dizayn edilmiştir. Bu algoritma temel olarak, çoğunluk sınıfları aşağı örneklerken, ilgili sınıfların dağılımsal ve istatiksel özelliklerini koruması açısından ön plana çıkmaktadır. IoT-IDS sistemlerinde, her saldırı türünün spesifik bir doğası olması nedeniyle, bu doğal yapının korunması ve bilgi kaybının olmaması ekstra önem arz etmektedir. Yapılan deneyler sonucunda, KLSampler algoritmasının gerçektende IoT-IDS veri setlerinde dağılımı koruruken, gerçekten veriyi dengeli hale getirdiği kanıtlanmıştır. Aynı zamanda bu algoritma daha genel bir konsepti içerdiğinden dolayı da, veri dengesizliğine sahip olacak diğer problemlere de uyarlanabilir. Bu durum da yine, Bölüm 6.2.'de incelenmiş ve farklı veri setleri, ve farklı problem uzaylarında da gayet başarılı sonuçlar vererek, veri dengesizliğinin giderilmesinde, ve bu esnada verinin dağılımı ve genel istatistiksel yapısının korunmasında etkili bir yöntem olarak karşımıza çıkmaktadır.

Araştırma Sorusu 3: IoT-IDS sistemlerinde bütünleşik bir sınıflandırma modeli oluşturulabilir mi?

Literatürdeki çoğu çalışmada olduğu gibi, IoT-IDS sistemleri kurulurken, öncelikli olarak bir veri seti seçilerek, yapılacak çalışmalar, deneyler ve sonuçlar, bu veri seti

üzerinde gerçekleştirilmektedir. Ancak gerçek dünya problemlerinde, ağ yapıları çok daha karmaşık, farklı seviyelerden dinamik yapılar taşıyabilir. Bu yüzden dolayı da, bu tez kapsamında farklı veri setleri birleşitirlerek, "global bir IoT-IDS sistemi kurulabilir mi?" sorusuna cevap aranmıştır. Buna istinaden, Bölüm 7.3.'te, farklı veri setleri çeşitli şekillerde kombinasyonlar ile birleştirilerek, tamamen bağımsız veri setleri üzerinden model performansları ölçülmüştür. Yapılan analizlerde, literatürde sıklıkla kullanılan, ToN_IoT, BoT_IoT ve UNSW-NB15 veri setleri kullanılmıştır. Modelleme çalışmaları öncesi yapılan, veri işlemesi sonrası ön analizlerde de farkedildiği üzere, bu veri setleri arasında çok yoğun bir yayılım kayması (drift shift) bulunmaktadır. Yayılım kayması fenomeni, esasen ele alınan veri setlerinin birbirilerinden oldukça farklı olması durumudur. Bu ise eğitilen makine öğrenmesi veya derin öğrenme modellerinin, eğitildiği veri ile test edildiği verinin temel olarak hiç bir ortak nokta taşımadığını gösterir. Yapılan çalışmalarda hem makine öğrenmesi modellerinin hem de derin öğrenme modellerinin, bu yayılım kaymasın fenomeninden oldukça güçlü bir şekilde etkilendiğini bizlere göstermiştir. Dolayısıyla şimdilik, bu tezde de gösterildiği biçimde, farklı veri setlerinin birleşitirlerek, global bir IDS oluşturulması çok kolay bir iş değildir. Sonuç olarak, bu araştırma sorusunun cevabı olarak, bu tez çalışması, bütünleşik bir sınıflandırma modelinin oluşturulamayacağını göstermiştir.

KAYNAKLAR

Abdaljabar, Z. H., Ucan, O. N., Alheeti, K. M. A. (2021). An Intrusion Detection System for IoT Using KNN and Decision-Tree Based Classification. *International Conference of Modern Trends in Information and Communication Technology Industry*, Sana'a, Yemen, 4–6 Aralık 2021.

Abdulhammed, R., Faezipour, M., Abuzneid, A., AbuMallouh, A. (2018). Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic. *IEEE Sensors Letters*, *3* (1), 1–4.

Aburomman, A. A., Reaz, M. B. I. (2016). A Novel SVM-kNN-PSO Ensemble Method for Intrusion Detection System. *Applied Soft Computing*, *38* (2016), 360–372.

Aburomman, A. A., Reaz, M. B. I. (2017). A Survey of Intrusion Detection Systems Based on Ensemble and Hybrid Classifiers. *Computers & Security*, *65* (2017), 135–152.

Abushwereb, M., Alkasassbeh, M., Almseidin, M., Mustafa, M. (2022). An Accurate IoT Intrusion Detection Framework Using Apache Spark. *arXiv preprint arXiv:2203.04347*.

Ackerman, S., Farchi, E., Raz, O., Zalmanovici, M., Dube, P. (2020). Detection of Data Drift and Outliers Affecting Machine Learning Model Performance Over Time. *arXiv preprint arXiv:2012.09258*.

Agrawal, A., Viktor, H. L., Paquet, E. (2015). SCUT: Multi-Class Imbalanced Data Classification Using SMOTE and Cluster-Based Undersampling. *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3k)*, Lisbon, Portugal, 12–14 Kasım 2015.

Ahmad, R., Alsmadi, I. (2021). Machine Learning Approaches to IoT Security: A Systematic Literature Review. *Internet of Things*, *14* (2021), 100365.

Akbar, N. A. F., Jaya, M. I., Razak, M. F. A., Zamri, N. A. (2025). A Systematic Review of Recurrent Neural Network Adoption in Missing Data Imputation. *International Journal of Computing*, *17* (1), 1–17.

Albulayhi, K., Smadi, A. A., Sheldon, F. T., Abercrombie, R. K. (2021). IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors*, *21* (19), 6432.

Altalhan, M., Algarni, A., Alouane, M. T. (2025). Imbalanced Data Problem in Machine Learning: A Review. *IEEE Access*, *13* (2025), 13686–13699.

Andriani, A., Meyliana, A., Maarif, V., Syukron, A. (2024). Resampling and Ensemble Method for Enhance Breast Cancer Classification Performance in

Imbalance Dataset. *International Conference on Advanced Information Scientific Development*, West Java, Indonesia, 25–26 Kasım 2024.

Anitha, A. A., Arockiam, L. (2022). A Review on Intrusion Detection Systems to Secure IoT Networks. *International Journal of Computer Networks and Applications*, *9* (1), 38–50.

Ashraf, E., Areed, N., Salem, H., Abdelhady, E., Farouk, A. (2022). IoT Based Intrusion Detection Systems from the Perspective of Machine and Deep Learning: A Survey and Comparative Study. *Delta University Scientific Journal*, *5* (2), 367–386.

Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.-K. R., Sarsour, M. (2024). Network Traffic Classification: Techniques, Datasets, and Challenges. *Digital Communications and Networks*, *10* (3), 676–692.

Batista, G. E. A. P., Prati, R. C., Monard, M. C. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD Explorations Newsletter*, *6* (1), 20–29.

Becker, B., Kohavi, R. (1996). *Census Income*, https://archive.ics.uci.edu/dataset/2/adult (Ziyaret Tarihi: 15 Aralık 2025).

Benkhelifa, E., Welsh, T., Hamouda, W. (2018). A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems. *IEEE Communications Surveys & Tutorials*, *20* (4), 3496–3509.

Bhattacharya, N., Kumar, P. (2025). A Comparative Analysis of Artificial Neural Network and Convolutional Neural Network Models: Performance Evaluation and Insights. *Mapping Human Data and Behavior With the Internet of Behavior (IoB)*, 163–188.

Bhattacharya, S., Krishnan, S. S. R., Maddikunta, P. K. R., Kaluri, R., Singh, S., Gadekallu, T. R., Alazab, M., Tariq, U. (2020). A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU. *Electronics*, *9* (2), 219.

Binitha, S., Sathya, S. S. (2012). A Survey of Bio-Inspired Optimization Algorithms. *International Journal of Soft Computing and Engineering*, *2* (2), 137–151.

Blastchar (2018). *Telco Customer Churn*, https://www.kaggle.com/blastchar/telco-customer-churn (Ziyaret Tarihi: 15 Aralık 2025).

Breiman, L., Friedman, J., Olshen, R. A., Stone, C. J. (2017). *Classification and Regression Trees* (Ebook. Bs.). New York: Chapman ve Hall/CRC.

Castro, E., Fonseca, I. E., Kumar, A., Alencar, M. S. (2010). A Packet Distribution Traffic Model for Computer Networks. *The International Telecommunications Symposium (ITS)*, Tokyo, Japan, 27–30 Haziran 2010.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys & Tutorials*, *21* (3), 2671–2701.

Chandrashekar, G., Sahin, F. (2014). A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, *40* (1), 16–28.

Chaudhary, P., Gupta, B. B. (2019). DDoS Detection Framework in Resource Constrained Internet of Things Domain. *Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, 15–18 Eylül 2019.

Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, *16* (2002), 321–357.

Choudhury, A., Kosorok, M. R. (2020). Missing Data Imputation for Classification Problems. *arXiv preprint arXiv:2002.10709*, (2020).

Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A. W. (2018). Time Series Feature Extraction on Basis of Scalable Hypothesis Tests (tsfresh: A Python Package). *Neurocomputing*, *307* (2018), 72–77.

Cisco (2025). *Cisco NetFlow*, https://www.cisco.com/site/us/en/products/networking/software/ios-nx-os/ios-netflow/index.html (Ziyaret Tarihi: 3 Aralık 2025).

Cox, D. R. (1958). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society Series B*, *20* (2), 215–232.

Davies, J. (2025). *A Walkthrough of a TLS 1.3 Handshake*, https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art080 (Ziyaret Tarihi: 23 Aralık 2025).

Dempster, A. P., Laird, N. M., Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B*, *39* (1), 1–22.

Dormann, C. F., Elith, J., Bacher, S., Buchmann, C., Carl, G., Carré, G., Marquéz, J. R. G., Gruber, B., Lafourcade, B., Leitão, P. J. (2013). Collinearity: A Review of Methods to Deal with It and a Simulation Study Evaluating Their Performance. *Ecography*, *36* (1), 27–46.

Dwyer, O. P., Marnerides, A. K., Giotsas, V., Mursch, T. (2019). Profiling IoT-Based Botnet Traffic Using DNS. *Global Communications Conference (GLOBECOM)*, Hawaii, United State, 9–12 Aralık 2019.

Elmobark, N., Elhishi, S. (2025). Comparing Cost-Sensitive and Data-Level Strategies to Address Extreme Class Imbalance in Educational Review Sentiment Analysis. *Research Square*, *Preprint* (-), 1–35.

Espinosa, F. C. M. (2023). Intrusion Detection in IoT Networks Using Machine Learning. Yüksek Lisans Tezi, Universitat Politècnica de Catalunya, Uygulamalı Telekomünikasyon ve Mühendislik Yönetimi, Barselona, İspanya.

Fatayer, T. S., Azara, M. N. (2019). IoT Secure Communication Using ANN Classification Algorithms. *International Conference on Promising Electronic Technologies (ICPET)*, Gaza, Palestine, 23–24 Ekim 2019.

Ferrag, M. A., Maglaras, L. (2019). DeepCoin: A Novel Deep Learning and Blockchain-Based Energy Exchange Framework for Smart Grids. *IEEE Transactions on Engineering Management*, *67* (4), 1285–1297.

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., Janicke, H. (2020). Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study. *Journal of Information Security and Applications*, *50* (2020), 102419.

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, *3* (Mar), 1289–1305.

Galli, S. (2024). *Feature Selection for Machine Learning*, https://github.com/solegalli/feature-selection-for-machine-learning (Ziyaret Tarihi: 23 Ekim 2025).

Garcia, S., Plohmann, A. (2020). *IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic (Version 1.0.0)*, https://doi.org/10.5281/zenodo.4743746 (Ziyaret Tarihi: 10 Kasım 2025).

Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., Robles-Kelly, A. (2019). Deep Learning-Based Intrusion Detection for IoT Networks. *Pacific Rim International Symposium on Dependable Computing (PRDC)*, Kyoto, Japan, 1–3 Aralık 2019.

Geurts, P., Ernst, D., Wehenkel, L. (2006). Extremely Randomized Trees. *Machine Learning*, *63* (1), 3–42.

Goldenberg, I., Webb, G. I. (2019). Survey of Distance Measures for Quantifying Concept Drift and Shift in Numeric Data. *Knowledge and Information Systems*, *60* (2), 591–615.

Goyal, M., Sahoo, I., Geethakumari, G. (2019). HTTP Botnet Detection in IoT Devices Using Network Traffic Analysis. *International Conference on Recent Advances in Energy-Efficient Computing and Communication (ICRAECC)*, Nagercoil, India, 7–8 Mart 2019.

Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, *29* (7), 1645–1660.

Guyon, I., Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3* (Mar), 1157–1182.

Guzmán-Ponce, A., Valdovinos, R. M., Sánchez, J. S., Marcial-Romero, J. R. (2020). A New Under-Sampling Method to Face Class Overlap and Imbalance. *Applied Sciences*, *10* (15), 5164.

Hamouda, D., Ferrag, M. A., Benhamida, N., Seridi, H. (2021). Intrusion Detection Systems for Industrial Internet of Things: A Survey. *International Conference on Theoretical and Applicative Aspects of Computer Science*, Skikda, Algeria, 15–16 Aralık 2021.

Heidari, A., Jamali, M. A. J. (2023). Internet of Things Intrusion Detection Systems: A Comprehensive Review and Future Directions. *Cluster Computing*, *26* (6), 3753–3780.

Huston, G. (2024). *The Size of Packets*, https://labs.apnic.net/index.php/2024/10/04/the-size-of-packets/ (Ziyaret Tarihi: 23 Aralık 2025).

Hwang, R.-H., Peng, M.-C., Nguyen, V.-L., Chang, Y.-L. (2019). An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level. *Applied Sciences*, *9* (16), 3414.

Idrissi, I., Azizi, M., Moussaoui, O. (2020). IoT Security with Deep Learning-Based Intrusion Detection Systems: A Systematic Literature Review. *International Conference on Intelligent Computing in Data Sciences*, Fez, Morocco, 21–23 Eylül 2020.

Ilic, I., Görgülü, B., Cevik, M., Baydoğan, M. G. G. (2021). Explainable Boosted Linear Regression for Time Series Forecasting. *Pattern Recognition*, *120* (2021), 108144.

Imperva Group (2025). *MTU and MSS: What You Need to Know*, https://www.imperva.com/learn/application-security/what-is-mtu-mss/ (Ziyaret Tarihi: 23 Aralık 2025).

Inayat, U., Zia, M. F., Mahmood, S., Khalid, H. M., Benbouzid, M. (2022). Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects. *Electronics*, *11* (9), 1502.

Ioannou, C., Vassiliou, V. (2019). Classifying Security Attacks in IoT Networks Using Supervised Learning. *International Conference on Distributed Computing in Sensor Systems*, Santorini, Greece, 29–31 Mayıs 2019.

Jarrah, O. Y. A., Hammdi, Y. A., Yoo, P. D., Muhaidat, S., Qutayri, M. A. (2018). Semi-Supervised Multi-Layered Clustering Model for Intrusion Detection. *Digital Communications and Networks*, *4* (4), 277–286.

Jing, D., Chen, H.-B. (2019). SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset. *International Conference on ASIC*, Chongqing, China, 29 Eylül–1 Ekim 2019.

Kanter, J. M., Veeramachaneni, K. (2015). Deep Feature Synthesis: Towards Automating Data Science Endeavors. *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Paris, France, 19–21 Ekim 2015.

Karlaš, B., Salimi, B., Schelter, S. (2025). Navigating Data Errors in Machine Learning Pipelines: Identify, Debug, and Learn. *International Conference on Data Engineering (ICDE)*, Hong Kong, Hong Kong, 19–23 Mayıs 2025.

Kasongo, S. M., Sun, Y. (2020). A Deep Learning Method with Wrapper Based Feature Extraction for Wireless Intrusion Detection System. *Computers & Security*, *92* (2020), 101752.

KDD Cup (1999). *KDD Cup 1999 Data*, https://www.kdd.org/kdd-cup/view/kdd-cup-1999/Data (Ziyaret Tarihi: 7 Aralık 2025).

Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., Turnbull, B., Zomaya, A. Y. (2023). An Explainable Deep Learning-Enabled Intrusion Detection Framework in IoT Networks. *Information Sciences*, *639* (2023), 119000.

Khalilia, M., Chakraborty, S., Popescu, M. (2011). Predicting Disease Risks from Highly Imbalanced Data Using Random Forest. *BMC Medical Informatics and Decision Making*, *11* (1), 51.

Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., Bahaj, S. A. (2022). Deep Learning for Intrusion Detection and Security of Internet of Things (IoT): Current Analysis, Challenges, and Possible Solutions. *Security and Communication Networks*, *2022* (1), 4016073.

Khatun, M. A., Chowdhury, N., Uddin, M. N. (2019). Malicious Nodes Detection Based on Artificial Neural Network in IoT Environments. *International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 18–20 Aralık 2019.

Khraisat, A., Alazab, A. (2021). A Critical Review of Intrusion Detection Systems in the Internet of Things: Techniques, Deployment Strategy, Validation Strategy, Attacks, Public Datasets and Challenges. *Cybersecurity*, *4* (1), 18.

Kikissagbe, B. R., Adda, M. (2024). Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review. *Electronics*, *13* (18), 3601.

Kilichev, D., Kim, W. (2023). Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO. *Mathematics*, *11* (17), 3724.

Kim, J., Kim, J., Kim, H., Shim, M., Choi, E. (2020). CNN-Based Network Intrusion Detection against Denial-of-Service Attacks. *Electronics*, *9* (6), 916.

Kolmogorov, A. N. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*, *4* (1933), 83–91.

Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B. (2019). Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset. *Future Generation Computer Systems*, *100* (2019), 779–796.

Kramer, O. (2016). Scikit-Learn. (Ed.), *Machine Learning for Evolution Strategies* (1. Bs.) (45–53). Gewerbestrasse, İsviçre: Springer Cham.

Kubat, M. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. *International Conference on Machine Learning*, San Francisco, CA, United States, 8–12 Temmuz 1997.

Kullback, S., Leibler, R. A. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics*, *22* (1), 79–86.

Kumar, A., Lim, T. J. (2019). EDIMA: Early Detection of IoT Malware Network Activity Using Machine Learning Techniques. *World Forum on Internet of Things (WF-IoT)*, Limerick, Ireland, 15–19 Nisan 2019.

Li, Y., Xu, Y., Liu, Z., Hou, H., Zheng, Y., Xin, Y., Zhao, Y., Cui, L. (2020). Robust Detection for Network Intrusion of Industrial IoT Based on Multi-CNN Fusion. *Measurement*, *154* (2020), 107450.

Liang, C., Shanmugam, B., Azam, S., Jonkman, M., Boer, F. D., Narayansamy, G. (2019). Intrusion Detection System for Internet of Things Based on a Machine Learning Approach. *International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*, Coimbatore, India, 30–31 Mart 2019.

Liang, X., Znati, T. (2019). A Long Short-Term Memory Enabled Framework for DDoS Detection. *Global Communications Conference (GLOBECOM)*, Hawaii, United States,

Liao, H., Murah, M. Z., Hasan, M. K., Aman, A. H. M., Fang, J., Hu, X., Khan, A. U. R. (2024). A Survey of Deep Learning Technologies for Intrusion Detection in Internet of Things. *IEEE Access*, *12* 4745–4761.

Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., Tung, K.-Y. (2013). Intrusion Detection System: A Comprehensive Review. *Journal of Network and Computer Applications*, *36* (1), 16–24.

Lin, J. (1991). Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, *37* (1), 145–151.

Liu, H., Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, *17* (4), 491–502.

Liu, X., Zhang, P., Xu, C., Zhang, D., Dong, Z., Li, M. (2025). Study of 3D Geological Suitability Evaluation via Machine Learning and Gray Correlation Analysis. *Earth Science Informatics*, *18* (4), 1–23.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., Lloret, J. (2019). Shallow Neural Network with Kernel Approximation for Prediction Problems in Highly Demanding Data Networks. *Expert Systems with Applications*, *124* (2019), 196–208.

M Sarhan and S Layeghy and M Portmann (2022). *Datasets, Towards a Standard Feature Set for Network Intrusion Detection System Datasets*, https://espace.library.uq.edu.au/view/UQ:275ac0d (Ziyaret Tarihi: 28 Kasım 2025).

MacDermott, Á., Kendrick, P., Idowu, I., Ashall, M., Shi, Q. (2019). Securing Things in the Healthcare Internet of Things. *Global IoT Summit (GIoTS)*, Aarhus, Denmark, 17–21 Haziran 2019.

Maheswaran, N., Logeswari, G., Bose, S., Anitha, T. (2023). A Critical Review on Intrusion Detection Systems in IoT Based on ML Approach: A Survey. *International Conference on Smart Technologies and Systems for Next Generation Computing*, Dublin, Ireland, 21–22 Nisan 2023.

Mani, I., Zhang, I. (2003). kNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. *Workshop on Learning from Imbalanced Datasets*, Washington, DC., United States, 21 Aralık 2003.

Manikandan, J., Harini, K. G., Arshini, M. N. (2024). Deep Learning-Based Lung Cancer Detection and Classification with Hybrid Sampling for Imbalanced Data. *International Conference on Smart Electronics and Communication Systems*, Kerala, India, 6–7 Aralık 2024.

Manivannan, D. (2024). Recent Endeavors in Machine Learning-Powered Intrusion Detection Systems for the Internet of Things. *Journal of Network and Computer Applications*, *229* (2024), 103925.

Mannapur, S. (2025). Understanding Data Drift and Concept Drift in Machine Learning Systems. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *11* (2025), 318–330.

Markov, S. (2017). *Skopt Documentation*, https://scikit-optimize.github.io/ (Ziyaret Tarihi: 12 Aralık 2025).

McHugh, J. (2000). Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, *3* (4), 262–294.

McKinney, W. (2011). pandas: A Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing*, *14* (9), 1–9.

McQueen, J. B. (1967). Some Methods of Classification and Analysis of Multivariate Observations. *Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, CA, United States,

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y. (2018). N-BaIoT: Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, *17* (3), 12–22.

Mishra, N., Pandya, S. (2021). Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions: A Systematic Review. *IEEE Access*, *9* (2021), 59353–59377.

Mliki, H., Kaceam, A. H., Chaari, L. (2021). A Comprehensive Survey on Intrusion Detection Based Machine Learning for IoT Networks. *EAI Endorsed Transactions on Security & Safety*, *8* (29), 1–23.

Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M. (2018). Deep Learning for IoT Big Data and Streaming Analytics: A Survey. *IEEE Communications Surveys & Tutorials*, *20* (4), 2923–2960.

Mohy-Eddine, M., Guezzaz, A., Benkirane, S., Azrour, M., Farhaoui, Y. (2023). An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security. *Big Data Mining and Analytics*, *6* (3), 273–287.

Montero-Manso, P., Hyndman, R. J. (2021). Principles and Algorithms for Forecasting Groups of Time Series: Locality and Globality. *International Journal of Forecasting*, *37* (4), 1632–1653.

Moraes, G. L., Angelo, R. M., Costa, A. C. S. (2024). Axiomatic Approach to Measures of Total Correlations. *Entropy*, *26* (12), 1098.

Moro, S., Cortez, P., Rita, P. (2014). A Data-Driven Approach to Predict the Success of Bank Telemarketing. *Decision Support Systems*, *62* (2014), 22–31.

Motamedi, B., Villányi, B. (2025). A Novel Hybrid Machine Learning Approach for Early Prediction of Parkinson's Disease Severity Using Optimized Feature Selection and Ensemble Learning. *Intelligence-Based Medicine*, *12* (2025), 100276.

Moustafa, N. (2021). A New Distributed Architecture for Evaluating AI-Based Security Systems at the Edge: Network TON_IoT Datasets. *Sustainable Cities and Society*, *72* (2021), 102994.

Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y., Tari, Z. (2023). Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions. *IEEE Communications Surveys & Tutorials*, *25* (3), 1775–1807.

Moustafa, N., Slay, J. (2015). UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems. *Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 10–12 Ekim 2015.

Muniyandi, A. P., Rajeswari, R., Rajaram, R. (2012). Network Anomaly Detection by Cascading k-Means Clustering and C4.5 Decision Tree Algorithm. *Procedia Engineering*, *30* (2012), 174–182.

Nagisetty, A., Gupta, G. P. (2019). Framework for Detection of Malicious Activities in IoT Networks Using Keras Deep Learning Library. *International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 27–29 Mart 2019.

Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A. A. (2023). CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment. *Sensors*, *23* (13), 5941.

Nsunza, W. W., Tetteh, A.-Q. R., Hei, X. (2018). Accelerating a Secure Programmable Edge Network System for Smart Classroom. *IEEE SmartWorld Congress (UIC-ATC)*, Guangzhou, China, 8–12 Eylül 2018.

Otoum, Y., Liu, D., Nayak, A. (2022). DL-IDS: A Deep Learning-Based Intrusion Detection Framework for Securing IoT. *Transactions on Emerging Telecommunications Technologies*, *33* (3), e3803.

Park, S. H., Park, H. J., Choi, Y.-J. (2020). RNN-Based Prediction for Network Intrusion Detection. *International Conference on Artificial Intelligence in Information and Communication*, Fukuoka, Japan, 19–21 Şubat 2020.

Patel, V. (2016). *A Practical Guide to Dimensionality Reduction Techniques*. PyData Conference, https : / / www . youtube . com / watch ? v = ioXKxulmwVQ (Ziyaret Tarihi: 6 Nisan 2025).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. (2011). Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12* (2011), 2825–2830.

Piramuthu, S., Sikora, R. T. (2009). Iterative Feature Construction for Improving Inductive Learning Algorithms. *Expert Systems with Applications*, *36* (2), 3401–3406.

Prajapati, N., Karki, B., Gopali, S., Namin, A. S. (2025). IoT Malware Network Traffic Detection using Deep Learning and GraphSAGE Models. *Computers, Software, ve Applications Conference (COMPSAC)*, Toronto, Canada, 8–11 Temmuz 2025.

Pratomo, B. A., Burnap, P., Theodorakopoulos, G. (2018). Unsupervised Approach for Detecting Low Rate Attacks on Network Traffic with Autoencoder. *International*

Conference on Cyber Security and Protection of Digital Services (Cyber Security), Glasgow, UK, 11–12 Haziran 2018.

Rahmani, K., Thapa, R., Tsou, P., Chetty, S. C., Barnes, G., Lam, C., Tso, C. F. (2023). Assessing the Effects of Data Drift on the Performance of Machine Learning Models Used in Clinical Sepsis Prediction. *International Journal of Medical Informatics*, *173* (2023), 104930.

Rajkarnikar, R. M. (2023). Intrusion Detection System for Internet of Things (IDS for IoT). Doktora Tezi, Selinus Üniversitesi, Bilgisayar Bilimleri Fakültesi, Amerika Birleşik Devletleri.

Roopak, M. (2021). Intrusion Detection System for IoT Networks for Detection of DDoS Attacks. Doktora Tezi, Newcastle Üniversitesi, Bilgisayar Bilimleri Fakültesi, Birleşik Krallık.

Roopak, M., Tian, G. Y., Chambers, J. (2019). Deep Learning Models for Cyber Security in IoT Networks. *Computing and Communication Workshop ve Conference (CCWC)*, Las Vegas, NV, United States, 7–9 Ocak 2019.

Roopak, M., Tian, G. Y., Chambers, J. (2020). An Intrusion Detection System against DDoS Attacks in IoT Networks. *Computing and Communication Workshop ve Conference (CCWC)*, Las Vegas, NV, United States, 6–8 Ocak 2020.

Roy, B., Cheung, H. (2018). A Deep Learning Approach for Intrusion Detection in Internet of Things Using Bi-Directional Long Short-Term Memory Recurrent Neural Network. *International Telecommunication Networks and Applications Conference*, Sydney, NSW, Australia, 21–23 Kasım 2018.

Saadouni, R., Gherbi, C., Aliouat, Z., Harbi, Y., Khacha, A. (2024). Intrusion Detection Systems for IoT Based on Bio-Inspired and Machine Learning Techniques: A Systematic Review of the Literature. *Cluster Computing*, *27* (7), 8655–8681.

Sahiner, B., Chen, W., Samala, R. K., Petrick, N. (2023). Data Drift in Medical Machine Learning: Implications and Potential Remedies. *The British Journal of Radiology*, *96* (1150), 20220878.

Sakar, C., Kastro, Y. (2018). *Online Shoppers Purchasing Intention Dataset*, https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset (Ziyaret Tarihi: 15 Aralık 2025).

Sanchez, R. A. M., Zaman, M., Goel, N., Naik, K., Joshi, R. (2022). Towards Developing a Robust Intrusion Detection Model Using Hadoop–Spark and Data Augmentation for IoT Networks. *Sensors*, *22* (20), 7726.

Santos, L., Rabadao, C., Gonçalves, R. (2018). Intrusion Detection Systems in Internet of Things: A Literature Review. *Iberian Conference on Information Systems and Technologies*, Caceres, Spain, 13–16 Haziran 2018.

Sarhan, M., Layeghy, S., Portmann, M. (2022). Towards a Standard Feature Set for Network Intrusion Detection System Datasets. *Mobile Networks and Applications*, *27* (1), 357–370.

Scikit-Learn (2025). *sklearn.cluster.MiniBatchKMeans — scikit-learn documentation*, ht tps://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans .html (Ziyaret Tarihi: 19 Aralık 2025).

Sculley, D. (2010). Web-Scale k-Means Clustering. *International Conference on World Wide Web*, Raleigh, North Carolina, United States, 26–30 Nisan 2010.

Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *International Conference on Information Systems Security and Privacy*, *1* (2018), 108–116.

Silverman, B. W. (2018). *Density Estimation for Statistics and Data Analysis*: Routledge.

Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., Sakurai, K. (2019a). A Sequential Scheme for Detecting Cyber Attacks in IoT Environment. *International Symposium on Dependable, Autonomic ve Secure Computing (DASC)*, Fukuoka, Japan, 5–8 Ağustos 2019.

Soe, Y. N., Santosa, P. I., Hartanto, R. (2019b). DDoS Attack Detection Based on Simple ANN with SMOTE for IoT Environment. *International Conference on Informatics and Computing (ICIC)*, Semarang, Indonesia, 16–17 Ekim 2019.

Sugiyama, M., Liu, S., Plessis, M. C. du, Yamanaka, M., Yamada, M., Suzuki, T., Kanamori, T. (2013). Direct Divergence Approximation between Probability Distributions and Its Applications in Machine Learning. *Journal of Computing Science and Engineering*, *7* (2), 99–111.

Sun, Y., Wong, A. K. C., Kamel, M. S. (2009). Classification of Imbalanced Data: A Review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23* (4), 687–719.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, 8–10 Temmuz 2009.

Telikani, A., Shen, J., Yang, J., Wang, P. (2022). Industrial IoT Intrusion Detection via Evolutionary Cost-Sensitive Learning and Fog Computing. *IEEE Internet of Things Journal*, *9* (22), 23260–23271.

Tian, Q., Li, J., Liu, H. (2019). A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning. *IEEE Access*, *7* (2019), 38688–38695.

Ugale, S., Potgantwar, A. (2025). Risk Classification of Docker Container Images Using Machine Learning and Vulnerability Metrics. *Engineering, Technology & Applied Science Research*, *15* (6), 30310–30316.

Ullah, F., Naeem, H., Jabbar, S., Khalid, S., Latif, M. A., Al-Turjman, F., Mostarda, L. (2019). Cyber Security Threats Detection in Internet of Things Using Deep Learning Approach. *IEEE Access*, *7* (2019), 124379–124389.

Villani, C. (2009). *Optimal Transport: Old and New*. Berlin, Heidelberg: Springer.

Walt, S. V. D., Colbert, S. C., Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, *13* (2), 22–30.

Wehbi, K., Hong, L., Al-Salah, T., Bhutta, A. A. (2019). A Survey on Machine Learning Based Detection on DDoS Attacks for IoT Systems. *IEEE SoutheastCon*, Huntsville, AL, United State, 11–14 Nisan 2019.

Wei, W., Li, J., Cao, L., Ou, Y., Chen, J. (2013). Effective Detection of Sophisticated Online Banking Fraud on Extremely Imbalanced Data. *World Wide Web*, *16* (4), 449–475.

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., Alvarenga, S. C. D. (2017). A Survey of Intrusion Detection in Internet of Things. *Journal of Network and Computer Applications*, *84* 25–37.

Zhang, Y., Li, P., Wang, X. (2019). Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access*, *7* 31711–31722.

Zhou, Y., Aryal, S., Bouadjenek, M. R. (2024). Review for Handling Missing Data with Special Missing Mechanism. *arXiv preprint*.

Zhou, Y., Han, M., Liu, L., He, J. S., Wang, Y. (2018). Deep Learning Approach for Cyberattack Detection. *IEEE INFOCOM Workshops*, Honolulu, HI, USA, 15–19 Nisan 2018.

Zhu, R., Ji, X., Yu, D., Tan, Z., Zhao, L., Li, J., Xia, X. (2020). KNN-Based Approximate Outlier Detection Algorithm over IoT Streaming Data. *IEEE Access*, *8* 42749–42759.

Zhu, S., Xu, X., Gao, H., Xiao, F. (2023). CMTSNN: A Deep Learning Model for Multiclassification of Abnormal and Encrypted Traffic of Internet of Things. *IEEE Internet of Things Journal*, *10* (13), 11773–11791.

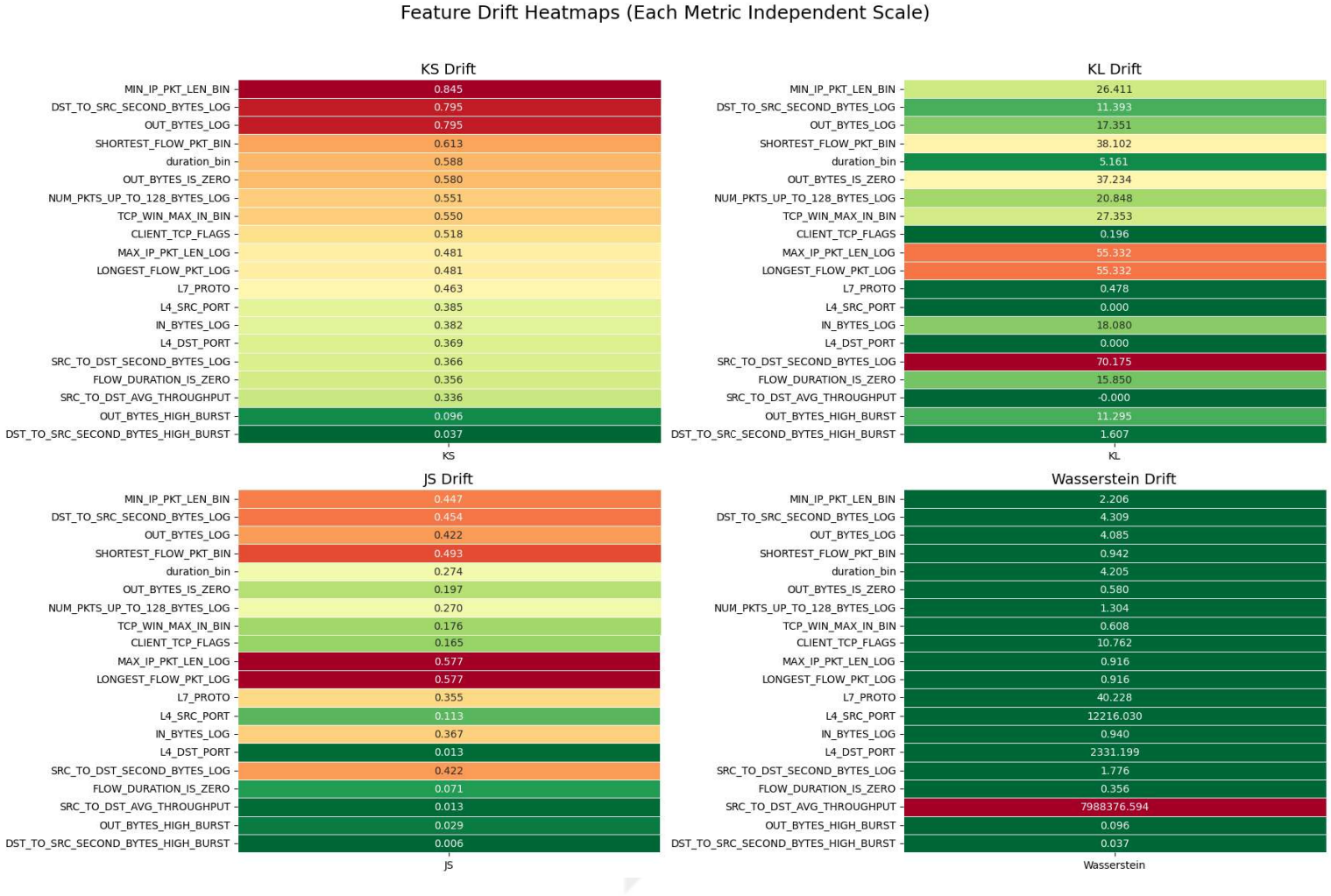
Zou, H., Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B*, *67* (2), 301–320.

EKLER

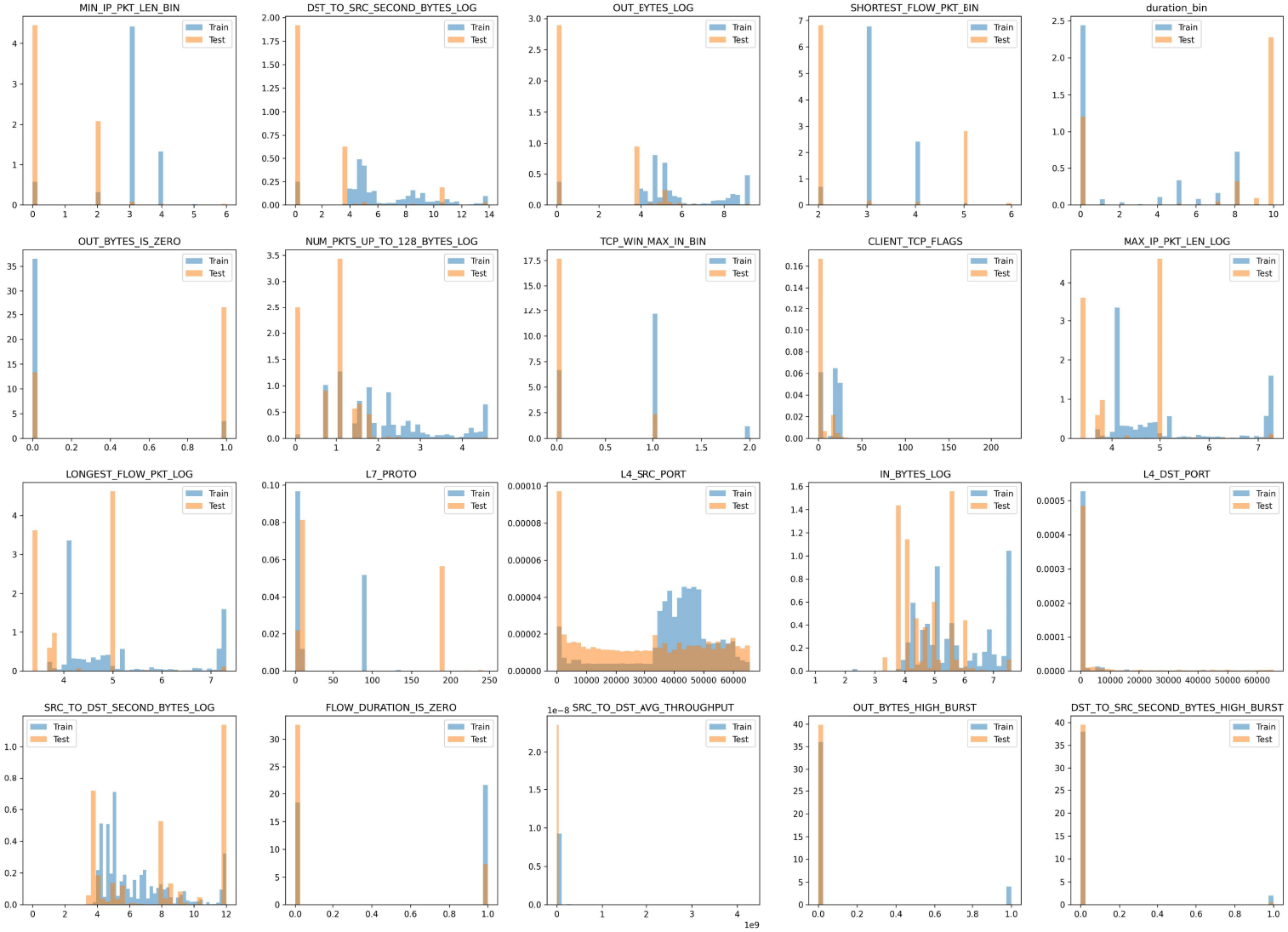
Ek-A

A.1 ToN-UNSW Eğitim ve BoT_IoT Test Seti için Yayılım Kayma Grafikleri

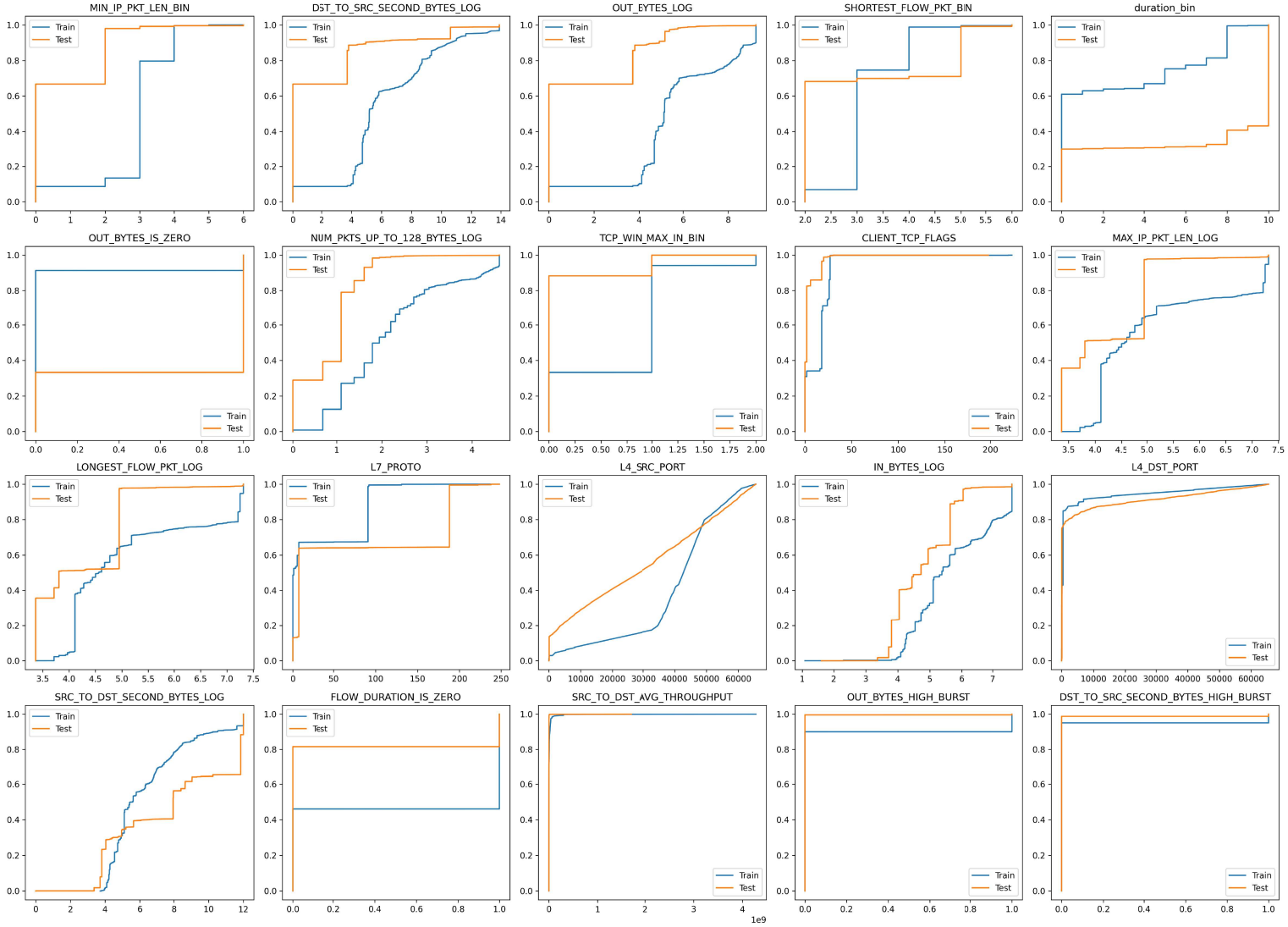
Bölüm 7.3.2.'de yapılan işlemlere benzer olarak, yayılım analizi, ToN_IoT ile UNSW-NB15 veri kümeleri birleştirilerek eğitim seti oluşturulup, test seti olarak ise BoT_IoT kullanılan senaryoya da uygulanmıştır. Şekil A.1-A.5 arasındaki görseller incelendiğinde, eğitim ve test veri setleri arasında çeşitli öznitelikler bazında belirgin yayılım kaymaları (distributional shift) gözlemlenmektedir. ECDF, histogram, KDE ve yoğunluk farkı analizleri, önceki bölümde detaylandırılan bulgularla benzer bir şekilde, bazı özniteliklerin test setinde baskınlaştığını veya tamamen farklı bir dağılım gösterdiğini ortaya koymaktadır. Görsel inceleme, özellikle trafik hacmi ve paket yoğunluğuna ilişkin özniteliklerde değişken davranışları açıkça ortaya koymaktadır.



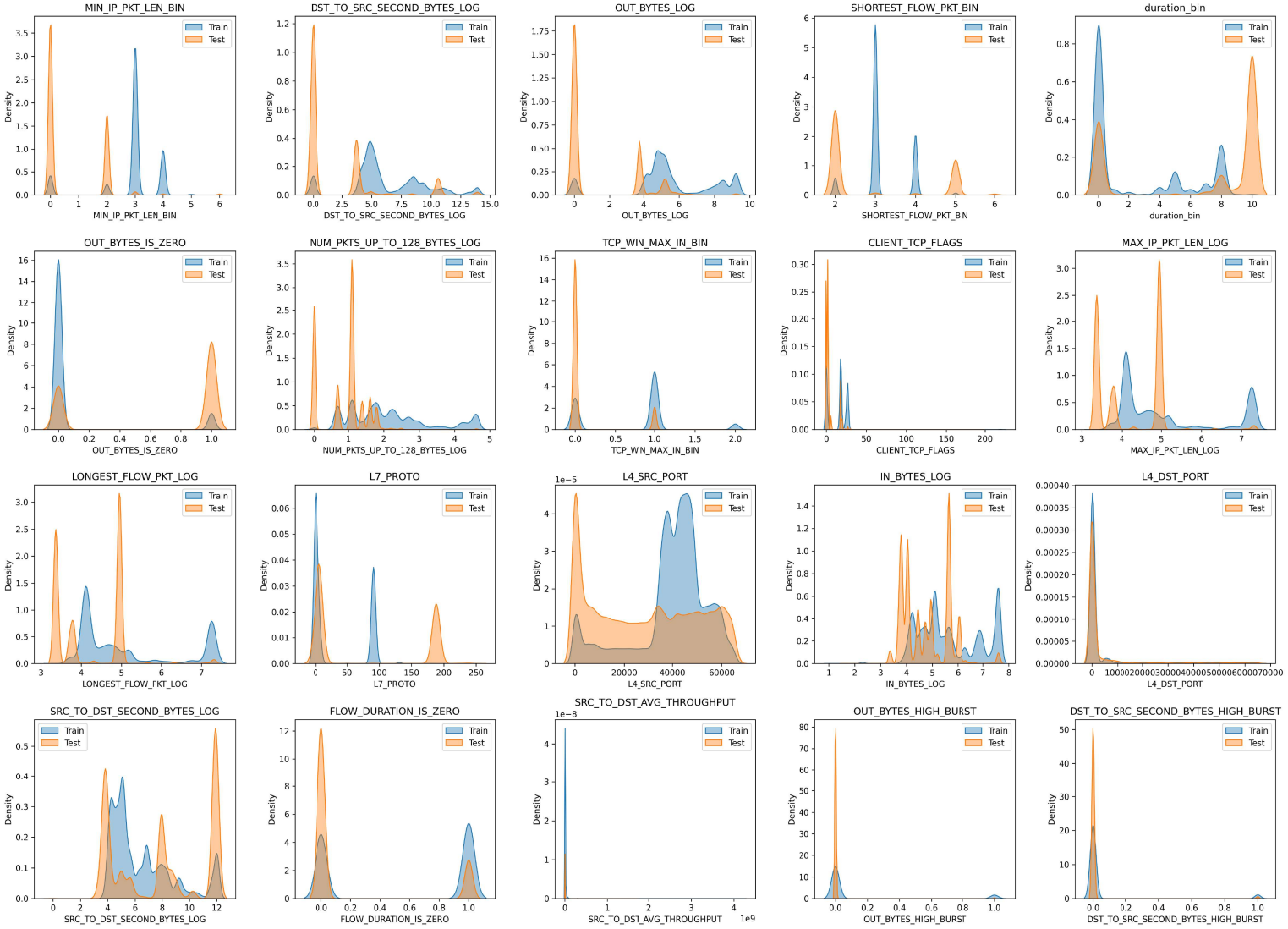
Şekil A.1. ToN-UNSW ve BoT_IoT için Yayılım Metrikleri



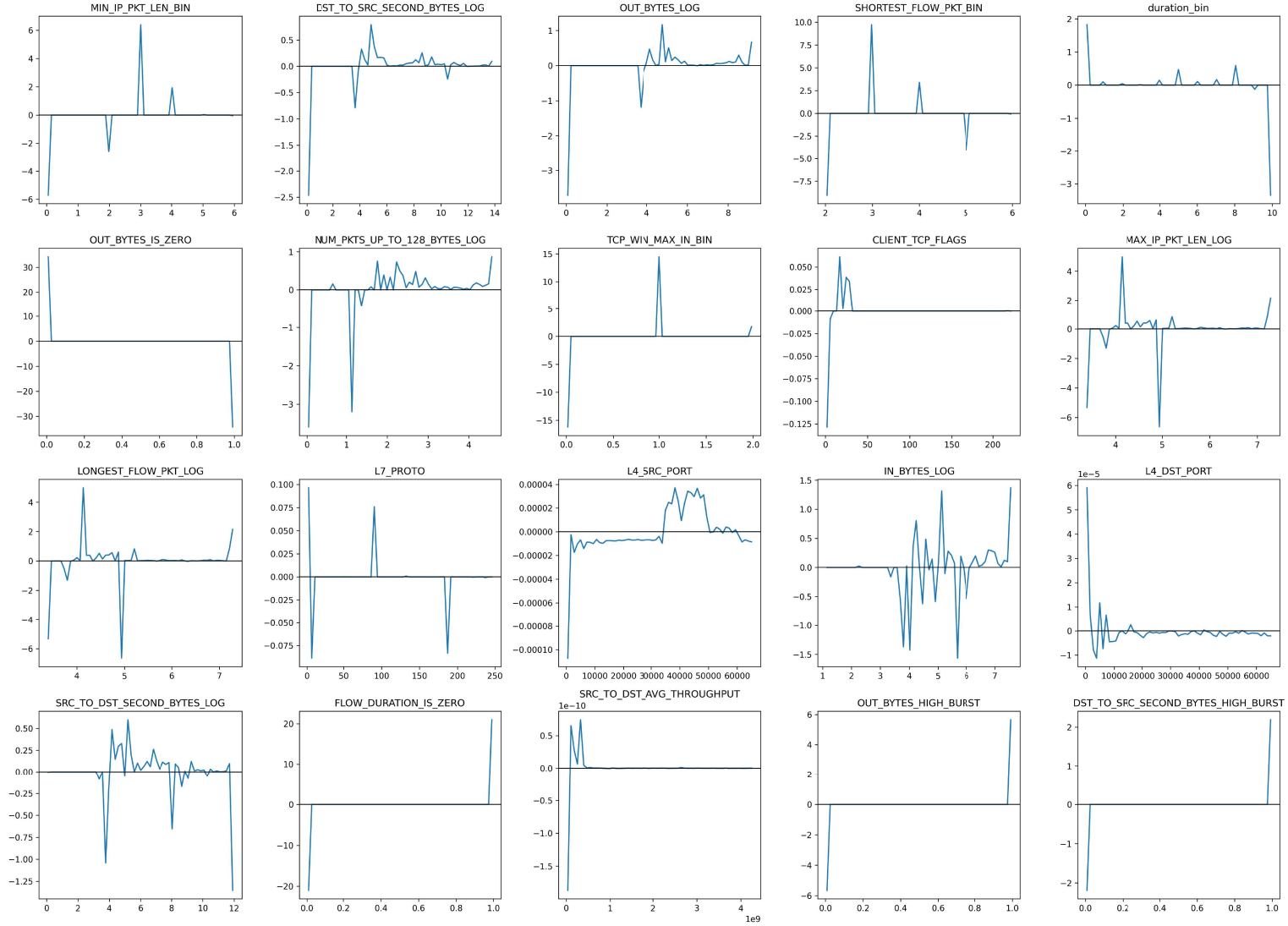
Şekil A.2. ToN-UNSW ve BoT_IoT için Histogramlar



Şekil A.3. ToN-UNSW ve BoT_IoT için ECDF Grafikleri



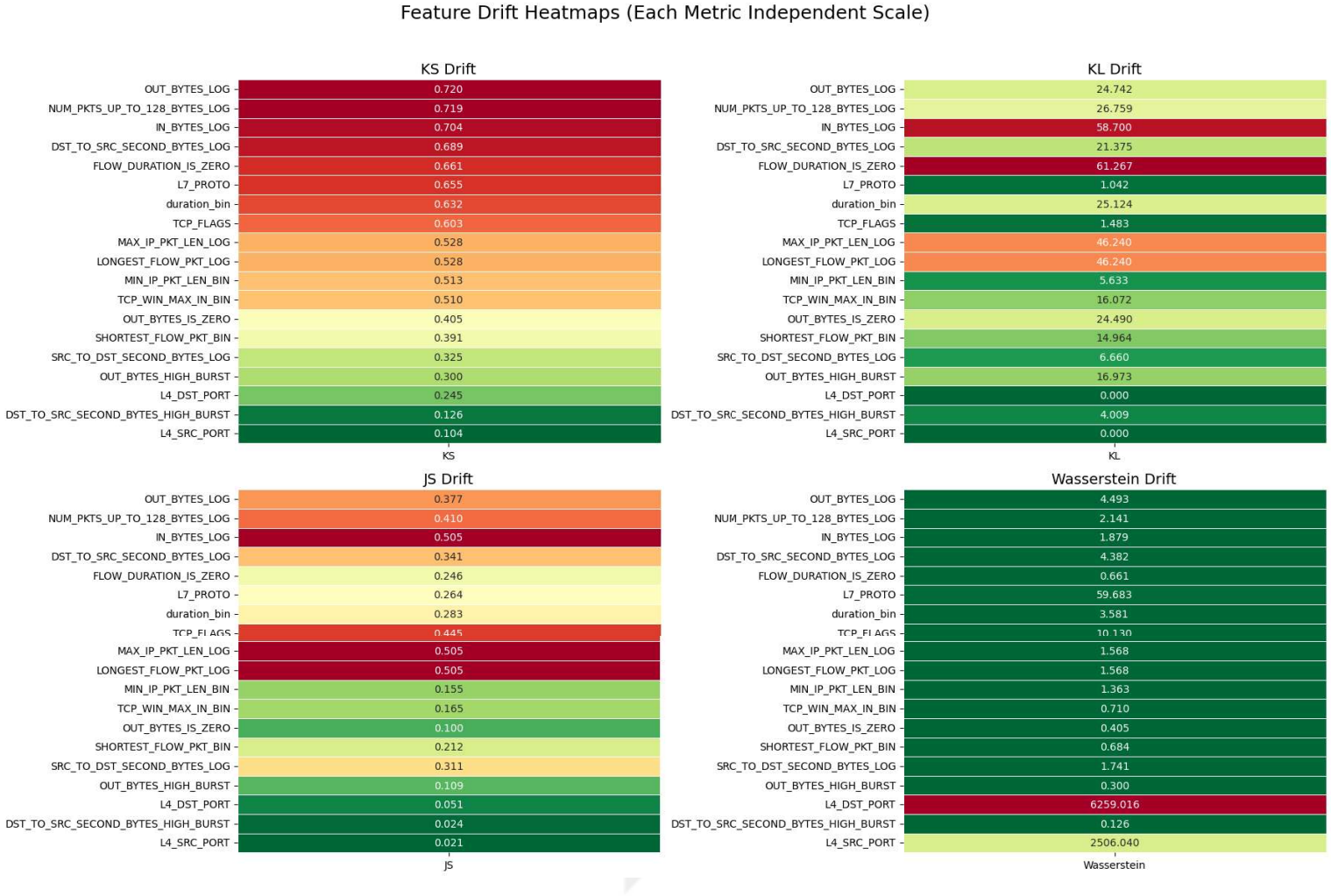
Şekil A.4. ToN-UNSW ve BoT_IoT için KDE Grafikleri



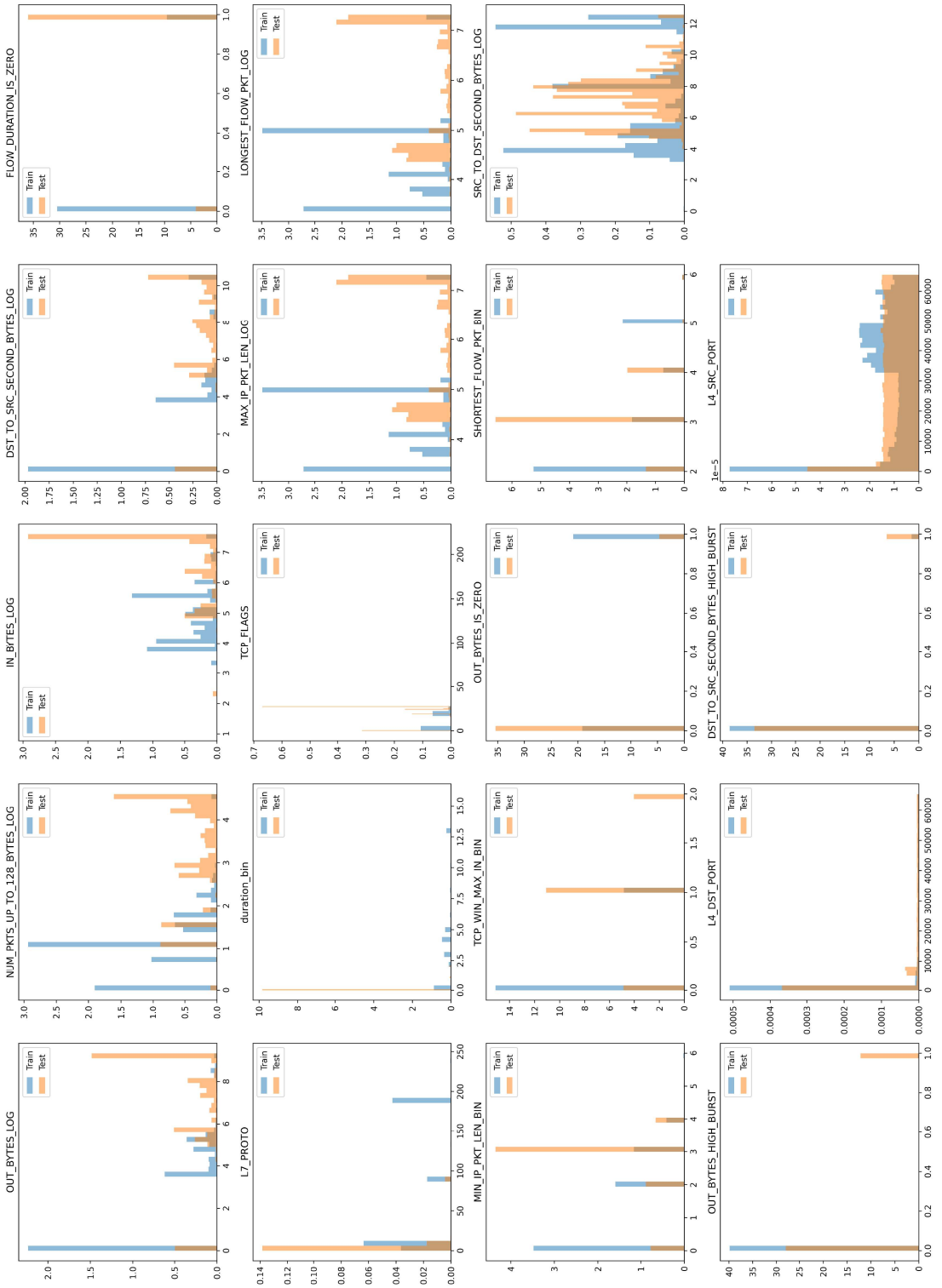
Şekil A.5. ToN-UNSW ve BoT_IoT için KDE Farkları

A.2 BoT-ToN Eğitim ve UNSW-NB15 Test Seti için Yayılım Kayma Grafikleri

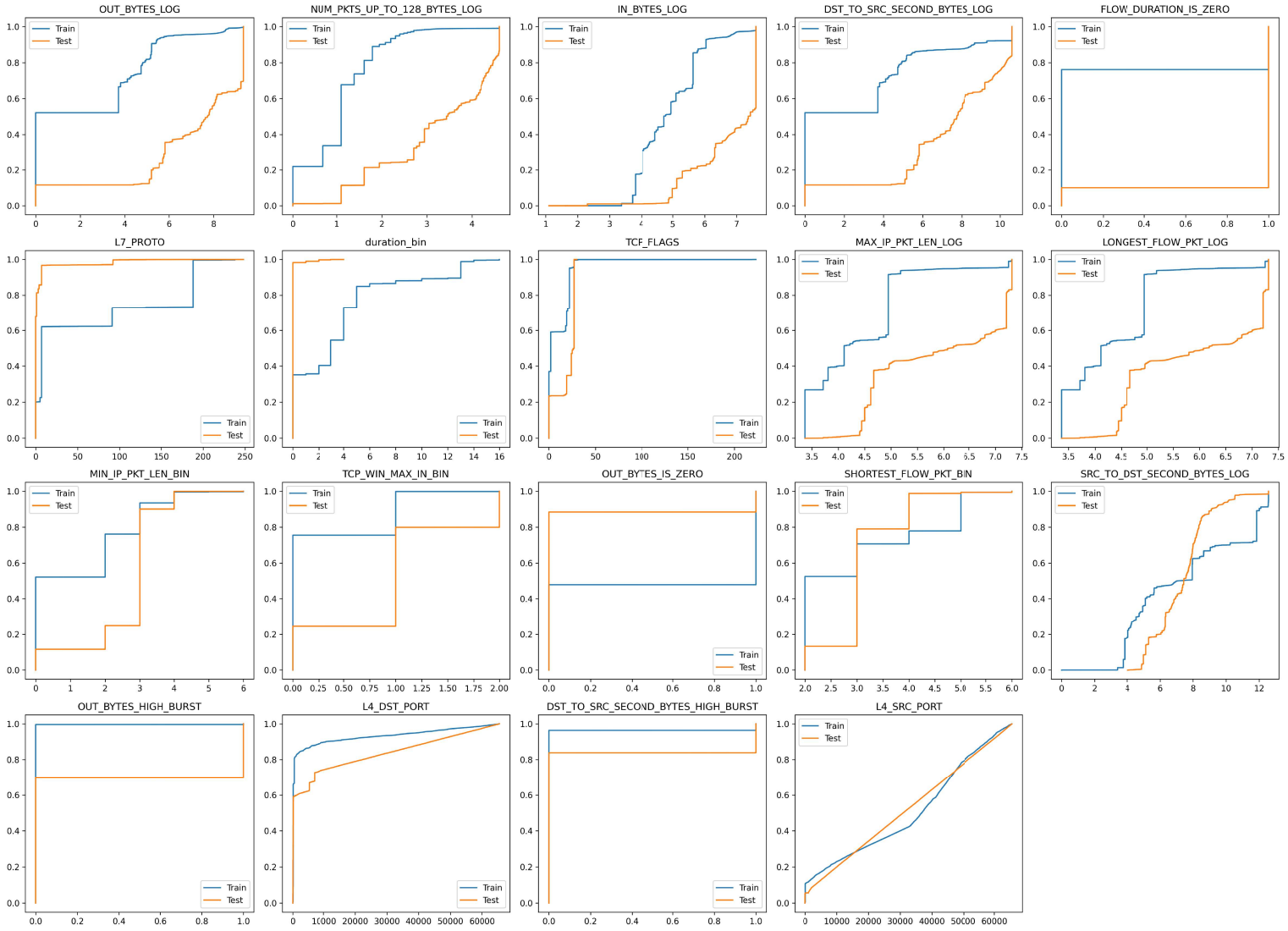
Bu senaryoda ise BoT_IoT ve ToN_IoT veri kümelerinin birleşimi eğitim seti olarak kullanılmış; UNSW-NB15 ise test verisi olarak kullanıldığı senaryo için bir yayılım analizi gerçekleştirilmiştir. Şekil A.6-A.10 arasındaki yer alan grafiklerde, dağılımsal yapıların önceki deneylerle benzer düzeyde tutarsızlık gösterdiği izlenmektedir. Bu durum, UNSW test verisinin de daha heterojen kaynaklardan geldiğini ve modelin genelleme yetisini zorlayabilecek nitelikte olduğunu göstermektedir. Genel olarak, değişken kayması bu senaryoda da önceki bulgularla tutarlı şekilde gözlemlenmiştir.



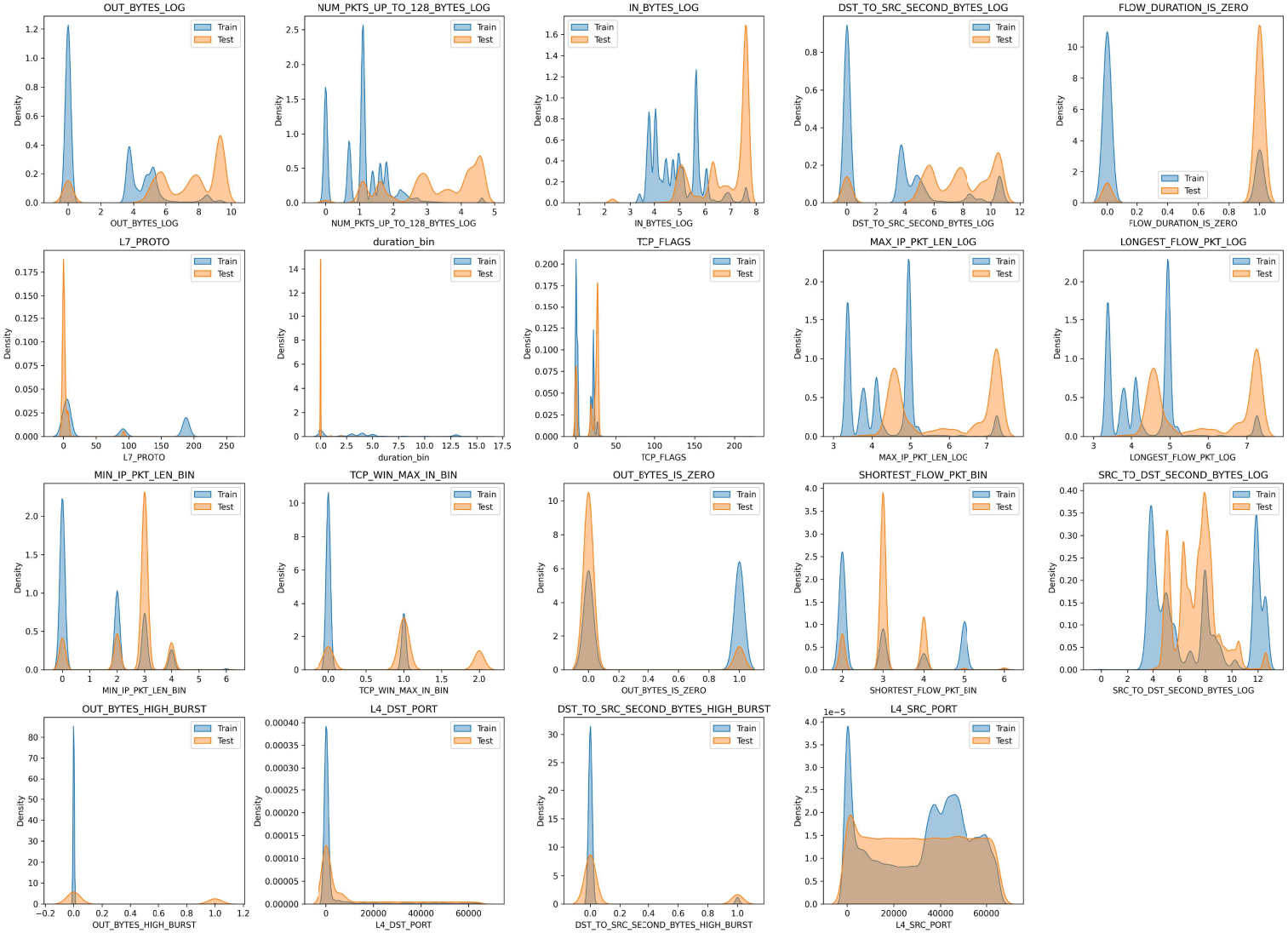
Şekil A.6. BoT-ToN ve UNSW-NB15 için Yayılım Metrikleri



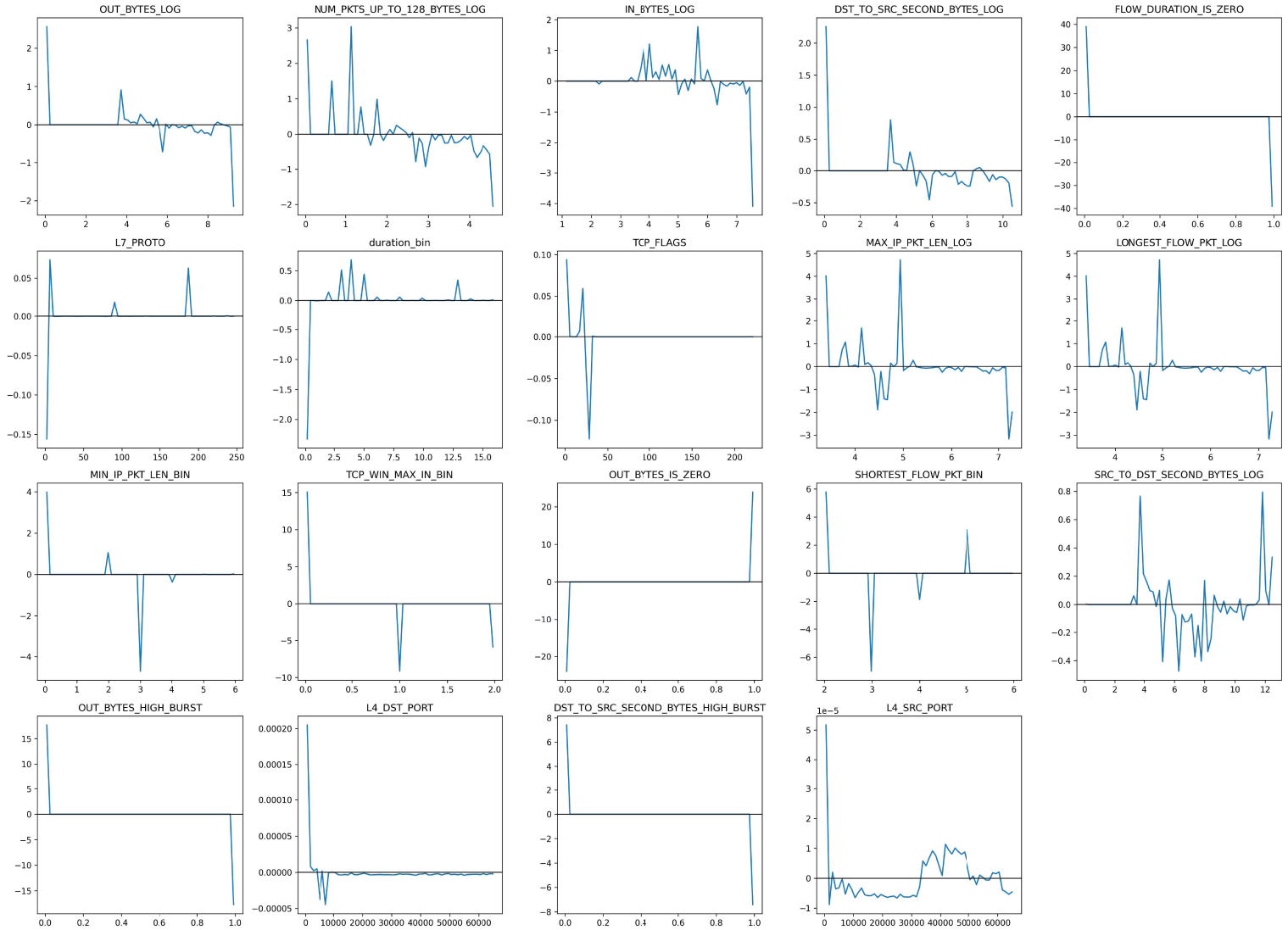
Şekil A.7. BoT-ToN ve UNSW-NB15 için Histogramlar



Şekil A.8. BoT-ToN ve UNSW-NB15 için ECDF Grafikleri



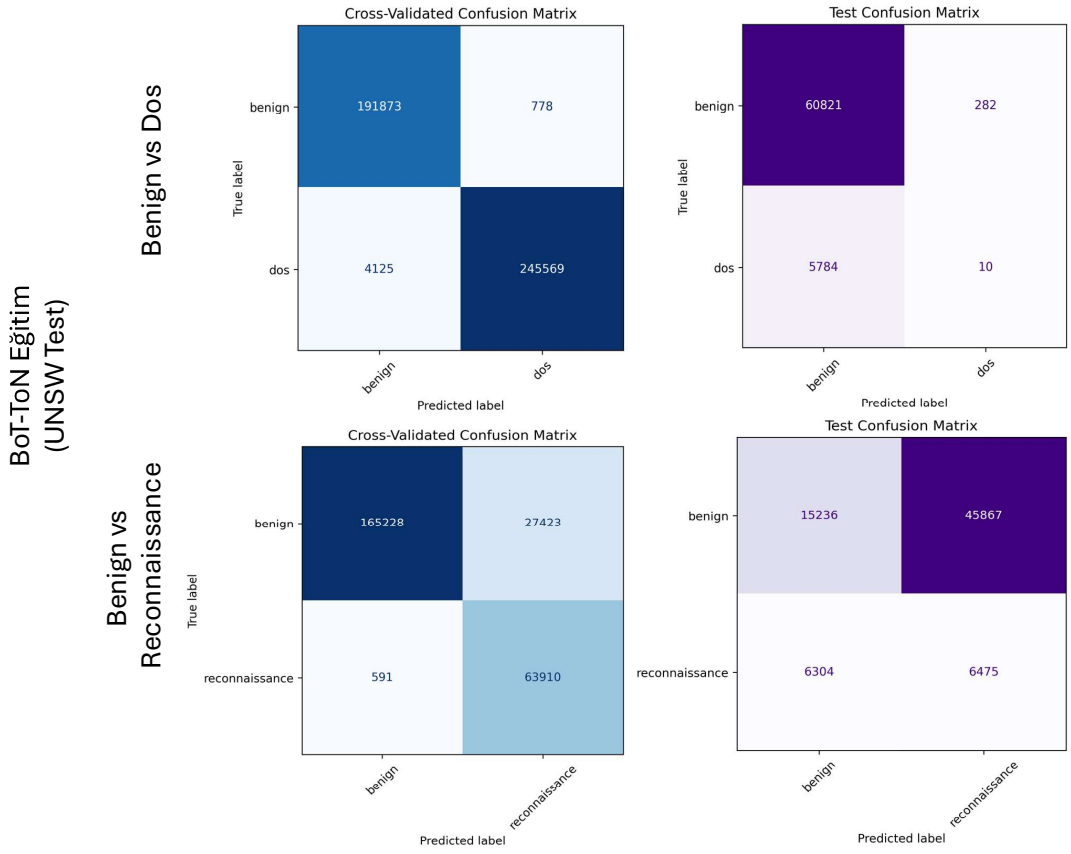
Şekil A.9. BoT-ToN ve UNSW-NB15 için KDE Grafikleri



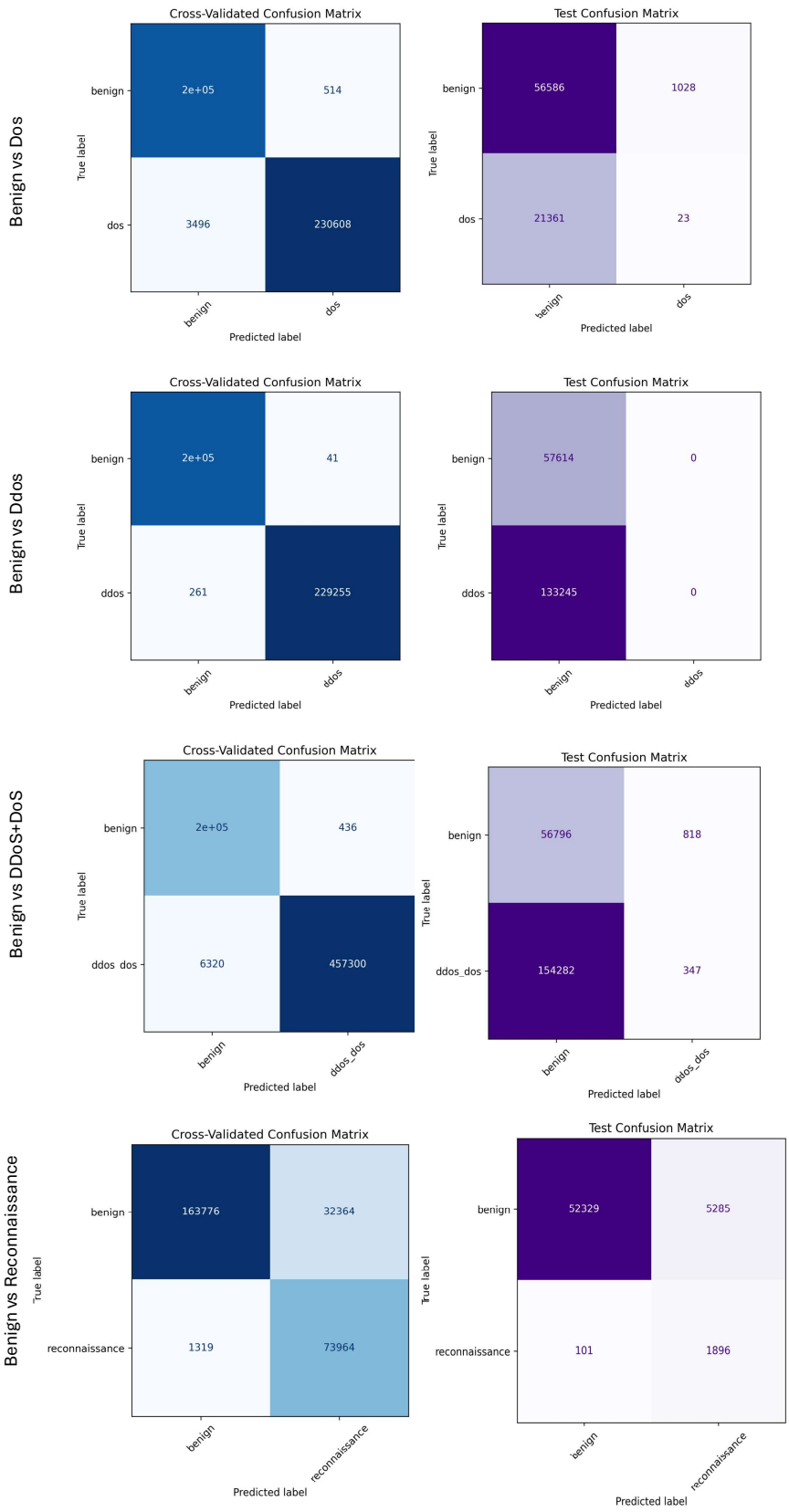
Şekil A.10. BoT-ToN ve UNSW-NB15 için KDE Farkları

A.3 Gelenksel Modeller için Katışıklık Matrisleri

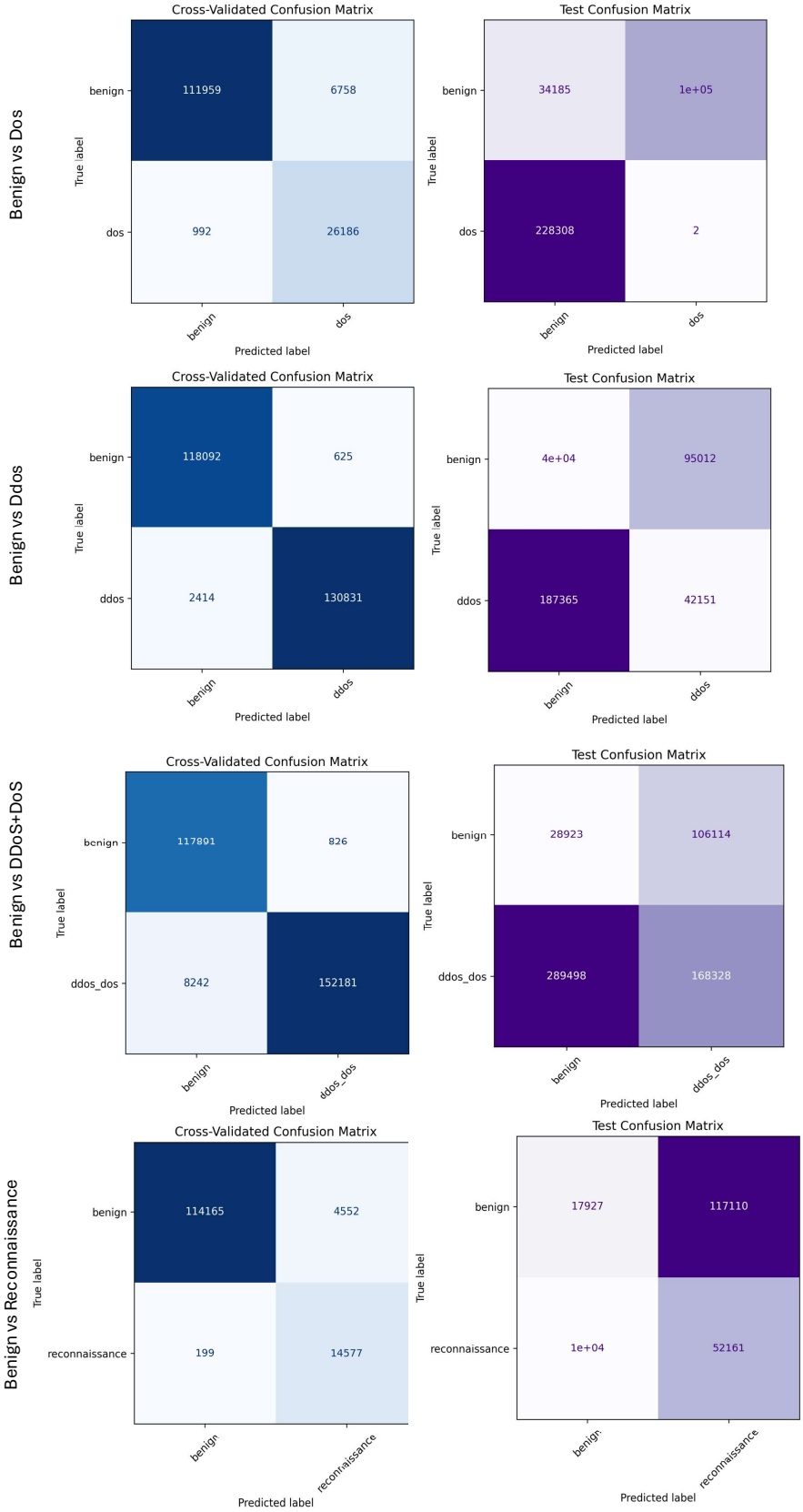
Bu bölümde, Bölüm 7.4.'te bahsedilen, değişken kayması etkisini gözlemlemek için eğitilen geleneksel makine öğrenmesi yöntemlerine dayalı modellere ait katışıklık matrisleri Şekil A.11 - Şekil A.13 arasında gösterilmiştir.



Şekil A.11. BoT-ToN Eğitim için Katışklık Matrisleri



Şekil A.12. BoT-UNSW Eğitim için Katışklık Matrisleri



Şekil A.13. ToN-UNSW Eğitim için Katışklık Matrisleri

KİŞİSEL YAYIN VE ESERLER

Eren, **K. K.**, Çınar, E., Karakurt, H. U., Özgür, A. (2023). Improving the filtering of false positive single nucleotide variations by combining genomic features with quality metrics. *Bioinformatics*, *39* (12), btad694.

Eren, **K. K.**, Küçük, K. (2023). Improving intrusion detection systems for iot devices using automated feature generation based on ton_iot dataset. *International Conference on Computer Science ve Engineering (UBMK)*, Burdur, Türkiye, 13–15 Eylül 2023.

Eren, **K. K.**, Küçük, K. (2025). Exploratory Data Analysis of the ToN_IoT Dataset to Improve the Accuracy of Machine Learning-Based Intrusion Detection Applications. (Ed.), *Lecture Notes in Networks and Systems (LNNS)* (1. Bs.) (456–466). Kolkata, India: Springer.

Eren, **K. K.**, Küçük, K., Bulut, S. Z., Yiğit, Y., Akbulut, O. (2025a). KLSampler: A Kullback-Leibler Divergence Based Undersampling Technique for Machine Learning Applications. *Black Sea Conference on Communications and Networking (BlackSeaCom)*, Chisinau, Moldova, 23–25 Haziran 2025.

Eren, **K. K.**, Küçük, K., Özyurt, F., Alhazmi, O. H. (2025b). Simple Yet Powerful: Machine Learning-Based IoT Intrusion System With Smart Preprocessing and Feature Generation Rivals Deep Learning. *IEEE Access*, *13* (2025), 41435–41455.

Kalifati, E., Küçük, K., Şolpan, Ş., **Eren**, **K. K.** (2025a). A Comparative Study of SMOTE-Enhanced Machine Learning Models for IoT Device Classification on the TON IoT Dataset. *Black Sea Conference on Communications and Networking (BlackSeaCom)*, Chisinau, Moldova, 23–25 Haziran 2025.

Kalifati, E., Şolpan, Ş., **Eren**, **K. K.**, Küçük, K. (2025b). Performance Analysis of SMOTE-Enhanced Machine Learning Approaches on IoT-Based TON IoT Dataset. *Signal Processing ve Communications Applications Conference (SIU)*, İstanbul, Türkiye, 25–28 Haziran 2025.

ÖZGEÇMİŞ

Kazım Kıvanç Eren, lisans (2016) ve yüksek lisans (2019) eğitimlerini Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümünde tamamlamıştır. Halen aynı üniversitede Bilgisayar Mühendisliği alanında doktora çalışmalarına devam etmektedir. Akademik ilgi alanları arasında makine öğrenmesi, açıklanabilir yapay zekâ, öznitelik mühendisliği, derin öğrenme ve zaman serisi analizi yer almaktadır.

Mesleki kariyerine 2017 yılında IDEA Teknoloji Çözümleri A.Ş.'de başlamış, burada Kıdemli Araştırma ve Geliştirme Mühendisi olarak görev almıştır. Bu pozisyonda makine öğrenmesi tabanlı yazılım efor tahmini ve biyoinformatik alanlarında çeşitli çalışmalarda yer almıştır. 2022–2024 yılları arasında Invent Analytics bünyesinde Yazılım ve Veri Mühendisi olarak çalışmış, büyük veri altyapıları, ETL süreçleri ve API geliştirme konularında sorumluluk üstlenmiştir. 2024 yılından bu yana Hitit Bilgisayar Hizmetleri A.Ş.'de Yapay Zeka Uzmanı olarak görev yapmaktadır. Bu görev kapsamında zaman serisi verilerine yönelik analiz ve kümelendirme yöntemleri üzerinde çalışmaktadır.

Eren'in çeşitli akademik çalışmaları bulunmaktadır. IEEE Access ve Bioinformatics dergilerinde yayımlanmış iki makalesine ek olarak, farklı ulusal ve uluslararası konferanslarda çeşitli bildirileri bildiriler yayınlamış ve sunmuştur. Bu yayınlarda IoT tabanlı siber güvenlik sistemleri, genomik varyant analizi ve örnekleme stratejileri gibi konular ele alınmaktadır. Ayrıca, genetik varyantların klinik değerlendirilmesine yönelik bir puanlama ve önceliklendirme yöntemi üzerine geliştirilmiş bir sistem için patent başvurusunda bulunmuştur.

Yürüttüğü çalışmalar ağırlıklı olarak veri işleme, özellik seçimi ve açıklanabilir yapay zekâ yöntemlerinin uygulanmasına odaklanmakta; akademik bilgi birikimini uygulamalı endüstriyel projelerle desteklemektedir.