T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SALDIRI TESPİT SİSTEMLERİNDE MAKİNE ÖĞRENİMİ TEKNİKLERİNİN KULLANIMI VE ANALİZİ

BERKSU ERTUĞRUL

YÜKSEK LİSANS TEZİ

HESAPLAMALI BİLİMLER ANABİLİM DALI

Tez Danışmanı: Dr. Öğr. Üyesi Derya ARDA

EDİRNE-2024

Berksu Ertuğrul'un hazırladığı **"Saldırı Tespit Sistemlerinde Makine Öğrenimi Tekniklerinin Kullanımı ve Analizi"** başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **Hesaplamalı Bilimler** Anabilim Dalında bir **Yüksek lisans tezi** olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Dr. Öğr. Üyesi Derya ARDA (Danışman)	……
Dr. Öğr. Üyesi Deniz TAŞKIN	……
Prof. Dr. Erdinç UZUN	……

Tez Savunma Tarihi: 26/06/2024

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Dr. Öğr. Üyesi Derya ARDA

Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

……

Prof. Dr. Muharrem Tolga SAKALLI

Fen Bilimleri Enstitüsü Müdürü

T. Ü. FEN BİLİMLERİ ENSTİTÜSÜ

HESAPLAMALI BİLİMLER ANABİLİM DALI YÜKSEK LİSANS PROGRAMI

DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

26 / 06 / 2024

Berksu ERTUĞRUL

İmza

Yüksek Lisans Tezi

Saldırı Tespit Sistemlerinde Makine Öğrenimi Tekniklerinin Kullanımı ve Analizi

T.Ü. Fen Bilimleri Enstitüsü

Hesaplamalı Bilimler Anabilim Dalı

ÖZET

Günümüzün teknoloji odaklı dünyasında kamu kurum ve kuruluşlar, sağlık kuruluşları, bankalar gibi sistemler siber güvenlik riski ve tehditleri ile karşılaşmaktadırlar. Bunun gibi internet ortamında olan sistem ve uygulamaların güvenliğinin sağlanmasında Saldırı Tespit Sistemlerinin kullanımı oldukça önemlidir. İnternet ağında meydana gelen hareketleri inceleyen ve anormal davranışları tespit edebilen ve tehditlerin üstesinden gelebilen saldırı tespit sistemleri, sistemin güvenliğini sağlamada başarılı bir şekilde kullanılmaktadır. Klasik Saldırı Tespit Sistemlerinin yanı sıra makine öğrenmesi yöntemleri ile siber saldırıların tespiti ve analizi ile sistemlerin güncel tutulabilmesi sağlanabilmektedir. Bu tezde, sisteme yapılan kötü amaçlı faaliyetleri tespit edebilen makine öğrenimi tabanlı bir saldırı tespit sisteminin başarımı için kapsamlı bir analiz çalışması yapılmıştır. Bu araştırmada veri seti olarak CSE-CIC-IDS-2018 veri seti kullanılmıştır. Çalışma gerçekleştirilirken her saldırı türüne ait veriler için öncelikle veri önişleme, ardından üç farklı öznitelik seçme yönteminden toplanan bilgiler analiz edilerek her bir saldırı için öznitelikler seçilmiştir. Daha sonra makine öğrenmesi teknikleri ile STS'nin başarısı ve performansı doğruluk, duyarlılık, kesinlik ve F-Skoru ve karmaşıklık matrisi gibi metriklerle değerlendirilmiştir. Yapılan çalışmalar sonucunda tüm saldırı çeşitlerinde ortalama olarak en iyi başarıyı yaklaşık %95 doğruluk oranıyla Rassal Orman algoritması sağlamıştır.

Yıl : 2024

Sayfa Sayısı : 122

Anahtar Kelimeler : Siber Güvenlik, Siber Saldırılar, Saldırı Tespit Sistemleri, Makine Öğrenimi, Anomali Tespit

Master's Thesis
Analysis and Use of Machine Learning Techniques in Intrusion Detection Systems
Trakya University Institute of Natural and Applied Sciences
Department of Computational Science

ABSTRACT

In today's technology-oriented world, systems such as public institutions and organizations, healthcare institutions and banks face cyber security risks and threats. The use of Intrusion Detection Systems is very important in ensuring the security of systems and applications on the internet. Intrusion detection systems, which examine the movements occurring on the Internet network, detect abnormal behavior and overcome threats, are used successfully to ensure the security of the system. In addition to Classical Intrusion Detection Systems, it is possible to keep systems up-to-date by detecting and analyzing cyber attacks with machine learning methods. In this thesis, a comprehensive analysis study was conducted for the performance of a machine learning-based intrusion detection system that can detect malicious activities on the system. CSE-CIC-IDS-2018 data set was used as the data set in this research. While carrying out the study, attributes were selected for each attack by first preprocessing the data for each attack type, and then analyzing the information collected from three different feature selection methods. Then, the performance and performance of STS were evaluated with machine learning techniques and metrics such as accuracy, sensitivity, precision and F-Score and confusion matrix. As a result of the studies, the Random Forest algorithm achieved the best success on average in all attack types, with an accuracy rate of approximately 95%.

Year : 2024
Number of Pages : 122
Keywords : Cyber Security, Cyber Attack, Intrusion Detection Systems, Machine Learning, Anomaly Detection

TEŞEKKÜR

Tez danışmanım Dr. Öğrt. Üyesi Derya ARDA'ya sonsuz teşekkürlerimi sunarım. Sabrı, bilgeliği ve rehberliği olmadan bu çalışma gerçekleştirilemezdi. Bana gösterdiği ilgi ve destek için minnettarım.

Bilgisayar Mühendisliği Bölümü Araştırma Görevlisi Yaren ÖZSOY'a bilgi paylaşımı ve yardımları için teşekkürlerimi sunuyorum.

Ayrıca, aileme sonsuz sevgi ve destekleri için teşekkür etmek istiyorum. Her zaman beni cesaretlendirdiler ve bu tez sürecindeki zorlukları aşmam için yanımda oldular.

İÇİNDEKİLER

ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
ŞEKİLLER DİZİNİ xi
ÇİZELGELER DİZİNİ xiii
SİMGELER VE KISALTMALAR DİZİNİ xiv
BÖLÜM 1 1
GİRİŞ 1
BÖLÜM 2 5
SİBER GÜVENLİK 5
2.1. Siber Güvenliğin Tanımı 5
2.2. Bilgi Güvenliği 7
2.3. Siber Güvenliğe Duyulan İhtiyaç 8
2.4. Saldırılar ve Saldırı Tipleri 9
2.4.1. Hizmeti Reddi (DoS / DDoS) Saldırıları 10
2.4.2. Oltalama Saldırıları 12
2.4.3. SQL Enjeksiyonu Saldırıları 12
2.4.4. IP Sahteciliği 13
2.4.5. Ortadaki Adam Saldırıları 13
2.4.6. ARP Zehirlenmesi 14

2.4.7. Bot Saldırıları 14
2.4.8. Kaba Kuvvet Saldırıları 15
2.4.9. Sıfır-gün Saldırıları 16
2.4.10. Zararlı Yazılımlar 17
2.5. Kayıt Altına Alınan Büyük Çaplı Saldırılar 17
2.5.1. SolarWinds Siber Saldırısı 17
2.5.2. WannaCry Fidye Yazılımı Saldırısı 18
2.5.3. Marriott International Veri İhlali 18
2.5.4. Equifax Veri İhlali 18
2.5.5. Capital One Veri İhlali 19
2.5.6. Facebook-Cambridge Analytica Veri İhlali 19
BÖLÜM 3 20
SALDIRI TESPİT SİSTEMLERİ VE MAKİNE ÖĞRENİMİ 20
3.1. Saldırı Tespit Sistemleri Tipleri 22
3.1.1. Saldırı Algılama Yöntemine göre Saldırı Tespit Sistemleri 22
3.1.2. Sistemdeki Konumuna Göre Saldırı Tespit Sistemleri 24
3.1.3. Veri İşleme Zamanına Göre Saldırı Tespit Sistemleri 26
3.2. Makine Öğrenimi 27
3.2.1. Denetimli Öğrenme 28
3.2.2. Denetimsiz Öğrenme 28
3.2.3. Pekiştirmeli Öğrenme 28
3.3. Akıllı Saldırı Tespit Sistemleri 29
BÖLÜM 4 32
LİTERATÜR 32
4.1. Benzer Çalışmaların Özeti 32

4.2. Daha Önce Yapılan Çalışmaların Bulguları 35
BÖLÜM 5 42
MATERYAL VE YÖNTEM 42
5.1. Önerilen Sistem 42
5.2. Kullanılan Algoritmalar 43
5.2.1. Naive Bayes 43
5.2.2. K-En Yakın Komşu 45
5.2.3. Rassal Ormanlar 46
5.2.4. Karar Ağaçları 47
5.2.5. AdaBoost 48
5.2.6. Çok Katmanlı Algılayıcı 49
5.2.7. Doğrusal Ayrımcılık Analizi 50
5.2.8. Lojistik Regresyon 52
5.2.9. Gradyan Artırıcı 53
5.3. CSE-CIC-IDS-2018 55
5.4. Kullanılan Dil ve Kütüphaneler 57
5.5. Veri Önişleme 58
5.6. Öznitelik Çıkarımı 59
5.6.1. DDoS saldırıları için seçilen öznitelikler 60
5.6.2. DoS saldırıları için seçilen öznitelikler 60
5.6.3. Kaba Kuvvet saldırıları için seçilen öznitelikler 60
5.6.4. Botnet saldırıları için seçilen öznitelikler 61
5.6.5. Sızma saldırıları için seçilen öznitelikler 61
5.6.6. Web saldırıları için seçilen öznitelikler 62
5.7. Kullanılan Performans Metrikleri 62

BÖLÜM 6 66
BULGULAR 66
6.1. Makine Öğrenimi Sonuçları 66
6.1.1. DDoS Saldırılarının Tespitinin Sınıflandırma Sonuçları 67
6.1.2. DoS Saldırılarının Tespitinin Sınıflandırma Sonuçları 69
6.1.3. Brute Force Saldırılarının Tespitinin Sınıflandırma Sonuçları 72
6.1.4. Botnet Saldırılarının Tespitinin Sınıflandırma Sonuçları 75
6.1.5. Sızma Saldırılarının Tespitinin Sınıflandırma Sonuçları 78
6.1.6. Web Attack Saldırılarının Tespitinin Sınıflandırma Sonuçları 81
6.2. Verilerin ve Bulguların Demografik Özellikleri ve Karşılaştırılması 84
BÖLÜM 7 89
TARTIŞMA VE SONUÇ 89
KAYNAKLAR 96
ÖZGEÇMİŞ 106

ŞEKİLLER DİZİNİ

Şekil 2.1. Siber güvenlik türleri 6
Şekil 2.2. Bilgi güvenliği 7
Şekil 3.1. STS içeren örnek topoloji 20
Şekil 3.2. STS tipleri 22
Şekil 3.3. Saldırı algılama yöntemine göre STS 23
Şekil 3.4. Sistemdeki konumuna göre STS 24
Şekil 3.5. Veri işleme zamanına göre STS 26
Şekil 3.6. Makine öğrenimi temel çeşitleri 28
Şekil 5.1. Önerilen sistem akış şeması 43
Şekil 6.1. DDoS veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları ... 69
Şekil 6.2. DoS veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları 72
Şekil 6.3. Brute Force veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları 75
Şekil 6.4. Botnet veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları .. 78
Şekil 6.5. Infilteration veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları 81
Şekil 6.6. Web Attack veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları 84
Şekil 6.7. Saldırı ve Normal verilerinin toplamının Saldırılara göre dağılımı 85
Şekil 6.8. Algoritmalara göre Doğruluk oranları 85
Şekil 6.9. Algoritmalara göre F Skoru oranları 86

Şekil 6.10. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi TP değerleri 86
Şekil 6.11. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi TN değerleri..... 87
Şekil 6.12. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi FP değerleri..... 87
Şekil 6.13. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi FN değerleri..... 88

ÇİZELGELER DİZİNİ

Çizelge 4.1. Yapılan çalışmaların bulguları 36
Çizelge 5.1. CSE-CIC-IDS-2018 saldırı senaryoları 55
Çizelge 5.2. CSE-CIC-IDS-2018 AWS bulutu network trafik dağılımı 56
Çizelge 5.3. CSE-CIC-IDS-2018 veri hazırlama sonrası demografik özellikler 59
Çizelge 5.4. DDoS veri setinde seçilen özellikler 60
Çizelge 5.5. DoS veri setinde seçilen özellikler 60
Çizelge 5.6. Brute Force veri setinde seçilen özellikler 61
Çizelge 5.7. Botnet veri setinde seçilen özellikler 61
Çizelge 5.8. Infilteration veri setinde seçilen özellikler 61
Çizelge 5.9. Web Attack veri setinde seçilen özellikler 62
Çizelge 5.10. Karmaşıklık Matrisi Modeli 62
Çizelge 5.11. Karmaşıklık Matrisi Örneği 63
Çizelge 6.1. DDoS sınıflandırma ölçütleri değerlendirmesi 67
Çizelge 6.2. DoS sınıflandırma ölçütleri değerlendirmesi 70
Çizelge 6.3. Brute Force sınıflandırma ölçütleri değerlendirmesi 73
Çizelge 6.4. Botnet sınıflandırma ölçütleri değerlendirmesi 75
Çizelge 6.5. Sızma saldırılarının sınıflandırma ölçütleri değerlendirmesi 78
Çizelge 6.6. Web Attack sınıflandırma ölçütleri değerlendirmesi 81
Çizelge 6.7. Algoritmalara göre tüm saldırılardan elde edilen ortalama doğruluk oranları ile diğer çalışmaların karşılaştırılması 91
Çizelge 6.8. RF Algoritması metriklerinin karşılaştırması 92

SİMGELER VE KISALTMALAR DİZİNİ

ADR:	Ortalama Algılama Oranı
ADA:	AdaBoost Algoritması
ARP:	Adres Çözümleme Protokolü
AUC:	Eğri Altında Kalan Alan
CDF:	Kümülatif Dağılım Fonksiyonu
CWGAN:	Şartlı Wasserstein Üretici Karşıtlı Ağ
DBM:	Derin Boltzmann Makinesi
DBN:	Derin İnanç Ağı
DBNIDS:	Derin İnanç Ağı Saldırı Tespit Sistemi
DNN:	Derin Sinir Ağı
DR:	Tespit Oranı
DT:	Karar Ağacı
EIDM:	Kanıta Dayalı Karar Verme
ET:	Ekstra Ağaçlar
FAR:	Yanlış Alarm Oranı
FNNIDS:	İleri Beslemeli Yapay Sinir Ağı Saldırı Tespit Sistemi
ForestPA:	Orman Tabanlı Gizlilik-Duyarlı Saldırı Tespit Sistemi
FP:	Yanlış Pozitif
FTP:	Dosya Transfer Protokolü
GB:	Gradyan Arttırma

GBD:	Gradyan Artırmalı Karar Ağaçları
LGBFS:	LightGBM Özellik Seçimi
HTTPS:	Güvenli Hipermetin Transfer Protokolü
HTTP:	Hipermetin Transfer Protokolü
IDS:	Saldırı Tespit Sistemleri
IoT:	Nesnelerin İnterneti
J-48	C4.5 Karar Ağacı Algoritmasının Bir Çeşidi
K-Best:	K-en İyi Öznitelikler
KNN:	K-En Yakın Komşu
LDA:	Doğrusal Ayrımcılık Analizi
LGBM:	Hafif Gradyan Artırma Makinesi
LR:	Lojistik Regresyon
LTSM:	Uzun Kısa Süreli Hafıza
MCC:	Matthews Korelasyon Katsayısı
MLP:	Çok Katmanlı Algılayıcı
ML:	Makine Öğrenmesi
NB:	Naive Bayes
PRC:	Doğruluk-Duyarlılık Eğrisi
PCA:	Temel Bileşen Analizi
RBM:	Kısıtlı Boltzmann Makinesi
RF:	Rassal Orman
RNN:	Tekrarlayan Sinir Ağı
RNNIDS:	Tekrarlayan Sinir Ağı Saldırı Tespit Sistemi
ROC:	İşletim Karakteristiği Eğrisi
SGD:	Stokastik Gradyan İnişi

SNNIDS:	İğnecikli Yapay Sinir Ağı Saldırı Tespit Sistemi
SSH:	Güvenlik Kabuğu
SQL:	Yapılandırılmış Sorhu Dili
STS:	Saldırı Tespit Sistemleri
SVM:	Destek Vektör Makinesi
SVMIDS:	Derin İnanç Ağı Saldırı Tespit Sistemi
TCP:	Aktarım Kontrol Protokolü
TN:	Doğru Negatif
TP:	Doğru Pozitif
XGBoost:	Aşırı Gradyan Artırma
YSA:	Yapay Sinir Ağları

BÖLÜM 1

GİRİŞ

Günümüzdeki küresel ortamda ağ ve cihazlarda gerçekleşen veri alışverişindeki artışla beraber sistemlerdeki açıklıklardan meydana gelen güvenlik olayları da doğru orantılı olarak artmıştır. 2023 yılında yapılan bir ankete göre, evden internet erişimi olan hane halkı oranı, bir önceki yıla göre 1,4 puan artarak %95,5'e yükseldi (TÜİK, 2023). İnternetin gelişimi ve ağlardaki haraketlilik sonucunda saldırganlar ve saldırı çeşitleri de artış göstermiştir.

Ağda bilgi alışverişi yapmak, veri toplamak veya ilişkili etkinlikleri izlemek için kullanılan araçlar saldırı ihtimalini artırır. 2018'de Massachusetts Institute of Technology tarafından gerçekleştirilen teknoloji incelemesi, bulut bilişim şirketleri, yapay zekâ silahlanması, IoT sistemlerine yönelik saldırılar, kripto para birimi madenciliği ve izinsiz erişim gibi ağ güvenliği ile ilgili büyük risklerin yanı sıra, insanların kişisel bilgilerinin fidye yazılımlarının ana hedefi olduğunu ve veri güvenliğini tehlikeye attığını iddia etmiştir (Knight, 2018). Bu da ağı güvenlik saldırılarına karşı koruyabilen verimli bir saldırı tespit sistemi (STS) oluşturmaya yönelik gereksinimi ortaya koymaktadır.

STS terimi, ağ ortamındaki herhangi bir anormal davranış üzerine alarm üreten sistem olarak tanımlanabilir. STS'nin birincil görevi, sistem güvenliğini tehlikeye atan davetsiz misafirleri engellemek ve kötü niyetli faaliyetleri önlemek için çeşitli işlem kombinasyonları uygulayarak bir sistemi korumaktır. İhlal herhangi bir biçimde olabilir. Örneğin, bir davetsiz misafir, kullanıcının parolasını çalarak, maskeleyerek, gizlice dinleyerek veya kötü amaçlı kod enjekte ederek kullanıcının hesabına yetkisiz erişim sağlayabilir. Ağ içindeki kişiler, sistemi tehlikeye atabilecek yazılım uygulamasının ve/veya sunucunun güvenlik açıklarını kullanarak bir sisteme zarar verebilir. Ağı herhangi bir tehdide karşı korumak için güvenlik duvarları, parola şifreleme, erişim

kontrolü ve saldırı önleme sistemleri gibi birçok araç ve hizmet kullanılır. Genellikle bir güvenlik uzmanı izinsiz girişlerin neden olduğu zararı azaltmak için gerekli önlemleri alabilir. Fakat, ağ ve bilgisayar üzerindeki tehditlerin listesi sonsuzdur ve sürekli gelişmektedir, bu nedenle saldırı tespit sistemleri aktif bir araştırma alanı olmaya devam etmektedir.

Dünya çapında bilgi çalmak, gayrimeşru yollardan gelir elde etmek ve yeni hedefler bulmak için motive olmuş çok sayıda siber suçlu vardır. Kötü amaçlı yazılımlar evrim geçirip daha da karmaşık hale geldikçe STS'lerin tasarımı da kritik bir zorluk teşkil eder. Kötü amaçlı yazılımları tespit etmek için yeni, karmaşık ve verimli bir saldırı tespit sistemi geliştirmeye ihtiyaç vardır. Geliştirilen sistemler, ağ trafiğini analiz edebilir ve kötü amaçlı yazılım, ağ sızıntıları ve hizmet reddi saldırıları gibi potansiyel güvenlik tehditlerini tanımlayabilirler. Ancak, ağ trafiğinin giderek artan karmaşıklığı ve çeşitliliği, geleneksel kural tabanlı STS kullanarak ağ trafiğini doğru bir şekilde sınıflandırmayı zorlaştırmıştır (Thakkar & Lohiya, 2022).

Bir STS'nin amacı, geleneksel bir güvenlik duvarı tarafından elde edilemeyen farklı kötü amaçlı yazılım türlerini olabildiğince erken tespit etmektir. Bilgisayardaki kötü amaçlı yazılım hacminin artmasıyla birlikte, iyileştirilmiş STS'lerinin geliştirilmesi son derece önemli hale gelmiştir. Son birkaç on yılda, izinsiz giriş tespitini iyileştirmek için makine öğrenimi kullanılmıştır ve şu anda da güncel yazılımlara ihtiyaç vardır.

Makine Öğrenimi (ML) teknikleri STS'lerde ağ trafiği sınıflandırması için geniş ölçüde benimsenmiştir. Bu teknikler, istatistiksel modellerin ve algoritmaların gücünden yararlanarak, güvenlik tehditlerinin belirteci olan anormal ağ trafiği desenlerini otomatik olarak öğrenip tespit edebilirler. ML algoritmaları, büyük miktarda veriyi analiz edebilir ve geleneksel kural tabanlı veya imza tabanlı STS tarafından kolayca algılanamayan kalıpları belirleyebilir.

Anormallik tespiti için yapılan çalışmaları ve ML ve STS'nin kullanıldığı farklı araştırma alanlarını incelemek; Chandola, Banerjee ve Kumar'ın ana odak noktasıydı (Chandola, Banerjee, & Kumar, 2009). Araştırma çalışmalarında, anormallik tespiti üzerine yapılan araştırmaların yapılandırılmış ve kapsamlı bir genel bakışını sağlamaya çalışmışlardır.

Özniteliklerin fazlalığı ve ilgisizliği gibi sorunların çözümü öznitelik seçim teknikleri kullanılarak sağlanır (John, Kohavi, & Pfleger, 1994). Gerçek ağ senaryolarında yüksek boyutlu ağ verilerinin ortaya çıkması nedeniyle öznitelik seçimi zorunlu hale gelmiştir. Bir STS, büyük miktarda veriyi işleme kapasitesine sahiptir, ancak alakasız ve gereksiz verilerin varlığı, STS'nin performansını bozabilir (Sung & Mukkamala, 2004). Sınıflandırma doğruluğunu artırmak için farklı ML ve veri madenciliği yöntemleri ile gelişmiş öznitelik seçim teknikleri kullanılır.

Tanrıkulu ve Sazlı çalışmalarında, ağ trafiği üzerinde hangi saldırı yöntemlerinin kullanıldığını belirlemeyi amaçladılar, bunu yaparken yapay sinir ağları kullandılar. İlk olarak, STS'lerin işleyişinden bahsettiler ve ardından çeşitli saldırı tipleri incelendi. Çalışmada, makine öğrenmesi yöntemlerinden biri olan yapay sinir ağı yöntemlerinden çok katmanlı algılayıcı (MLP) kullanıldı ve elde edilen sonuçlar Matlab ortamında değerlendirildi. Yapılan araştırma sonucunda, DoS saldırılarının sıkça görülen motifler olduğu belirlendi ve YSA'nın tüm saldırıları başarıyla öğrenebildiği sonucuna varıldı (Tanrıkulu & Sazlı, 2009).

Soni ve Sharma (2014), KDD'99 veri setinin %10'unu kullanarak yaptıkları çalışmada, Karar Ağaçları (DT) ve YSA yöntemlerini kullanarak özelliklerin modelin başarısına etkisini incelemişlerdir. U2R ve Probe saldırıları için sırasıyla %99,99 ve %99,94 doğruluk oranları elde edilmiştir. DoS ve R2L saldırıları için ise sırasıyla %99,94 ve %99,92 doğruluk oranlarına ulaşılmıştır (Soni & Sharma, 2014).

Özekes ve Karakoç'un (2019) çalışmasında, DT ve Rassal Orman (RF) yöntemleri kullanılarak gerçekleştirilen araştırmada, DDoS, DoS ve Port Tarama saldırıları anormal olarak etiketlenmiş ve deney sonuçlarına göre RF yöntemi %99,966 doğruluk oranıyla daha başarılı bulunmuştur (Ozekes & Karakoc, 2019). Dikkat çeken bir başka çalışma olan Takaoğlu ve Özer'in (2019) ortak çalışmasında, çeşitli saldırı verileri bir araya getirilmiş ve bu veri seti destek vektör makineleri (SVM) ve Naive Bayes (NB) algoritmalarıyla değerlendirilmiştir. SVM'nin, NB sınıflandırıcılarına göre daha yüksek doğruluk ve tahmin oranları sunduğu gözlemlenmiştir. Çalışmanın sonucunda, öğrenme tekniklerinin hibrit bir model olarak kullanılmasının da faydalı olabileceği önerilmektedir (Takaoğlu & Özer, 2019).

Sharafaldin, Lashkari ve Ghorbani (2019) çalışmalarında her tehdidi belirleyen en iyi dört özniteliği belirlemek amacıyla RF kullanılarak KNN, RF, ID3, Adaboost, MLP, NB ve ikinci dereceden ayırım analizi, ML algoritmalarıyla eğitim yapmıştır. Bu eğitim sonucunda, RF ve ID3 algoritmalarının %98 doğrulukla en iyi tahminleri gerçekleştirdiği gözlemlenmiştir (Sharafaldin, Lashkari, & Ghorbani, 2019).

Genel olarak, STS'ler modern bilgisayar ağlarının güvenliğini sağlamak için gereklidir. Araştırma çalışmaları, STS'lerin doğruluğunu, verimliliğini ve etkililiğini geliştirmeye önemli ölçüde katkıda bulunmuştur. Teknolojinin hızlı evrimi ile henüz bilinmeyen, karmaşık ve gelişmiş saldırıların tespitinde yapay zekâ algoritmalarından faydalanılmakladır. Bu tez çalışmasında da modern bilgisayar ağlarının güvenliğini sağlamak için kullanılan STS'lerin doğruluğunu, verimliliğini ve etkililiğini artırmak amacıyla yapay zekâ algoritmalarının rolü incelenecektir.

Bu tezde, öncelikle Siber Güvenlikle ilgili temel kavramlar kısaca açıklanmıştır. Ardından, Saldırılar ve saldırı tipleri konuları ele alınmış ve bununla birlikte saldırı tespit sistemleri hakkında bilgi verilmiştir. Daha sonraki bölümde, literatür altında önceki çalışmaların bulguları incelenmiş ve özetlenmiştir. Materyal ve yöntem kısmında ise uygulama için seçilen CSE-CIC-IDS2018 veri seti tanıtılmış ve öznitelikleri açıklanmıştır. Yöntem bölümünde, makine öğrenmesi üzerine odaklanılmış ve kullanılan ML algoritmalarıyla birlikte algoritmaların işleyişi açıklanmıştır. Önerilen sistem olarak, veri hazırlık ve öznitelik çıkarımı adımları incelenmiş ve son olarak bulgular bölümünde Yapay Sinir Ağları (MLP), Naive Bayes (NB), K-En Yakın Komşu (KNN), Karar Ağacı (DT), Adaboost (ADA), Doğrusal Ayırma Analizi (LDA), Lojistik Regresyon (LR), Gradyan Artırma (GB) ve Rassal Orman (RF) gibi sınıflandırma algoritmalarının karşılaştırmalı sonuçları sunulmuştur. Tartışma ve sonuç kısmında ise yapılan analizlerle en iyi sonuçları veren algoritmaların başarı oranları kıyaslanmış ve performans değerlendirmeleri yapılmıştır.

BÖLÜM 2

SİBER GÜVENLİK

Teknolojiye ve internete olan bağımlılığımızın artmasıyla birlikte, siber güvenlik endişesi giderek daha önemli hale gelmiştir. Kötü niyetli yazılımlar, kimlik avı saldırıları ve bilgisayar korsanlığı gibi siber tehditler, maddi kayıplara, veri sızıntılarına ve hatta fiziksel zararlara neden olabilir.

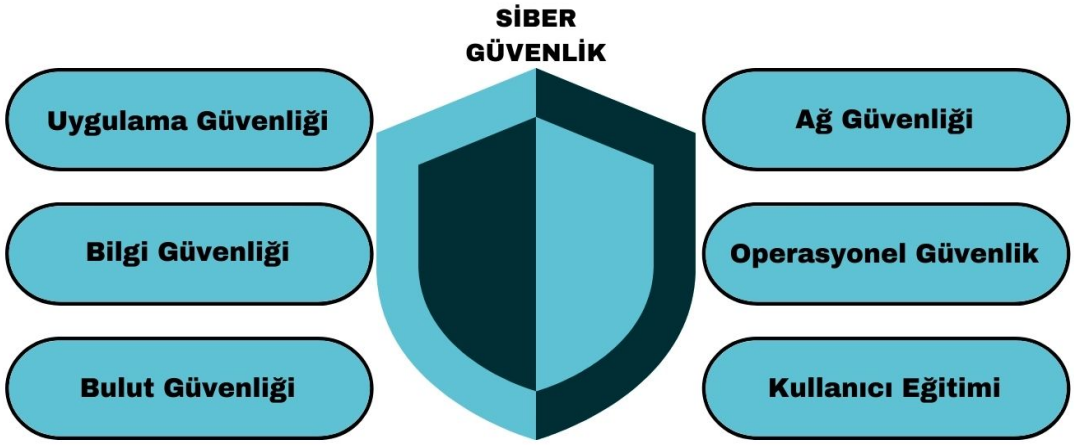
Siber güvenlik, bilgi, ağlar ve verileri hem iç hem de dış tehditlere karşı korumak için kullanılan pratik önlemleri kapsar. Siber güvenlik uzmanları; ağları, sunucuları, intranetleri ve bilgisayar sistemlerini korur. Bu, sadece yetkili kişilerin bu bilgilere erişmesini sağlayarak gerçekleşir. Daha etkili bir koruma sağlamak için farklı siber güvenlik türlerini anlamak önemlidir. Şekil 2.1, çeşitli siber güvenlik türlerini göstermektedir. Bölüm 2'de; siber güvenliğin genel tanımı, bilgi güvenliği ve siber güvenliğe duyulan ihtiyaç incelenecektir.

2.1. Siber Güvenliğin Tanımı

Bir bilgisayar sistemindeki veya mobil cihazlardaki olası ihlaller ve güvenlik ihlalleri arasında, birkaç isim vermek gerekirse, muhtemelen zarar verme niyetiyle bilgilere yetkisiz erişim sağlanması, bilgilerin imhası ve değiştirilmesi yer alır. Bahsedilen tüm güvenlik ihlallerinin olası risk ve tehlikesi tehdit olarak adlandırılır ve herhangi bir ihlal yapmaya yönelik herhangi bir girişim saldırı olarak adlandırılır. Siber güvenlik birkaç şekilde tanımlanabilir.

Siber güvenlik; veri, yazılım, donanım, sistem, süreç, ve kapasite güvenliğinin siber dünyada korunmasıdır. Genel olarak bilişim sistemlerinin korunmasını ve bu ortamın gizliliğinin, bütünlüğünün ve erişilebilirliğinin sağlanması, meydana gelebilecek herhangi bir saldırının tespiti ve önlenebilmesidir (Sağıroğlu & Akleylek, Siber Güvenlik ve Savunma, 2018).

Kaspersky'nin (2020) siber güvenlik tanımı, bilgisayarlara, sunuculara ve bir bilgisayar ağındaki ve mobil cihazdaki verilere yönelik kötü niyetli saldırılara karşı bir savunma mekanizmasına sahip olmayı içerir (Kaspersky, ty). Kaspersky ayrıca siber güvenliği ağ güvenliği, bilgi güvenliği ve diğer kategorilere ayırmıştır. Daha iyi koruma için siber güvenlik türlerini bilmek gerekir. Şekil 2.1'de farklı siber güvenlik türleri gösterilmektedir.



Şekil 2.1. Siber güvenlik türleri

Ağ Güvenliği: Ağ güvenliği, bilgisayar ağını kötü amaçlı yazılım veya bilgisayar korsanlığı olabilen bozuculardan korur. Ağ güvenliği, kuruluşların bilgisayar ağlarını; bilgisayar korsanlarının, organize saldırganların ve kötü amaçlı yazılımların erişiminden uzak tutmasını sağlayan bir dizi çözümdür.

Uygulama Güvenliği: Donanım ve yazılım (anti-virüs programları, şifreleme ve güvenlik duvarları) kullanma sistemi uygulama geliştirmeye müdahale edebilecek dış tehditlere karşı korur.

Bilgi Güvenliği: Fiziksel ve dijital verilere yetkisiz erişime, ifşaya, kötüye kullanıma, yetkisiz değişikliklere ve silinmeye karşı korur. Bilgi güvenliği konusu Başlık 2.2'de daha detaylı olarak açıklanacaktır.

Operasyonel Güvenlik: Verileri kontrol etmek ve korumak için alınan süreçleri ve kararları içerir. Örneğin, ağa erişirken kullanıcı izinleri veya bilgilerin ne zaman ve nerede depolanabileceğini veya paylaşılabileceğini belirten işlemlerdir.

Bulut Güvenliği: Buluttaki bilgileri korur (yazılıma dayalı olarak) ve yerinde saldırı risklerini ortadan kaldırır.

Kullanıcı eğitimi: Siber güvenliğin öngörülemeyen yönlerini, yani bireyleri ifade eder. Herkes yanlışlıkla güvenlik sistemine virüs bulaştırabilir. Kullanıcıya anonim USB'lere bağlanmamayı, e-postadaki şüpheli ekleri kaldırmayı öğretmek ve diğer kritik sorunlar, herhangi bir şirketin kurumsal güvenlik planının parçası olmalıdır.

2.2. Bilgi Güvenliği

Bilgi güvenliği, bilgi teknolojileri ve iletişim sistemlerinin kullanımıyla ilişkili risklerin tanımlanması, değerlendirilmesi ve yönetilmesini sağlayan bir disiplindir. Uluslararası Standartlar Örgütü (ISO) tarafından tanımlanan bilgi güvenliği Şekil 2.2'de görüldüğü gibi, "Bilginin gizliliği, bütünlüğü ve erişilebilirliğini" korumayı amaçlar.



Şekil 2.2. Bilgi güvenliği

Bu tanım, "ISO/IEC 27000 Ailesi- Bilgi Güvenliği Yönetim Sistemleri" standartlarında detaylandırılmıştır. Bu standartlar, organizasyonların bilgi varlıklarını korumak için uygulanabilecek en iyi uygulamaları belirler ve bilgi güvenliği yönetim sistemlerinin kurulması, uygulanması, denetlenmesi ve sürekli iyileştirilmesi için bir

çerçeve sunar. Bilgi güvenliği, sadece teknik önlemlerle değil, aynı zamanda politika, eğitim, farkındalık ve sürekli iyileştirme gibi alanlarda da dikkate alınmalıdır (ISO, 2018).

Çeşitli araştırmalara göre, bilgi güvenliği genellikle insan, teknoloji ve organizasyon bileşenlerine ayrılmaktadır. Cherdantseva ve Hilton'un (2013) çalışmalarına göre ise, bilgi güvenliği bileşenleri dört ana başlık altında incelenmektedir. Bu bileşenler; organizasyonel (politika, denetim, strateji vb.), teknolojik (biyometri, güvenlik duvarı, dijital imza vb.), insan odaklı (eğitim, kültür, farkındalık vb.) ve yasal (mevzuat, servis seviyesi anlaşmaları vb.) unsurları içermektedir (Cherdantseva & Hilton, 2013).

Hem bileşenlerin hem de bu bileşenlere yönelik alınabilecek tedbirlerin sınıflandırılmasında insan unsurunun en kritik öneme sahip olduğu kabul edilir. Günümüzde, teknolojik, yasal ve kurumsal alanlarda yapılan birçok çalışma varken, insan odaklı çalışmalar genellikle göz ardı ediliyor. Ancak, bu çok katmanlı yapının, bilgi güvenliği açısından önemli bir bilgi birikimi ve farkındalık düzeyi sunmasıyla birlikte, insan faktörünün de farklılaştığı düşünülmektedir.

2.3. Siber Güvenliğe Duyulan İhtiyaç

Sürekli artan siber saldırı tehdidi, veri ihlalleri ve kimlik hırsızlığı, hassas bilgilerin korunmasına yönelik proaktif bir yaklaşım gerektirir. Bu başlık altında, siber güvenliğe duyulan ihtiyacı ve dijital varlıklarımızı korumanın neden önemli olduğunu keşfedeceğiz.

Siber güvenliğe duyulan ihtiyacın başlıca nedenlerinden biri, bireylerin maruz kaldığı siber tehditlerin sayısındaki artıştır. Bilgisayar korsanları ve siber suçlular, bilgisayar sistemlerine, ağlara ve kişisel cihazlara yetkisiz erişim elde etmek için sürekli olarak yeni ve gelişmiş yollar geliştiriyor. Cybersecurity Ventures tarafından hazırlanan bir rapora göre, siber suçların 2025 yılına kadar dünyaya yılda 10,5 trilyon dolara mal olması bekleniyor (Cybercrime Magazine, 2020).

Mali kayıpların yanı sıra, siber saldırılar bir kuruluşun itibarının zedelenmesine ve müşteri güveninin kaybına da neden olabilir. Bir veri ihlali, kişisel ve finansal veriler, ticari sırlar ve gizli ticari bilgiler gibi hassas bilgilerin açığa çıkmasına neden olabilir. Bu tür olayların bir kuruluşun operasyonları ve kârlılığı üzerinde önemli bir etkisi olabilir.

Ponemon Institute tarafından yapılan bir araştırmaya göre, bir ABD şirketi için bir veri ihlalinin ortalama maliyeti 2020'de 8,64 milyon dolardı (IBM, 2023).

Siber güvenlik ihtiyacı yalnızca kurum ve işletmelerle sınırlı değildir. Bireyler olarak, banka hesabı bilgilerimiz, kredi kartı bilgilerimiz ve diğer hassas veriler dahil olmak üzere cihazlarımızda çok büyük miktarda kişisel bilgi depolarız. Siber suçlular bu bilgileri kimlik hırsızlığı ve finansal dolandırıcılık için kullanabilir. Bu nedenle, bireylerin kişisel bilgilerini siber tehditlerden korumak için gerekli önlemleri almaları önemlidir. Siber tehditlerle ilişkili riskleri azaltmak için kuruluşların ve bireylerin etkili siber güvenlik önlemleri alması gerekir. Buna güvenlik duvarları, anti virüs yazılımı, iki faktörlü kimlik doğrulama, şifreleme ve düzenli güvenlik güncellemeleri kullanımı da dahildir. Aynı zamanda siber güvenlik farkındalığı eğitimi, çalışanları ve kullanıcıları çevrimiçi ortamda güvende kalmalarına yönelik en iyi uygulamalar konusunda eğitmek için de önemlidir.

Siber güvenlik tehditleri yalnızca işletmeleri ve bireyleri etkilemekle kalmaz, aynı zamanda ulusal güvenlikle ilgili sonuçları da olabilir. ABD Savunma Bakanlığı tarafından hazırlanmış bir raporda, bu tür saldırılara karşı korunmak için geliştirilmiş siber güvenlik önlemlerine duyulan ihtiyaç vurgulanmıştır (U.S. Department of Defense, 2023).

Sonuç olarak, siber güvenlik günümüzün dijital ortamında kritik bir ihtiyaçtır. Artan siber tehditler ve internete bağlanan cihaz sayısının artması, bireylerin ve kuruluşların dijital varlıklarını korumak için gerekli önlemleri almasını zorunlu kılmaktadır. Etkili siber güvenlik önlemleri alarak siber saldırı risklerini azaltabilir ve hassas bilgilerimizin yanlış ellere geçmesini engelleyebiliriz.

2.4. Saldırılar ve Saldırı Tipleri

Saldırı tespit sistemleri Anderson tarafından 1980'de ortaya atıldığında, siber saldırı; bilgiye erişmek, bilgiyi değiştirmek veya bir sistemi güvenilmez veya kullanılamaz hale getirmek şeklinde tanımlanmıştır (Anderson, 1980). Siber saldırılar amaçlarına göre ikiye ayrılmaktadır. Birinci tür aktif saldırı olarak da bilinir; uygulanan saldırıda, sistem kaynaklarının değiştirilmesi çalışılır. İkinci tür pasif saldırılardır. Sistem kaynakları etkilenmeyecek şekilde sistemden bilgilerin öğrenilip kullanılması için saldırılar yapılır.

Siber saldırılar, saldırıların yapıldığı yere göre de ikiye ayrılmaktadır. Güvenlik çevresi içinden yapılan saldırılar iç saldırılardır. İçeriden gelebilecek tehdidi bilebilmek gerçekten çok zordur çünkü içerideki kişinin imkân ve kabiliyetleri siber güvenlik uzmanlarından daha iyi olabilmektedir. Bu tehditlerle başa çıkmanın en önemli yollarından biri karakter analizi yaparak daha önce meydana gelen olaylarda bireysel ve ortak özellikleri bulabilmektir. Sistem dışında kalan bir kullanıcı tarafından yapılan saldırılar ise dış saldırılardır.

Yaygın siber saldırı yöntemlerinden Hizmeti Reddi (DoS / DDoS) Saldırıları, Oltalama Saldırıları, SQL Enjeksiyonu Saldırıları, IP Sahteciliği, Ortadaki Adam Saldırıları, ARP Zehirlenmesi, Bot Saldırıları, Kaba Kuvvet Saldırıları, Sıfır-gün Saldırıları ve Zararlı yazılımlar alt bölümlerde açıklanmıştır. Ayrıca tezin uygulama kısmında kullanılan veri setinin içerdiği saldırılar da başlıklar altında ayrıca açıklanmıştır.

2.4.1. Hizmeti Reddi (DoS / DDoS) Saldırıları

Hizmet Reddi (DoS) ve Dağıtılmış Hizmet Reddi (DDoS) saldırıları, kuruluşların günümüzde karşı karşıya kaldığı en yaygın ve tehlikeli siber tehditlerden bazılarıdır. Bu saldırılar, bir şirketin itibarına önemli ölçüde zarar verebilir, mali kayıplara yol açabilir ve hatta kritik hizmetlerin kapanmasına neden olabilir.

DoS Saldırıları: Bir sunucuya veya ağ kaynağına, erişimi bozmayı veya engellemeyi amaçlayan bir siber saldı çeşididir. Sunucuyu veya ağ kaynağını meşru kullanıcılar için kullanılamaz hale getirir. DoS saldırıları yazılım veya işletim sistemlerindeki güvenlik açıklarından yararlanarak, CPU veya bellek gibi kaynakları tüketerek veya hedef ağı trafikle doldurmak da dahil olmak üzere çeşitli şekillerde gerçekleştirilebilir.

DoS Saldırı çeşitlerinden olan Slow Read DoS (Slow Http) saldırısı, saldırganların web sunucusuna düşük bant genişliğine sahip ancak uzun süreli HTTP istekleri göndererek gerçekleştirdiği bir türdür. Bu istekler, sunucunun kaynaklarını yavaşça tüketir, böylece sunucunun diğer meşru istemcilere hizmet verme yeteneğini ciddi şekilde azaltır veya tamamen engeller. HULK Aracı, web sunucusuna yönelik bir tür DoS saldırısı için kullanılan bir araçtır. HTTP GET veya POST isteklerini kullanarak sunucunun kaynaklarını tüketir ve bu sayede sunucunun normal işlevselliğini etkiler.

Sunucunun aşırı yüklenmesi sonucunda çökme veya hizmet veremez hale gelme riski bulunmaktadır.

Slowloris saldırısı, saldırganların web sunucusuna bağlantı açarak HTTP başlıklarını yavaşça göndererek gerçekleştirdiği bir türdür. Bu, sunucunun bağlantıyı açık tutmasını ve diğer meşru istemcilere hizmet vermeye devam etmesini engeller.

Goldeneye saldırısı, saldırganların web sunucusuna çok sayıda TCP bağlantısı açarak her bağlantı üzerinden çok sayıda HTTP GET veya POST isteği göndererek gerçekleştirdiği bir türdür. Bu, sunucunun aşırı yüklenmesine ve normal kullanıcıların isteklerine yanıt verme yeteneğinin azalmasına neden olur.

DDoS Saldırıları: Hedefi trafikle doldurmak için birden fazla kaynak kullanan ve engellemeyi veya filtrelemeyi zorlaştıran bir DoS saldırısı türüdür. Saldırganlar, saldırıyı başlatmak için Botnet olarak da bilinen virüslü makinelerden oluşan bir ağ kullanır. Botnet, kötü amaçlı yazılım tarafından tehlikeye atılan ve saldırganın bunları uzaktan kontrol etmesine olanak tanıyan cihazlardan oluşur. DDoS saldırıları, saldırıyı başlatmak için birden çok kaynak kullandıklarından DoS saldırılarına göre daha güçlü ve hafifletilmesi daha zordur (Kumar & Somani, 2022).

HOIC, dağıtılmış hizmet reddi saldırılarını gerçekleştirmek için kullanılan açık kaynaklı bir araçtır. Saldırganlar, HOIC aracılığıyla çok sayıda TCP veya UDP isteği göndererek hedeflenen sunucuları aşırı yüklemeye çalışır. UDP Flood saldırıları, düşük gecikmeli, kayıp toleranslı bağlantılar için kullanılan UDP protokolünü kullanarak hedef sistemlere çok sayıda UDP paketi göndererek sistem kaynaklarını tüketir. HTTP Flood saldırıları ise genellikle sunucuları hedef alır ve çoğu web sunucusunun çok sayıda eş zamanlı bağlantıyı kabul etme ve işleme kapasitesine sahip olması nedeniyle etkilidir. Bu saldırılar, sunucuların kaynaklarını tüketerek web sitelerine erişimi engellemek veya performanslarını önemli ölçüde düşürmek için kullanılabilir.

Kuruluşlar, DoS ve DDoS saldırılarına karşı savunma yapmak için çeşitli korunma teknikleri kullanabilir. Yaklaşımlardan biri, kötü niyetli trafiği algılayabilen ve filtreleyebilen özel bir DoS koruma hizmeti kullanmaktır. Başka bir yaklaşım, saldırı trafiğini tespit etmek ve engellemek için güvenlik duvarları ve saldırı tespit/önleme sistemleri gibi ağ güvenlik cihazlarının uygulanmasıdır (Shrivastava, Sharma, & Rai, 2010).

2.4.2. Oltalama Saldırıları

Oltalama saldırıları, saldırganların kullanıcılardan hassas bilgileri çalmak için kullandıkları en yaygın siber saldırılardan biridir. Bu saldırılar, kullanıcıları oturum açmak için kullandıkları kimlik bilgilerini, kişisel bilgilerini veya finansal ayrıntılarını vermeleri için kandırmak gibi sosyal mühendislik taktiklerine dayanır.

Oltalama saldırıları, e-posta, metin mesajları veya hassas bilgilerini sağlamaları için kullanıcıları kandırmaya yönelik diğer iletişim biçimlerindendir. Saldırganlar genellikle sahte bir giriş sayfası veya güvenilir bir kaynaktan (banka veya e-posta sağlayıcısı gibi) geliyormuş gibi görünen sahte bir e-posta oluşturur. Kullanıcı bilgilerini girdikten sonra, saldırgan bu bilgileri kimlik hırsızlığı veya mali dolandırıcılık gibi çeşitli kötü amaçlar için kullanabilir (Jagatic, Johnson, Jakobsson, & Menczer, 2007).

Kötü amaçlı e-postaları veya mesajları algılayabilen ve engelleyebilen kimlik avı önleme yazılımı kullanmak Phishing saldırılarından korunma yöntemlerinden biridir. Başka bir yaklaşım da kullanıcıları kimlik avının riskleri konusunda eğitmek ve onlara bu saldırıları nasıl belirleyecekleri ve önleyecekleri konusunda yönergeler sağlamaktır (Hadnagy, 2010).

2.4.3. SQL Enjeksiyonu Saldırıları

SQL enjeksiyon saldırıları, hassas bilgilere erişim elde etmek veya veri tabanında kötü amaçlı komutlar yürütmek için bir uygulamanın giriş alanlarına kötü amaçlı SQL ifadeleri eklemeyi içeren bir enjeksiyon saldırısı türüdür. Bu saldırılar, giriş doğrulama veya web uygulamasının kodundaki güvenlik açıklarından yararlanma gibi çeşitli şekillerde gerçekleştirilebilir (Halfond, Viegas, Orso, & others, 2006).

Hata tabanlı saldırılar, uygulamanın SQL kodunu yanıt olarak hatalar üretmesini sağlayarak veri tabanıyla ilgili bilgilerin ifşa edilmesine neden olur. Union tabanlı saldırılar ise uygulama kodundaki güvenlik açıklarından yararlanarak saldırganın birden çok veri tabanı tablosundan bilgi almasına izin veren SQL kodunu enjekte eder. SQL enjeksiyon saldırılarına karşı korunmak için çeşitli önlemler alınabilir. Bu önlemler, giriş doğrulamasını kullanarak kullanıcı girişlerini kabul edilebilir sınırlar içinde tutmayı, veri tabanı kullanıcılarının ayrıcalıklarını sınırlayarak saldırganın verebileceği zararı en aza indirmeyi ve yazılım güncellemeleriyle bilinen güvenlik açıklarını gidermeyi içerir (Halfond, Viegas, Orso, & others, 2006).

2.4.4. IP Sahteciliği

IP sahteciliği, saldırganlar tarafından gerçek kimliklerini gizlemek veya başka bir bilgisayar sistemini taklit etmek için kullanılan bir tekniktir. Bu, internet üzerinden gönderilen paketlerin kaynak IP adresini değiştirerek gerçekleştirilir. Saldırgan, IP adresini yanıltarak, paketlerin farklı bir bilgisayar sisteminden geliyormuş gibi görünmesini sağlayabilir (Ali, 2007).

Kuruluşların kendilerini IP sahtekarlığı saldırılarına karşı korumak için alabilecekleri birkaç önlem vardır. Bunlar, sahte paketleri tanımlamak ve engellemek için güvenlik duvarları ve izinsiz giriş tespit sistemlerini kullanmayı, sistemlere yetkisiz erişimi önlemek için katı erişim kontrolleri uygulamayı ve şüpheli etkinlik belirtileri için ağ trafiğini izlemeyi içerir.

2.4.5. Ortadaki Adam Saldırıları

Ortadaki adam saldırısında (MITM), saldırgan iki taraf arasındaki iletişimi keser ve bilgileri ileri geri iletir. Saldırgan, iletişim akışındaki yeni verileri okuyabilir, değiştirebilir veya enjekte edebilir. Bu tür saldırılar özellikle tehlikelidir çünkü tespit edilmesi zor olabilir ve saldırgan hassas bilgilere tespit edilmeden erişebilir (Taormina, ve diğerleri).

MITM saldırılarının çeşitli türleri vardır. Wi-Fi gizli dinleme, saldırganların bir hileli erişim noktası kurarak cihazlar ile meşru erişim noktası arasındaki kablosuz iletişimi engellediği bir yöntemdir. Bu saldırı türü, iletişimi gizlice dinleyen ve potansiyel olarak hassas bilgileri çalan saldırganlar için bir fırsat yaratır. Öte yandan, DNS sahtekarlığı saldırganların DNS kayıtlarını değiştirerek kullanıcıları sahte bir web sitesine yönlendirdiği bir yöntemdir. Bu, kullanıcıların oturum açma bilgileri gibi hassas verilerini toplamak için kullanılabilir.

MITM saldırılarına karşı korunma teknikleri mevcuttur. Örneğin, güvenli iletişim protokolleri olan HTTPS gibi yöntemler, iletişimi şifreler ve saldırganların verilere müdahale etmesini zorlaştırır. Ayrıca, kullanıcılar Wi-Fi ağlarına bağlanırken dikkatli olmalı ve hassas işlemler için halka açık Wi-Fi ağlarını kullanmaktan kaçınmalıdır. Bu önlemler hem kuruluşların hem de bireylerin MITM saldırılarına karşı daha iyi korunmasına yardımcı olabilir.

2.4.6. ARP Zehirlenmesi

Bir ARP zehirleme saldırısında, bir saldırgan yerel bir ağa yanlış ARP mesajları gönderir. Bu mesajlar, saldırganın makinesiyle eşleşen sahte bir MAC adresi içerir. Ağdaki diğer makinelerin ARP önbelleği, sahte MAC adresiyle güncellendiğinde, o makineye yönelik tüm trafik, saldırganın makinesine yönlendirilir. Bu, saldırganın iletişim akışına yeni verileri yakalamasına, değiştirmesine veya enjekte etmesine olanak tanır (Govindarajan & Chandrasekaran, 2011).

Alınabilecek önlemler şu şekilde sıralanabilir: İki taraf arasındaki iletişimi şifreleyen ve saldırganların verileri ele geçirmesini ve değiştirmesini zorlaştıran HTTPS gibi güvenli ağ protokollerini uygulamak ;Yalnızca yetkili cihazların ağ üzerinde iletişim kurmasına izin verilmesini sağlayarak ARP zehirlenmesi saldırılarını önlemeye yardımcı olabilecek statik ARP tablolarının uygulanması; Yöneticileri şüpheli ağ etkinliklerini algılayabilen ve uyarabilen ağ saldırı tespit sistemlerini kullanma.

2.4.7. Bot Saldırıları

Bir bot saldırısında, bir saldırgan hedef sistem veya ağa bir bilgisayarda çeşitli kötü amaçlı faaliyetler gerçekleştirmek için botları kullanır. Botlar, diğerlerinin yanı sıra DDoS saldırıları, kaba kuvvet saldırıları, spam gönderme ve kimlik avı gibi çok çeşitli görevleri gerçekleştirmek üzere programlanabilir. Merkezi bir komuta ve kontrol sunucusu tarafından kontrol edilen güvenliği ihlal edilmiş bilgisayarların ağları olan bot ağları, hassas bilgileri ifşa etmeleri için kullanıcıları kandırmak amacıyla kullanılabilir (Feily, Shahrestani, & Ramadass, 2009).

Kötü amaçlı yazılımdan koruma yazılımı yüklemek, kuruluşların ve bireylerin bot saldırılarından kaynaklanabilecek kötü amaçlı yazılımları algılayıp kaldırmasına yardımcı olabilir. Güçlü parolalar kullanmak, botların kullanıcı kimlik bilgilerini tahmin etmesini zorlaştırarak hesapları daha güvenli hale getirebilir. Ağ trafiği filtreleme uygulamak, botlar tarafından oluşturulan trafikleri algılayıp engelleyerek ağın güvenliğini artırabilir. Ağ trafiğini izlemek, bir bot saldırısının belirtisi olabilecek anormal trafik kalıplarını tespit etmeye yardımcı olabilir. Son olarak, hız sınırlamaları kullanmak, bir hedef sisteme veya ağa gönderilebilecek trafik miktarını sınırlayarak DDoS saldırılarını önlemeye yardımcı olabilir. Bu önlemler, çeşitli güvenlik tehditlerine

karşı korunma sağlayarak sistem ve ağların güvenliğini artırabilir (Feily, Shahrestani, & Ramadass, 2009).

2.4.8. Kaba Kuvvet Saldırıları

Bir kaba kuvvet saldırısında, saldırgan bir parolayı veya şifreleme anahtarını kırmak için olası her karakter kombinasyonunu oluşturmak üzere otomatikleştirilmiş yazılım kullanır. Bu zaman alıcı bir süreç olabilir, ancak parola zayıfsa veya saldırganın güçlü bir bilgisayar veya botnet'e erişimi varsa etkili olabilir (Najafabadi, Khoshgoftaar, Kemp, Seliya, & Zuech, 2014).

Kaba Kuvvet Saldırıları çeşitli protokolleri ve uygulamaları hedef alabilir. Örneğin, SSH protokolü, saldırganların oturum açma bilgilerini tahmin etmek için kaba kuvvet saldırılarına maruz kalabilir. Bu tür saldırılarda, yaygın kullanılan kullanıcı adı ve şifre kombinasyonları, otomatik araçlar veya yazılımlar kullanılarak denenebilir. Benzer şekilde, FTP sunucuları da benzer bir risk altındadır. Saldırganlar, otomatize edilmiş yazılımlar aracılığıyla büyük bir kullanıcı adı ve şifre listesini FTP sunucusuna uygularlar.

Web tabanlı Kaba Kuvvet Saldırıları da oldukça yaygındır. Örneğin, Brute Web saldırıları, otomatik araçlar veya yazılımlar kullanılarak bir web sitesine veya web uygulamasına oturum açmak için şifreleri kaba kuvvetle deneme işlemidir. Bu saldırı türü, yetkili bir kullanıcı olmak veya özel bilgilere erişmek amacıyla gerçekleştirilebilir. Bir diğer tür olan Brute XSS saldırıları ise, saldırganların web uygulamalarındaki güvenlik açıklarını keşfetmek için geniş bir giriş yelpazesini otomatik olarak test etmelerini sağlar. Bu saldırılar, form alanları veya metin kutuları gibi giriş noktalarını hedef alarak gerçekleştirilir ve XSS saldırıları için uygun bir zayıf nokta bulunmaya çalışılır.

Kaba kuvvet saldırılarına karşı korunma teknikleri oldukça önemlidir ve kuruluşlar ile bireyler için çeşitli önlemler mevcuttur. Güçlü parolalar kullanmak, saldırganların kaba kuvvet yöntemleriyle parolayı tahmin etmesini zorlaştırır. Hesap kilitleme politikalarının uygulanması, oturum açma girişimlerinin sayısını sınırlar ve birden çok kez yanlış parola giren kullanıcıları kilitleyebilir. Çok faktörlü kimlik doğrulama kullanmak, sisteme veya ağa erişimi zorlaştıran ek bir güvenlik katmanı sağlar. Hassas verileri şifrelemek ise, saldırganların parolayı kırabilseler bile verileri elde

etmelerini zorlaştırır. Bu önlemler, kaba kuvvet saldırılarına karşı etkili bir koruma sağlar ve güvenlik açıklarının minimize edilmesine yardımcı olur.

2.4.9. Sıfır-gün Saldırıları

Sıfır gün saldırısı, yazılım geliştiricisi tarafından keşfedilmeden ve yamalanmadan önce yazılımdaki bir güvenlik açığından yararlanan bir siber saldırı türüdür. Bu güvenlik açığı, bir bilgisayarda veya mobil cihazda çalışan işletim sistemlerinde, uygulamalarda veya diğer yazılımlarda bulunabilir. "Sıfır gün" terimi, yazılım geliştiricinin güvenlik açığını yamalamak için sıfır günü olduğu gerçeğini ifade eder (Bilge & Dumitraş, 2012).

Sıfır gün saldırıları, genellikle güvenlik açığından yararlanmak için kötü amaçlı kod veya kötü amaçlı yazılım kullanımını içerir. Bu kod e-posta, anlık mesajlaşma veya bir web sitesi aracılığıyla gönderilebilir. Kötü amaçlı yazılım yürütüldükten sonra, hassas verilere erişmek, ek olarak kötü amaçlı yazılım yüklemek veya sistemin kontrolünü ele geçirmek için güvenlik açığından yararlanabilir.

Sıfırıncı gün saldırıları, bireyler ve kuruluşlar için ciddi sonuçlar doğurabilir. Hassas verilerin çalınmasına, finansal kayıplara ve itibarın zarar görmesine yol açabilirler. Ek olarak, güvenlik açığına yönelik bilinen bir yama veya çözüm bulunmadığından bu saldırıların tespit edilmesi ve hafifletilmesi zor olabilir.

Korunma teknikleri, bilgisayar sistemlerini sıfır gün saldırılarına karşı güvende tutmanın önemli adımlarını içerir. Yazılım güncel tutma, bilinen güvenlik açıklarının saldırıya uğramadan önce düzeltilmesini sağlayarak sistemi korumada kritik bir rol oynar. Kötü amaçlı yazılımdan korunma yazılımı kullanmak, sıfır gün saldırıları tarafından kullanılabilecek kötü niyetli kodları tespit etme ve kaldırma konusunda etkilidir. Kullanıcı ayrıcalıklarını sınırlama, kötü amaçlı yazılımların hassas verilere erişmesini veya sistem kontrolünü ele geçirmesini önlemeye yardımcı olur. Ağ segmentasyonu uygulamak, virüslü sistemleri ağın geri kalanından izole ederek sıfır gün saldırılarının oluşturabileceği hasarı sınırlamaya yardımcı olur. Bu önlemler, bilgisayar sistemlerinin güvenliğini artırmak ve potansiyel tehditlere karşı daha dirençli hale getirmek için alınmalıdır.

2.4.10. Zararlı Yazılımlar

Kötü amaçlı yazılım, bir bilgisayar sistemine veya ağına zarar vermek için tasarlanmış herhangi bir program veya koddur. Kötü amaçlı yazılımlar, hassas verileri çalmak, sistem işlemlerini bozmak veya bir sistemin kontrolünü ele geçirmek gibi çeşitli amaçlar için kullanılabilir.

Kötü amaçlı yazılım türlerinden biri virüslerdir. Virüsler, kendilerini çoğaltabilen ve diğer programlara ve sistemlere bulaşabilen programlardır. Genellikle e-posta ekleri, virüslü dosyalar veya virüslü web siteleri aracılığıyla yayılırlar. Kullanıcıların sistemindeki güncellemeleri takip etmesi ve güvenlik açıklarını düzeltmesi önemlidir.

Truva atları, bir başka kötü amaçlı yazılım türüdür. Truva atları, meşru yazılım kılığına giren ve kötü amaçlı faaliyetler yürütmek üzere tasarlanmış programlardır. Hassas verileri çalmak için kullanılabilirler, bir sistemin kontrolünü ele geçirir ve gelecekteki saldırılar için bir arka kapı oluşturabilirler. Kullanıcılar, bilinmeyen kaynaklardan yazılım indirmekten kaçınmalı ve güvenilir kaynaklardan yazılım yüklemelidirler.

Fidye yazılımı, kullanıcının dosyalarını şifreleyen ve şifre çözme anahtarı karşılığında fidye talep eden bir kötü amaçlı yazılım türüdür. Bu tür saldırılara karşı korunmak için düzenli yedekleme yapmak ve güvenilir güvenlik yazılımları kullanmak önemlidir.

Reklam yazılımı, bir kullanıcının sisteminde reklam görüntülemek için tasarlanmış yazılımdır. Adware doğası gereği kötü amaçlı olmasa da sistem performansını yavaşlatabilir ve kullanıcı gizliliğini tehlikeye atabilir. Kullanıcılar, güvenilir anti virüs ve anti-malware programları kullanarak sistemlerini korumalı ve şüpheli reklamlardan kaçınmalıdırlar.

2.5. Kayıt Altına Alınan Büyük Çaplı Saldırılar

Bu bölümde, kayıt altına alınan ve geniş çaplı etkisi olan küresel saldırılardan bazıları yer almaktadır.

2.5.1. SolarWinds Siber Saldırısı

SolarWinds saldırısı, dünya çapında devlet kurumlarını, kritik altyapıyı ve özel şirketleri etkileyen, tarihteki en büyük siber saldırılardan biridir. SolarWinds, Orion

yazılımına erişim sağlayan ve SolarWinds müşterilerinin sistemlerine erişmelerine izin veren kötü amaçlı bir kod sokan Rus bilgisayar korsanları tarafından gerçekleştirilmiştir. Aralık 2020'de keşfedilmiştir ve İç Güvenlik Bakanlığı, Hazine Bakanlığı ve Adalet Bakanlığı gibi ABD devlet kurumları dahil en az 18.000 SolarWinds müşterisini etkilemiştir (Alkhadra, Abuzaid, AlShammari, & Mohammad, 2021).

2.5.2. WannaCry Fidye Yazılımı Saldırısı

WannaCry fidye yazılımı saldırısı, 150'den fazla ülkede 200.000'den fazla bilgisayarı etkileyen küresel bir siber saldırıydı. Saldırı 12 Mayıs 2017'de başladı ve Microsoft Windows'taki bir güvenlik açığından yararlanan bir solucan kullanılarak gerçekleştirildi. Saldırganlar, etkilenen sistemlerin kilidinin açılması karşılığında Bitcoin ile ödeme talep etti. Saldırı, ameliyatları iptal etmek ve hastaları başka hastanelere yönlendirmek zorunda kalan İngiltere'deki Ulusal Sağlık Servisi (NHS) dahil olmak üzere birçok kuruluşu etkilemiştir (Mohurle & Patil, 2017).

2.5.3. Marriott International Veri İhlali

Kasım 2018'de Marriott International, 500 milyona kadar müşterinin kişisel bilgilerini tehlikeye atan bir veri ihlaline maruz kaldığını duyurdu. İhlal, Marriott 'un 2016 yılında Starwood Hotels; Resorts Worldwide'ı satın aldığında edindiği rezervasyon veri tabanındaki bir güvenlik açığından kaynaklanmıştır. İhlal, isimleri, adresleri, telefon numaralarını, e-posta adreslerini, pasaport numaralarını ve diğer hassas bilgileri içeriyordu. İhlal, devlet kurumları tarafından soruşturmalara ve etkilenen müşteriler tarafından davalara yol açmıştır (The New York Times, 2018).

2.5.4. Equifax Veri İhlali

Eylül 2017'de ABD'deki en büyük kredi raporlama ajanslarından biri olan Equifax, 143 milyondan fazla kişiyi etkileyen bir veri ihlali yaşadı. İhlale, Equifax'ın web uygulaması çerçevesindeki ve saldırganların Sosyal Güvenlik numaraları, doğum tarihleri ve ehliyet numaraları dahil olmak üzere hassas kişisel ve finansal bilgilere erişmesine izin veren bir güvenlik açığına neden olmuştur. İhlal, Equifax'ın güvenlik uygulamalarına yönelik yaygın eleştirilere yol açmıştır ve etkilenen tüketicilerle 700 milyon dolarlık bir uzlaşmayla sonuçlanmıştır (Reuters, 2019).

2.5.5. Capital One Veri İhlali

Temmuz 2019'da Capital One, ABD ve Kanada'da 100 milyondan fazla müşterinin ve başvuranın kişisel bilgilerini tehlikeye atan bir veri ihlaline maruz kaldığını duyurdu. İhlal, Capital One'ın web uygulaması güvenlik duvarındaki ve saldırganın isimler, adresler, telefon numaraları, e-posta adresleri, kredi puanları ve Sosyal Güvenlik numaraları dahil olmak üzere hassas bilgilere erişmesine izin veren bir güvenlik açığından kaynaklanmıştır (The New York Times, 2019).

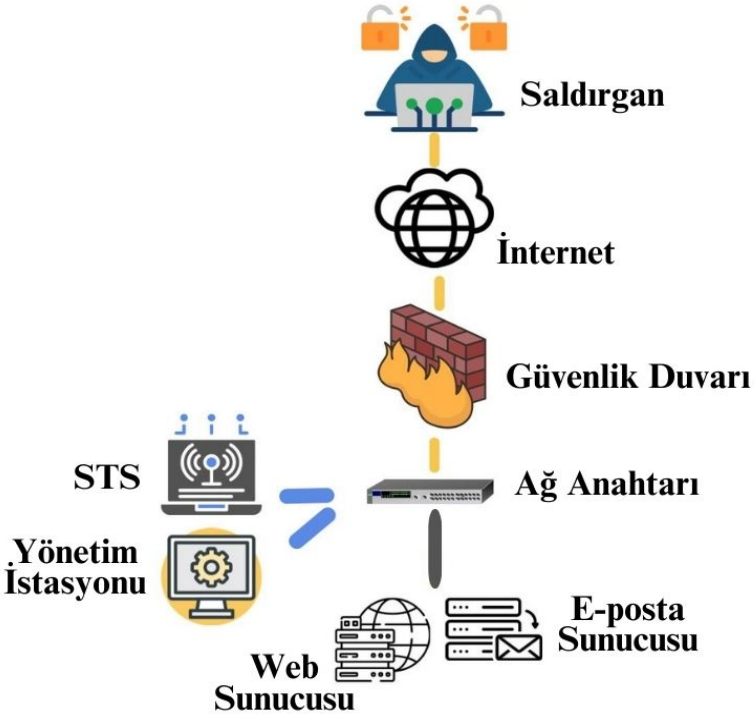
2.5.6. Facebook-Cambridge Analytica Veri İhlali

Mart 2018'de siyasi danışmanlık firması Cambridge Analytica'nın 87 milyona yakın Facebook kullanıcısının kişisel verilerini rızaları olmadan elde ettiği ortaya çıktı. Veriler, 2016 ABD başkanlık seçimlerini ve Brexit referandumunu etkilemek için kullanılmıştır. Saldırı, Facebook'un veri uygulamalarına yönelik yaygın eleştirilere yol açmıştır ve sosyal medya şirketlerinin daha fazla düzenlenmesi çağrısında bulunulmuştur (Hinds, Williams, & Joinson, 2020).

BÖLÜM 3

SALDIRI TESPİT SİSTEMLERİ VE MAKİNE ÖĞRENİMİ

Bilgisayar ağlarında gerçekleştirilen yetkisiz erişimlerin tespiti ve ağ içindeki anormal davranışların belirlenmesi, akan trafiğin izlenmesi ve analiz edilmesiyle mümkündür. Bu durumların tespiti için bilgisayar sistemlerinde STS kullanılmaktadır. STS'ler, bilgisayar sistemlerinde veya ağlarda meydana gelen olayları izlemek ve güvenlik sorunları açısından analiz etmek amacıyla kullanılır. Saldırı tespit sistemlerinin ana işlevi, bilgisayar sistemleri veya ağlardaki saldırıları belirlemek ve bu sistemlere destek sağlamaktır (Ashoor & Gore, 2012).



Şekil 3.1. STS içeren örnek topoloji

Saldırı tespit sistemlerinin görevleri oldukça çeşitlidir. STS içeren örnek bir topoloji Şekil 3.1'de görülmektedir. Bu sistemler, kullanıcıların ve sistem faaliyetlerini sürekli olarak izler ve analiz ederler. Ayrıca, sistem yapılandırmasını ve güvenlik açıklarını değerlendirerek potansiyel riskleri belirlerler. Dosya bütünlüğünü kontrol eder ve sistemde meydana gelen herhangi bir değişikliği izlerler. Saldırıların tipik örüntülerini tanıyarak, potansiyel tehditleri erken aşamada tespit ederler. Anormal olayları izleyerek, sistemin normal işleyişinden sapmaları belirler ve bu durumları analiz ederler. Son olarak, kullanıcıların güvenlik politikası ihlallerini tespit ederek, gerektiğinde önlemler alınmasını sağlarlar. Bu şekilde, saldırı tespit sistemleri, sistem güvenliğini korumak ve olası tehditlere karşı önlem almak için kritik bir rol oynarlar.

1980 yılında James Anderson, "Bilgisayar güvenlik tehditleri izleme ve gözetim" başlıklı makalesinde saldırı tespit sistemi fikrini öne sürmüştür. Anderson, kullanıcı davranışlarını anlamak ve kötüye kullanımı izlemek için denetim izlerinin (log kayıtları) önemli bilgiler barındırabileceğini belirtmiştir. Bu fikir, denetim verilerinin öneminin anlaşılmasını sağlamış ve neredeyse tüm işletim sistemlerinin alt sistemlerinin denetlenmesinde yeni gelişmelere yol açmıştır. Anderson'un hipotezi, gelecekteki saldırı tespit sistemlerinin tasarımı ve geliştirilmesi çalışmalarına temel teşkil etmiştir (Anderson, 1980).

Bununla birlikte STS'nin çalışma döngüsü Bace ve Mell'e (2001) göre; bilgilerin toplanması, analizlerin yapılması ve buna karşılık bir cevabın oluşturulması olacak şekilde 3 aşamadır (Bace & Mell, 2001).

Denning, bilgisayar sistemlerindeki saldırıların tespit edilmesinin gerekliliğini şu şekilde sıralar; Pek çok mevcut sistemde güvenlik açıkları bulunmaktadır ve bu açıkların tespit edilip giderilmesi, teknik veya ekonomik sebeplerden ötürü mümkün olmamaktadır. Güvenlik zaafları bilinen sistemler, daha güvenli alternatifleriyle değiştirilememektedir. Bunun başlıca nedeni ekonomik sebepler veya mevcut sistemlerdeki bazı özelliklerin alternatif sistemlerde bulunmamasıdır. Tam anlamıyla güvenli bir sistem geliştirmek zor, hatta neredeyse imkansızdır. Bu yüzden mevcut sistemlerin korunması daha mantıklıdır. En üst düzey güvenlik önlemlerine sahip sistemler bile, kullanıcıların yetkilerini kötüye kullanması durumunda risk altında olabilir (Denning, 1987).

STS'nin kullanımının faydalarını üç ana başlıkta inceleyebiliriz (Dayıoğlu & Özgit, 2001). Birincisi, STS'ler herhangi bir ihlali sistem yöneticilerinden daha önce fark edebilir ve bu sayede sistem yöneticisinin önceden bilgilendirilip olaya müdahale etmesi sağlanır. İkincisi, sistem yöneticileri, geçmişte veya halen devam eden saldırılarla ilgili detaylı bilgi edinebilir. Bu bilgiler, saldırının kaynağı, büyüklüğü ve hedefler üzerindeki etkileri konusunda kritik öneme sahiptir. Üçüncüsü, sistem tarafından toplanan bilgiler herhangi bir durumda kanıt olarak kullanılabilir.

3.1. Saldırı Tespit Sistemleri Tipleri

Saldırı tespit sistemleri, genellikle üç grupta incelenir; saldırıları algılama yöntemine göre, sistemdeki konumuna göre ve veri işleme zamanına göre. Bu sınıflandırma, Şekil 3.2'de gösterilmektedir.



Şekil 3.2. STS tipleri

3.1.1. Saldırı Algılama Yöntemine göre Saldırı Tespit Sistemleri

Saldırı algılama yöntemine göre STS'ler iki grupta incelenir (Şekil 3.3). Bunlar imza temelli ve anormallik tespiti temelli yaklaşımlardır.



Şekil 3.3. Saldırı algılama yöntemine göre STS

3.1.1.1. İmza Temelli STS

İmza temelli saldırı tespit sistemleri, yetkisiz erişimleri ve bilinen saldırı örüntülerini tespit etmek amacıyla tasarlanmıştır. Bu sistemler, kötüye kullanım temelli STS veya bilgi temelli STS olarak da bilinir. Bilinen saldırıların imzalarını kaydeder ve yeni gelen istekleri bu kayıtlarla karşılaştırarak "saldırı" veya "normal davranış" olarak sınıflandırır (Kaya & Oktay, 2015). Bu imzalar, sistemin şimdiye kadar tespit ettiği saldırıların karakteristik özelliklerini içerir. Ağ paketlerinin protokol türü, protokol işaretleri, kaynak veya hedef IP adresleri, port numaraları ve paketin veri kısmı gibi özellikler bu imzaların karakteristik özelliklerindendir.

İmza temelli STS'lerin en büyük dezavantajı, yeni saldırı türlerini algılama yeteneğinden yoksun olmalarıdır. Yeni bir saldırı türüne ait bilgiler imza veri tabanına eklenmedikçe, sistem bu tür saldırıları tespit edemez. Bu nedenle, imza temelli STS'ler tek başına yeterli bir çözüm sunamamaktadır. Bu eksikliği gidermek amacıyla, araştırmacılar anormallik tespiti temelli STS'leri geliştirmişlerdir.

3.1.1.2. Anormallik Tespiti Temelli STS

Anormallik Tespiti yöntemini kullanan bir STS yaklaşımında sistem davranışları incelenerek normal davranış durumlarının dışına çıkılması halinde alarm üretilir. Sistem davranışlarının incelenmesine ve bu davranışların normal veya anormal olarak etiketlenmesine dayanan bir çalışma prensibi bulunmaktadır. Bir ağ yapısı dahilindeki bir kullanıcının çalışma saatleri sabitse ve bu kullanıcı mesai saatleri dışında sisteme giriş yaptıysa bu durum anormal olarak etiketlenerek sistem yöneticisine alarm üretebilir veya yine bir kullanıcı yetkisi olmadığı halde sistem dosyalarına erişmeye veya değişiklik

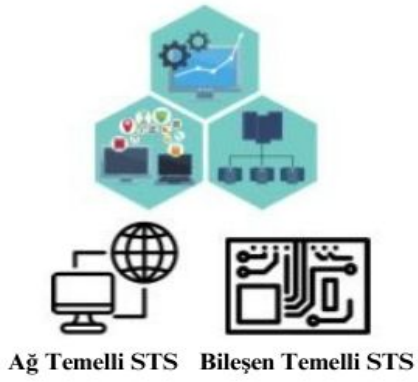
yapmaya çalışıyorsa, bir programa çekirdek erişimi istiyorsa bu durumlar da anormal olarak etiketlenebilir. Bu bakımdan Anormallik Tespiti karşılaşacağı yeni saldırılar karşısında başarı sağlayabilir. Bu başarıyı sağlayabilmek için normal durumların çok iyi planlanması gerekmektedir.

Anormallik Tespiti yaklaşımının zafiyet gösterdiği nokta ise çok fazla yanlış alarm üretmesidir. Günümüz dinamik dünyasında karşılaşılabilecek her durumun önceden belirlenerek normal olarak tanımlanabilmesi mümkün olamamaktadır. Muhakkak ki çok defa ilk defa karşılaşılan veya plansız şekilde gerçekleştirilmesi gereken durumlar olabilmektedir. Anormallik Tespiti yaklaşımı bu nedenle zararlı olmasa dahi birçok durum için anormal etiketlemesi yapmakta ve bu durum çok sayıda yanlış alarm üretilmesine neden olmaktadır. Yukarıda verilen örnek paralelinde düşünülürse mesai saatleri dışındaki kullanıcı aktivitelerini anormal olarak niteleyen ve anormallik tespiti yapan bir STS yaklaşımında zararlı aktivitede bulunmayan bir kullanıcı da anormal yani zararlı olarak nitelendirilebilir.

Bu yöntemin zorluklarından biri, normal sistem özelliklerinin belirlenmesinde yaşanır. Yanlış veya yetersiz modelleme sonucunda, normal işlemler yanlışlıkla saldırı olarak tanımlanabilir. Ancak, bu yöntemin avantajı, yeni saldırı türlerini tespit edebilmesidir (Axelsson, 2000).

3.1.2. Sistemdeki Konumuna Göre Saldırı Tespit Sistemleri

Bilgilerin görüntülenme ve analiz edilme kaynağına göre sınıflandırılması iki gruba ayrılır: ağ temelli ve bileşen temelli (Şekil 3.4). Her iki STS türü de problem formülasyonu ve çözümü açısından kendine özgüdür.



Şekil 3.4. Sistemdeki konumuna göre STS

3.1.2.1. Ağ Temelli STS

Ağ Tabanlı Saldırı Tespitini akıllı ağ donanımı olarak düşünebiliriz. Bu donanımlar ağ trafiğini monitör eder ve bazı işaretlere göre bu trafiği analiz eder. Olay kaynağı olarak ağ üzerinden akan paketler kullanılır. Ağ üzerindeki kritik noktalara yerleştirilmiş ve algılayıcılar ile ağ trafiğini izleyen sistemlerdir. Ağ Temelli STS aygıtının bulunduğu ağın kritik noktalarına konumlandırılması önemlidir (Can & Sahingoz, 2015).

Ağ tabanlı STS'lerin en belirgin avantajları arasında, ağ performansını olumsuz etkilememesi, saldırıları anında algılayarak sistem yöneticisine hızlı bildirimde bulunabilmesi, tanımlanmamış şüpheli saldırıları tespit edebilmesi ve işletim sisteminden bağımsız çalışabilmesi yer alır. Bununla birlikte, ağ tabanlı STS'lerin bazı dezavantajları da vardır. Belirli uygulamalar için karmaşık protokoller gerektirebilir, şifrelenmiş trafiği izlemekte güçlük çekebilir ve ağ trafik izlerinin hacmi sistemin kapasitesini aştığında işlevselliğini yitirebilir.

3.1.2.2. Bileşen (Host) Temelli STS

Bileşen temelli STS, sistemlerin dinamik durumunu ve davranışlarını izlemek için kullanılır. Denetim ajanlarından gelen verileri veya log analiz dosyalarını kullanarak dosya sistemlerindeki değişimleri, kullanıcı erişim kontrollerini, sistem proseslerinin davranışlarını ve diğer kaynakların kullanımını izler ve değerlendirir. Ayrıca, ağa bağlı makinelerden gelen paketlerin kısa ve basit bir denetimini gerçekleştirir.

Bileşen temelli STS'lerin diğer bir özelliği, dosya sistemlerinin hash değerlerini tutmasıdır. Bu sistem, önemli olarak kabul edilen dosyaların hash değerlerini içeren bir veri tabanı oluşturur. Bu veri tabanında kayıtlı olan hash değerleri, sistemde herhangi bir güvenlik ihlali tespit edildiğinde kullanılır. STS, güvenlik ihlali olduğunu algıladığında, etkilenen dosyanın yeni hash değerini hesaplar ve veri tabanında kayıtlı olan eski hash değeri ile karşılaştırır. Eğer iki hash değeri birbirinden farklı ise, dosyada değişiklik yapıldığı sonucuna varır. Hash değeri doğrulaması, zararlı yazılımlara karşı oldukça etkili bir yöntemdir (Pathan, 2014).

Bileşen temelli STS'lerin en önemli avantajları, programlar ve kullanıcılar arasında güçlü bir ilişki kurabilmesi, saldırıların potansiyel hedeflerini doğru bir şekilde belirlemek için saldırı sırasında değerli bilgiler sunabilmesi ve ek donanım

gerektirmemesi dolayısıyla kolay kurulabilmesidir. Ancak, bu sistemlerin bazı dezavantajları da bulunmaktadır. Sistemin ölçeklenebilirliği ve yönetimi (özellikle yapılandırma açısından) oldukça karmaşıktır ve derin veri analizi ile detaylı bilgi toplama işlemleri, host performansını olumsuz etkileyebilir.

3.1.3. Veri İşleme Zamanına Göre Saldırı Tespit Sistemleri

STS'ler için veri işleme zamanı, bir saldırı veya güvenlik ihlalinin tespit edilip sistem yöneticisine bildirilmesi arasındaki süreyi ifade eder. STS'ler, veri işleme zamanına göre aktif (gerçek zamanlı) ve pasif (gerçek zamanlı olmayan) olmak üzere iki kategoriye ayrılır (Şekil 3.5).



Şekil 3.5. Veri işleme zamanına göre STS

3.1.3.1. Gerçek Zamanlı (Aktif) STS

Gerçek zamanlı saldırı tespit sistemlerinde, algılanan saldırıyı tespit ettikten hemen sonra sistem, mevcut saldırı tehdidine karşı aktif bir tepki gösterir. Bu tür STS'ler günümüzde araştırılmakta olup, saldırı engelleme sistemi olarak adlandırılmaktadır. Hemen yanıt verilmesi gereken ticari uygulamalarda tek çözüm olarak kabul edilirler. Ancak, gerçek zamanlı STS'lerin yoğun ağ trafiği olan sistemlerde uygulanması, performans açısından zayıf kalabilir ve maliyetli olabilir.

3.1.3.2. Gerçek Zamanlı Olmayan (Pasif) STS

Gerçek zamanlı olmayan (pasif) STS'ler, herhangi bir saldırı modeli tespit ettiğinde uyarı üretir ve bu uyarıları sistem yöneticisine iletir. Pasif sistemler, depolama ve iletme prensibiyle çalışır. Tanımlanan saldırılar ve muhtemel tehditler kaydedilir ve daha sonra incelenmek üzere saklanır.

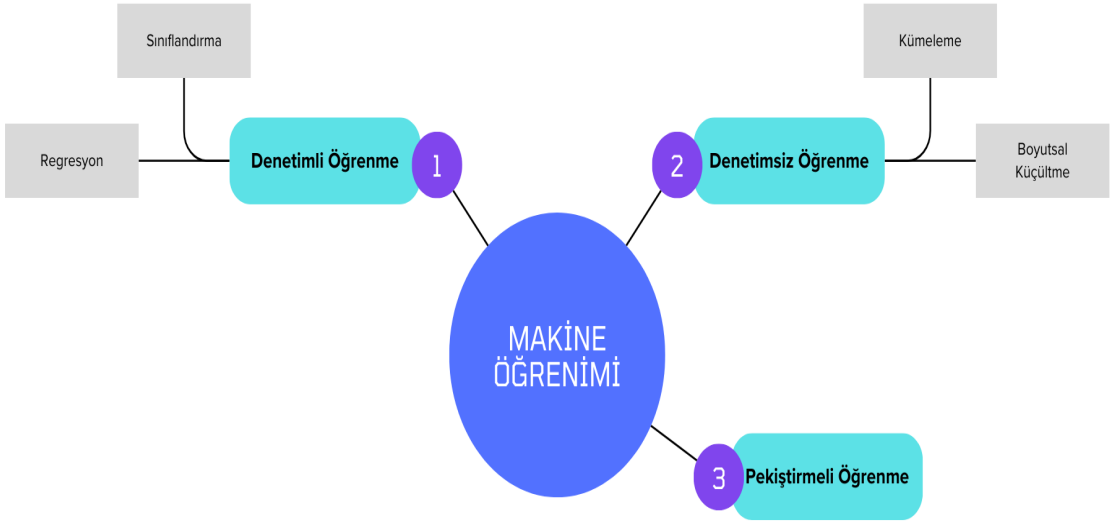
3.2. Makine Öğrenimi

Makine öğrenimi, bir bilgisayarın belirli bir görevi belirli bir performans ölçütü üzerinden optimize etmek için veri kullanarak öğrenme yeteneği kazanması sürecidir. Bu, bir algoritmanın veriyle beslenerek istatistiksel model oluşturmasını ve bu modeli kullanarak gelecekteki verilere dayalı tahminler yapmasını içerir.

Günümüzdeki siber güvenlik odaklı ortamlarda tehlikeli desenleri ve davranışları anlamak için ML, siber güvenlik uzmanlarına trendleri öngörme ve ilgili algoritmaları uygulama konusunda yardımcı olabilir. ML ayrıca gelecekteki saldırıları önlemede ve mevcut olanlara yanıt vermede de önemli bir araç olabilir. Naive Bayes, Karar Ağacı, Rassal Orman ve Gradyan Artırma gibi birçok bilinen ML tekniği, siber güvenlikte kullanılmakta ve güvenlik açıklarını ve tehditleri tespit etmek ve önlemek için kullanılmaktadır. ML yaklaşımları, büyük veri kümelerinde kötü amaçlı yazılım, spam ve sahtekarlık gibi konuları tahmin etmek ve tanımlamak için kullanılabilir (Shahriar, ve diğerleri, 2019). Örnek olarak, bir spam filtresi oluştururken, makine öğrenimi algoritması e-postaları analiz eder, spam ve normal e-postalar arasındaki farklılıkları öğrenir ve sonunda yeni e-postaları spam veya normal olarak sınıflandırır.

Bilgi eksikliğimizi veriyle telafi ediyoruz. Binlerce örnek mesaja kolayca erişim sağlayabiliriz, bazılarının spam olduğunu ve bazılarının olmadığı bilinebilir, bizim istediğimiz şey, spamı neyin oluşturduğunu bu mesajlardan "öğrenmek"'tir. Başka bir deyişle, bilgisayarın (makinenin) bu görev için otomatik olarak algoritmayı çıkarmasını istemekteyiz. Sayıları sıralamayı öğrenmemize gerek yoktur çünkü bunun için zaten algoritmalar vardır, ancak birçok uygulama için henüz bir algoritma olmasa da çok miktarda veriye sahibiz (Alpaydin, 2020).

Makine öğreniminde, birçok metodoloji ve algoritmanın mevcut olmasına rağmen, temelde denetimli, denetimsiz ve pekiştirmeli öğrenme olmak üzere üç ana türe ayrılmaktadır (Şekil 3.6).



Şekil 3.6. Makine öğrenimi temel çeşitleri

3.2.1. Denetimli Öğrenme

Örnek girdi-çıktı çiftlerine dayanarak bir girdiyi bir çıktıya eşleyen bir fonksiyonu öğrenme işlemidir. Etiketli eğitim verilerinden bir fonksiyon türetir; bu veriler, bir eğitim örnekler kümesinden gelir. Denetimli makine öğrenimi algoritmaları dış yardım gerektiren algoritmalardır. Giriş veri kümesi eğitim ve test veri kümelerine ayrılır. Eğitim veri kümesinde tahmin edilmesi veya sınıflandırılması gereken bir çıktı değişkeni bulunur. Tüm algoritmalar, eğitim veri kümesinden bazı desenler öğrenir ve bunları tahmin veya sınıflandırma için test veri kümesine uygular (Mahesh, 2020). Sınıflandırma ve regresyon olmak üzere iki ana türü vardır. Sınıflandırma, girdi verilerini belirli bir sınıfa veya kategoride sınıflandırmayı amaçlar, örneğin spam e-postalarını tespit etme. Regresyon ise, sürekli bir çıktı değişkeni tahmin etmeyi hedefler, örneğin bir evin fiyatını tahmin etme.

3.2.2. Denetimsiz Öğrenme

Etiketlenmemiş veri kümesi üzerinde desenler ve ilişkiler bulmak için kullanılan bir makine öğrenimi yöntemidir. Denetimsiz ML teknikleri, etiketlenmemiş verilerden analitik iç görüler üretmeye yardımcı olarak ham veri kümelerinin analizini kolaylaştırır.

3.2.3. Pekiştirmeli Öğrenme

Bir ajanın bir ortamda belirli bir görevi optimize etmek için deneyimlerini kullanarak öğrenmesini içerir. Ajan, çevresel bir ortamla etkileşimde bulunur, bir eylem gerçekleştirir, bu eylemin sonucunu gözlemler ve ardından bu bilgiyi kullanarak gelecekte daha iyi kararlar alır. Ajan, bir ödül veya ceza mekanizması kullanarak en iyi eylemi seçmeyi öğrenir. Bu yöntem, robotik, oyun teorisi, otomatik ticaret ve diğer birçok alan için kullanılır (Sutton & Barto, 2018).

Ayrıca bir başka kavram olan Derin öğrenme, hiyerarşik mimarilerin kullanıldığı ve denetimsiz veya denetimli özellik seçiminin yapıldığı bir makine öğrenimi sınıfıdır ve bu öğrenilmiş özellikler daha sonra sınıflandırma ve diğer ilgili görevlerde kullanılır (Usama, ve diğerleri, 2019). Büyük ve karmaşık veri kümelerinden anlamlı bilgileri çıkarmak için kullanılan bir makine öğrenimi alt dalıdır. Derin öğrenme, yapay sinir ağlarının çok katmanlı ve karmaşık yapılarına dayanır ve bu ağlar genellikle birbiri ardına gelen katmanlar şeklinde organize edilir. Örnekler arasında yaygın olarak kullanılan denetimsiz öğrenme teknikleri arasında kümeleme algoritmaları (örneğin, K-ortalama), boyut azaltma teknikleri (örneğin, PCA), ve birlikte öğrenme yöntemleri (örneğin, birliktelik kuralı madenciliği) bulunmaktadır. Bu teknikler, veri analitiği, pazar araştırması, biyoinformatik ve diğer birçok alanda yaygın olarak kullanılır.

Çalışmamızda denetimli makine öğrenimi algoritmalarından K-En Yakın Komşu, Karar Ağaçları, Naive Bayes, Rassal Orman, Gradyan Arttırma, AdaBoost, Lojistik Regresyon ve Doğrusal Ayrımcılık Analizi ile derin öğrenme alanında kullanılan yapay sinir ağlarının kullanıldığı algoritmalarından olan Çok Katmanlı Algılayıcı kullanılmıştır. Kullanılan algoritmalar Başlık 5.2 altında açıklanmıştır.

3.3. Akıllı Saldırı Tespit Sistemleri

Teknolojinin yaygınlaşmasıyla birlikte, internet ve diğer ağlara çok sayıda bağlantı noktası olduğu ve çeşitli kanallardan gelen tehditlerin arttığı gözlemlenmiştir. Mevcut sistemlerde, tehditlerin tespit edilip önlenmesi için kullanılan güvenlik duvarlarının ve şifreleme tekniklerinin yetersiz olduğu görülmüş, bu durum farklı tespit ve önleme sistemlerinin kullanılmasına yol açmıştır. Bu ihtiyaçtan hareketle, Saldırı Tespit Sistemleri tasarlanmıştır. Geleneksel olan Saldırı Tespit Sistemleri, teknolojik gelişmelere ayak uyduramamış ve durağan kalmıştır. Bu nedenle, daha dinamik bir yapıya sahip, saldırıları daha hızlı ve doğru bir şekilde tespit edebilen, insan faktörünün

daha az önemli olduğu bir sisteme geçiş önem kazanmıştır. Bu amaçla, Akıllı Saldırı Tespit Sistemlerinin tasarımı kaçınılmaz hale gelmiştir.

Siber saldırılara karşı ilk savunma mekanizmalarından biri olan STS, siber savunma alanında önemli bir rol oynamaktadır. Özellikle büyük şirketler, devletler ve askeri kuruluşlar gibi karmaşık ve büyük siber altyapılara sahip olan kurumlar, bu altyapılarını korumak için STS'lerini etkin bir şekilde kullanmaktadır. Saldırı Tespit Sistemlerinin önemli bir gücü, karmaşık ağ yapılarını koruyabilmesi ve saldırıları tespit edebilmesidir, böylelikle sistem büyük zarar görmeden ve geri dönülemez kayıplar yaşanmadan önce müdahale edilebilir. Bu sistemler, devlet ve askeri ağ sistemleri gibi kritik ve riskli bilgiler içeren verilerin korunmasında kullanılabilir ve gelecek saldırıların özelliklerini belirlemek için analiz yapabilirler.

Siber saldırıların önlenmesinde önemli bir rol oynayan STS, bazen saldırganların ataklarını tamamen engelleyememektedir, çünkü sıklıkla saldırganlar önceden hareket ederler. Bu durumun ana nedeni, geleneksel STS'lerin sınırlı veri ve yazılıma sahip olmaları, hızlı karar alamamaları ve saldırıları anlık ve proaktif olarak tespit etmede zorluk yaşamalarıdır. Bu noktada, akıllı bir algoritma ile geliştirilmiş STS devreye girer ve değişen saldırıları erken aşamada algılayabilecek yapay zekâ teknolojilerini kullanır.

Akıllı STS'lerin Avantajlarını birkaç madde özetlersek;

1.Anlık Tespit Yeteneği: Akıllı STS, makine öğrenimi ve yapay zekâ gibi ileri teknolojileri kullanarak potansiyel tehditleri hemen tespit edebilir. Bu, güvenlik personelinin hızlı bir şekilde müdahale etmesini ve güvenlik ihlallerinin artmasını önleyebilir.

2.Yanlış Alarm Oranlarının Azaltılması: Bu sistemler, makine öğrenimi algoritmalarını kullanarak verileri analiz eder ve olası tehditleri belirler. Bu sayede yanlış pozitiflerin sayısı azalır ve güvenlik personeli yalnızca gerçek tehditlere odaklanabilir.

3. Sürekli İzleme: Akıllı STS'ler, güvenlik personelinin bulunmadığı zamanlarda dahi tesislerin ve ağların sürekli olarak izlenmesini sağlar. Bu, güvenlik ihlallerinin mesai saatleri dışında veya personelin meşgul olduğu zamanlarda dahi önlenebilmesine yardımcı olur.

Sonuç olarak, akıllı saldırı tespit sistemleri, geleneksel sistemlerin kısıtlamalarını aşarak daha etkin bir siber güvenlik çözümü sunmaktadır. Yapay zekâ ve makine öğrenimi gibi ileri teknolojilerin entegrasyonuyla, bu sistemler siber saldırıları daha hızlı ve doğru bir şekilde tespit edebilmekte ve önleyici tedbirler alabilmektedir.

BÖLÜM 4

LİTERATÜR

Literatürde bu tez çalışmasında olduğu gibi farklı makine öğrenmesi teknikleriyle geliştirilmiş birçok saldırı tespit sistemi mevcuttur. Bu çalışmaları daha iyi anlamak adına Başlık 4.1'de 2003-2019 yılları arasında yapılan çalışmalar özetlenmiştir. Başlık 4.2'de ise 2019-2024 yılları arasında yapılan çalışmaların bulguları, karşılaştırmalı analizi kolaylaştırmak adına Çizelge 4.1'de verilmiştir.

4.1. Benzer Çalışmaların Özeti

Sabhnani ve Serpen (2003), KDD 1999 Cup veri seti üzerinde gerçekleştirdikleri çalışmada, belirli saldırı kategorilerinde bazı sınıflandırıcı algoritmalarının daha iyi performans sergilediğini göstermiştir. Önerilen çoklu sınıflandırıcı model, tüm saldırı kategorileri için daha iyi tespit ve daha düşük yanlış alarm oranları sağlamıştır (Sabhnani & Serpen, 2003).

Yang Li ve Li Guo (2007) yaptığı deneylerde, önerilen yeni bir denetimli ağ saldırı tespit yöntemi olan TCM-KNN (K-En Yakın Komşular için Transdüktif Güven Makineleri) ve aktif öğrenmeye dayalı eğitim verileri seçme yönteminin, mevcut teknolojilere göre daha sağlam ve etkili olduğunu göstermiştir. Bu yöntem, izinsiz girişleri daha etkili bir şekilde tespit edebilmiştir (Li & Guo, 2007).

"Zeki Saldırı Tespit Sistemi ve Gerçekleştirilmesi" Adlı çalışmada KDD Cup99 veri kümesini Çok Katmanlı Algılayıcı modeli ile eğitmek suretiyle başarı oranını ve öğrenme hızını incelediler. Bu araştırmanın sonuçlarına göre; 65.536 test verisi için işlem süresi 22 saniye ve her bir veri için sonuç üretme süresi 0,00048 saniye olarak ölçüldü. Başarı oranı ise en düşük %81,93 ve en yüksek %97,92 olarak belirlendi (Sağıroğlu, Yolaçan, & Yavanoğlu, 2011).

Meng'in (2011) çalışmasında, J48, SMO (SVM) ve RBFnetwork (YSA modeli) gibi sınıflandırıcıların genel olarak yüksek bir tespit performansı sergilediği, ancak bu performansın farklı özellik alt kümeleri ve farklı sınıflandırıcılarla değişkenlik gösterdiği belirtilmektedir. Örneğin, U2R için tespit oranlarının (%32,69 ile %67,27) ve Probe için tespit oranlarının (%67,2 ile %99,34) özellik seçimi yöntemlerine göre (farklı alt kümeler kullanılarak) büyük ölçüde değiştiği görülmektedir (Meng, 2011).

Govindarajan ve Chandrasekaran (2011) tarafından yapılan araştırmada, hibrit bir yapı önerilmiştir. Deneysel sonuçlardan, hibrit modelin normal ve anormal saldırı veri setlerini sırasıyla MLP için %98,81 ve %93,93, RBF (YSA modeli) için %94,20 ve %99,02 doğrulukla tespit edebildiği görülmüştür. Sonuçlar, veri madenciliği yönteminin sınıflandırma doğruluğu üzerinde önemli bir etkisi olduğunu göstermektedir (Govindarajan & Chandrasekaran, 2011).

Özgür ve Erdem'in (2012) çalışmalarında, WEKA ortamında KDD Cup99 veri kümesini kullanarak; Saldırı tespiti için Makine Öğreniminin kullanılmasının etkinliğini araştırmışlardır. Karar Ağaçları, Yapay Sinir Ağları, Destek Vektör Makineleri ve AdaBoost algoritmalarının sonuçlarının %90 üzerinde başarı sağladığı belirtilmiştir (Ozgur & Erdem, 2012).

Sindhu vd. (2012) yaptıkları deneyde, STS'nin performansının; eğitim verilerine, makine öğrenme tekniklerine ve sızma tespiti için seçilen özniteliklere bağlı olduğunu belirtmiştir. Önerilen sistemin ortalama sınıflandırma oranı 98.4%'dir, literatürde tartışılan mevcut sistemlere göre daha iyi sonuç elde edilmiştir. Eğitim ve test veri seti, KDD'den toplanan ve yinelenen örnekleri kaldırmak için önceden işlenmiş bir şekilde normal ve anormal örneklerin daha büyük bir numarası toplanarak elde edilmiştir. Sistem kör bir şekilde çalışır ve belirli saldırı türlerini tespit edebilecek şekildedir (Sindhu, Geetha, & Kannan, 2012).

Belavagi ve Muniyal (2016) ve diğerleri tarafından NSL-KDD veri seti üzerinde denenen Lojistik Regresyon, Naive Bayes, Destek Vektör Makinesi ve Rassal Orman gibi denetimli makine öğrenimi algoritmalarının performansı incelenmiştir. Rassal Orman sınıflandırıcısının diğer yöntemleri aştığı ve en yüksek doğruluk oranına sahip olduğu görülmektedir. Çalışmalarında SVM en düşük doğruluk oranına sahipken, Lojistik

Regresyon algoritması Naive Bayes ve SVM'den daha iyi doğruluk oranına sahiptir (Belavagi & Muniyal, 2016).

Aburomman ve Reaz (2016) KDD99 veri setinden beş rastgele alt küme seçmişlerdir. Yeni yaklaşımların yanı sıra ağırlıklı çoğunluk algoritması (WMA) yaklaşımını kullanarak birden fazla öğrenme modelinin veya sınıflandırıcının bir araya getirerek oluşturulan sınıflandırıcılar oluşturulmuştur. Deneysel sonuçlar, yeni yaklaşımın sınıflandırma doğruluğu açısından WMA'yı geride bırakan yeni sistemler üretebileceğini öne sürmektedir. Gelecekteki çalışmalarının amacını, ağırlıkların üretilmesi için farklı optimizasyon algoritmalarını uygulamak ve karşılaştırmak olarak sunmuşlardır (Aburomman & Reaz, 2016).

Al-Yaseen, Othman ve Ahmad Nazri (2017) KDD Cup 1999 veri seti üzerinde yaptıkları deneyler sonucunda, önerilen modelin doğruluk açısından performansı %95,75'e kadar ulaşmış ve yanlış alarm oranı %1,87 olarak belirlenmiştir. Bu sistem KDD Cup 1999 veri setiyle uygulanmıştır. Bu sistem eğitim veri setinde görünmeyen yeni saldırıların tespit oranını %40,02'ye ulaştırarak yeni saldırıların tespitini iyileştirebilmektedir (Al-Yaseen, Othman, & Nazri, 2017).

Belouch, El Hadaj ve Idhammad (2018) UNSW-NB15 veri seti kullanılarak yaptıkları deneyde, ağ trafiğindeki saldırıları tespit etmek amacıyla Naive Bayes, Destek Vektör Makineleri, Karar Ağacı ve Rassal Orman temelli yöntemler kullanılmıştır. Farklı tespit algoritmalarının performansı Apache Spark kullanarak değerlendirilmiştir. Çalışmalarında Rassal Orman sınıflandırıcısı, doğruluk, hassasiyet, özgüllük ve işlem süresi açısından en iyi performansı verirken Karar Ağacı ve Naive Bayes, %74.19 ile en kötü tespit doğruluğunu vermiştir. Gelecekte, tespit ve eğitim süresini azaltmak için kullanılan öznitelik sayısını azaltmak adına öznitelik seçme teknikleri uygulayarak çalışmayı amaçlamışlardır (Belouch, El Hadaj, & Idhammad).

Sharafaldin, Habibi Lashkari ve Ghorbani (2018), gerçek dünya kriterlerini karşılayan ve kamuoyuna açık olan zararsız ve yedi yaygın saldırı ağ akışlarını içeren güvenilir bir veri seti oluşturmuşlardır (CIC-IDS-2017). Bu veri seti, belirli saldırı kategorilerini tespit etmek için en iyi özellik setini belirlemek amacıyla kapsamlı ağ trafiği özellikleri ve makine öğrenimi algoritmaları setinin performansını değerlendirmektedir. Veri setinden 80 öznitelik çıkarılmış ve her saldırı ailesini algılamak

için en iyi kısa öznitelik setini belirlemek için Randam Forest Regressor algoritması kullanılmıştır. Seçilen özniteliklerin performansı ve doğruluğu yedi yaygın makine öğrenimi algoritması ile incelenmiştir (Sharafaldin, Lashkari, Ghorbani, & diğerleri, 2018).

4.2. Daha Önce Yapılan Çalışmaların Bulguları

2019-2024 yılları arasında saldırı veri setlerinde yapay zekâ uygulamaları kullanarak saldırı tespiti yapan çalışmaların özetini içeren Çizelge 4.1, bu alanda yapılan araştırmaların ayrıntılarını kapsamaktadır. Bu çizelgede, çalışmaları gerçekleştiren kişilere “Geliştiren" sütununda yer verilmiştir. Araştırmacıların isimlerinin yanı sıra, her çalışmada kullanılan saldırı veri setleri "Veri seti" sütununda detaylandırılmıştır. Ayrıca, öznitelik seçimi yapılmışsa, hangi tekniklerin kullanıldığı "Öznitelik seçimi" sütununda belirtilmiştir. Çizelgenin "Algoritmalar" kısmında, çalışmalarda kullanılan makine öğrenimi algoritmalarının çeşitleri listelenmiştir. Çalışmaların sonuçlarının değerlendirilmesi için kullanılan ölçütler ise "Ölçütler" sütununda yer almaktadır. Bu sütunda, araştırmacıların sonuçları nasıl ölçtüğü ve hangi metrikleri kullandığı verilmektedir.

Son olarak, "Açıklama" sütunu altında, elde edilen sonuçlar ve araştırmaların bulguları hakkında bilgiler bulunmaktadır. Bu bölüm, çalışmalardan elde edilen verilerin analizi ve sonuçların yorumlanması gibi detayları içermektedir.

Çizelge 4.1, saldırı tespiti konusunda yapay zekâ ve makine öğrenimi tekniklerinin nasıl uygulandığını ve bu uygulamaların sonuçlarını göstermektedir. Her bir çalışma, farklı veri setlerini kullanarak, çeşitli öznitelik seçimi teknikleri ve algoritmalarla saldırı tespiti yapmıştır. Bu çalışmalar, kullanılan veri setleri ve yöntemler açısından çeşitlilik göstermekte olup, saldırı tespiti alanında farklı yaklaşımlar ve tekniklerin etkili bir şekilde değerlendirildiği bir kaynak sunmaktadır. Bu çizelge, araştırmacılara doğru veri seti, öznitelik seçimi ve algoritma kombinasyonlarını belirlemelerinde yardımcı olabilecek nitelikte olup, gelecekteki çalışmalar bir rehber olarak hizmet etmektedir.

Çizelge 4.1. Yapılan çalışmaların bulguları

Geliştiren	Veri seti	Öznitelik seçimi	Algoritmalar	Ölçütler	Açıklama
(Cavusoglu & Kacar, 2019)	NSL-KDD	Genetik Algoritma (7 öznitelikli bir grup), PCA (6,11 ve 21 öznitelikli 3 grup)	NB, J-48, KNN	Doğruluk, Duyarlılık, F Skoru, MCC, ROC ve PRC	PCA 21 grubu ile yapılan KNN algoritması en iyi sonucu vermiştir (Doğruluk: 99,6871).
(Kanimozhi & Jacob, 2019)	CSE-CIC-IDS2018	10 kat çapraz doğrulama	MLP	Doğruluk, Duyarlılık, F Skoru, ROC ve AUC	En iyi F skoru: 0.9991678456370812 olarak bulunmuştur. (Seçilen Parametreler; {'alpha': 1e-05, 'hidden_layer_sizes': (9, 4)})
(Zhou, Cheng, Jiang, & Dai, 2020)	NSL-KDD, AWID, CIC-IDS2017	Bat Algoritması ile Korelasyon	DT, RF, ForestPA, Birleştirilmiş modeller	Doğruluk, Duyarlılık, F-Skoru, ADR, FAR	NSL-KDD veri seti için 10 özellikli alt kümesi ile %99,81 doğruluk, %99,8 DR ve %0,08 FAR ve AWID için yalnızca 8 özellikten oluşan bir alt küme ile %99,52 doğruluk ve %0,15 FAR, CIC-IDS2017 veri seti için 13 özellikli alt kümesinde %99,89 doğruluk ve %99,9 DR ile en yüksek doğruluk elde edilmiştir (Algoritmalar için ortalama oranlar).
(Atay, Odabaş, & Pehlivanoğlu, 2019)	CSE-CIC-IDS2018	Korelasyon Analizi	CNN, RF, LGBM, CNN+RF, LGBM+RF, RF+RF	Doğruluk, Duyarlılık, F Skoru	%98 doğruluk oranı ve 0.86 makro F skoru ile (CNN + RF) hibrit modelinin en iyi saldırı tespiti yaptığı görülmüştür.
(Emhan & Akın, 2019)	NSL-KDD	Filtreleme, Kazanç Oranı, Simetrik Belirsizlik Katsayısı, Ki-Kare, One-R ve Korelasyon	KNN, RF	İşlem süresi, Doğruluk, Pozitif doğru oranı, Pozitif yanlış oranı ve ROC	Bilgi Kazancı, Ki-Kare, One-R yöntemleri ile elde seçilen özniteliklerin RF yöntemi ile sınıflandırılmasının en optimum yöntem olduğu gözlemlenmiştir.

Geliştiren	Veri seti	Öznitelik seçimi	Algoritmalar	Ölçütler	Açıklama
(Lima Filho, Silveira, de Medeiros Brito Junior, Vargas-Solar, & Silveira, 2019)	CIC-DoS, CICIDS2017 ve CSE-CIC-IDS2018 (DoS/DDoS için)	Çapraz Doğrulamalı Yinelemeli Öznitelik Elemesi	RF, LR, ADA, SGD, DT ve YSA	DR, FAR ve Doğruluk	Önerilen sistem, CIC-DoS ve CSE-CIC-IDS2018 veri setleri için %93'ten yüksek DR ve kesinlik elde etmiştir ve FAR %1'den azdır. Sonuçlar, RF'nin diğer algoritmaların üzerinde daha yüksek doğruluk elde ettiğini göstermektedir.
(Ferrag, Maglaras, Moschoyiannis, & Janicke, 2020)	CSE-CICIDS2018, Bot-IoT	-	DNN, RNN, CNN, RBM, DBN, DBM ve Diskriminant Analizi	DR, FAR ve Doğruluk	DA modeli, CSE-CIC-IDS2018 veri setinde %98,18 ve Bot-IoT veri setinde %96,84'lük en yüksek genel tespit oranını vermiştir. NB, YSA, DVM ve RF algoritmaları ile DR oranı ile karşılaştırma yapılıp bu dört ML algoritmasına göre daha iyi sonuçlar elde edilmiştir.
(Liu, Kantarci, & Adams, 2020)	NSL-KDD	Ağ saldırısı veri kümesi oluşturucusu (Rajasinghe, Samarabandu, & Wang, 2018)	RF, DT, Bagging, SVM, NB, ADA, XGBoost	TP, FP, MCC, AUC	XGBoost, %97 doğruluk, %90,5 MCC ve %99,6 AUC performansıyla birinci sırayı almaktadır.
(Manimurugan, ve diğerleri, 2020)	CICIDS2017	Azınlık saldırı sınıflarını benzer davranış ve özelliklere sahip olarak birleştirilmiştir.	DBNIDS, SVMIDS, RNNIDS, SNNIDS, FNNIDS	Doğruluk, kesinlik, duyarlılık, F Skoru	Önerilen yöntem, normal sınıf için %99,37 doğruluk, Botnet sınıfı için %97,93, Kaba Kuvvet saldırısı sınıfı için %97,71, DoS/DDoS sınıfı için %96,67, Sızma saldırısı sınıfı için %96,37, Port tarama sınıfı için %97,71 ve Web saldırısı için %98,37 doğruluk elde etmiştir.
(Cil, Yildiz, & Buldu, 2021)	CIC-DDoS2019	Reflection ve Exploitation olarak iki kategoriye ayrılıp analiz edilmiştir	DNN	Doğruluk, Kesinlik, Duyarlılık, F Skoru	DDoS saldırılarını neredeyse %95 doğrulukla sınıflandırarak ve DDoS saldırılarını neredeyse %100 doğrulukla tespit etmeyi başarmıştır.

Geliştiren	Veri seti	Öznitelik seçimi	Algoritmalar	Ölçütler	Açıklama
(Injadat, Moubayed, Nassif, & Shami, 2020)	CICIDS 2017, UNSW-NB15	IGBFS ve Korelasyon	KNN, RF	Doğruluk, Kesinlik, Duyarlılık, F Skoru, FAR	Önerilen model, gereken eğitim örneği boyutunu (%74'e kadar) ve özellik seti boyutunu (%50'ye kadar) önemli ölçüde azaltır. Dahası, hiper-parametre optimizasyonu ile model performansı geliştirilir ve her iki veri seti için de %99'un üzerinde tespit doğruluğu elde etmiştir.
(Islam, ve diğerleri, 2021)	IOT DEV NET, NSL-KDD, DS2OS, IOT BOTNET	-	DT, RF, SVM, DNN, DBN, LSTM, Stacked LSTM, Bi-LSTM	Doğruluk, F Skoru, Kesinlik, Duyarlılık	ML modelleri (DT, RF ve SVM) için, SVM NSL-KDD, IoTDevNet ve DS2OS'a karşılık gelen en yüksek eğitim ve test doğruluğunu sağlar, sırasıyla; (99.81%, 99.81%), (99.82%, 99.84%) ve (99.42%, 99.44%). Diğer iki veri seti, IoTID20 ve IoT Botnet için, DT en yüksek eğitim ve test doğruluğunu elde eder, sırasıyla; (100%, 100%) ve (99.71%, 99.71%). Bi-LSTM, NSL-KDD, IoTDevNet, DS2OS, IoTID20 ve IoT Botnet veri setlerine karşılık gelen eğitim ve test doğruluğu açısından diğer dört DL modelini (DNN, DBN, LSTM ve Stacked LSTM) geride bırakır, sırasıyla; (99.27%, 99.27%), (99.97%, 99.97%), (99.43%, 99.39%), (99.996%, 99.99%) ve (99.998%, 99.991%).
(Pai, Adesh, & diğerleri, 2021)	NSL-KDD	Öznitelik oranı	RF, J48, LR, DT, SVM, NB	Doğruluk, Kesinlik, Duyarlılık, F Skoru, ROC	En iyi sınıflandırıcı DoS saldırısı için %99.997 doğrulukla Rassal Orman sınıflandırıcısı iken, benzer şekilde U2R saldırısı için %99.947 doğrulukla J48 en iyi sınıflandırıcıdır.
(Liu, Yang, Lian, & Li, 2021)	UNSW-NB15	Pearson Korelasyonu, LGBM	PSO-LGBM, LR, NB, KNN, ADA, SVM	Doğruluk, DR, FAR	Geleneksel makine öğrenimi algoritmaları ile STS, Normal ve Generic gibi yüksek frekanslı veriler için yüksek tespit oranları sağlar, ancak Backdoor, Shellcode ve Worms gibi verileri neredeyse algılamak imkansızdır. PSO-LGBM, yüksek frekanslı verilerin tespit yeteneğini azaltmazken, Backdoor, Shellcode ve Worms'un algılama oranını sırasıyla %51.28, %64.47 ve %77.78'e artırır.

Geliştiren	Veri seti	Öznitelik seçimi	Algoritmalar	Ölçütler	Açıklama
(Saba, Sadad, Rehman, Mehmood, & Javaid, 2021)	NSL-KDD	Genetik Algoritma	SVM, DT	Doğruluk	10-kat çapraz doğrulama tekniği kullanılarak bir topluluk sınıflandırıcı ile NSL veri setinde doğruluk açısından daha yüksek bir sınıflandırma sonucu gözlemlenmiştir (99.8%).
(Ahmad, ve diğerleri, 2021)	UNSW-NB15	Flow&MQTT (En Az Mesaj Kuyruk Telemetri Taşımacılığı), TCP	RF, SVM, YSA	Doğruluk, Karmaşıklık Matrisi	RF kullanarak, sırasıyla ikili ve çoklu sınıflandırmada %98.67 ve %97.37 doğruluk elde edilmiştir. Küme tabanlı tekniklerde, RF kullanarak Flow ve MQTT özniteliklerinde %96.96, TCP özniteliklerinde %91.4 ve her iki kümeden en iyi özelliklerde %97.54 sınıflandırma doğruluğu sağlanmıştır.
(Yu, Nguyen, & Park, 2022)	NSL-KDD 99, UNSW-NB15, Kyoto 2006, CIDDS	RF ile sınıflandırma, RNN	LR, DT, KNN, RF, MLP, NB, GB	Doğruluk, ROC	RF, veri sınıflandırması için yüksek performansı nedeniyle gerçek zamanlı tespit sağlamamızı sağlar, RNN ise doğal sekanssal veri ayrıştırmasını kullanarak yüksek doğruluk sağlar. Önerilen sistem, verilen sistem parametreleri ile üretilen modeli adaptif ve akıllıca ayarlar. Bu parametreler, saldırganın gerçek zamanlı olarak karşılaştırma tekniklerine karşı savunma amacıyla güvenlik parametreleri olarak kullanılabilir.
(Garg, Kumar, & Payyavula, 2022)	CIC-DDoS2019	PCA ve ANOVA	XGB, LGBM, CDF	Doğruluk	Boosting yöntemleri ile elde edilen doğruluk, CDF ile elde edilen doğruluğun en az %16 daha yüksektir. Boosting algoritmaları, CDF'den en az 240 kat daha hızlı olduğu sonucuna ulaşılmıştır.
(Khan, ve diğerleri, 2022)	HVAC system	İleriye doğru öznitelik seçimi tekniği	DT, RF, GB, ADA, LGBM, XGBoosst, CatBoost	AUC, Doğruluk, Kesinlik, Duyarlılık, F Skoru	Sonuçlar, XGBoost sınıflandırıcısının en iyi sonucu ürettiğini gösterdi. Bu sonuca göre, 0.9999 AUC, 0.9998 doğruluk, 0.9996 duyarlılık, 1.000 kesinlik ve 0.9998 F Skoru elde edilmiştir.

Geliştiren	Veri seti	Öznitelik seçimi	Algoritmalar	Ölçütler	Açıklama
(Ma, ve diğerleri, 2022)	NSL-KDD, UNSW-NB15	Spierman Korelasyonu (XGBoost ile)	BiLTSM	Doğruluk, Duyarlılık, Kesinlik, F Skoru	NSL-KDD ve UNSW-NB15 veri setlerinde, önerilen modelin doğruluk oranı sırasıyla %99.57 ve %85.59'a ulaşmıştır, bu da aynı tür CWGAN ve derin sinir ağı (CWGAN-DNN) modeline göre sırasıyla %1.44 ve %2.98 daha yüksektir.
(Saied Essa & Kamal Guirguis, 2023)	N-BaIoT	Veri kümesinin bir alt kümesi, dengeli bir ikili etiketli veri kümesi oluşturmak için seçilmiştir.	DT, RF, BMC, ADB, GDB ve XGBoost	Doğruluk, Duyarlılık, Kesinlik, F Skoru	Deneysel değerlendirme sonuçları, IoT ortamlarında ağ ihlallerini tespit etme konusunda ağaç tabanlı ML algoritmalarının önemli potansiyeline işaret etmiştir. RF algoritması, %99.9982 doğruluk oranıyla en iyi performansı elde etmiştir.
(Awotunde, ve diğerleri, 2023)	TON_IoT	Ki-Kare	XGBoost, Bagging, ET, RF, ADA	Doğruluk, Duyarlılık, Kesinlik, F Skoru ve Karmaşıklık Matrisi	Sonuçlar, IIoT saldırılarını tespit etme ve sınıflandırmada diğer modellere göre XGBoost topluluğunun en yüksek doğruluğu gösterdiğini işaret etmektedir.
(Türk, 2023)	UNSW-NB15, NSL-KDD	Pearson Korelasyonu	LR, KNN, RF, DT, MLP-LTSM	Doğruluk, Duyarlılık, Kesinlik, F Skoru	UNSW-NB15 veri seti için ikili sınıflandırma ve çoklu sınıflandırma için sırasıyla %98.6 ve %98.3 doğruluk elde edilmiş olup, NSL-KDD veri setinde ise sırasıyla %97.8 ve %93.4 doğruluk elde edilmiştir.
(Henry, ve diğerleri, 2023)	CICIDS-2017	Pearson Korelasyonu	CNN-GRU	Duyarlılık, Kesinlik, FP, TP	Düşük bir FP (yanlış pozitif) oranı ve %98,73 doğruluk oranı elde edilmiştir. Diğer tüm sınıflandırıcılarla karşılaştırıldığında, kullanılan teknik neredeyse toplam özniteliklerin yarısını (her alt veri seti için toplamın %58'inden az) kullanmaktadır.

Geliştiren	Veri seti	Öznitelik seçimi	Algoritmalar	Ölçütler	Açıklama
(Elnakib, Shaaban, Mahmoud, & Emara, 2023)	CICIDS2017	-	MLP, CNN, LTSM, CNN+LTSM, EIDM	Doğruluk, Duyarlılık, Kesinlik, F Skoru	EIDM, diğer dört modeli geride bırakarak %99.48 doğruluk oranıyla en iyi performansı sergilemiştir.
(Thockchom, Singh, & Nandi, 2023)	KDD Cup 1999, UNSW-NB15 ve CIC-IDS2017.	Ki-Kare	NB, LR, DT ve SGD adı verilen bir meta-sınıflandırıcı kullanıldı.	Doğruluk, Duyarlılık, Kesinlik, F Skoru	Deneysel sonuçlar, topluluk sınıflandırıcısının kullanılmasının son derece faydalı olabileceğini kesin olarak kanıtlamaktadır.
(Mohy-Eddine, Guezzaz, Benkirane, & Azrour, 2023)	Bot-IoT	PCA, K-Best, Genetik Algoritma	KNN	Doğruluk, DR, FAR ve hesaplama süresi	Önerilen model, seçilen on öznitelik için %99.99 doğruluk oranı sağlamıştır.
(Umara, Chenb, Shuaiba, & Liuc, 2023)	NSL-KDD, UNSW-NB15	Wrapper yöntemi (DT temelli)	SVM, KNN, RF, NB, YSA	Doğruluk, Tespit oranı, Yanlış alarm oranı, Hesaplama süresi	RF, hem NSL-KDD hem de UNSW-NB15 veri setlerinde en iyi performansı göstermiştir. Sırasıyla %99.87 ve %98.5 tahmin doğruluklarına ve %99.79 ve %99.17 tespit oranlarına sahiptir.
(Talukder, ve diğerleri, 2024)	UNSW-NB15, CIC-IDS2017, CIC-IDS2018	Çapraz Doğrulamalı Yinelemeli Öznitelik Elemesi ve PCA	DT, RF, ET, XGBoost	Kesinlik, Duyarlılık, Karmaşıklık matrisi, Doğruluk, F Skoru, ROC	RF ve ET'nin sırasıyla UNSW-NB15 veri setinde %99.59 ve %99.95 doğruluk oranları elde ettiği gösterilmektedir. Ayrıca, model DT, RF ve ET'nin CIC-IDS2017 veri setinde %99.99 doğruluk oranına ulaştığını ve CICIDS2018 veri seti için DT ve RF modelleri kullanılarak %99.94 doğruluk oranı elde ettiği gösterilmektedir.

BÖLÜM 5

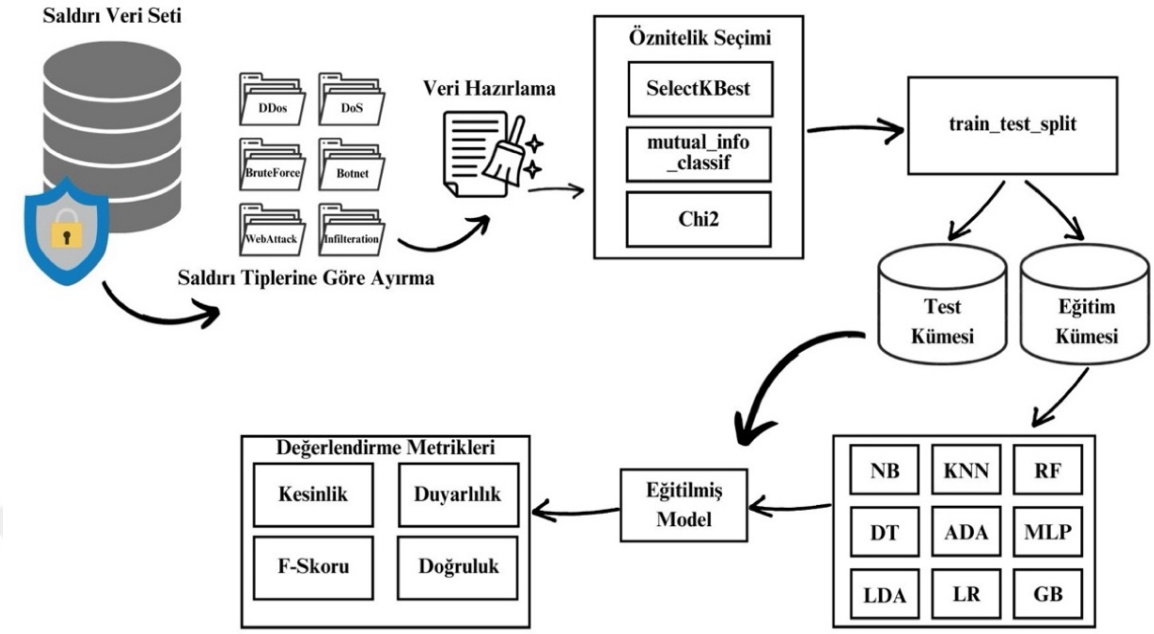
MATERYAL VE YÖNTEM

5.1. Önerilen Sistem

Bu tez çalışmasında, ağlarda gerçek zamanlı olarak kötü amaçlı faaliyetleri tespit edebilen makine öğrenimi tabanlı bir STS için kapsamlı bir analiz çalışması yapılmıştır. Saldırı tespiti araştırmalarında, sistemin saldırı altındayken davranışını incelemek için bir veri seti kullanılır. Bu araştırmada, veri seti olarak CSE-CIC-IDS-2018 veri seti kullanılmıştır.

Çalışma gerçekleştirilirken öncelikle her saldırı türüne ait veriler farklı dosyalara yazdırılmıştır. Veri önişleme aşaması her bir dosyaya ayrı ayrı uygulanmıştır. Ardından 3 farklı öznitelik seçme yönteminden toplanan bilgiler analiz edilerek her bir dosya için öznitelikler seçilmiştir. Dosyaların sağladığı verilerin özelliklerine göre gerekli görüldüğünde sentetik veri arttırma (SMOTE) ve rastgele alt örnekleme (Random Under Sample) yöntemleri kullanılmıştır. Bu tezde yapılan test ve analizler için Python programlama dili, makine öğrenimi ile veri işleme araçlarından Pandas, Mathplotlib, Numpy, Seaborn, Scikit-learn ve Imblearn kütüphaneleri kullanılmıştır.

STS'lerdeki Makine öğrenmesi tekniklerinin performans analizi NB, KNN, RF, DT, Adaboost, MLP, LDA, GB ve LR ile gerçekleştirilen testlerde sınıflandırma başarısı doğruluk, duyarlılık, kesinlik, F skoru ve karmaşıklık matrisi kullanılarak hesaplanmıştır. Akış şeması Şekil 5.1' de verilmiştir.



Şekil 5.1. Önerilen sistem akış şeması

5.2. Kullanılan Algoritmalar

Bu başlık altında, çalışmada kullanılan algoritmaların kapsamlı bir açıklaması sunulmuştur. Bu algoritmalar, çalışmanın temelini oluşturmakta ve verilen problemleri çözmek için benzersiz yeteneklere sahiptirler. Algoritmaların doğru bir şekilde anlaşılması, sonuçların anlaşılmasını ve örüntülerin değerlendirilmesini artırarak daha iyi yorumlar yapılmasını sağlar.

5.2.1. Naive Bayes

Naive Bayes sınıflandırıcısı, İngiliz matematikçi Thomas Bayes ‘ten (yak. 1701-7 Nisan 1761) adını almıştır. Naive Bayes algoritması, Bayes teoreminin bir uygulamasıdır bu da önceki bir olayın veya sonucun meydana gelmesine dayanarak bir olayın veya sonucun gerçekleşme olasılığını tanımlayan koşullu olasılığa dayanır. Bayes teoremi, bir olayın gerçekleşme olasılığını, olayın gerçekleştiği zamanda mevcut olan koşullu olasılıklarla ilişkilendirir. Naive Bayes algoritması, Bayes teoremini kullanarak bir belgenin verilen bir sınıfa ait olma olasılığını hesaplar. Temel varsayım, her özelliğin sınıflandırmaya bağımsız bir katkıda bulunduğudur, bu nedenle "naif" olarak adlandırılır (Barber, 2012).

Naive Bayes algoritması, bir sınıflandırma problemi için olasılıksal bir model oluşturur. Temelde, veri setindeki özelliklerin bir sınıfa ait olma olasılığını hesaplamak için Bayes teoremi kullanılır. Bayes teoremi Denklem (5.1)'de görüldüğü gibi bir olayın gerçekleşme olasılığını koşullu olasılıklar aracılığıyla hesaplar. Veri seti içinde n özellik (veya değişken) $X = (x_1, x_2, \dots, x_n)$ ve bir hedef değişken C olduğu varsayılır. Hedef değişken C farklı sınıflara ait olabilir.

$$P(C|X) = \frac{P(X|C).P(C)}{P(X)} \tag{5.1}$$

Burada:

$P(C|X)$: Verilen özellikler X altında sınıf C'ye ait olma olasılığı

$P(X|C)$: Sınıf C'ye ait olduğu varsayılan bir örneğin özelliklerinin gözlemlenmesi olasılığı

$P(C)$: Sınıf C'nin önceki olasılığı

$P(X)$: Verilen özelliklerin gözlemlenme olasılığı

Naive Bayes algoritması, özellikler arasındaki bağımsızlık varsayımını yapar, yani her özellik sınıf etiketi bilindiğinde diğerlerinden bağımsız olarak kabul edilir. Bu varsayım altında, Bayes teoremini şu şekilde yeniden yazabiliriz:

$$P(C|X) = \frac{P(x_1|C) * P(x_2|C) * \dots * P(x_n|C) * P(C)}{P(X)} \tag{5.2}$$

Bu formül, verilen bir örneğin her bir özelliğinin belirli bir sınıfa ait olma olasılığını birlikte çarparak sınıfın olasılığını tahmin eder.

Sürekli veri ile uğraşırken, tipik bir varsayım her bir sınıfla ilişkilendirilen sürekli değerlerin normal (veya Gauss) dağılıma göre dağıldığıdır. Bu çalışmada da Gaussian Naive Bayes kullanılmıştır. Gaussian Naive Bayes, Naive Bayes sınıflandırıcısının bir türüdür ve özelliklerin bir normal dağılıma sahip olduğu varsayımını yapar. Yani, özelliklerin normal dağılıma sahip olduğu varsayımı altında, sınıf etiketi bilindiğinde her bir özelliğin olasılığı, bu özelliğin o sınıfa ait olduğu normal dağılımın olasılığını kullanarak hesaplanır. Bu algoritma, sınıflandırma problemleri için kullanılır ve özelliklerin sürekli olduğu durumlar için uygun bir seçenektir.

5.2.2. K-En Yakın Komşu

K-En Yakın Komşu, basit fakat çoğu durumda etkili olan parametrik olmayan bir sınıflandırma yöntemidir. Bir t veri kaydının sınıflandırılması için, onun en yakın K komşusu alınır ve bu, t'nin bir komşuluğunu oluşturur. Komşuluktaki veri kayıtları arasındaki çoğunluk oyu genellikle mesafeye dayalı ağırlıklandırma dikkate alınarak veya dikkate alınmayarak sınıflandırmaya karar vermek için kullanılır. Ancak KNN'yi uygularken K için uygun bir değer seçmemiz gerekir ve sınıflandırmanın başarısı büyük ölçüde bu değere bağlıdır. Bir bakıma KNN yöntemi K değerine bağımlıdır. Bu değerini seçmenin birçok yolu vardır, ancak basit olanı, algoritmayı farklı K değerleriyle birçok kez çalıştırmak ve en iyi performansa sahip olanı seçmektir. KNN algoritmasında k'nın seçimine daha az bağımlı olmak için; Wang (1998), tek bir K-en yakın komşular kümesi yerine birden fazla en yakın komşular kümesine bakmayı önerdi. Önerilen formalizm bağlamsal olasılığa dayalıdır ve fikir çeşitli sınıflar için birden fazla en yakın komşu kümesinin desteğini bir araya getirerek gerçek t sınıfını daha iyi ortaya koyan daha güvenilir bir destek değeri vermektir (Guo, Wang, Bell, Bi, & Greer, 2003).

Bir KNN modeli, iki nokta arasındaki mesafeyi kullanarak benzerliği hesaplar. Noktalar arasındaki mesafe ne kadar büyükse, o kadar az benzerdirler. Mesafe ölçümü için birçok yöntem vardır, ancak en yaygın olanı Öklid mesafesidir.

KNN, sınıflandırma problemleri için yeni veri noktasına sınıf etiketini atarken en yakın K komşusunun oy çoğunluğuna dayanır. Örneğin, eğer en yakın K komşunun çoğunluğu A sınıfındaysa, algoritma yeni veri noktasını da A sınıfından olarak tahmin eder.

Regresyon problemleri için ise KNN, sayısal değerleri tahmin eder. Yeni veri noktası için hedef değişkenin değerini belirlemek için K en yakın komşusunun hedef değerlerinin ortalamasını veya ağırlıklı ortalamasını kullanır.

K seçiminde, küçük bir K değeri (örneğin, K=1), algoritmanın gürültülü veri noktalarına aşırı duyarlı hale gelmesine ve aşırı uyum oluşmasına neden olabilir. Diğer yandan, büyük bir K değeri, algoritmanın verideki yerel kalıpları göz ardı etmesine ve yetersiz uyum oluşmasına yol açabilir. Bu nedenle, optimum performansı elde etmek için uygun bir K değerinin seçilmesi önemlidir.

Öklid mesafesi, iki nokta arasındaki 2-boyutlu uzaklığı hesaplamak için yaygın olarak kullanılan bir metriktir. Matematiksel olarak, iki nokta arasındaki Öklid mesafesi, koordinat düzlemindeki iki nokta arasındaki uzunluğu hesaplamak için kullanılan "Pitonagoras teoremi" ile hesaplanır. İki nokta arasındaki Öklid mesafesi d şu formülle hesaplanır:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5.3}$$

Burada, (x_1, y_1) ve (x_2, y_2) sırasıyla iki noktanın koordinatlarıdır. Bu formül, iki boyutlu uzayda çalışırken kullanılan bir Öklid mesafesi örneğidir. Ancak, KNN gibi çok boyutlu veri setleriyle çalışırken, mesafeyi hesaplarken her bir boyuttaki farklar için karelerin toplamını alarak genişletilebilir. Örneğin, üç boyutlu bir uzayda (x, y, z), iki nokta arasındaki Öklid mesafesi şu şekilde hesaplanır:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{5.4}$$

Bu genel formül, her boyuttaki koordinat değerlerinin farkların karelerini alır, toplar ve son olarak karekökünü alarak iki nokta arasındaki uzaklığı hesaplar. Bu, KNN algoritmasının çok boyutlu veri setlerinde de çalışmasını sağlar.

5.2.3. Rassal Ormanlar

Adından da anlaşılacağı gibi RF, daha doğru sınıflandırmalar üretmek için birçok karar ağacını birleştirir. RF hesaplamalarının yan ürünleri, değişken öneme sahip ölçümleri ve kümeleme, çok boyutlu ölçeklendirme, grafiksel gösterim ve eksik değer atama için kullanılabilecek veri noktalarının benzerlik ölçümlerini içerir (Cutler, ve diğerleri, 2007).

Rassal Orman algoritması, ağaçları büyütürken modele rastgelelik katmanın bir yoludur. Bu yöntem, düğümleri bölmeden önce belirli bir özellik alt kümesini rastgele seçer. Bunun yerine, her özellik için en iyi eşiği aramak yerine, her bir özelliğin rastgele eşiklerini kullanarak geniş bir çeşitlilik elde edilir. Bu yaklaşım, genellikle daha iyi bir model elde etmek için çeşitli ağaçların bir araya getirilmesiyle sonuçlanır. Bu sayede Rassal Orman, her ağacın kendi rastgele doğasını koruyarak, daha güçlü ve dengeli bir model oluşturur.

Öncelikle, eğitim ver kümesi olan D kullanılır. Bu veri kümesi, n örneğe ve m özelliğe sahip $(X_1, y_1), (X_2, y_2), \dots, (X_n, X_n)$ şeklindeki örneklerden oluşur. Burada X_i

özelliklerin ve y_i hedef değişkenin vektörleridir. RF algoritması, belirli bir sayıda karar ağacı oluşturur. Bu ağaçlar genellikle aynı türde olup, ancak rastgele örneklemeler ve özelliklerle eğitilirler. Her bir ağaç için, rastgele bir alt örneklem D_i oluşturulur. Bu, orijinal veri kümesinden rastgele örnekler alınarak oluşturulur. Rastgele bir özellik alt kümesi seçilir. Bu, toplam özelliklerin bir alt kümesini rastgele seçerek yapılır. Seçilen alt örneklem ve özellikler kullanılarak bir karar ağacı oluşturulur. Oluşturulan tüm karar ağaçlarının tahminleri bir araya getirilir. Sınıflandırma problemlerinde, genellikle sınıfların modu (en çok tekrar eden sınıf) veya sınıfların olasılıklarının ortalaması kullanılır. Regresyon problemlerinde ise genellikle tahminlerin ortalaması alınır.

Random Forest algoritması, her bir ağacın rastgele seçilen örnekler ve özelliklerle eğitilmesi sayesinde overfitting'e (aşırı uyum) karşı daha dirençli bir model elde etmeyi amaçlar. Ayrıca, birçok ağacın bir araya getirilmesiyle oluşturulan ensemble, genellikle daha güçlü ve genelleştirilmiş tahminler yapar.

5.2.4. Karar Ağaçları

Karar ağaçları, bir dizi basit testi mantıksal olarak birleştiren sıralı modellerdir; her test, sayısal bir özelliği bir eşik değeriyle veya nominal bir özelliği bir dizi olası değerle karşılaştırır. Bir karar ağacı tarafından takip edilen mantıksal kuralların yorumlanması, bir sinir ağındaki düğümler arasındaki bağlantıların sayısal ağırlıklarından çok daha kolaydır. Karar vericiler anlayabilecekleri modelleri kullanma konusunda kendilerini daha rahat hissetme eğilimindedir. Bir veri noktası bölümlenmiş bir bölgeye düştüğünde, karar ağacı onu en sık karşılaşılan sınıfa ait olarak sınıflandırır.

Karar ağaçlarının otomatik olarak oluşturulmasını gerçekleştiren birçok program geliştirilmiştir. Bu programlar, bir dizi etiketli örnek, yani her bir örneğin sınıf etiketinin belirlendiği önceden edinilmiş verilerin bir koleksiyonunu gerektirir. Algoritma verileri genelleştirmeye veya verilerdeki kalıpları bulmaya çalışır. Bunu, hangi testlerin(soruların) örnekleri en iyi şekilde ayrı sınıflara böldüğünü belirleyerek bir ağaç oluşturarak yapar. Daha sonra örnekler bölünmeye göre bölümlenir ve bu prosedür, bir düğümdeki tüm örnekler aynı sınıfa ait olana kadar yinelemeli olarak uygulanır. Diğer model sınıflandırma paradigmalarında olduğu gibi, daha karmaşık modeller (daha büyük karar ağaçları) daha zayıf genelleme performansı üretme eğilimindedir. O halde daha

küçük boyutlu karar ağaçları oluşturmak için büyük miktarda çaba harcanması şaşırtıcı değildir (Kotsiantis, 2013).

Öncelikle, eğitim veri kümesi olan D kullanılır. Bu veri kümesi, n örneğe ve m özelliğe sahip $(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)$ şeklindeki örneklerden oluşur. Burada X_i özelliklerin ve y_i hedef değişkenin vektörleridir.

Karar ağacı, veri kümesini bölerek hedef değişkenin tahminini yapacak bir dizi karar kuralı veya koşul oluşturur. Ağaç oluşturulurken genellikle en iyi bölünmeyi sağlayan özellik ve eşik değeri bulunur. Bu, genellikle bilgi kazancı, Gini impurity veya diğer ölçütler kullanılarak yapılır. Karar ağacı oluşturulurken her düğüm için en iyi bölünme seçilir ve bu şekilde ağaç adım adım büyütülür. Bu işlem, belirli bir durdurma kriteri (örneğin, ağacın maksimum derinliği, minimum örnekleme sayısı gibi) sağlanana kadar devam eder.

Oluşturulan karar ağacı kullanılarak yeni veri noktaları için tahminler yapılır. Yeni bir veri noktası ağaçta kök düğümden başlayarak aşağıya doğru ilerler ve her düğümde karar kuralıyla karşılaşır. Bu kuralların sonucunda, veri noktası bir yaprak düğümüne ulaşır ve bu düğümün tahmin ettiği hedef değişken değeri alınır.

5.2.5. AdaBoost

Boosting, nispeten zayıf ve yanlış birçok kuralı birleştirerek son derece doğru tahmin kuralı oluşturma fikrine dayanan bir makine öğrenimi yaklaşımıdır. Freund ve Schapire'ın (1997) AdaBoost algoritması ilk pratik güçlendirme algoritmasıdır ve birçok alanda uygulamalarıyla en yaygın kullanılan ve üzerinde çalışılan algoritmalardan biri olmaya devam etmektedir (Freund & Schapire, 1997). AdaBoost, ilk boosting algoritması olarak sayılır.

Bu modelde eğitim seti, öncelikle bir zayıf öğrenici ile eğitilir. Ardından, eğitim sonucunda yanlış tahminlenen örnekler bu algoritma için kritiktir. Bir sonraki eğitimde, ilk tahminlemelerde yanlış öğrenilen verilere daha fazla önem verilerek, yani ağırlıkları artırılarak tekrar eğitim sağlanır. Bu süreçte, zayıf öğrenicilerin çıktıları diğer öğrenicilere giriş sağlayacak şekilde eğitilir ve en sonunda elde edilen sonuçlar birleştirilerek nihai karar sınırları oluşturulur. Eğitim setinden eğitilmiş, maksimum derinliğe sahip karar ağaçları kullanılmaz. Bunun yerine, zayıf öğrenici olarak, AdaBoost ile bir nod ve iki

yapraktan oluşan, derinliği bir olan karar ağaçları tercih edilir. Bu kümülatif yapıda, sıralama önemlidir. İlk karar ağacının hataları, ikinci ağacın ağırlıklarını etkiler.

5.2.6. Çok Katmanlı Algılayıcı

Çok Katmanlı Algılayıcı, yapay sinir ağlarının en temel ve yaygın biçimidir. Bu model, insan beyninin bilgi işleme sürecini taklit etmeye çalışan matematiksel bir yapıdır. MLP, birbirine bağlı yapay sinir hücrelerinden oluşan birden fazla katmana sahiptir. Bu katmanlar genellikle şu şekilde düzenlenir: bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıktı katmanı.

Çok katmanlı algılayıcıların yetenekleri, düğümlerde kullanılan doğrusal olmayan öğelerden kaynaklanır. Eğer düğümler doğrusal öğeler olsaydı, uygun şekilde seçilmiş ağırlıklarla tek katmanlı bir ağ, herhangi birçok katmanlı ağ tarafından yapılan hesaplamaları tam olarak kopyalayabilirdi (Lippmann, 1988).

MLP algoritmasında giriş vektörü; $X = [x_1, x_2, \dots, x_n]$ olmak üzere; Her gizli katmanda, önceki katmandan gelen çıktılar üzerine ağırlıklı toplam işlemi uygulanır;

$$Z^{(i)} = W^{(i)} . A^{(i-1)} + b^{(i)} \tag{5.5}$$

Sonra aktivasyon fonksiyonu f uygulanarak gizli katmanın çıktısı elde edilir;

$$A^{(i)} = f\left(Z^{(i)}\right) \tag{5.6}$$

Son gizli katmandan gelen çıktı üzerine ağırlıklı toplam işlemi uygulanır;

$$Z^{(l)} = W^{(L)} . A^{(L-1)} + b^{(L)} \tag{5.7}$$

Son olarak, çıktı katmanının çıktısı elde edilir:

$$\hat{Y} = A^{(L)} = f\left(Z^{(L)}\right) \tag{5.8}$$

Burada,

$W^{(i)}$ ve $b^{(i)}$, i-inci katmandaki ağırlık matrisi ve bias vektörüdür.

$A^{(i)}$, i-inci katmandaki aktivasyonları içeren vektördür.

$Z^{(i)}$, i-inci katmandaki toplam girdileri içeren vektördür.

f , aktivasyon fonksiyonudur. (ReLU, sigmoid, tanh gibi.)

L , toplam katman sayısıdır.

Eğitim aşamasında, gerçek çıktı Y ile tahmini çıktı $\hat{Y}$ arasındaki hatayı minimize etmek için geriye yayılım algoritması kullanılır. Bu algoritma, türev zinciri kuralını kullanarak ağdaki hata kaynağını geriye doğru yayarak ağırlıkları günceller.

5.2.7. Doğrusal Ayrımcılık Analizi

Doğrusal Ayrımcılık Analizi, çok sınıflı sınıflandırma problemleri için kullanılan bir boyut indirgeme ve özellik seçimi tekniğidir. Temel amacı, farklı sınıflar arasındaki ayrımı maksimize etmektir. LDA, sınıflar arası ayrımı en üst düzeye çıkaran ve sınıf içi varyansı minimize eden bir yöntemdir. Bu şekilde, verilerin sınıflar arasında daha net ayrılabilmesi ve sınıflandırma performansının artırılabilmesi sağlanır.

Veri kümeleri dönüştürülebilir ve test vektörleri dönüştürülmüş uzayda iki farklı yaklaşımla sınıflandırılabilir.

Sınıfa Bağımlı Dönüşüm: Bu yaklaşım, sınıflar arasındaki varyansın oranını maksimize etmeyi hedefler. Yani, her bir sınıf için ayrı ayrı dönüşüm yapılır ve sınıf içi varyansı minimize edip, sınıflar arası ayrımı maksimize edecek şekilde optimizasyon yapılır. Bu şekilde, her sınıfın ayrılabilirliği en üst düzeye çıkarılmaya çalışılır.

Sınıftan Bağımsız Dönüşüm: Bu yaklaşım, genel varyansın sınıf içi varyansa oranını maksimize etmeyi amaçlar. Bu yöntemde, tüm veri noktaları sınıf kimliklerine bakılmaksızın aynı dönüşüm kullanılarak dönüştürülür. Yani, veri kümesi genelindeki varyansı maksimize edecek tek bir dönüşüm bulunmaya çalışılır.

Her iki yaklaşım da LDA içinde kullanılabilir ve sınıflandırma performansını artırmak için veri dönüşümü sağlar (Balakrishnama & Ganapathiraju, 1998).

Veri seti standartlaştırılarak ortalama değeri 0, varyansı ise 1 olacak şekilde normalize edilir. Bu adım, verinin işlenmesini kolaylaştırır. Her bir sınıf için özelliklerin ortalaması (merkez) hesaplanır. Bu, sınıfların merkez noktalarını temsil eden vektörlerdir. Sınıf k için merkez vektörü:

$$m_k = \frac{1}{n_k} \sum_{i \in C_k} x_i \tag{5.9}$$

Burada,

n_k: Sınıf k içindeki örneklemlerin sayısıdır.

C_k: Sınıf k içindeki örneklemlerin indisleridir.

x_i: i-inci örnek vektörüdür.

Sınıf içi dağılım matrisi, her bir sınıfın içindeki veri noktalarının dağılımını tanımlayan bir matristir:

$$S_W = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - m_k)(x_i - m_k)^T \tag{5.10}$$

Sınıflar arası dağılım matrisi, sınıflar arasındaki farklılıkları tanımlayan bir matristir:

$$S_B = \sum_{k=1}^{K} n_k (m_k - m)(m_k - m)^T \tag{5.11}$$

m = Tüm veri seti için ortak merkez vektörü olmak üzere,

$$m = \frac{1}{N} \sum_{k=1}^{K} n_k m_k \tag{5.12}$$

Burada,

N: Toplam örnek sayısıdır.

LDA, sınıflar arası dağılımı maksimize edip, sınıf içi dağılımı minimize ederek optimize edilir. Bu kriter, sınıf merkezlerinin birbirlerine yakın olmasını ve aynı zamanda sınıflar arasındaki mesafenin de en büyük olmasını hedefler.

Fisher kriterini optimize etmek için en büyük kuvvet yöntemi kullanılır. Bu yöntemde, $S_W^{-1} S_B$ matrisinin öz vektörleri ve öz değerleri hesaplanır. Bu öz vektörler, veri setini en iyi ayıran doğrusal bileşenlerdir.

Son olarak, en önemli öz vektörler ile veri seti dönüştürülür:

$$y = W^T x \tag{5.13}$$

Burada,

W: Öz vektör matrisidir

y: Yeni dönüşüm veri setidir.

Bu adımlar, LDA matematiksel temelini oluşturur. LDA, sınıflar arasındaki ayrımı en üst düzeye çıkararak veri setlerini sınıflandırmak ve boyut indirgemek için

güçlü bir araç sağlar. LDA dönüşümü uygulandıktan sonra, genellikle sınıflandırma modeli eğitilir. Bu model, dönüştürülmüş veri üzerinde sınıfları doğru bir şekilde tahmin etmeye çalışır.

5.2.8. Lojistik Regresyon

Regresyon yöntemleri, bir yanıt değişkeni ile bir veya daha fazla açıklayıcı değişken arasındaki ilişkinin tanımlanmasıyla ilgilenen herhangi bir veri analizinin ayrılmaz bir bileşeni haline gelmiştir. Çoğunlukla sonuç değişkeni ayrıktır ve iki veya daha fazla olası değeri alır. Lojistik regresyon modeli, bu verilerin analizi için en sık kullanılan regresyon modelidir.

Lojistik regresyon modelini doğrusal regresyondan ayıran şey, lojistik regresyondaki sonuç değişkeninin ikili olmasıdır. Lojistik ve doğrusal regresyon arasındaki bu fark hem modelin biçiminde hem de varsayımlarında yansıtılmaktadır. Bu fark hesaba katıldığında, lojistik regresyon kullanan bir analizde kullanılan yöntemler, aşağı yukarı doğrusal regresyonda kullanılan genel prensiplerin aynısını izler. Dolayısıyla doğrusal regresyon analizinde kullanılan teknikler lojistik regresyona yaklaşımı motive etmektedir (Hosmer Jr, Lemeshow, & Sturdivant, 2013).

Lojistik regresyon, bağımsız değişkenlerin doğrusal kombinasyonunu sınıflandırmak için bir lojistik (sigmoid) fonksiyon ile bağlayan bir modeldir. Temelde, doğrusal regresyon modelinin logit dönüşümüdür.

Her bir örnek x_i için, lineer kombinasyon veya logit değeri z_i şu şekildedir:

$$z_i = w^T x_i + b \tag{5.14}$$

Burada,

w: Ağırlık vektörü

b: Bias terimi

Lineer kombinasyon sonucu elde edilen z_i değeri, sigmoid fonksiyonuna $\sigma(z_i)$ uygulanarak 0 ile 1 arasında bir olasılık değeri elde edilir:

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}} \tag{5.15}$$

Modelin eğitimi için kullanılan olabilirlik fonksiyonu, veri setindeki her bir örneğin doğru sınıf olasılığını maksimize etmeyi hedefler:

$$L(w,b) = \prod_{i=1}^{N} P(y_i|x_i;w,b) \tag{5.16}$$

Burada:

$P(y_i|x_i;w,b) = \sigma(z_i)^{y_i}(1 - \sigma(z_i))^{1-y_i}$

y_i: Gerçek sınıf etiketi

Eğitim sırasında kullanılan kayıp fonksiyonu, modelin tahminlerinin gerçek etiketlerden ne kadar uzak olduğunu ölçer. Çapraz entropi yaygın olarak kullanılır:

$$J(w,b) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i)] \tag{5.17}$$

Model parametreleri, genellikle gradyan azaltma gibi optimizasyon algoritmaları kullanılarak güncellenir:

$$w^{(t+1)} = w^{(t)} - \alpha \nabla_w J\left(w^{(t)}, b^{(t)}\right) \tag{5.18}$$

$$b^{(t+1)} = b^{(t)} - \alpha \frac{\partial J\left(w^{(t)}, b^{(t)}\right)}{\partial b} \tag{5.19}$$

Burada,

α: Öğrenme oranı

$\nabla_w J$: Kayıp fonksiyonunun ağırlık vektörüne göre gradyanı

$\frac{\partial J}{\partial b}$: Kayıp fonksiyonunun bias terimine göre türevi

Model eğitildikten sonra, yeni veri noktaları için sınıflandırma tahminleri yapılır

5.2.9. Gradyan Artırıcı

Gradyan Artırıcı, topluluk tabanlı bir makine öğrenimi tekniğidir ve küçük karar ağaçlarını aşamalı olarak birleştirerek modeller oluşturur. Her ardışık ağaç, önceki tüm eğitilmiş ağaçların genel tahmin gücünü artırmaya çalışır.

Gradyan artırmanın temel prensibi, genel amaç fonksiyonunu en aza indirmek için tahmin edicileri sırayla eğitmektir. Bu teknik, zayıf öğrenicileri güçlü bir öğrenicide birleştiren bir çalışan artırma tekniği kullanır. Zayıf öğreniciler, rastgele tahminden biraz

daha iyi olan tahmincilerdir. Buradaki temel fikir, güçlendirme tekniklerini kullanarak birden fazla zayıf öğreniciyi tek bir, daha güçlü öğrenicide birleştirmektir.

Güçlendirme sürecinde ilk olarak bir temel veya zayıf öğrenici seçilir. Sonraki her adımda, boosting tekniği farklı ağırlıklar vererek zayıf öğrenicileri tek tek güçlendirir, böylece tek başlarına ve bir bütün olarak daha etkili hale gelirler. Güçlendirme sürecinin sonunda, zayıf öğrenicilerin birleşimi ile çok daha güçlü bir tahminci oluşturulur.

Fonksiyon tahmininde veya "tahmin edici öğrenme" probleminde, rastgele bir "çıktı" veya "yanıt" değişkeni ve bir dizi rastgele "girdi" veya "açıklayıcı" değişkenden oluşan bir sistem vardır (Friedman, 2001).

Eğitim verisi $\{(x_i, y_i)\}_{i=1}^{n}$ içerir, burada x_i özellik vektörü y_i hedef değişkendir. Hedef değişken, regresyon için gerçek sayılar veya sınıflandırma için etiketler olabilir.

Hedef, bir genel amaç fonksiyonu (kayıp fonksiyonu) $L(y, F(x))$ ile minimize edilir, burada $F(x)$ tahminciyi temsil eder. Tahminci $F(x)$, zayıf öğrenicilerin (genellikle karar ağaçları) toplamı olarak ifade edilir:

$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{5.20}$$

Burada, $f_m(x)$ her bir zayıf öğrencidir ve M toplam ağaç sayısını temsil eder.

GB, gradyan azaltma kullanarak bir sonraki tahminciyi eğitir. Kayıp fonksiyonunun L gradyanı $\nabla_{F(x)} L(y, F(x))$ alınır ve bu gradyanı tahminciyi geliştirmek için kullanır. Başlangıçta, tahminci $F_0(x)$ genellikle bir sabit değer (örneğin, veri setinin ortalaması) olarak belirlenir. Ardışık olarak, her iterasyonda şu adımlar gerçekleştirilir:

Hata(residual) Hesaplama:

$$r_{im} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}, \qquad i = 1, \dots, n \tag{5.21}$$

Yeni ağaç eğitimi,

$$f_m = arg\min_{f} \sum_{i=1}^{n} L\big(y_i, F_{m-1}(x_i) + f(x_i)\big) \tag{5.22}$$

Ağacı Tahminciye ekleme,

$$F_m(x) = F_{m-1}(x) + v.F_m(x) \quad (5.23)$$

Burada v bir öğrenme oranıdır, genellikle küçük bir değerdir (0,01 ile 0,1 arasında).

Bu işlem, belirtilen sayıda ağaç (M) eklenecek kadar devam eder. Son tahminci, $F_M(x)$, tüm ağaçların toplamıdır ve bu, hedef değişkenin tahmini olarak kullanılır.

5.3. CSE-CIC-IDS-2018

CSE-CIC-IDS-2018 veri seti, İletişim Güvenliği Kurumu (CSE) ile Kanada Siber Güvenlik Enstitüsü (CIC) arasındaki iş birliği projesi kapsamında geliştirilmiştir (Canadian Institute for Cybersecurity, 2018). Veri, on günlük bir dönem boyunca elde edilmiş olup, elli dört sütun içermekte ve on beş farklı türde saldırıyı içermektedir: FTP-Brute Force, SSH-Brute Force, DoS saldırıları-GoldenEye, DoS saldırıları-Slowloris, DoS saldırıları-Hulk, DoS saldırıları-SlowHTTPTest, DDoS saldırıları-LOIC-HTTP, DDoS saldırısı-HOIC, DDoS saldırısı-LOIC-UDP, Brute Force-Web, Brute Force-XSS, SQL enjeksiyonu, sızma, etiket ve bot. Bu saldırılar çalışmamızda da Çizelge 5.1'deki gibi altı farklı saldırı senaryoya ayrılmaktadır;

Çizelge 5.1. CSE-CIC-IDS-2018 saldırı senaryoları

Senaryo	Saldırı Tipleri	Toplam
Normal	Benign	13,484,708
DDoS Saldırısı	Hoic Loic-UDP Loic-HTTP	1,263,933
DoS Saldırısı	HULK GoldenEye Slowloris SlowHTTPTest	654,300
Kaba Kuvvet Saldırısı	FTP SSH	380,949
Bot Saldırıları	Botnet	286,191
Sızma Saldırıları	Infilteration	161,934
Web Saldırıları	Web XSS SQL Injection	923

Saldırı altyapısında 50 makine bulunurken, mağdurlar toplamda 30 sunucu ve 420 makineden oluşan beş bölümden meydana gelmektedir. Veriler üç ay süresince 10 ayrı gün boyunca toplanmıştır. AWS bulutundan indirilebilecek 10 CSV dosyasının gün gün dağılımı Çizelge 5.2'de gösterilmiştir.

Çizelge 5.2. CSE-CIC-IDS-2018 AWS bulutu network trafik dağılımı

Gün	Dosya Adı	Özellik sayısı	Saldırı Tipi	Saldırı Tipleri Sayısı	Toplam
1	Wednesday-14-02-2018	80	Benign FTP SSH	667,626 193,360 187,589	1,048,575
2	Thursday-15-02-2018	80	Benign GoldenEye Slowloris	996,077 41,508 10,990	1,048,575
3	Friday-16-02-2018	80	Benign Hulk SlowHTTPTest	446,772 461,912 139,890	1,048,574
4	Thursday-20-02-2018	84	Benign LOIC-HTTP	7,372,557 576,191	7,948,748
5	Wednesday-21-02-2018	80	Benign HOIC LOIC-UDP	360,833 686,012 1730	1,048,575
6	Thursday-22-02-2018	80	Benign Brute Force-Web Brute Force-XSS SQL Injection	1,048,213 249 79 34	1,048,575
7	Friday-23-02-2018	80	Benign Brute Force-Web Brute Force-XSS SQL Injection	1,048,009 362 151 53	1,048,575
8	Wednesday-28-02-2018	80	Benign Infilteration	544,200 68,871	613,071
9	Thursday-01-03-2018	80	Benign Infilteration	238,037 93,063	331,100
10	Friday-02-03-2018	80	Benign Botnet	762,384 286,191	1,048,575

Toplam satır (Örneklem)= 16,232,943; Toplam sütun (Özellik) = 84.

Veri seti daha önce derlenen CICIDS-2017 veri kümesiyle aynı özellikleri taşımaktadır (Canadian Institute for Cybersecurity, 2017). Toplanan veriler gerçek dünya trafiğini andırmaktadır ve ağ anomalisi tespiti bağlamında kullanılmak üzere uygun bir

veri kümesidir. Bu veri setinin seçilmesindeki en önemli sebepler; Veri kümesinde tekrarlanan veri azdır, NaN (belirsiz) veri neredeyse yok gibidir ve CSV formatında kullanıma hazırdır şeklinde sıralanabilir.

5.4. Kullanılan Dil ve Kütüphaneler

Python: Python yüksek seviyeli, nesneye yönelik ve yorumlamalı bir programlama dilidir. Python'ın basit, öğrenmesi kolay sözdizimi vardır. Kodun parçalanmasını ve kodun tekrar kullanılabilirliğini teşvik eden modülleri ve paketleri destekler. Python yorumlayıcısı ve kapsamlı standart kütüphanesi, tüm önemli platformlar için ücretsiz olarak sunulur ve özgürce dağıtılabilir (Python).

Pandas: Python'da pratik, gerçek dünya veri analizi yapmayı sağlayan açık kaynaklı veri analizi/manipülasyon aracıdır (Pandas).

Matplotlib: Python'da statik, animasyonlu ve etkileşimli görselleştirmeler oluşturmak için kullanılan kapsamlı bir kütüphanedir (Matplotlib).

Numpy: Python'da bilimsel hesaplamalar için kullanılan temel pakettir. Çok boyutlu bir dizi nesnesi, matrisler, diziler üzerinde hızlı işlemler için çeşitli rutinler sunan bir Python kütüphanesidir. Bu rutinler arasında matematiksel işlemler, mantıksal işlemler, şekil manipülasyonu, sıralama, seçme, G/Ç işlemleri, ayrık Fourier dönüşümleri, temel lineer cebir, temel istatistiksel işlemler ve çok daha fazlası bulunmaktadır (Numpy).

Seaborn: Seaborn, matplotlib'e dayalı bir Python veri görselleştirme kütüphanesidir. Çekici ve bilgilendirici istatistiksel grafikler çizmek için yüksek seviyeli bir arayüz sağlar (Seaborn).

Scikit-learn: Python'da makine öğrenimi için kullanılan açık kaynaklı bir kütüphanedir. Sınıflandırma, regresyon, kümeleme ve boyut azaltma gibi makine öğrenimi ve istatistiksel modelleme için etkili araçlar seçeneği sunar ve bunları Python'da tutarlı bir arayüz üzerinden sağlar. Bu kütüphane, büyük ölçüde Python'da yazılmış olup, NumPy, SciPy ve Matplotlib üzerine inşa edilmiştir (Scikit-learn).

Imblearn: Dengesiz sınıflara sahip veri setleriyle çalışırken sınıflandırma modellerini dengelemek için kullanılan bir Python kütüphanesidir (Imbalanced-learn).

5.5. Veri Önişleme

Çalışmada kullanılan CSE-CIC-IDS-2018 veri seti ML algoritmaları ile uyumlu bir şekilde çalışabilmesi için Python programlama dili kullanılarak bazı ön işlemlerden geçirilmiştir. İlk olarak Çizelge 5.2'de görülen 10 csv dosyası indirildikten sonra 1.gün Brute Force saldırıları için, 2. ve 3.gün DoS saldırıları için, 4. ve 5.gün DDoS saldırıları için, 6. ve 7.gün Web saldırıları için 8. ve 9.gün Infilteration saldırıları için ve 10.gün Botnet saldırıları için olacak şekilde 6 farklı veri seti oluşturulmuştur. Oluşturulan veri setlerine ayrı ayrı aşağıdaki adımlar uygulanmıştır;

1) Veri setinde bulunan değerlerin tipleri uygun şekilde düzenlenmiştir. "Timestamp" ve "Label" sütunları dışındaki değerler eğer sayısal değilse uygun formda sayısal değerlere dönüştürülmüştür.
2) Sonsuz ve eksik (NaN) değerler temizlenmiştir.
3) Gereksiz olarak görülen "Timestamp" sütunu kaldırılmıştır.
4) Farklı çeşitteki saldırılar tümü "Label" sütunu altında temsil ettikleri saldırı tipinin adını almışlardır. Böylece "Label" sütunu altında "Saldırı adı" ve "Normal" olacak şekilde sadece iki değer olması sağlanmıştır.
5) "LabelEncoder" fonksiyonu ile "Label" sütunundaki değerlere saldırılar "1" normal sınıf "0" olacak şekilde etiketlendirme yapılarak bu etiketlerden oluşan "Binary_label" sütunu oluşturulmuştur.
6) "Label" sütunu silinerek böylelikle bütün veri setindeki verilerin sayısal değerler olması sağlanmıştır.
7) "Train_test_split" fonksiyonu ile veri %20 test ve %80 eğitim kümesi olacak şekilde ayrılmıştır.

Adımlardan sonra oluşan veri setlerinin örneklem değerleri Çizelge 5.3'te gözlemlenmektedir.

Çizelge 5.3. CSE-CIC-IDS-2018 veri hazırlama sonrası demografik özellikler

Saldırı Tipi	Hazırlık Öncesi	Hazırlık Sonrası	Test Seti	Eğitim Seti
DDoS	8,997,323	8,937,870	1,787,574	7,150,296
DoS	2,097,149	2,089,122	417,825	1,671,297
Brute Force	1,048,575	1,044,751	208,951	835,800
Bot	1,048,575	1,039072	207,815	831,257
Infilteration	944,171	935,083	186,062	744,244
Web	2,097,150	2,085,832	415,991	1,663,961

5.6. Öznitelik Çıkarımı

Öznitelik seçimi, orijinal veri setini doğru bir şekilde yansıtan faydalı özellikleri çıkarma tekniğidir. Bu, gereksiz veya gürültülü özellikleri silerek model performansını artırırken, orijinal veri desenini bozmadan veya temel özellikleri çıkarmadan yapılır. Veride en uygun özniteliklerin seçilmesi, doğruluk ve hesaplama verimliliği açısından tespit verimliliğini artırdığı gösterilmiştir. Bu nedenle, anormallik tabanlı bir IDS'nin tasarımında kritik bir rol oynamaktadır (Torabi, Udzir, Abdullah, & Yaakob, 2021).

Çalışmamızda 3 farklı öznitelik seçme tekniği kullanılmıştır;

SelectKBest: Scikit-learn kütüphanesinde bulunan bir öznitelik seçimi yöntemidir. Veri setindeki en iyi K özniteliği seçmek için çeşitli istatistiksel testleri kullanır. Bu uygulamada f_classif skorlama fonksiyonu ile belirli bir adım boyutunda (batch_size) veri seti üzerinde dolaşarak en iyi K öznitelik seçilmiştir**.**

Mutual_info_classif: Scikit-learn kütüphanesindeki bir öznitelik seçimi yöntemidir. Bu yöntem, özelliklerin hedef değişkenle (sınıflandırma problemlerinde) olan karşılıklı bilgi paylaşımını ölçerek özelliklerin önemini belirler. Özelliklerin hedef değişkenle ne kadar bilgi paylaştığını ölçmek için kullanılan bir metriktir.

Chi2: Ki-kare kullanılarak öznitelik seçimi yapılırken, her özellik için Ki-kare istatistiği hesaplanır. Bu istatistik, özelliğin hedef değişkenle bağımsız olduğu durumda alınabilecek varyans miktarını temsil eder. Özelliklerin hedef değişkenle ne kadar bilgi paylaştığını ölçer ve bu sayede en bilgilendirici özellikleri seçmek için kullanılır.

Her bir öznitelik seçme algoritmasından ilk 10 özellik alınmıştır ve toplam 30 öznitelik arasında analiz yapılarak en iyi öznitelikler seçilmiştir. Bu işlem ayrı dosyalara kaydedilen her bir saldırı veri seti için ayrı ayrı uygulanmıştır.

5.6.1. DDoS saldırıları için seçilen öznitelikler

DDoS saldırılarının oluşturduğu veri setinde öznitelik seçimi sonrası seçilen özellikler ve seçilen özelliklerin açıklamaları Çizelge 5.4'te verilmiştir

Çizelge 5.4. DDoS veri setinde seçilen özellikler

Seçilen Özellik	Açıklama
Init Fwd Win Byts	İleri yönde ilk pencere içinde gönderilen bayt sayısı
Fwd Seg Size Avg	İleri yönde gözlenen ortalama boyut
Pkt Len Var	Bir akışın uzunluk varyansı

5.6.2. DoS saldırıları için seçilen öznitelikler

DoS saldırılarının oluşturduğu veri setinde öznitelik seçimi sonrası seçilen özellikler ve seçilen özelliklerin açıklamaları Çizelge 5.5'te verilmiştir.

Çizelge 5.5. DoS veri setinde seçilen özellikler

Seçilen Özellik	Açıklama
Dst Port	Hedef Port
ACK Flag Cnt	ACK içeren paket sayısı
Init Fwd Win Byts	İleri yönde ilk pencere içinde gönderilen bayt sayısı
Fwd Seg Size Min	İleri yönde gözlenen minimum segment boyutu
Fwd IAT Tot	İleri yönde gönderilen iki paket arasındaki toplam zaman
Fwd Seg Size Avg	İleri yönde gözlenen ortalama boyut
Fwd Header Len	İleri yöndeki başlıklar için kullanılan toplam byte
Fwd Pkt Len Max	İleri yönde paketlerin maksimum uzunluğu
Fwd Pkt Len Mean	İleri yönde paketlerin ortalama uzunluğu

5.6.3. Kaba Kuvvet saldırıları için seçilen öznitelikler

Kaba Kuvvet saldırılarının oluşturduğu veri setinde öznitelik seçimi sonrası seçilen özellikler ve seçilen özelliklerin açıklamaları Çizelge 5.6'da verilmiştir.

Çizelge 5.6. Brute Force veri setinde seçilen özellikler

Seçilen Özellik	Açıklama
Fwd Seg Size Min	İleri yönde gözlenen minimum segment boyutu
Bwd Pkts/s	Saniyedeki geri yön paket sayısı
Flow Duration	Mikro-saniyedeki akış süresi
Bwd IAT Max	Geri yönde gönderilen iki paket arasındaki maksimum zaman
Flow Byts/s	Saniyede akan byte sayısı
Flow Pkts/s	Saniyede akan paket sayısı
Init Fwd Win Byts	İleri yönde ilk pencere içinde gönderilen bayt sayısı
Pkt Len Var	Bir akışın uzunluk varyansı

5.6.4. Botnet saldırıları için seçilen öznitelikler

Botnet saldırılarının oluşturduğu veri setinde öznitelik seçimi sonrası seçilen özellikler ve seçilen özelliklerin açıklamaları Çizelge 5.7'de verilmiştir.

Çizelge 5.7. Botnet veri setinde seçilen özellikler

Seçilen Özellik	Açıklama
Flow Pkts/s	Saniyede akan paket sayısı
Flow IAT Mean	Paketlerin ortalama varış zamanı
Fwd IAT Mean	İleri yönde gönderilen iki paket arasındaki ortalama zaman
Fwd IAT Min	İleri yönde gönderilen iki paket arasındaki minimum zaman
Fwd Pkts/s	Saniyedeki ileri yön paket sayısı

5.6.5. Sızma saldırıları için seçilen öznitelikler

Sızma saldırılarının oluşturduğu veri setinde öznitelik seçimi sonrası seçilen özellikler ve seçilen özelliklerin açıklamaları Çizelge 5.8'de verilmiştir.

Çizelge 5.8. Infilteration veri setinde seçilen özellikler

Seçilen Özellik	Açıklama
Fwd Pkt Len Std	İleri yönde paket uzunluklarının standart sapması
Flow Pkts/s	Saniyede akan paket sayısı
Bwd Pkts/s	Saniyedeki geri yön paket sayısı
ACK Flag Cnt	ACK içeren paket sayısı
Down/Up Ratio	İndirme ve yükleme oranı
Init Bwd Win Byts	Geri yönde ilk pencere içinde gönderilen bayt sayısı
Fwd Pkt Len Mean	İleri yönde paketlerin ortalama uzunluğu
Flow Byts/s	Saniyede akan byte sayısı
Bwd Pkt Len Max	Geri yönde paketlerin maksimum uzunluğu
Fwd Pkts/s	Saniyedeki ileri yön paket sayısı

5.6.6. Web saldırıları için seçilen öznitelikler

Web saldırılarının oluşturduğu veri setinde öznitelik seçimi sonrası seçilen özellikler ve seçilen özelliklerin açıklamaları Çizelge 5.9'da verilmiştir.

Çizelge 5.9. Web Attack veri setinde seçilen özellikler

Seçilen Özellik	Açıklama
Flow IAT Max	Paketlerin maksimum varış zamanı
Fwd Pkt Len Mean	İleri yönde paketlerin ortalama uzunluğu
Flow Duration	Mikro-saniyedeki akış süresi
Flow Byts/s	Saniyede akan byte sayısı
Fwd Seg Size Min	İleri yönde gözlenen minimum segment boyutu
Fwd Act Data Pkts	İleri yönde en az 1 bayt TCP veri yüküne sahip paket sayısı
Subflow Fwd Byts	İleri yönde bir alt akıştaki ortalama byte sayısı

5.7. Kullanılan Performans Metrikleri

Sınıflandırma modellerinin ne kadar doğru sınıflandırdığını ve yanlış sınıflandırma oranlarını göstermek için Karmaşıklık Matrisi kullanılır. Aynı zamanda Hata Matrisi olarak da bilinen bu ölçüm aracı, tahmin başarısının hesaplanmasında yaygın olarak kullanılır. Basit ve anlaşılır olması nedeniyle, birçok STS çalışmasında ve araştırmada tercih edilmiştir. Bu çalışmada da Makine Öğrenimi modellerinin başarısını ölçmede Karmaşıklık Matrisi ve bu matristen elde edilebilen değerlendirme kriterlerinden modelin doğruluğu, kesinliği, duyarlılık ve F skoru kullanılmıştır. Karmaşıklık Matrisi Çizelge 5.10'da görülebileceği üzere binary (ikili) sınıflandırma için kullanılan 2x2 boyutunda bir matristir. Bir eksende gerçek değerler, diğer eksende tahmin edilen değerler bulunur.

Çizelge 5.10. Karmaşıklık Matrisi Modeli

<table>
<tr><td colspan="2" rowspan="2">Karmaşıklık Matrisi</td><td colspan="2">Tahmin Değerleri</td></tr>
<tr><td>Pozitif (1)</td><td>Negatif (0)</td></tr>
<tr><td rowspan="2">Gerçek Değerler</td><td>Pozitif (1)</td><td>TP</td><td>FP</td></tr>
<tr><td>Negatif (0)</td><td>FN</td><td>TN</td></tr>
</table>

Matristeki Pozitif (1) ve Negatif (0) terimleri doğruluk veya yanlışlığı değil ayrıştırılmak istenen sınıfları temsil eder. Doğru pozitif (TP), Doğru Negatif (TN), Yanlış Negatif (FN) ve Yanlış Pozitif (FP) parametreleri bir örnekle açıklayacak olursak;

Bir makine öğrenimi modeli, bir e-ticaret web sitesindeki kullanıcıların bir ürünü satın alıp almama olasılığını tahmin etmek için eğitilsin. Test veri seti, 100 farklı kullanıcının çevrimiçi alışveriş davranışlarını içeriyor olsun ve bu davranışların sonucunda, her bir kullanıcının ürünü satın alıp almadığı tahmin edilip sonuçları Çizelge 5.11'de verilsin.

Çizelge 5.11. Karmaşıklık Matrisi Örneği

Karmaşıklık Matrisi		Tahmin Değerleri	
		Pozitif (1)	Negatif (0)
Gerçek Değerler	Pozitif (1)	15	20
	Negatif (0)	10	55

Pozitif (1) ürünü alma durumu ve Negatif (0) ürünü almama durumu olmak üzere;

TP; Model, pozitif sınıfı (tahmin ve gerçek her ikisinde de pozitif olanlar) doğru bir şekilde tahmin eder. Yukarıdaki örnekte, model tarafından pozitif olarak tahmin edilen 15 kişi, aslında ürünü satın alan kişilerdir.

TN; Model, negatif sınıfı (tahmin ve gerçek her ikisinde de negatif olanlar) doğru bir şekilde tahmin eder. Yukarıdaki örnekte, model tarafından negatif olarak tahmin edilen 55 kişi, aslında ürünü almayan kişilerdir.

FP; Model, negatif sınıfı yanlış bir şekilde tahmin eder (tahmin-pozitif, gerçek-negatif). Yukarıdaki örnekte, 20 kişinin ürünü aldığı tahmin edilmiştir, ancak aslında ürünü satıl almayan kişilerdir.

FN; Model, pozitif sınıfı yanlış bir şekilde tahmin eder (tahmin-negatif, gerçek-pozitif). Yukarıdaki örnekte, 10 kişinin ürünü almadığı tahmin edilmiştir, ancak aslında ürünü satın alan kişilerdir.

Bu dört parametreden faydalanarak doğruluk, kesinlik, duyarlılık ve F ölçütü değerlerini açıklayalım;

Doğruluk değeri doğru sınıflandırma yeteneğini ölçer. Yani, doğru olarak tahmin edilen pozitif ve negatif sınıfların toplam veri sayısına oranıdır. Yüksek doğruluk değeri, modelin genel olarak doğru tahminler yaptığı anlamına gelirken, düşük doğruluk değeri modelin yanlış tahminler yaptığını gösterir. (5.24) denkleminde verildiği gibi elde edilir.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.24}$$

Kesinlik değeri bir modelin pozitif olarak tahmin ettiği örneklerin ne kadarının gerçekte pozitif olduğunu gösterir. Yüksek kesinlik değeri, modelin pozitif olarak tahmin ettiği örneklerin çoğunun gerçekten pozitif olduğunu ve yanlış pozitiflerin az olduğunu gösterir ve (5.25) denkleminde verildiği gibi bulunur.

$$Kesinlik = \frac{TP}{TP + FP} \tag{5.25}$$

Duyarlılık değeri modelin gerçek pozitifleri ne kadar başarılı bir şekilde tespit ettiğini gösterir denklem (5.26)'teki gibi hesaplanır. Yüksek duyarlılık değeri, gerçek pozitiflerin çoğunun doğru bir şekilde tespit edildiğini ve yanlış negatiflerin az olduğunu gösterir.

$$Duyarlılık = \frac{TP}{TP + FN} \tag{5.26}$$

F-Skoru, kesinlik ve duyarlılık metriklerinin harmonik ortalamasıdır (5.27). F1 Skoru hem modelin kesinlik hem de duyarlılık performansını dikkate alarak bir denge sağlar. Yüksek bir F1 Skoru hem yanlış pozitiflerin hem de yanlış negatiflerin düşük olduğunu ve modelin hem doğru pozitifleri hem de doğru negatifleri iyi bir şekilde sınıflandırdığını gösterir.

$$F\ Skoru = 2 * \frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \tag{5.27}$$

Örneğimiz için verilen metrikler; Doğruluk değeri 0.70, Kesinlik değeri 0.42, Duyarlılık değeri 0.6 ve F Skoru 0,5 olarak bulunur. Bulunan değerler şu şekilde yorumlanabilir:

Doğruluk için doğru sınıflandırma oranı %70'tir. Bu, modelin toplam veri setinin %70'ini doğru bir şekilde sınıflandırdığını gösterir.

Kesinlik için Pozitif olarak tahmin edilenlerin gerçekten pozitif olma oranı %42.86'dır. Bu, modelin pozitif olarak tahmin ettiği örneklerin yaklaşık yarısının gerçekte pozitif olduğunu ve diğer yarısının yanlış pozitif olduğunu gösterir.

Duyarlılık için Gerçek pozitiflerin doğru bir şekilde tespit edilme oranı %60'tır. Bu, modelin gerçek pozitiflerin %60'ını doğru bir şekilde tespit ettiğini, geri kalan %40'ını ise yanlış negatif olarak tahmin ettiğini gösterir.

F Skoru için Hassasiyet ve duyarlılığın harmonik ortalaması olan F Skoru, modelin dengeli bir performans sergilediğini gösterir. F1 Skoru 0,5 olduğuna göre, modelin hassasiyeti ve duyarlılığı dengeli bir şekilde performans gösteriyor ancak geliştirilmesi gerekebilir.

BÖLÜM 6

BULGULAR

6.1. Makine Öğrenimi Sonuçları

Bu bölümde oluşturulan 6 farklı saldırı senaryosu veri setinde seçilen algoritmalar ile ikili sınıflandırma sonuçları verilmiştir.

Veri kümelerinde sınıf dengesizliği sıkça görülmektedir. Bu durumda, modelin dengeli bir şekilde eğitilmesini sağlamak için örnekleme yöntemleri kullanılabilir.

Rastgele Alt Örnekleme (Random under sample) yöntemi, sınıf dengesizliği olan veri setlerinde kullanılan bir örnekleme yöntemidir. Bu yöntem, çoğunluk sınıfına ait örnekleri rastgele seçerek azaltır, böylece sınıf dengesizliğini azaltır ve daha dengeli bir veri seti elde edilir. DDoS ve DoS saldırılarını tahmin etmek için kullanılmıştır, çoğunluk sınıfı genellikle normal trafik örneklerinden oluşurken, azınlık sınıfı saldırı örneklerini içerir. Random under sample yöntemi, azınlık sınıfındaki saldırı örneklerinin sayısını azaltarak, modelin dengeli bir şekilde eğitilmesine yardımcı olur.

Öte yandan, SMOTE (Sentetik Azınlık Aşırı Örnekleme Tekniği) yöntemi, azınlık sınıfındaki örneklerin sayısını artırmak için sentetik veri noktaları oluşturur. Bu yöntem, azınlık sınıfındaki örnekleme eksikliğini giderir ve daha dengeli bir veri seti elde edilmesini sağlar. BruteForce, Botnet, Infiltration ve WebAttack veri kümelerinde kullanılmıştır, azınlık sınıfındaki saldırı örneklerinin sayısını artırmak için SMOTE yöntemi kullanılır. Bu da azınlık sınıfındaki nadir olayları daha iyi öğrenmek için model performansını artırabilir.

6.1.1. DDoS Saldırılarının Tespitinin Sınıflandırma Sonuçları

DDoS saldırısının tespiti için DDoS veri setine uygulanan algoritmaların başarı metriklerinin değerleri Çizelge 6.1'de gösterilmektedir. Kullanılan algoritmaya göre oluşturulan karmaşıklık matrisleri de Şekil 6.1'de bulunmaktadır.

Çizelge 6.1. DDoS sınıflandırma ölçütleri değerlendirmesi

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F Skoru
NB	0.90597	0.73873	0.52274	0.61225
KNN	0.96752	0.99999	0.77129	0.87087
RF	0.98057	0.99325	0.86908	0.92703
DT	0.95896	0.77666	0.99798	0.87352
ADA	0.95986	0.78183	0.99508	0.87566
MLP	0.90106	0.62235	0.77129	0.68886
LDA	0.92494	0.72006	0.77128	0.74479
LR	0.90311	0.63000	0.76981	0.69292
GB	0.99807	0.99945	0.98696	0.99317

Doğruluk: Tüm algoritmalar oldukça yüksek doğruluk değerlerine sahiptir. En yüksek doğruluk skorunu GB algoritması elde etmiştir %99.81, en düşük değer ise MLP algoritmasında %90.11 görülmüştür.

Kesinlik: Kesinlik, algoritmaların saldırı olarak tespit ettiği örneklerin ne kadarının gerçekten saldırı olduğunu gösterir. Bu metrik, FP (Yanlış Pozitifler) üzerinde odaklanır. Bu tabloda en yüksek kesinlik skoru KNN algoritmasında %99.99, en düşük ise MLP algoritmasında %62.24 gözlemlenmiştir.

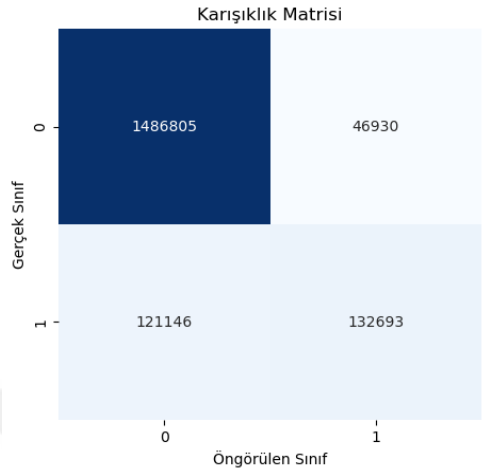
Duyarlılık: Duyarlılık, gerçekte saldırı olan örneklerin ne kadarının doğru bir şekilde tespit edildiğini gösterir. Bu metrik, FN (Yanlış Negatifler) üzerinde odaklanır. En yüksek duyarlılık skoru DT ve ADA algoritmalarında %99.80 ve %99.51, en düşük ise MLP ve LDA algoritmalarında gözlemlenmiştir.

F Skoru: F skoru, hassasiyet ve duyarlılığı dengeler. En yüksek F skorunu GB algoritması elde etmiştir %99.32, en düşük değer ise NB algoritmasında %61.22 görülmüştür.

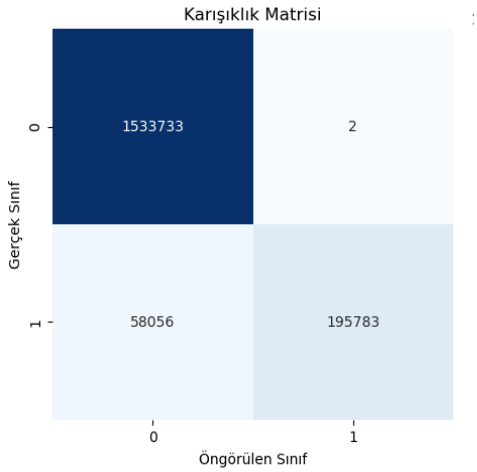
Verilen metriklerle değerlendirildiğinde GB algoritmasının en başarılı ve MLP algoritmasının en başarısız olduğu söylenebilir. Bu değerlendirme, her algoritmanın

belirli bir veri setinde farklı performans gösterebileceğini ve doğru algoritmanın seçilmesinin uygulamanın gereksinimlerine bağlı olduğunu göstermektedir.

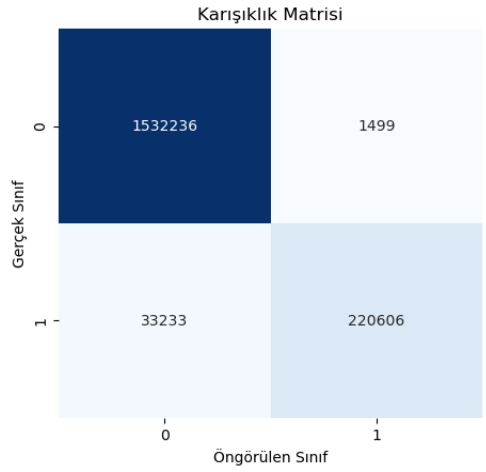
NB



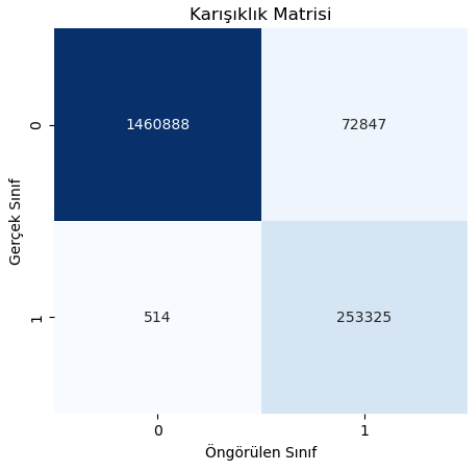
KNN



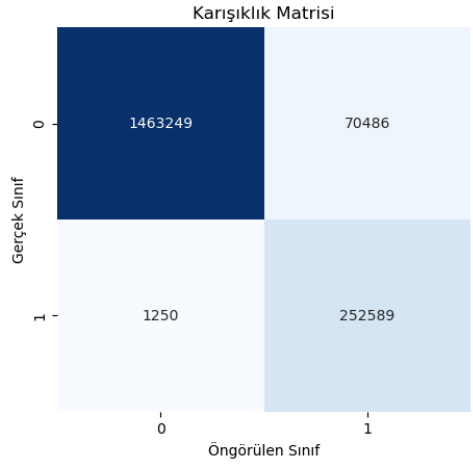
RF



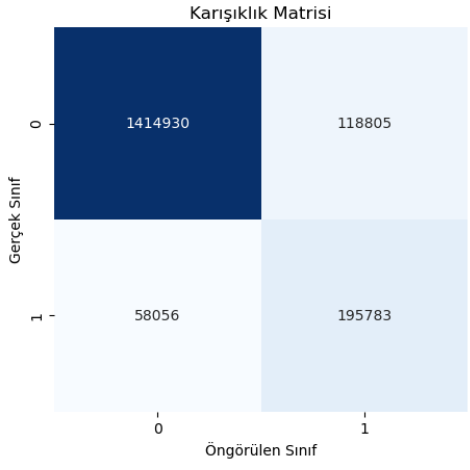
DT



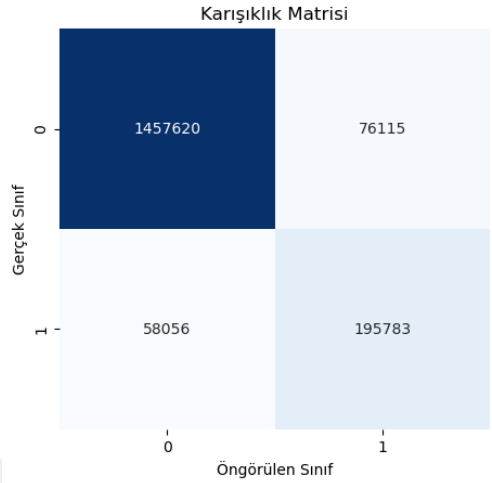
ADA



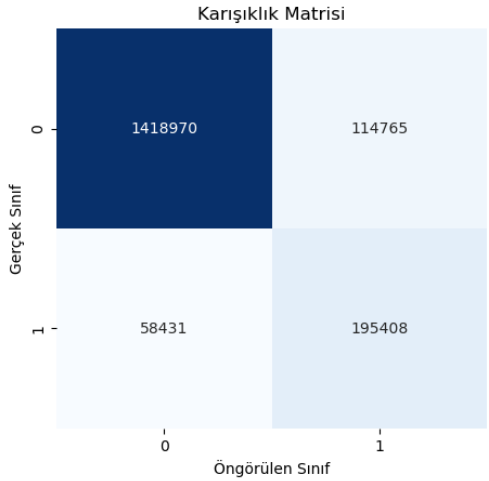
MLP



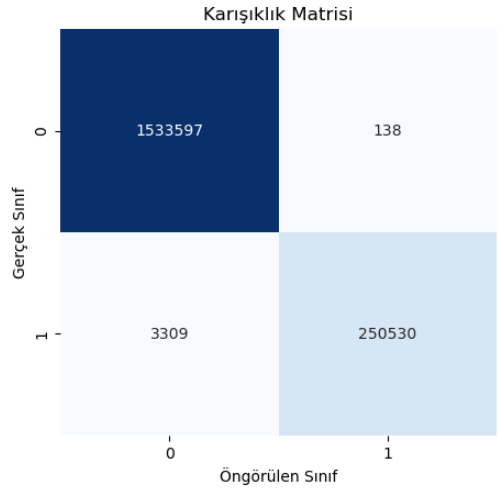
LDA



LR



GB



Şekil 6.1. DDoS veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları

6.1.2. DoS Saldırılarının Tespitinin Sınıflandırma Sonuçları

DoS saldırısının tespiti için DoS veri setine uygulanan algoritmaların başarı metriklerinin değerleri Çizelge 6.2'de sunulmuştur. Kullanılan algoritmaya göre oluşturulan karmaşıklık matrisleri ise Şekil 6.2'de bulunmaktadır.

Çizelge 6.2. DoS sınıflandırma ölçütleri değerlendirmesi

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F Skoru
NB	0.69588	0.50765	0.98306	0.66955
KNN	0.99725	0.99421	0.99704	0.99565
RF	0.94273	0.91049	0.90637	0.90843
DT	0.97721	0.95269	0.97575	0.96408
ADA	0.99898	0.99892	0.99782	0.99837
MLP	0.80134	0.99958	0.36627	0.53611
LDA	0.98941	0.97353	0.99321	0.98327
LR	0.95442	0.92872	0.92491	0.92681
GB	0.99881	0.99835	0.99782	0.99808

Doğruluk: Tüm algoritmalar yüksek doğruluk değerlerine sahiptir. En yüksek doğruluk skoru ADA ve GB algoritmalarında %99.89 ve %99.88, en düşük değer ise NB algoritmasında %69.59 gözlemlenmiştir.

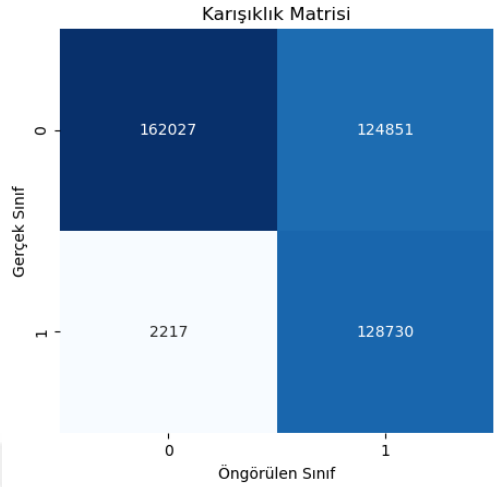
Kesinlik: Kesinlik, algoritmaların DoS saldırısı olarak tespit ettiği örneklerin ne kadarının gerçekten DoS saldırısı olduğunu gösterir. Bu tabloda en yüksek kesinlik skoru MLP algoritmasında %99.95, en düşük ise NB algoritmasında %50.76 görülmüştür.

Duyarlılık: Duyarlılık, gerçekte DoS saldırısı olan örneklerin ne kadarının doğru bir şekilde tespit edildiğini gösterir. Bu metrikte bir çok yüksek değer vardır, en düşük değer ise MLP algoritmasında %36.27 elde edilmiştir.

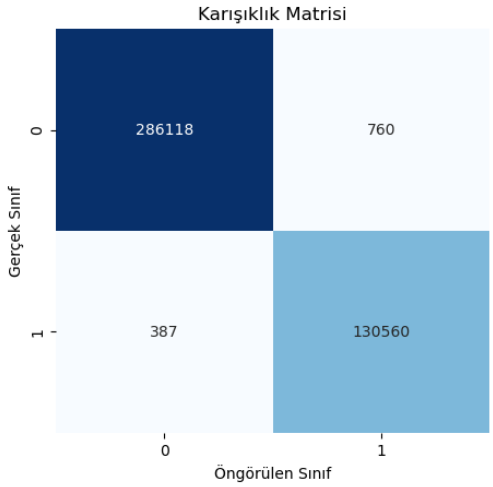
F Skoru: F skoru, hassasiyet ve duyarlılığı dengeler. En yüksek F skoru ADA algoritmasında %99.84, en düşük değer ise MLP algoritmasında %53.61 gözlemlenmiştir.

Bu değerlendirmeye göre, ADA algoritması en başarılı olarak değerlendirilebilir. En yüksek doğruluk, kesinlik, duyarlılık ve F skoru değerlerine sahiptir. En başarısız algoritma ise MLP olarak görünmektedir. MLP'nin diğer algoritmalara kıyasla daha düşük değerlere sahip olduğu gözlemlenmiştir.

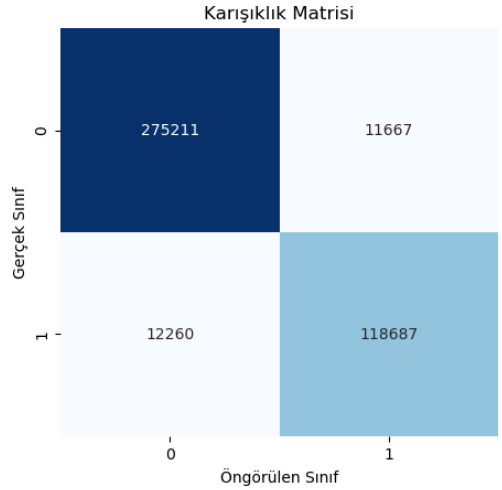
NB



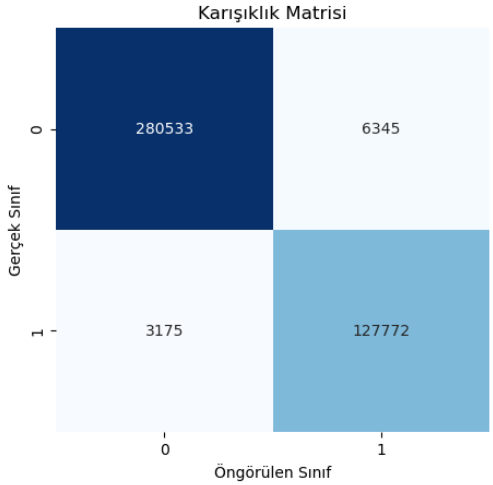
KNN



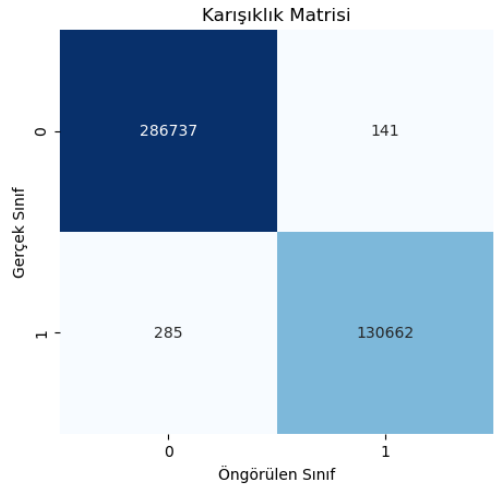
RF



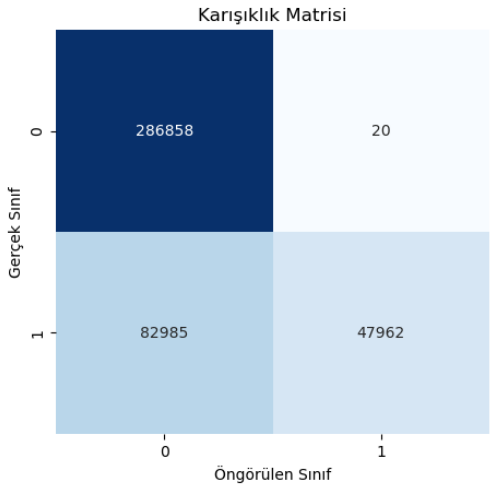
DT

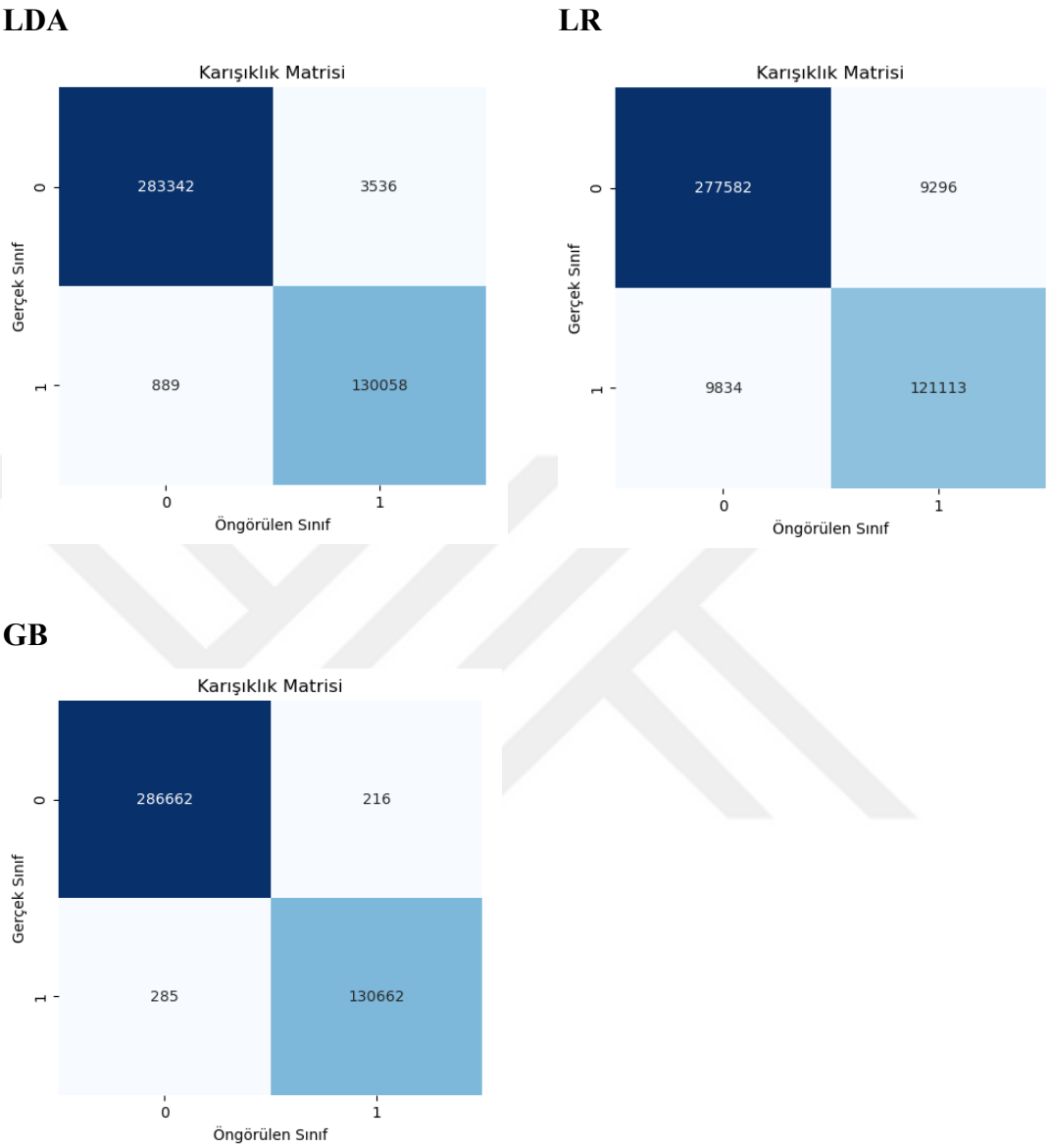


ADA



MLP





Şekil 6.2. DoS veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları

6.1.3. Brute Force Saldırılarının Tespitinin Sınıflandırma Sonuçları

Brute Force saldırısının tespiti için Brute Force veri setine uygulanan algoritmaların başarı metriklerinin değerleri Çizelge 6.3'te gösterilmektedir. Kullanılan algoritmaya göre oluşturulan karmaşıklık matrisleri de Şekil 6.3'te bulunmaktadır.

Çizelge 6.3. Brute Force sınıflandırma ölçütleri değerlendirmesi

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F Skoru
NB	0.86467	0.72936	0.99992	0.84249
KNN	0.99941	0.99857	0.99979	0.99918
RF	0.99957	0.99886	0.99997	0.99942
DT	0.99945	0.99927	0.99923	0.99925
ADA	0.99954	0.99877	0.99999	0.99938
MLP	0.99963	0.99902	0.99997	0.99950
LDA	0.99645	0.99036	1.00000	0.99516
LR	0.99786	0.99417	1.00000	0.99708
GB	0.99955	0.99885	0.99992	0.99938

Doğruluk: Tüm algoritmalar son derece yüksek doğruluk değerlerine sahiptir. En düşük değer NB algoritmasında %86.47 gözlemlenmiştir.

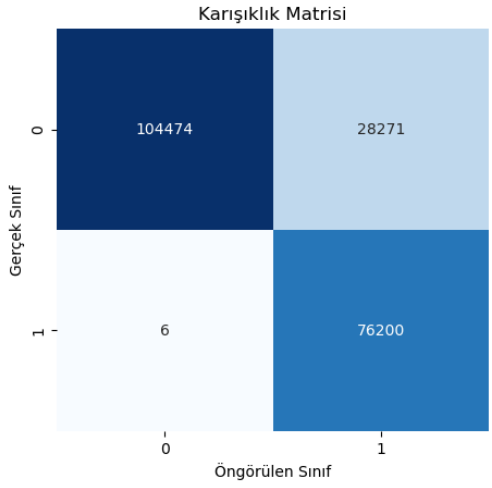
Kesinlik: Tüm algoritmalar yüksek kesinlik değerlerine sahiptir. En düşük değer ise NB algoritmasında (%72.94) gözlemlenmiştir.

Duyarlılık: Duyarlılık oranları genel anlamda oldukça yüksektir.

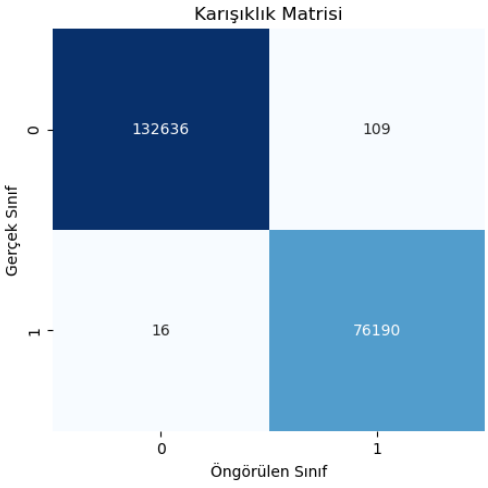
F Skoru: F skoru, hassasiyet ve duyarlılığı dengeler. Tüm algoritmalar yüksek F skorlarına sahiptir. En düşük değer ise NB algoritmasında %84.25 görülmüştür.

Bu sonuçlara göre, tüm algoritmaların sınıflandırma performansı oldukça yüksektir. NB algoritması aralarında en düşük performansı sergilemektedir.

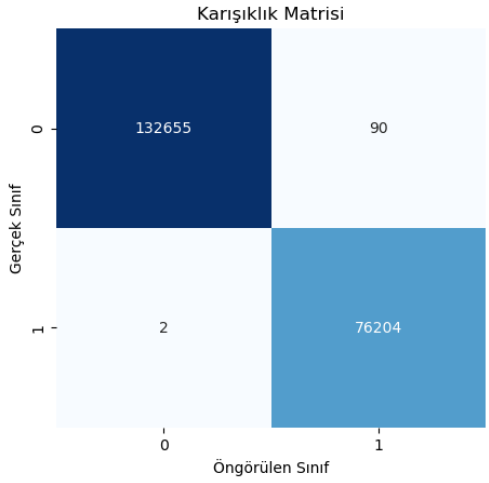
NB



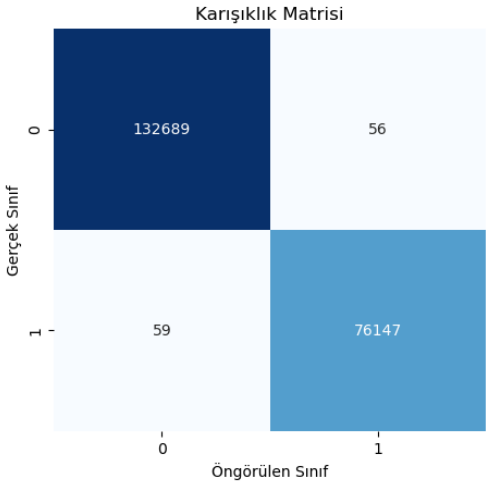
KNN



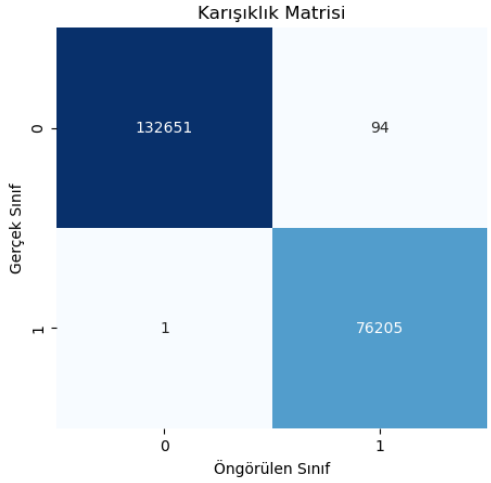
RF



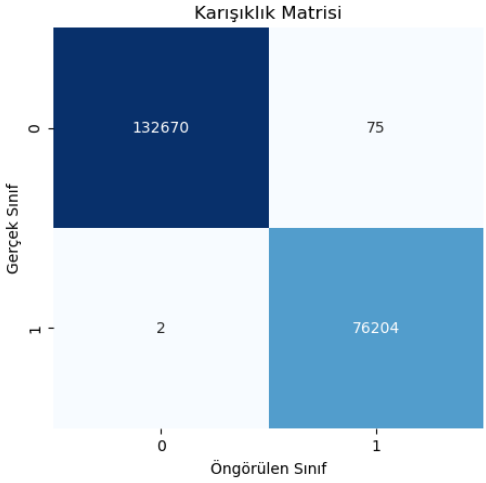
DT



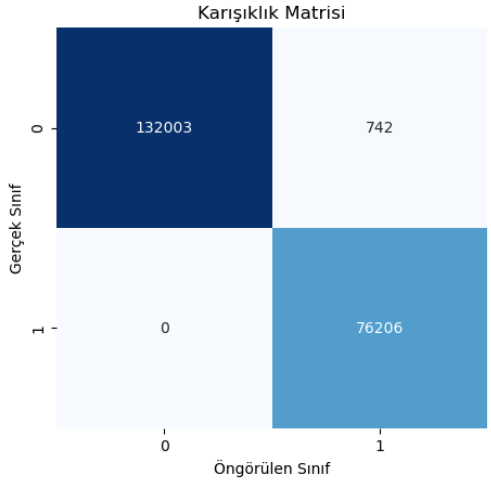
ADA



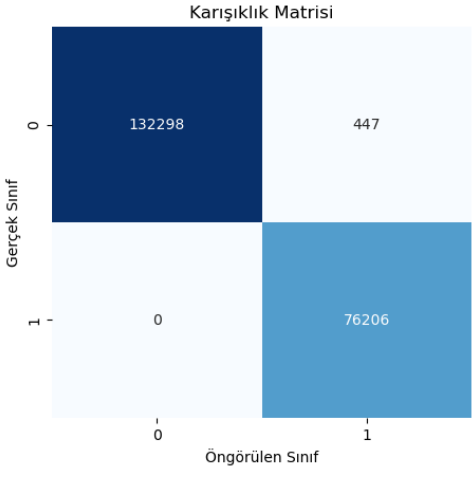
MLP



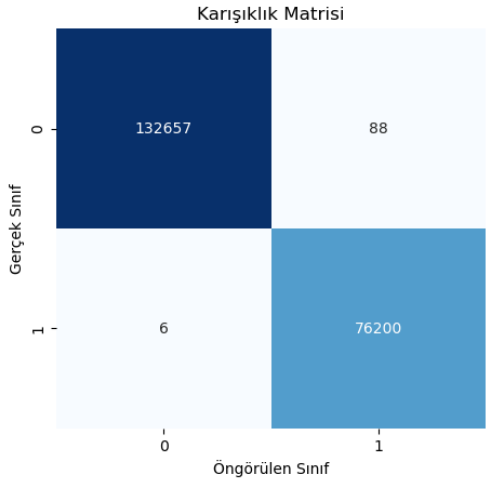
LDA



LR



GB



Şekil 6.3. Brute Force veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları

6.1.4. Botnet Saldırılarının Tespitinin Sınıflandırma Sonuçları

Botnet saldırısının tespiti için Botnet veri setine uygulanan algoritmaların başarı metriklerinin değerleri Çizelge 6.4'te gösterilmektedir. Kullanılan algoritmaya göre oluşturulan karmaşıklık matrisleri de Şekil 6.4'te bulunmaktadır.

Çizelge 6.4. Botnet sınıflandırma ölçütleri değerlendirmesi

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F Skoru
NB	0.76395	0.53567	0.99963	0.69755
KNN	0.99776	0.99370	0.99813	0.99591
RF	0.99837	0.99517	0.99886	0.99702
DT	0.99329	0.98359	0.99192	0.98774
ADA	0.98766	0.96318	0.99261	0.97767
MLP	0.99183	0.98306	0.98701	0.98503
LDA	0.86286	0.97984	0.50682	0.66807
LR	0.89483	0.72148	0.99973	0.83811
GB	0.99714	0.99302	0.99650	0.99476

Doğruluk: Tüm algoritmalar yüksek doğruluk değerlerine sahiptir. En yüksek doğruluk skoru RF algoritmasında %99.84, en düşük değer ise NB algoritmasında %76.39 gözlemlenmiştir.

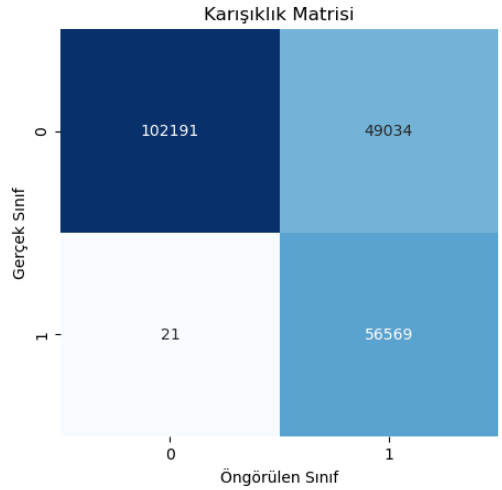
Kesinlik: Kesinlik, algoritmaların botnet saldırısı olarak tespit ettiği örneklerin ne kadarının gerçekten botnet saldırısı olduğunu gösterir. En yüksek kesinlik skoru RF algoritmasında (%99.51), en düşük ise NB algoritmasında (%53.57) gözlemlenmiştir.

Duyarlılık: Duyarlılık, gerçekte Botnet saldırısı olan örneklerin ne kadarının doğru bir şekilde tespit edildiğini gösterir. En yüksek duyarlılık skoru LR algoritmasında %99.97, en düşük ise LDA algoritmasında %50.68 elde edilmiştir.

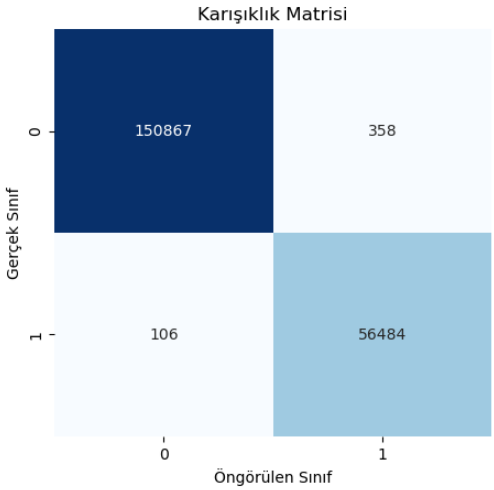
F Skoru: F skoru, hassasiyet ve duyarlılığı dengeler. En yüksek F skoru RF algoritmasında %99.70, en düşük ise LDA algoritmasında %66.81 görülmüştür.

Bu değerlendirmeye göre, RF algoritması en başarılı olarak değerlendirilebilir. En yüksek doğruluk, kesinlik ve F skoru değerlerine sahiptir. Diğer yandan, LDA algoritması en düşük performansı sergilemektedir. LDA'nın düşük duyarlılık ve F skoru değerleri, bu algoritmanın Botnet saldırılarını sınıflandırma konusunda diğerlerine kıyasla daha zayıf olduğunu göstermektedir.

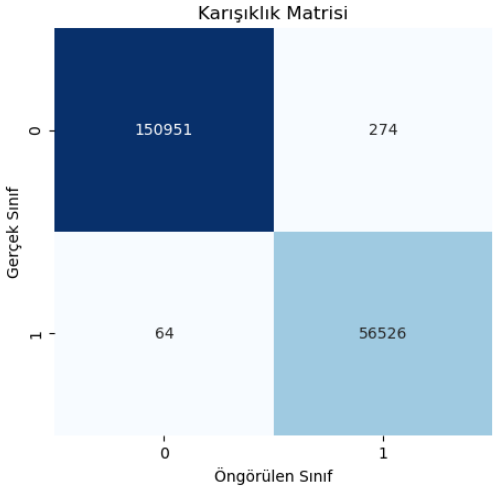
NB



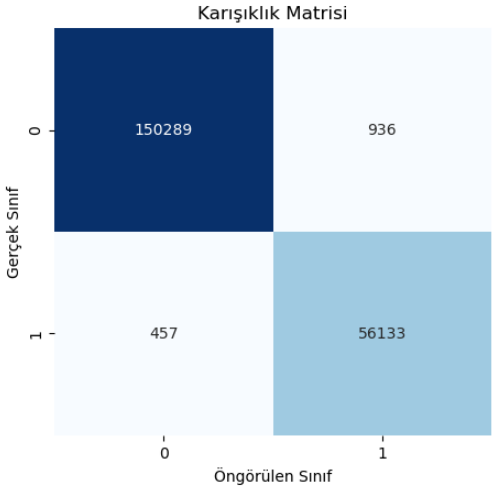
KNN



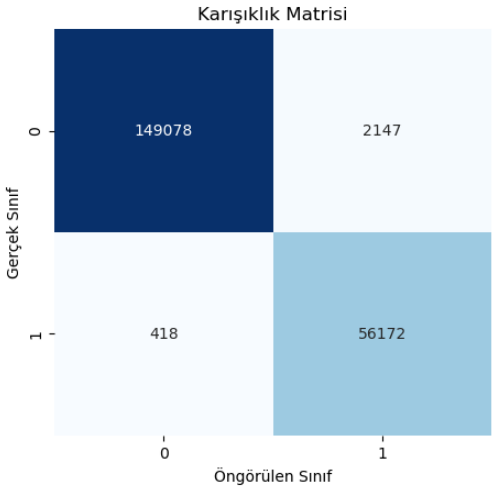
RF



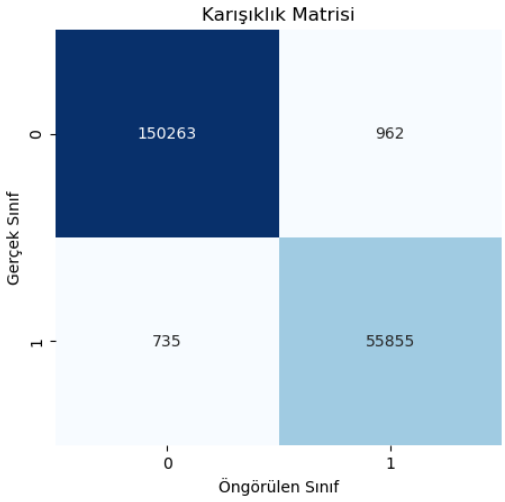
DT



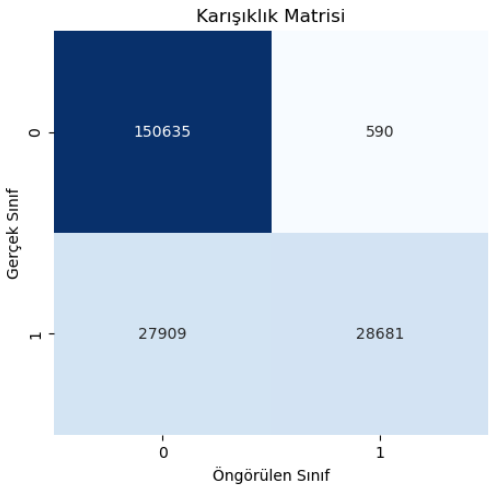
ADA



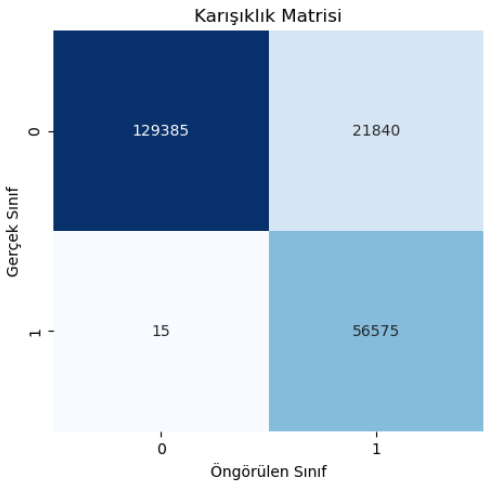
MLP



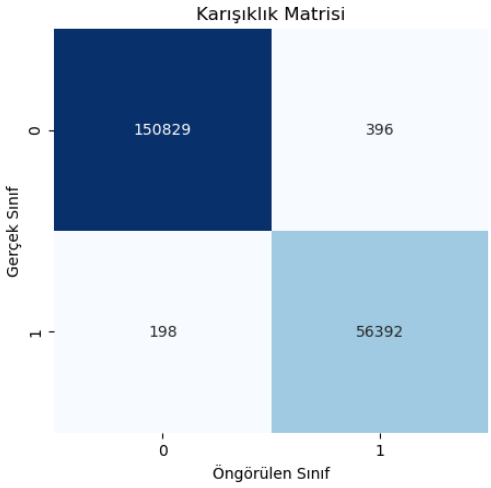
LDA



LR



GB



Şekil 6.4. Botnet veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları

6.1.5. Sızma Saldırılarının Tespitinin Sınıflandırma Sonuçları

Sızma saldırılarının tespiti için Infilteration veri setine uygulanan algoritmaların başarı metriklerinin değerleri Çizelge 6.5'te gösterilmektedir. Kullanılan algoritmaya göre oluşturulan karmaşıklık matrisleri de Şekil 6.5'te bulunmaktadır.

Çizelge 6.5. Sızma saldırılarının sınıflandırma ölçütleri değerlendirmesi

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F Skoru
NB	0.80454	0.25992	0.07788	0.11985
KNN	0.79458	0.35399	0.24505	0.28961
RF	0.76662	0.22419	0.14867	0.17878
DT	0.76497	0.22442	0.15286	0.18185
ADA	0.82910	0.49065	0.00328	0.00652
MLP	0.82874	0.47417	0.02068	0.03964
LDA	0.82864	0.41818	0.00719	0.01415
LR	0.82858	0.41707	0.00794	0.01559
GB	0.80047	0.35318	0.20168	0.25675

Doğruluk: Tüm algoritmaların doğruluk değerleri diğer saldırılara göre düşüktür. En yüksek doğruluk skoru ADA algoritmasında (%82.91), en düşük değer ise DT algoritmasında (%76.49) gözlemlenmiştir.

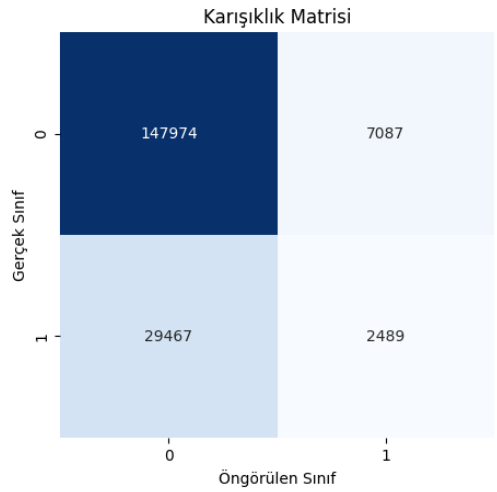
Kesinlik: Kesinlik, algoritmaların sızma saldırısı olarak tespit ettiği örneklerin ne kadarının gerçekten sızma saldırısı olduğunu gösterir. En yüksek kesinlik skoru ADA algoritmasında %49.07, en düşük ise RF algoritmasında %22.42 gözlemlenmiştir.

Duyarlılık: Duyarlılık, gerçekte sızma saldırısı olan örneklerin ne kadarının doğru bir şekilde tespit edildiğini gösterir. Tüm algoritmaların duyarlılık değerleri oldukça düşüktür.

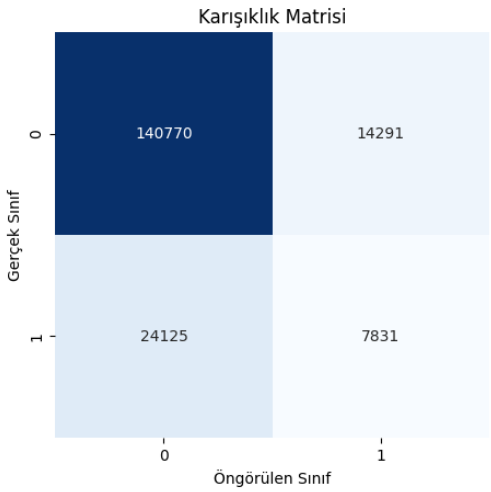
F Skoru: F skoru, hassasiyet ve duyarlılığı dengeler. En düşük F skoru ADA algoritmasında, en yüksek ise KNN algoritmasında görülmüştür.

Bu sonuçlara göre, sızma saldırılarının sınıflandırma performansı oldukça zayıf görünmektedir. Tüm algoritmaların düşük doğruluk, kesinlik, duyarlılık ve F skoru değerleri, bu algoritmaların sızma saldırılarının tespiti konusunda yetersiz olduğunu göstermektedir. Bu durum, veri setinin özelliklerine, dağılımına ve algoritma seçimine dikkat edilmesi gerektiğini göstermektedir. Belki de daha karmaşık veya özelleştirilmiş bir model kullanmak, sızma saldırılarının tespiti konusunda daha iyi sonuçlar verebilir

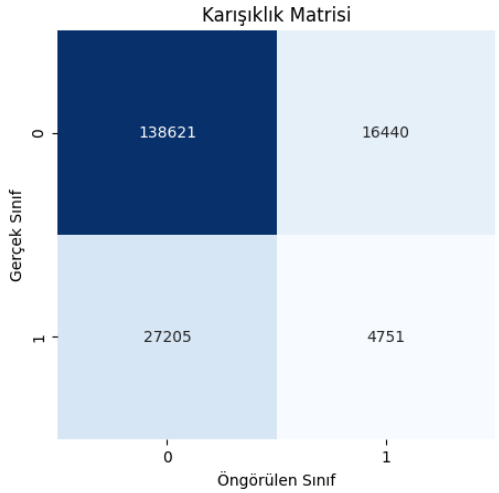
NB



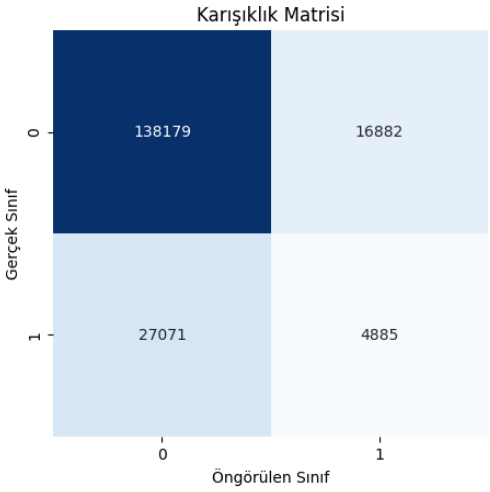
KNN



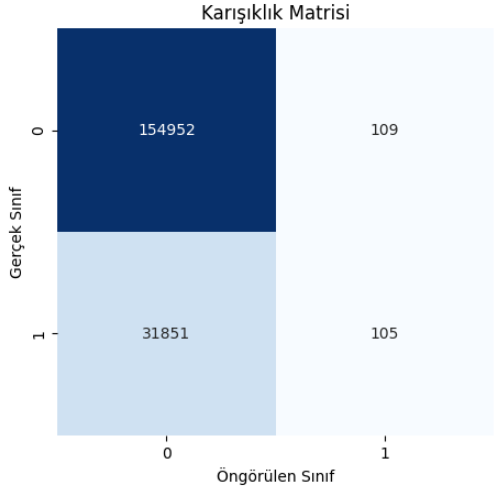
RF



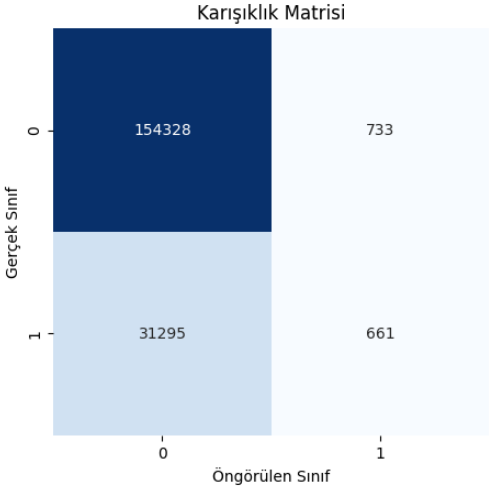
DT



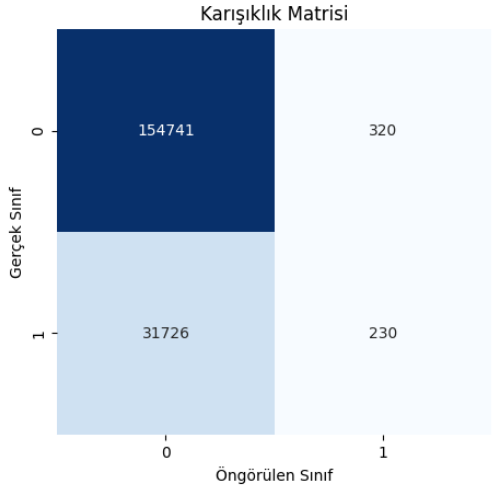
ADA



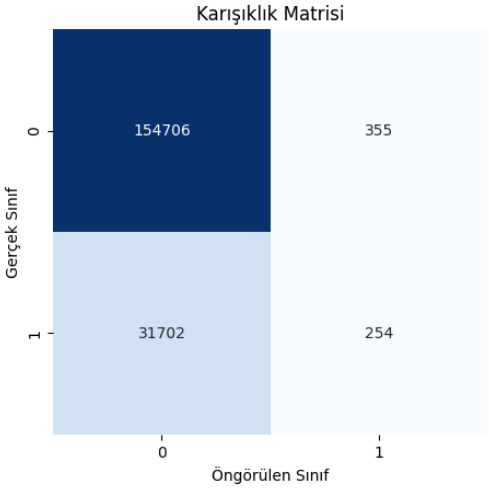
MLP



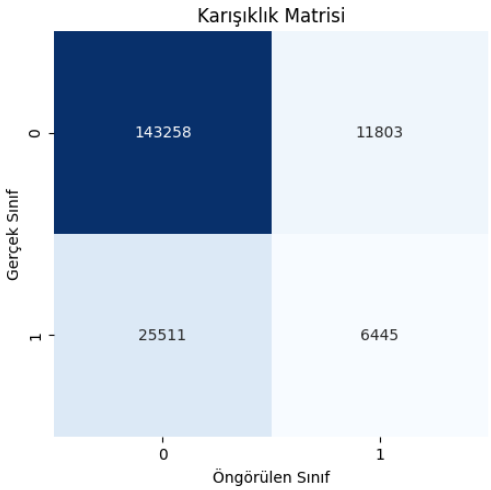
LDA



LR



GB



Şekil 6.5. Infilteration veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları

6.1.6. Web Attack Saldırılarının Tespitinin Sınıflandırma Sonuçları

Web saldırılarının tespiti için Web Attack veri setine uygulanan algoritmaların başarı metriklerinin değerleri Çizelge 6.6'da gösterilmektedir. Kullanılan algoritmalara göre oluşturulan karmaşıklık matrisleri de Şekil 6.6'da verilmiştir.

Çizelge 6.6. Web Attack sınıflandırma ölçütleri değerlendirmesi

Algoritma	Doğruluk	Kesinlik	Duyarlılık	F Skoru
NB	0.99305	0.02749	0.43169	0.05170
KNN	0.99498	0.05206	0.60655	0.09589
RF	0.99975	0.88571	0.50819	0.64583
DT	0.96711	0.01034	0.78142	0.02042
ADA	0.94508	0.00642	0.80874	0.01275
MLP	0.97028	0.00381	0.25683	0.00752
LDA	0.98402	0.01322	0.48087	0.02573
LR	0.98934	0.01852	0.44808	0.03558
GB	0.84167	0.00258	0.93442	0.00515

Doğruluk: Tüm algoritmaların doğruluk değerleri yüksektir. En yüksek doğruluk skoru RF algoritmasında %99.98, en düşük değer ise GB algoritmasında %84.17 gözlemlenmiştir.

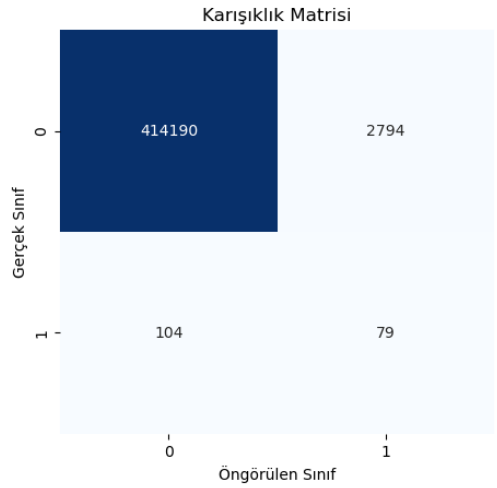
Kesinlik: Kesinlik, algoritmaların web saldırısı olarak tespit ettiği örneklerin ne kadarının gerçekten web saldırısı olduğunu gösterir. En yüksek kesinlik skoru RF algoritmasında %88.57, en düşük ise MLP algoritmasında gözlemlenmiştir.

Duyarlılık: Duyarlılık, gerçekte web saldırısı olan örneklerin ne kadarının doğru bir şekilde tespit edildiğini gösterir. Tüm algoritmaların duyarlılık değerleri düşüktür. En yüksek duyarlılık skoru GB algoritmasında %93.44, en düşük ise MLP algoritmasında %25.68 elde edilmiştir.

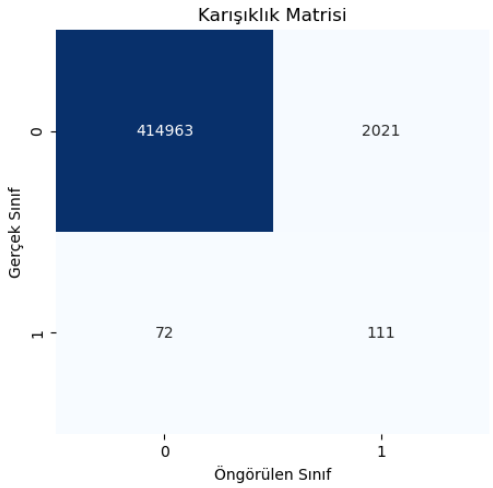
F Skoru: F skoru, hassasiyet ve duyarlılığı dengeler. En yüksek F skoru RF algoritmasında %64.58, en düşük ise MLP algoritmasında görülmüştür.

Bu sonuçlara göre, RF algoritması en başarılı olarak değerlendirilebilir. Yüksek doğruluk, kesinlik ve F skoru değerlerine sahiptir. Diğer yandan, MLP algoritması en düşük performansı sergilemektedir. MLP'nin düşük kesinlik, duyarlılık ve F skoru değerleri, bu algoritmanın web saldırısı sınıflandırması konusunda diğerlerine kıyasla daha zayıf olduğunu göstermektedir.

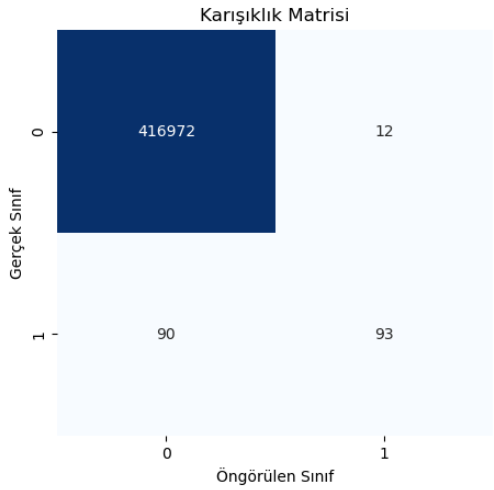
NB



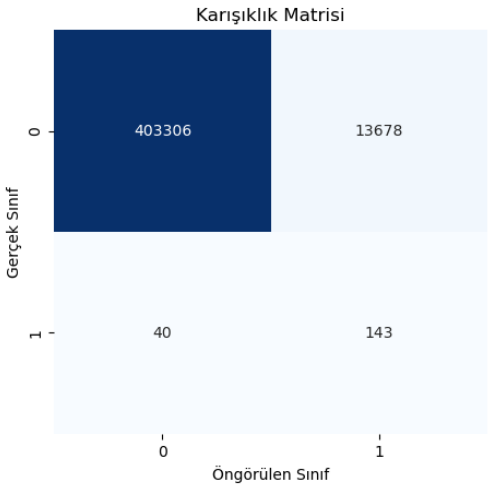
KNN



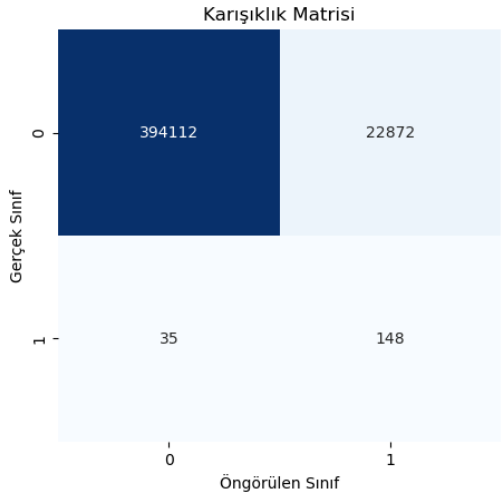
RF



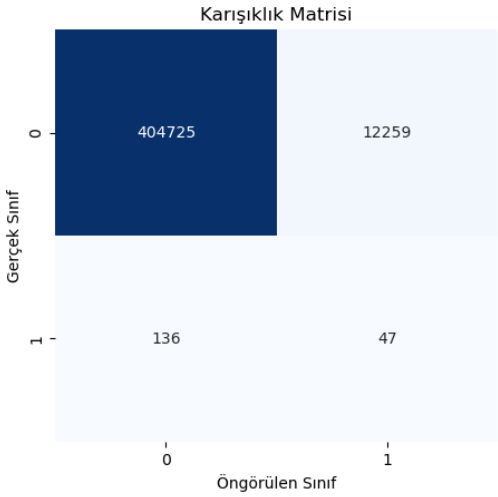
DT



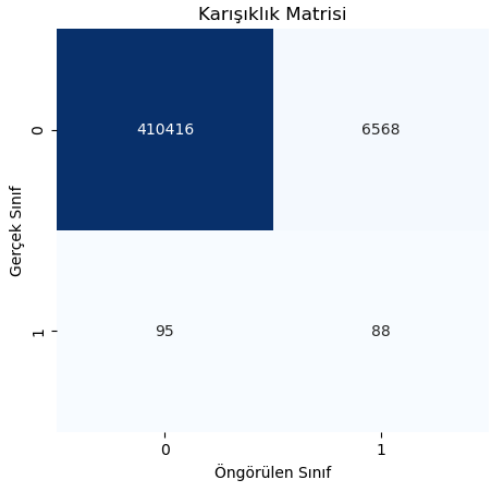
ADA



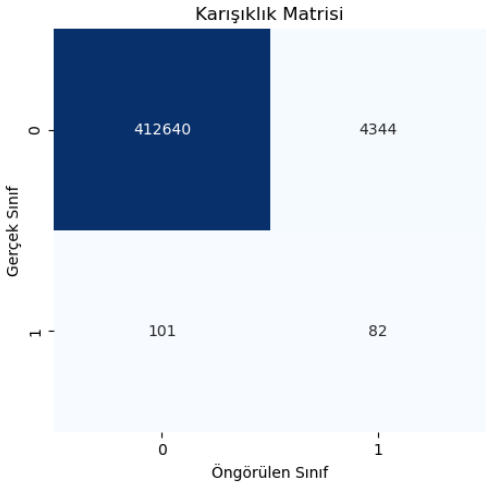
MLP



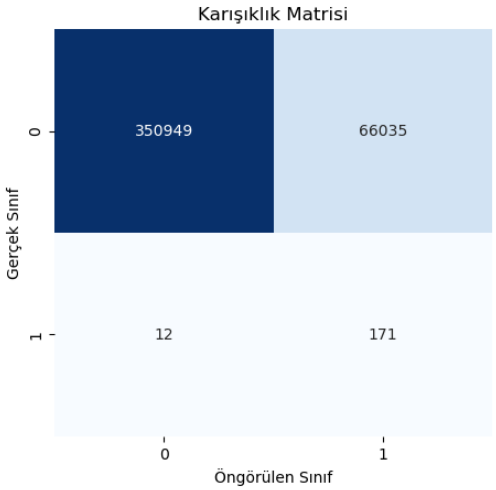
LDA



LR



GB

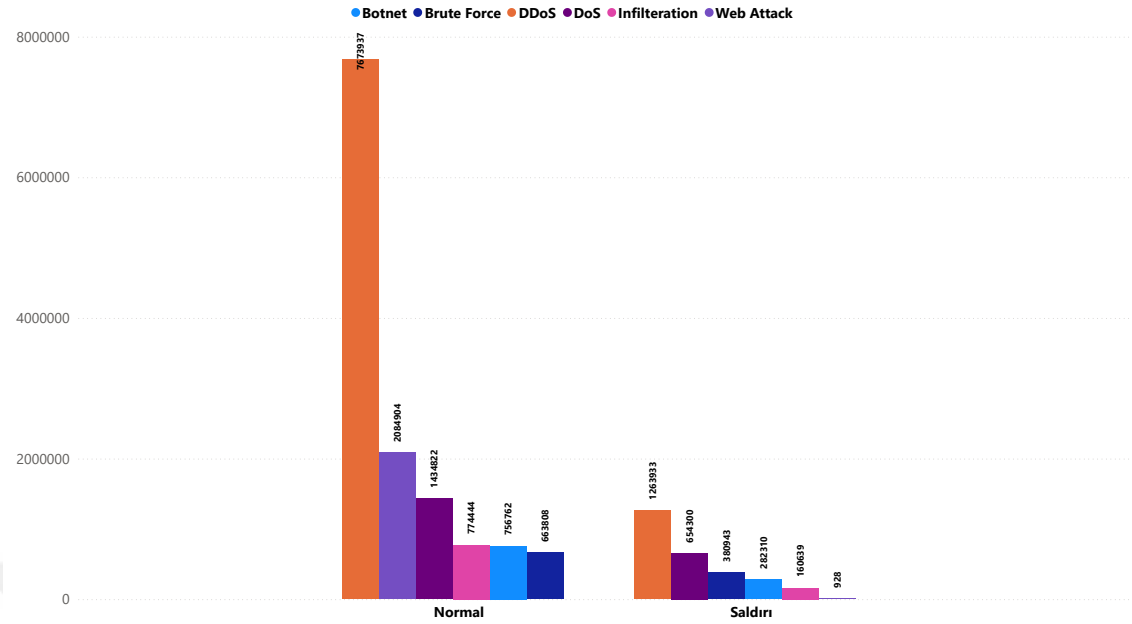


Şekil 6.6. Web Attack veri setinde seçilen algoritmaya göre karmaşıklık matrisi sonuçları

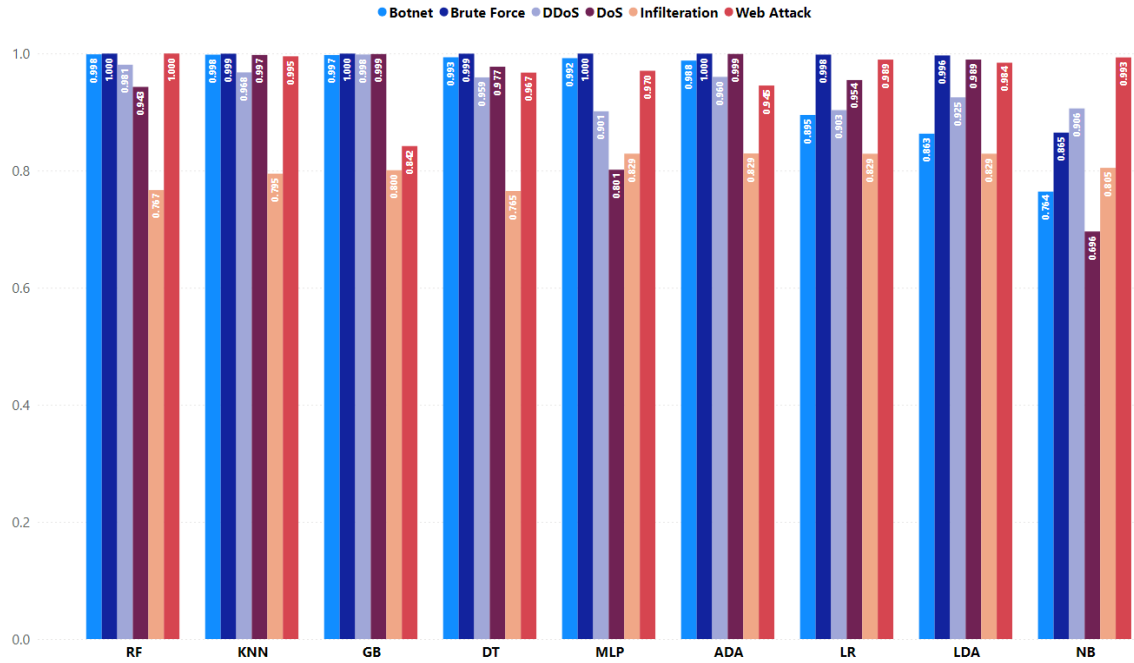
6.2. Verilerin ve Bulguların Demografik Özellikleri ve Karşılaştırılması

Bu bölümde, makine öğrenimi algoritmalarının çeşitli saldırı türlerine karşı performanslarını değerlendirmek amacıyla elde edilen metriklerin görselleştirilmesi yapılmıştır. Bu değerlendirme, saldırı türlerine göre algoritmaların etkinliğini daha iyi anlamamıza yardımcı olacaktır. Şekil 6.7'de, veri setinde normal ve saldırı olarak etiketlenmiş verilerin dağılımı ayrıntılı bir şekilde gösterilmektedir. Bu görselleştirme, farklı saldırı türlerinin veri setindeki temsil oranlarını net bir biçimde ortaya koymaktadır.

Bunun yanı sıra, kullanılan makine öğrenimi algoritmalarının çeşitli saldırı türlerine karşı sağladıkları doğruluk metrikleri de analiz edilmiştir. Şekil 6.8'de bu algoritmaların her bir saldırı türüne karşı elde ettikleri doğruluk oranları ayrıntılı olarak sunulmuştur. Bu doğruluk metrikleri, hangi algoritmaların hangi saldırı türlerine karşı daha etkili olduğunu belirlememize olanak tanımaktadır.

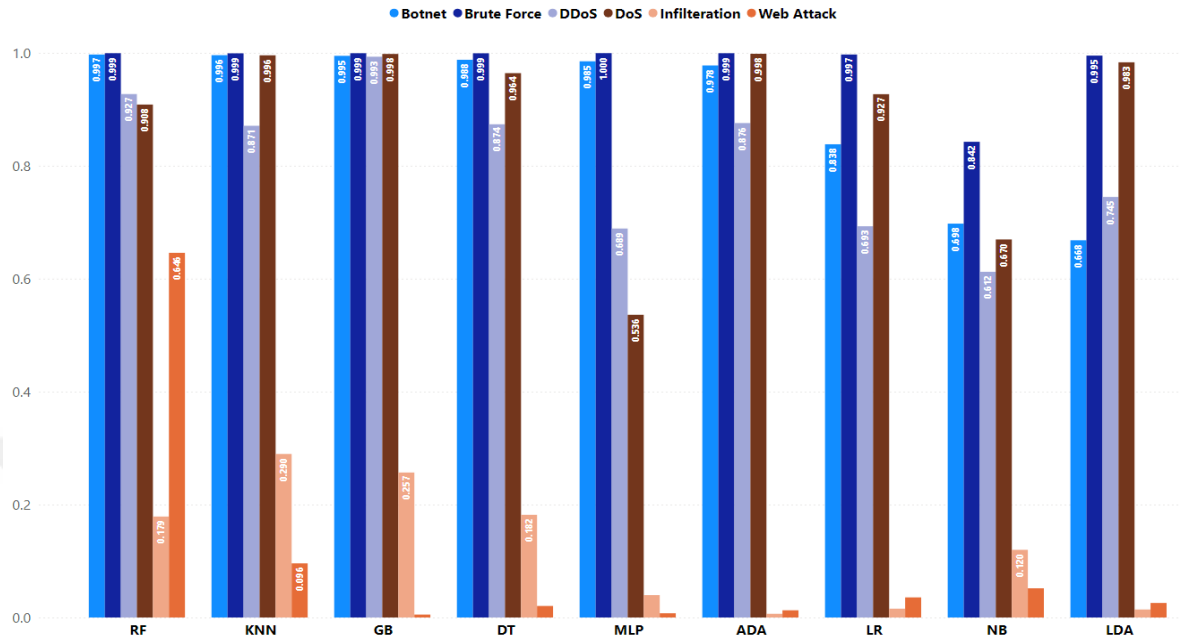


Şekil 6.7. Saldırı ve Normal verilerinin toplamının Saldırılara göre dağılımı



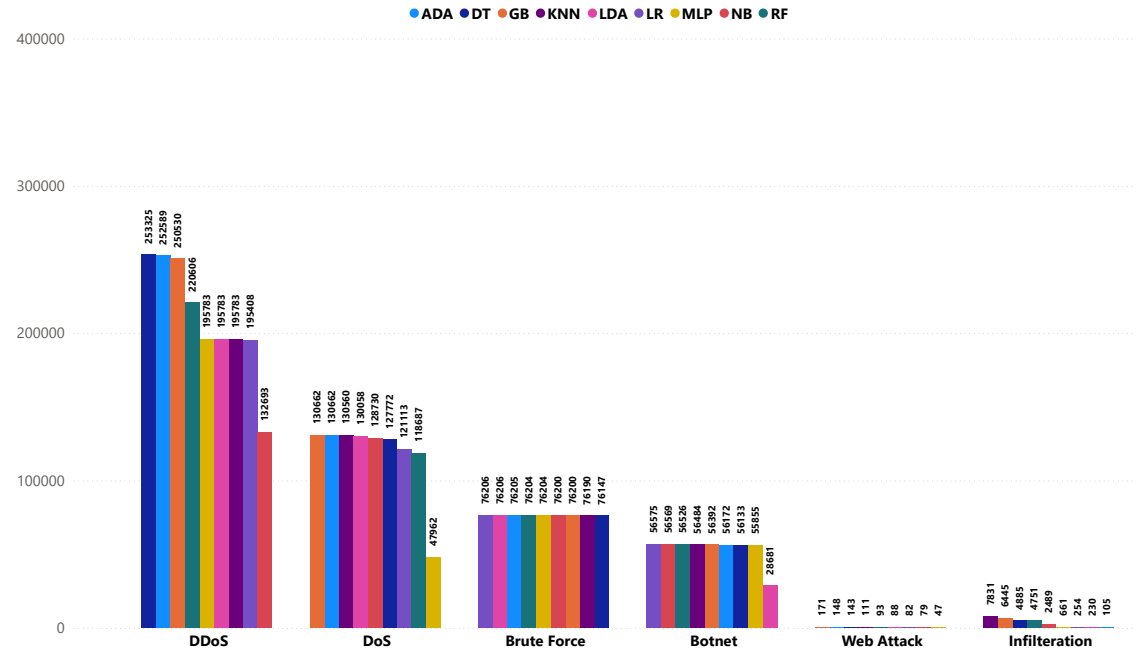
Şekil 6.8. Algoritmalara göre Doğruluk oranları

Algoritmaların saldırı çeşitlerine göre elde ettiği F skoru metriğinin değerleri Şekil 6.9'da gösterilmektedir.

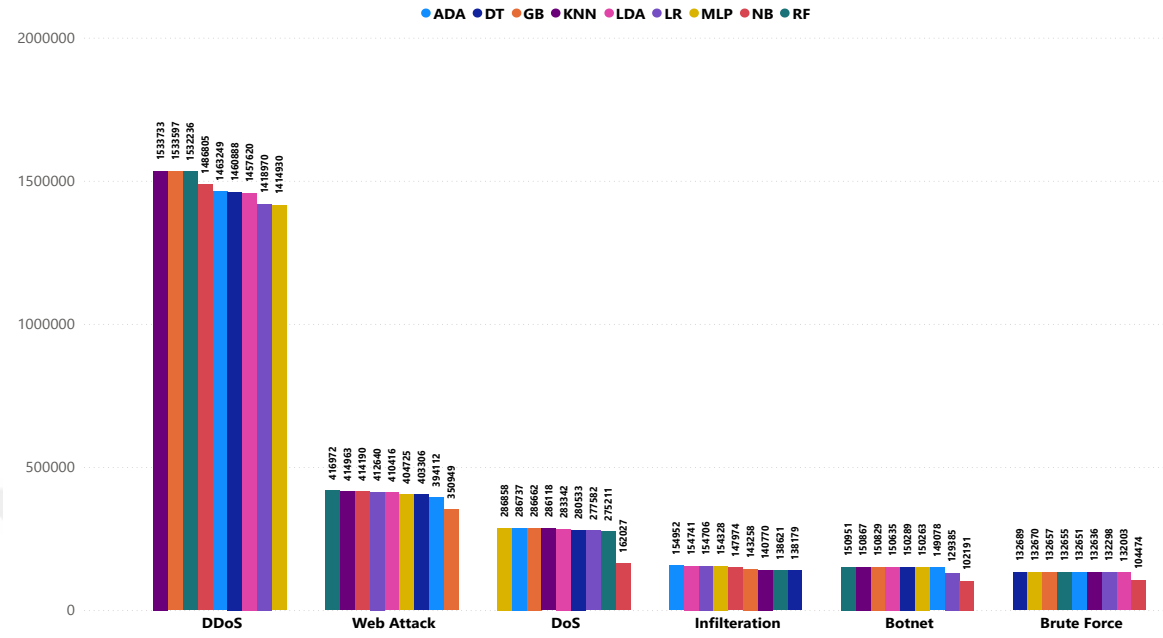


Şekil 6.9. Algoritmalara göre F Skoru oranları

Saldırı çeşitlerine göre algoritmaların karmaşıklık matrislerinde bulunan TP, TN değerleri sırasıyla Şekil 6.10, Şekil 6.11'de verilmiştir.

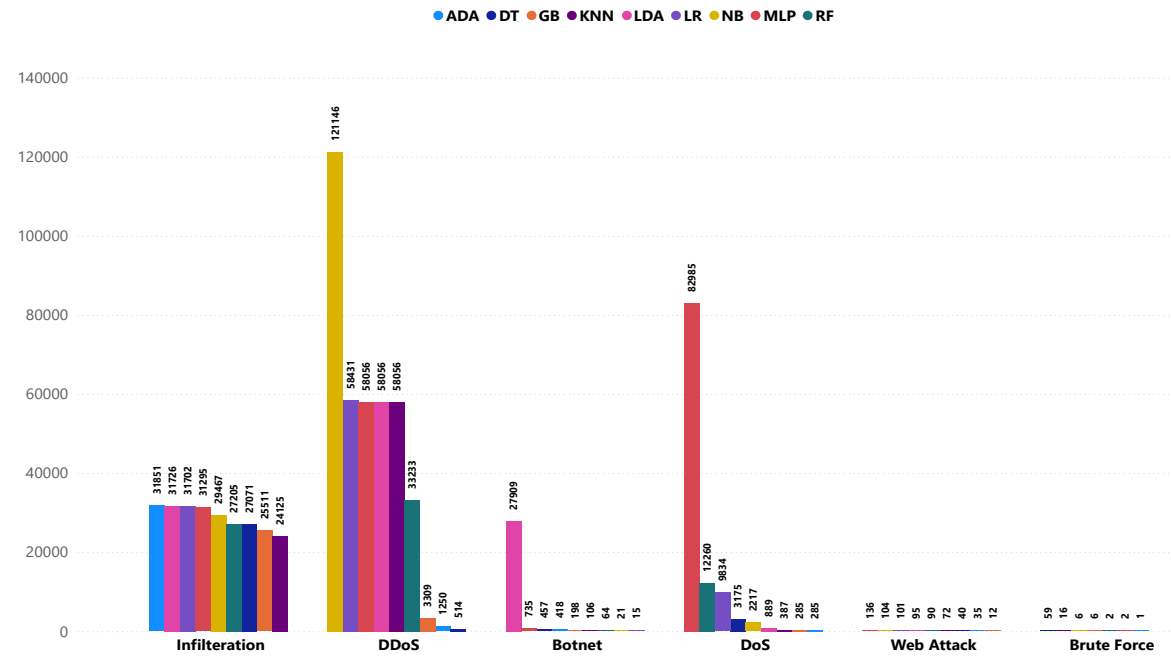


Şekil 6.10. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi TP değerleri

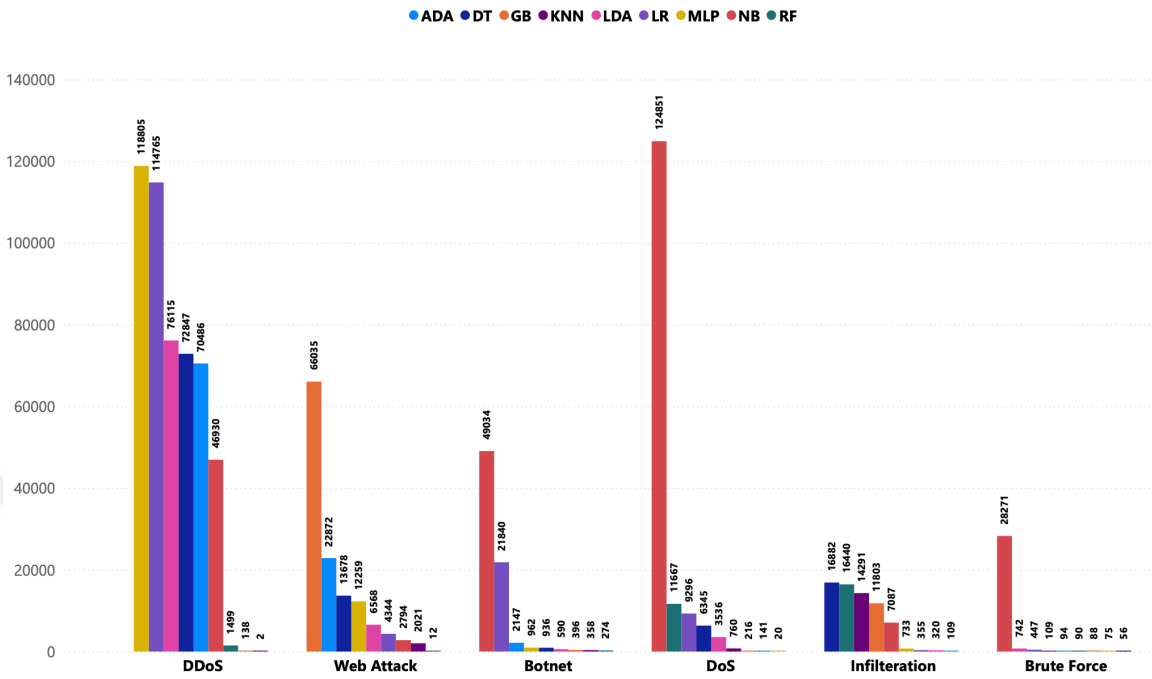


Şekil 6.11. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi TN değerleri

Saldırı çeşitlerine göre algoritmaların karmaşıklık matrislerinde bulunan FP ve FN değerleri sırasıyla Şekil 6.12 ve Şekil 6.13'te verilmiştir.



Şekil 6.12. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi FP değerleri



Şekil 6.13. Saldırılara ve Algoritmalara göre Karmaşıklık Matrisi FN değerleri

BÖLÜM 7

TARTIŞMA VE SONUÇ

Saldırı Tespit Sistemlerinin önemli görevleri arasında ağları düzenli olarak denetlemek, saldırıları tanımak, bu saldırılarla ilgili kayıtlar tutmak ve mevcut tehditlere karşı önlem almak gibi kritik işlevler bulunmaktadır. STS'ler, zaman zaman kuruluşların güvenlik politikalarındaki eksiklikleri de ortaya çıkarabilirler. Ayrıca, STS'ler karmaşık sistemlerin sadece şifreleme ve güvenlik duvarlarıyla yeterince korunamayacağını açıkça göstermişlerdir. Bu durum, ağ trafiğinin düzenli olarak izlenmesi, analiz edilmesi ve kontrol edilmesi gerekliliğini vurgular.

Yeni ve sofistike siber saldırıların tespitini artırmak için, mevcut STS'lerin zayıf yönlerinin giderilmesi ve bulut bilişim, makine öğrenimi ve derin öğrenme gibi yeni teknolojilerle entegre edilmesi gerekmektedir. Bu şekilde, STS'ler daha etkin hale gelirken, organizasyonlar da siber tehditlere karşı daha güçlü bir savunma oluşturabilirler.

Bilgisayar ağlarında özellikle veri merkezlerinde STS kullanımı, siber saldırılar tarafından neden olunan zararı azaltmak için önemlidir. Sunucularda ve ağa bağlı cihazlarda saldırı önleme, zamanında ve doğru tanımlamanın en önemli yönlerinden biridir. Saldırı Tespit Sistemleri, tespit için gereken özelliklerin azaltılmasıyla çok daha verimli hale gelir. Bu azalma, aşırı bellek ve CPU tüketimini azaltarak ve algılama hızını artırarak sızma algılama işlemlerinde olası yavaşlamaları önler. Bu nedenle, bilgisayar ağlarında saldırıları mümkün olduğunca çabuk tespit etmek, ciddi zararların önlenmesi açısından kritiktir. Dahası, daha yüksek tehdit algılama doğruluğuna ulaşmak, ağ

uzmanlarına daha fazla güven verir ve kaynakları insan müdahalesinden Saldırı Tespit Sistemlerinin güçlü yeteneklerine yönlendirerek maliyet tasarrufu sağlama fırsatı sunar.

Bu durumun farkında olarak, çeşitli saldırı tespit veri kümeleri oluşturulmuştur, bu veri kümeleri tahmin modellerini eğitmek için kullanılmaktadır. Bu çalışmada, bilgisayar ağlarında Saldırı Tespit Sistemlerini değerlendirmek için CSE-CIC-IDS2018 veri seti kullanılmıştır. Kanada Siber Güvenlik Enstitüsü tarafından paylaşılan veri seti, daha önceki çalışmalarda kullanılan veri setlerindeki eksiklikler göz önünde bulundurularak oluşturulmuştur. 80 özellikten ve 16 milyonun üzerinde kayıttan oluşan bu veri seti, bilgisayar bilimine önemli bir katkı sağlamaktadır.

Veri setinde bulunan saldırı çeşitlerinden 6 adet ayrı veri seti oluşturulmuştur. Ek olarak veri önişleme ve üç ayrı öznitelik seçimi yaklaşımını içermektedir. Ayrıca veri setindeki dengesizlik oranını azaltmak için, sentetik veri örnekleme modellerine başvurulmuştur. Bilgisayar ağlarında Saldırı tespiti alanında çeşitli yaklaşımların birleştirilmesiyle yeni metodolojiler sunarak bilgisayar biliminde ilerlemelere katkı sağlanmıştır. Başlangıç veri işleme aşaması, seçim özelliklerini engelleyebilecek rahatsız edici ve gürültülü verilerin elemesi açısından önemlidir. Veri temizleme aşamasından sonraki adım, değerli özelliklerin seçimini optimize etmeye odaklanmıştır.

Oluşturulan her bir veri setine 9 farklı Makine öğrenimi algoritması uygulanmıştır. İkili yaklaşım için belirlenen özelliklerle model eğitimi için NB, KNN, RF, DT, ADA, MLP, LDA, LR ve GB algoritmaları kullanılmıştır. Bu aşamada yalnızca doğruluk oranına bakmak CSE-CIC-IDS2018 gibi dengesiz bir veri setinde yanıltıcı bir metrik olabileceğinden kesinlik, duyarlılık, F Skor değerleri ve ayrıca karmaşıklık matrisi incelenmiştir.

Genel olarak herhangi bir tehdidin tespitinde en iyi sonucu RF ardından KNN modelinin verdiği ve en kötü sonucu NB modelinin verdiği gözlemlenmiştir. F skorları değerleri Şekil 6.9'da görülmektedir; Infilteration ve Web Attack saldırılarının oluşturduğu veri setlerinde uygulanan modellerin F skorlarının epey düşük olması model eğitiminde her ne kadar doğruluk oranı yüksek olan modeller bulunsa da veri setindeki dengesizlik sebebiyle sağlıklı sonuçlar elde edilemeyeceği olasılıklar arasındadır. Diğer saldırıların bulunduğu veri setlerinde makine öğrenimi değerlendirme sonuçlarının başarılı olduğu gözlemlenmektedir.

Çalışmamızın ortalama doğruluk oranları bir başka yapılan 4 çalışmanın doğruluk oranları ile Çizelge 6.7'de verilmiştir.

Çizelge 6.7. Algoritmalara göre tüm saldırılardan elde edilen ortalama doğruluk oranları ile diğer çalışmaların karşılaştırılması

Algoritmalar	**Ortalama Doğruluk Oranları**	(Chua & Salam, 2022)	(Shams & diğerleri, 2020)	(Karataş, 2020)	(Lin, Ye, & Xu, 2019)
NB	0.83816	0.4972	0.46	0.5711	0.554
KNN	0.95866	-	0.95	0.9852	0.942
RF	0.94816	0.5949	0.94	0.9921	0.942
DT	0.94333	0.5942	0.95	0.9966	0.928
ADA	0.9535	-	0.94	0.9969	-
MLP	0.91549	0.7000	0.73	-	-
LDA	0.93099	-	-	0.9080	-
LR	0.92799	-	-	-	-
GB	0.93933	-	-	0.9911	-

Genel olarak, Çizelge 6.7'de verildiği gibi kendi çalışmamızın sonuçları ve diğer çalışmalar incelendiğinde karar ağacı tabanlı algoritmaların dikkate değer yüksek doğruluk oranlarına ulaşabildiğini göstermektedir. Çizelgede çalışmamızın daha iyi olduğu iki çalışma ardından daha nispeten daha yüksek sonuçlar elde etmiş bir çalışma ve sonuçlarımızın neredeyse aynı sonuçları elde etmiş bir çalışma bulunmakta. Çalışmanızın sonuçlarının diğer iki çalışmadan daha iyi olması, kullandığınız veri önişleme, öznitelik seçimi ve makine öğrenmesi yöntemlerinin etkili olduğunu göstermektedir. Bu, yaptığınız çalışmanın bu alanda belirli bir gelişim ve iyileştirme sağladığını gösterir. Nispeten daha yüksek sonuçlar elde eden bir çalışmanın varlığı, alanınızdaki benchmark (karşılaştırma standardı) olarak kullanılabilir. Bu, kendi çalışmanızı bu yüksek sonuçlara ulaşmak için nasıl daha da iyileştirebileceğinizi analiz etmenize yardımcı olabilir. Sonuçlarınızın neredeyse aynı olduğu bir çalışmanın varlığı, çalışmanızın sonuçlarının güvenilir ve doğrulanabilir olduğunu gösterir. Bu, farklı metodolojiler veya veri setleri kullanılsa bile benzer sonuçların elde edilebileceğini ve çalışmanızın geçerliliğini arttırır.

Çalışmamızdaki en başarılı algoritma olarak görülen RF algoritması sonuçları, aynı algoritmayla yapılan başka bir çalışmanın sonuçları ve bir başka çalışmanın F skoru değerleri ile Çizelge 6.8'de verilmiştir.

Çizelge 6.8. RF Algoritması metriklerinin karşılaştırması

Saldırı Türü	Bizim Çalışmamız			(Seth, Chahal, & Singh, 2021)			(Labonne, 2020)
	Kesinlik	Duyarlılık	F Skoru	Kesinlik	Duyarlılık	F Skoru	F Skoru
DDoS	99.33%	86.91%	98.88%	99.84%	98.62%	99.23%	-
DoS	91.04%	90.64%	90.84%	77.33%	99.93%	87.19%	86.4%
Brute Force	99.88%	99.99%	99.93%	99.94%	72.78%	84.23%	99.9%
Botnet	99.51%	99.88%	99.70%	99.63%	99.91%	99.77%	65.5%
Infilteration	22.41%	14.86%	17.87%	31.87%	45.91%	37.62%	-
Web Attack	88.57%	50.82%	64.58%	98.81%	90.37%	94.40%	45.2%

Öncelikle, üç çalışmada da RF algoritmasının genelde yüksek performans gösterdiği görülüyor. Saldırı türlerinde kesinlik ve duyarlılık oranlarının oldukça yüksek olduğunu görmekteyiz.

Ancak, RF algoritmasının Brute Force ve DoS saldırılarında karşılaştırılan ölçütler açısından kendi çalışmamızda daha iyi yönde bir performans farkı gözlenmiştir. Bu farklılık algoritmanın veri setiyle nasıl etkileşime girdiği veya özelliklerin nasıl seçildiği gibi faktörlere bağlı olabilir.

Karar Ağacı ve Rassal Orman yöntemleri kullanılarak yapılan çalışmalardan biri olan (Ozekes & Karakoc, 2019), Hizmet Engelleme, Dağıtılmış Hizmet Reddi ve Port Tarama saldırıları anormal olarak etiketlenmiştir. Deney sonuçlarına göre, Rassal Orman yöntemi %99,966 doğruluk oranıyla daha başarılı sonuç vermiştir. Her tehdidi belirleyen en iyi dört özelliği bulmak için KNN, RF, ID3, ADA, MLP, NB ve ikinci dereceden ayırım analizi yöntemleri kullanılan başka bir çalışmada ise eğitim ve test süreleri ile doğruluk değerleri dikkate alındığında, Rassal Orman yöntemi en iyi sonucu vermiştir (Sharafaldin, Lashkari, & Ghorbani, 2019). RF algoritmasının bizim çalışmamızdaki gibi başka çalışmalarda da başarılı sonuçlar sağladığı gözlemlenmektedir.

Ayrıca karmaşıklık matrislerinde algoritmaların performansını değerlendirmek için kullanılan TP, TN, FP ve FN değerleri önemli göstergelerdir. TP, modelin pozitif olarak doğru tahmin ettiği örnekleri ifade eder ve yüksek olması, modelin pozitifleri doğru tespit etme yeteneğinin iyi olduğunu gösterir. NB ve MLP algoritmaları dışında, diğer algoritmaların TP değerleri veri setinin örneklem sayısına göre oldukça iyi performans sergilemektedir. TN, modelin negatif olarak doğru tahmin ettiği örnekleri ifade eder ve yüksek bir TN değeri, modelin negatifleri doğru tespit etme yeteneğini

gösterir. Bu açıdan NB algoritması biraz geride kalmıştır. FP, modelin yanlış pozitif (tip 1 hata) tahminlerini ifade eder; yüksek FP değeri, modelin yanlış pozitif yapma eğilimini gösterir. Yüksek F skoru ve doğruluk genel performansın iyiliğini gösterirken, yüksek FP değeri, yanlış pozitiflerin sıklığını ve bu hataların potansiyel etkilerini vurgular. NB ve MLP algoritmalarının yüksek hata üretme potansiyeline sahip olduğu görülmektedir, buna karşın ADA algoritması bu konuda en başarılı algoritma olarak öne çıkmaktadır. ADA algoritması, zayıf öğrenicileri birleştirerek güçlü bir sınıflandırıcı oluşturur ve FP ve FN hatalarını dengeli bir şekilde azaltmaya çalışır. FN, modelin yanlış negatif (tip 2 hata) tahminlerini ifade eder; yüksek FN değeri, modelin yanlış negatif yapma eğilimini gösterir. NB'nin tüm saldırılarda yüksek FN değerine sahip olması, veri kümesindeki dengesizliklerden kaynaklanabilir ve bu, pozitif örnekleri yeterince iyi öğrenememesine neden olabilir. Genel olarak, RF ve KNN algoritmalarının düşük FN değeri verdiği ve bu algoritmaların daha az yanlış negatif tahmini yaptığı görülmektedir.

Yaptığımız çalışma ve literatür araştırmalar sonucundan Saldırı tespit sistemlerinde makine öğrenimi algoritmalarının kullanılmasının sağladığı katkıları şu şekilde sıralayabiliriz.

- Makine öğrenimi algoritmaları, belirli verileri sınıflandırarak siber saldırıları yüksek hassasiyetle tespit edebilir.
- Makine Öğrenimi algoritmaları, değişen tehditlere hızlı bir şekilde uyum sağlayarak yeni saldırıları tespit edebilir ve sistemleri güncel tutabilir.
- Makine öğrenimi, büyük veri kümelerini etkin bir şekilde işleyerek anlamlı bilgiler çıkarabilir.
- Makine öğrenimi, kullanıcıların normal davranışlarını öğrenerek anormallikleri tespit edebilir, iç tehditler ve yetkisiz erişim gibi içsel riskleri azaltmada yardımcı olabilir. Bu katkılar, siber güvenlik alanında makine öğrenimi algoritmalarının kullanımının önemini ve etkinliğini vurgular.
- Makine öğrenimi artan tehdit farkındalığı sayesinde ekiplerin yeni tehditleri öngörmesini ve gecikmeyi azaltmasını kolaylaştırır.
- Makine öğrenimi gibi yapay zekâ uygulamaları personel gereksinimdeki azalmayla maliyet ve zaman tasarrufu sağlar.

Sonuç olarak, makine öğrenimi sınıflandırma tekniklerinin, saldırı girişimlerini değerlendirmek ve öngörmek için kullanılabilir olduğu kabul edilebilir. Özellikle, ön işleme ve öznitelik seçimi adımlarının başarılı bir şekilde tamamlanmasıyla, saldırı tahmini görevlerini yerine getiren modellerin geliştirilebileceği sonucuna varılabilir.

Çalışma, özellikle dengesiz veri kümelerinden kaynaklanan sorunları ele almak için mevcut modellerin gelecekteki iyileştirmelere ihtiyaç duyduğunu kabul eder. Ayrıca, modellerin değişen gerçek dünya ortamlarına uyum sağlama yeteneği konusunda çalışmalar yapıldığında daha iyi performans gösterebileceklerini belirtir. Yöntem, kontrollü koşullar altında başarılı bir performans göstermesine rağmen gerçek dünya ortamlarında uygulanması için daha fazla sağlamlık ve geniş kullanılabilirlik için iyileştirme ve detaylı test gerekmektedir.

Gelecekte oluşturulacak Evrensel bir saldırı tespit sistemi, gerçek zamanlı olarak herhangi bir anormalliği tespit edebilme yeteneğine sahip olmalıdır. Bu sistemin geliştirilmesi gereken bazı özelliklere sahip olması gerekmektedir. İlk olarak çok katmanlı olmalıdır, bu da ön işleme, özellik çıkarma, ikili sınıflandırma, çok sınıflı sınıflandırma gibi modüllerden oluşur. Gerçek zamanlı olmalıdır ve Apache Spark gibi bulut teknolojileri ve makine öğrenimi kütüphaneleri kullanarak verileri işleyebilmelidir. Kendi kendine öğrenme yeteneği olmalıdır; yani aktif öğrenme veya pekiştirmeli öğrenme gibi yaklaşımlar kullanılarak, modelin yeni tip anormalliği tanıma yeteneği artırılmalıdır. Çoklu sınıflandırma katmanlarına sahip olmalıdır, başlangıçta ikili sınıflandırma yaparak normal ve saldırı verileri arasında ayrım yapabilmeli ve anormallik algılandığında daha fazla analiz için çok sınıflı sınıflandırma katmanına geçiş yapabilmelidir. Son olarak, hızlı ve verimli olmalıdır, bu nedenle hızlı ve güçlü sınıflandırıcılar ikili sınıflandırma için kullanılmalı ve çok sınıflı sınıflandırma için daha güçlü ancak daha yavaş algoritmalar tercih edilmelidir.

Makine öğrenmesiyle oluşturulan modellerde veri, kritik bir unsurdur. Doğru, anlamlı ve çeşitli veri setleri, çalışmanın başarısını büyük ölçüde etkiler. Dengesiz veri kümeleri, genellikle çoğunluk sınıfına yatkınlık gösterir ve bazen azınlık sınıflarını göz ardı eder. Ancak bu azınlık sınıfları genellikle tespit edilmesi gereken sınıflardır. Bu nedenle dengesizlik oranının azaltılması, sistemin verimliliğini artırırken ortalama

doğruluğu azaltabilir. Gelecekte, bu çalışmada kullanılan modellerin farklı veri setleriyle test edilmesi, modellerin daha geniş bir geçerlilik alanına sahip olmasını sağlayacaktır.

Gelecekteki çalışmalarda, dengeli, çeşitli ve gerçek zamanlı veri setleri üzerinde saldırı tespiti başarısını ve performansını artırmak amacıyla, tek bir makine öğrenimi yönteminin tahmin performansı yerine birden fazla makine öğrenimi yönteminin tahminlerini birleştiren bir toplu öğrenme modeli önerilmektedir. Ayrıca önerilen modelin Derin Öğrenme yaklaşımları kullanılarak karşılaştırmalı bir analizinin yapılması ve bu modelin gerçek dünya kullanılabilirliği açısından çalışma süreleriyle karşılaştırılması planlanmaktadır.

KAYNAKLAR

Aburomman, A. A., & Reaz, M. B. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing, 38*, 360-372.

Ahmad, M., Riaz, Q., Zeeshan, M., Tahir, H., Haider, S. A., & Khan, M. S. (2021). Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *EURASIP Journal on Wireless Communications and Networking, 2021*, 1-23.

Ali, F. (2007). IP spoofing. *The Internet Protocol Journal, 10*(4), 1-9.

Alkhadra, R., Abuzaid, J., AlShammari, M., & Mohammad, N. (2021). Solar winds hack: In-depth analysis and countermeasures. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (s. 1-7). IEEE.

Alpaydin, E. (2020). *Introduction to machine learning.* MIT press.

Al-Yaseen, W. L., Othman, Z. A., & Nazri, M. Z. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications, 67*, 296-303.

Anderson, J. P. (1980). *Computer security threat monitoring and surveillance.*

Ashoor, A. S., & Gore, S. (2012). Intrusion detection system (IDS) & intrusion prevention system (IPS): Case study. *Internatioanl Journal of Scientific & Engineering Research, 2*.

Atay, R., Odabaş, D. E., & Pehlivanoğlu, M. K. (2019). ki Seviyeli Hibrit Makine Öğrenmesi Yöntemi İle Saldırı Tespiti. *Gazi Mühendislik Bilimleri Dergisi, 5*(3), 258-272.

Awotunde, J. B., Folorunso, S. O., Imoize, A. L., Odunuga, J. O., Lee, C.-C., Li, C.-T., & Do, D.-T. (2023). An ensemble tree-based model for intrusion detection in industrial internet of things networks. *Applied Sciences, 13*(4).

Axelsson, S. (2000). *Intrusion detection systems: A survey and taxonomy.* Citeseer.

Bace, R., & Mell, P. (2001). NIST special publication on intrusion detection

systems. *National Institute of Standards and Technology, 16*.

Balakrishnama, S., & Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing, 18*, 1-8.

Barber, D. (2012). *Bayesian reasoning and machine learning.* Cambridge University Press.

Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science, 89*, 117-123.

Belouch, M., El Hadaj, S., & Idhammad, M. (tarih yok). Performance evaluation of intrusion detection based on machine learning using Apache Spark. *Procedia Computer Science, 127*, 1-6.

Bilge, L., & Dumitraş, T. (2012). Before we knew it: an empirical study of zero-day attacks in the real world. *Proceedings of the 2012 ACM conference on Computer and communications security*, (s. 833-844).

Can, O., & Sahingoz, O. K. (2015). A survey of intrusion detection systems in wireless sensor networks. *2015 6th international conference on modeling, simulation, and applied optimization (ICMSAO)* (s. 1-6). IEEE.

Canadian Institute for Cybersecurity. (2017). *Intrusion detection evaluation dataset (CIC-IDS2017)*. https://www.unb.ca/cic/datasets/ids-2017.html adresinden alındı

Canadian Institute for Cybersecurity. (2018). *CSE-CIC-IDS2018 on AWS*. https://www.unb.ca/cic/datasets/ids-2018.html adresinden alındı

Cavusoglu, Ü., & Kacar, S. (2019). Anormal trafik tespiti için veri madenciliği algoritmalarının performans analizi. *Academic Platform-Journal of Engineering and Science, 7*(2), 205-216.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. New York, NY, USA: Association for Computing Machinery.

Cherdantseva, Y., & Hilton, J. (2013). A reference model of information assurance & security. *2013 international conference on availability, reliability and security* (s. 546-555). IEEE.

Chua, T.-H., & Salam, I. (2022). Evaluation of machine learning algorithms in network-based intrusion detection system. *arXiv preprint arXiv:2203.05232*.

Cil, A. E., Yildiz, K., & Buldu, A. (2021). Detection of DDoS attacks with feed

forward based deep neural network model. *Expert Systems with Applications, 169*, 114520.

Cutler, D. R., Edwards Jr, T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., & Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology, 88*(11), 2783-2792.

Cybercrime Magazine. (2020). *Cybercrime To Cost The World $10.5 Trillion Annually By 2025*. cybersecurityventures: https://cybersecurityventures.com/cybercrime-damage-costs-10-trillion-by-2025/ adresinden alındı

Dayıoğlu, B., & Özgit, A. (2001). Internet'de Saldırı Tespiti Teknolojileri. *İletişim Teknolojleri, 1*, 17-21.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on software engineering*(2), 222-232.

Elnakib, O., Shaaban, E., Mahmoud, M., & Emara, K. (2023). EIDM: Deep learning model for IoT intrusion detection systems. *The Journal of Supercomputing, 79*(12), 3241-13261.

Emhan, Ö., & Akın, M. (2019). Filtreleme tabanlı öznitelik seçme yöntemlerinin anomali tabanlı ağ saldırısı tespit sistemlerine etkisi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10*(2), 549-559.

Feily, M., Shahrestani, A., & Ramadass, S. (2009). A survey of botnet and botnet detection. *2009 Third International Conference on Emerging Security Information, Systems and Technologies* (s. 268-273). IEEE.

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*, 102419.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences, 55*(1), 119-139.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Garg, S., Kumar, V., & Payyavula, S. R. (2022). Identification of internet of things (IoT) attacks using gradient boosting: a cross dataset approach. *Telematique, 21*(1), 6982-7012.

Govindarajan, M., & Chandrasekaran, R. (2011). Intrusion detection using neural based hybrid classification methods. *Computer networks, 55*(8), 1662-1671.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE* (s. 986-996). Springer.

Hadnagy, C. (2010). *Social engineering: The art of human hacking.* John Wiley & Sons.

Halfond, W. G., Viegas, J., Orso, A., & others. (2006). A classification of SQL-injection attacks and countermeasures. *Proceedings of the IEEE international symposium on secure software engineering. 1*, s. 13-15. IEEE.

Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., . . . Chowdhury, S. (2023). Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors, 23*(2), 890.

Hinds, J., Williams, E. J., & Joinson, A. N. (2020). "It wouldn't happen to me": Privacy concerns and perspectives following the Cambridge Analytica scandal. *International Journal of Human-Computer Studies, 143*, 102498.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression.* John Wiley & Sons.

IBM. (2023). *Cost of a Data Breach Report 2023*. https://www.ibm.com/reports/data-breach adresinden alındı

Imbalanced-learn. (tarih yok). https://imbalanced-learn.org/stable/ adresinden alındı

Injadat, M., Moubayed, A., Nassif, A. B., & Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Transactions on Network and Service Management, 18*(2), 1803-1816.

Islam, N., Farhin, F., Sultana, I., Kaiser, M. S., Rahman, M. S., Mahmud, M., . . . Cho, G. H. (2021). Towards Machine Learning Based Intrusion Detection in IoT Networks. *Computers, Materials & Continua, 69*(2).

ISO. (2018). https://www.iso.org/standard/iso-iec-27000-family adresinden alındı

Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM, 50*(10), 94-100.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. *achine learning proceedings 1994* (s. 121-129). Elsevier.

Kanimozhi, V., & Jacob, T. P. (2019). Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. *2019 international conference on communication and signal processing (ICCSP)* (s. 33-36). IEEE.

Karataş, G. (2020). Derin öğrenme tabanlı saldırı tespit sistemi.

Kaspersky. (ty). *What is Cybersecurity? Types, Threats and Cyber Safety Tips*. Kaspersky: https://www.kaspersky.com/resource-center/definitions/what-is-cyber-security adresinden alındı

Kaya, Ç., & Oktay, Y. (2015). Intrusion detection with machine learning techniques: Comparative analysis. *Marmara Fen Bilimleri Dergisi, 26*(3), 89-104.

Khan, I. U., Aslam, N., AlShedayed, R., AlFrayan, D., AlEssa, R., AlShuail, N. A., & Al Safwan, A. (2022). A proactive attack detection for heating, ventilation, and air conditioning (HVAC) system using explainable extreme gradient boosting model (XGBoost). *Sensors, 22*(23), 9235.

Knight, W. (2018). *Serious quantum computers are finally here. What are we going to do with them?* MIT Technology Review: https://www.technologyreview.com/2018/02/21/145300/serious-quantum-computers-are-finally-here-what-are-we-going-to-do-with-them/ adresinden alındı

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review, 39*, 261-283.

Kumar, A., & Somani, G. (2022). DDoS attack mitigation in cloud targets using scale-inside out assisted container separation. *IEEE INFOCOM 2022-IEEE conference on computer communications workshops (INFOCOM WKSHPS)* (s. 1-6). IEEE.

Labonne, M. (2020). Anomaly-based network intrusion detection using machine learning. Institut polytechnique de Paris.

Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computers & security*, 459-467.

Lima Filho, F. S., Silveira, F. A., de Medeiros Brito Junior, A., Vargas-Solar, G., & Silveira, L. F. (2019). Smart detection: an online approach for DoS/DDoS attack detection using machine learning. *Security and Communication Networks*, 1-15.

Lin, P., Ye, K., & Xu, C.-Z. (2019). Dynamic network anomaly detection system by using deep learning techniques. *Cloud Computing-CLOUD 2019: 12th International Conference, Held as Part of the Services Conference Federation* (s. 161-176). Springer.

Lippmann, R. P. (1988). An introduction to computing with neural nets. *ACM SIGARCH Computer Architecture News, 16*(1), 7-25.

Liu, J., Kantarci, B., & Adams, C. (2020). Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset. *Proceedings of the 2nd ACM workshop on wireless security and machine learning*, (s. 25-30).

Liu, J., Yang, D., Lian, M., & Li, M. (2021). Research on intrusion detection based on particle swarm optimization in IoT. *IEEE Access, 9*, 38254-38268.

Ma, Z., Li, J., Song, Y., Wu, X., Chen, C., & diğerleri. (2022). Network intrusion detection method based on FCWGAN and BiLSTM. *Computational Intelligence and Neuroscience, 2022*.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet], 9*(1), 381-386.

Manimurugan, S., Al-Mutairi, S., Aborokbah, M. M., Chilamkurti, N., Ganesan, S., & Patan, R. (2020). Effective attack detection in internet of medical things smart environment using a deep belief neural network. *IEEE Access, 8*, 77396-77404.

Matplotlib. (tarih yok). https://matplotlib.org adresinden alındı

Meng, Y.-X. (2011). The practice on using machine learning for network anomaly intrusion detection. *2011 International Conference on Machine Learning and Cybernetics. 2*, s. 576-581. IEEE.

Mohurle, S., & Patil, M. (2017). A brief study of wannacry threat: Ransomware attack 2017. *International journal of advanced research in computer science, 8*(5), 1938-1940.

Mohy-Eddine, M., Guezzaz, A., Benkirane, S., & Azrour, M. (2023). An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. *Multimedia Tools and Applications, 82*(15), 23615-23633.

Najafabadi, M. M., Khoshgoftaar, T. M., Kemp, C., Seliya, N., & Zuech, R. (2014). Machine learning for detecting brute force attacks at the network level. *2014 IEEE International Conference on Bioinformatics and Bioengineering* (s. 379-385). IEEE.

Numpy. (tarih yok). https://numpy.org/doc/stable/ adresinden alındı

Ozekes, S., & Karakoc, E. N. (2019). Makine öğrenmesi yöntemleriyle anormal ağ trafiğinin tespit edilmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 7*(1), 566-576.

Ozgur, A., & Erdem, H. (2012). Saldırı tespit sistemlerinde kullanılan kolay erişilen makine öğrenme algoritmalarının karşılaştırılması. *Bilişim Teknolojileri Dergisi, 5*(2), 41-48.

Pai, V., Adesh, N., & diğerleri. (2021). Comparative analysis of machine learning algorithms for intrusion detection. *IOP Conference Series: Materials Science and Engineering. 1013*, s. 012038. IOP Publishing.

Pandas. (tarih yok). https://pandas.pydata.org adresinden alındı

Pathan, A.-S. K. (2014). *The state of the art in intrusion prevention and detection* (Cilt 44). CRC press Hoboken, NJ, USA.

Python. (tarih yok). *What is Python? Executive Summary*. https://www.python.org/doc/essays/blurb/ adresinden alındı

Rajasinghe, N., Samarabandu, J., & Wang, X. (2018). INSecS-DCS: a highly customizable network intrusion dataset creation framework. *2018 IEEE Canadian Conference on Electrical \& Computer Engineering (CCECE)* (s. 1-4). IEEE.

Reuters. (2019). *Equifax's $700 million data breach settlement spurs criticism, calls for new rules*. https://www.reuters.com/article/us-equifax-cyber-settlement-idUKKCN1UH16Y/ adresinden alındı

Saba, T., Sadad, T., Rehman, A., Mehmood, Z., & Javaid, Q. (2021). Intrusion detection system through advance machine learning for the internet of things networks. *IT professional, 23*(2), 58-64.

Sabhnani, M., & Serpen, G. (2003). Application of machine learning algorithms to KDD intrusion detection dataset within misuse detection context. *MLMTA*, 209-215.

Saied Essa, M., & Kamal Guirguis, S. (2023). Evaluation of Tree-Based Machine Learning Algorithms for Network Intrusion Detection in the Internet of Things. *IT Professional, 25*(5), 45-56.

Sağıroğlu, Ş., & Akleylek, S. (2018). *Siber Güvenlik ve Savunma.* Ankara: Grafiker Yayınları.

Sağıroğlu, Ş., Yolaçan, E., & Yavanoğlu, U. (2011). Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 26*(2).

Scikit-learn. (tarih yok). https://scikit-learn.org/stable/index.html adresinden alındı

Seaborn. (tarih yok). https://seaborn.pydata.org adresinden alındı

Seth, S., Chahal, K. K., & Singh, G. (2021). A novel ensemble framework for an intelligent intrusion detection system. *IEEE Access, 9*, 138451-138467.

Shahriar, H., Whitman, M., Lo, D. C.-T., Wu, F., Thomas, C., & Cuzzocrea, A. (2019). Experiential learning: Case study-based portable hands-on regression labware for cyber fraud prediction. *2019 IEEE International Conference on Big Data (Big Data)* (s. 5814-5818). IEEE.

Shams, M., & diğerleri. (2020). Ağ Anomalisi Tespitinde Makine Öğrenmesi Algoritmalarının Kullanımı ve Karşılaştırmalı Analizi . *Master's thesis, Sakarya Üniversitesi*.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2019). A detailed analysis of the cicids2017 data set. *Information Systems Security and Privacy: 4th International Conference* (s. 172-188). Springer.

Sharafaldin, I., Lashkari, A. H., Ghorbani, A. A., & diğerleri. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp, 1*, 108-116.

Shrivastava, G., Sharma, K., & Rai, S. (2010). The detection & defense of DoS & DDos attack: a technical overview. *Proceeding of ICC, 27*, 28.

Sindhu, S. S., Geetha, S., & Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. *Expert Systems with applications, 39*(1), 129-141.

Soni, P., & Sharma, P. (2014). An intrusion detection system based on KDD-99 data using data mining techniques and feature selection. *International Journal of Soft Computing and Engineering (IJSCE), 4*(3), 2231-2307.

Sung, A. H., & Mukkamala, S. (2004). The feature selection and intrusion

detection problems. *Annual Asian Computing Science Conference* (s. 468-482). Springer.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction.* MIT press.

Türk, F. (2023). Türk, F. (2023). Analysis of intrusion detection systems in UNSW-NB15 and NSL-KDD datasets with machine learning algorithms. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 12*(2), 465-477.

Takaoğlu, M., & Özer, Ç. (2019). SALDIRI TESPİT SİSTEMLERİNE MAKİNE ÖĞRENME ETKİSİ. *Uluslararası Yönetim Bilişim Sistemleri ve Bilgisayar Bilimleri Dergisi*, 11-22.

Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data, 11*(1), 33.

Tanrıkulu, H., & Sazlı, M. H. (2009). Saldırı tespit sistemlerinde yapay sinir ağlarının kullanılması. *Yüksek Lisans Tezi*. Ankara: Ankara Üniversitesi Fen Bilimleri Enstitüsü.

Taormina, R., Galelli, S., Tippenhauer, N. O., Salomons, E., Ostfeld, A., Eliades, D. G., . . . diğerleri. (tarih yok). Battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. *Journal of Water Resources Planning and Management, 144*(8), 04018048.

Thakkar, A., & Lohiya, R. (2022). A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. *Artificial Intelligence Review*, 453-563.

The New York Times. (2018). *Marriott Hacking Exposes Data of Up to 500 Million Guests*. https://www.nytimes.com/2018/11/30/business/marriott-data-breach.html adresinden alındı

The New York Times. (2019). *Capital One Data Breach Compromises Data of Over 100 Million*. https://www.nytimes.com/2019/07/29/business/capital-one-data-breach-hacked.html adresinden alındı

Thockchom, N., Singh, M. M., & Nandi, U. (2023). A novel ensemble learning-based model for network intrusion detection. *Complex & Intelligent Systems, 9*(5), 5693-5714.

TÜİK. (2023). *Survey on Information and Communication Technology (ICT) Usage in Households and by Individuals*. Türkiye İstatistik Kurumu: https://data.tuik.gov.tr/Bulten/Index?p=Survey-on-Information-and-Communication-Technology-(ICT)-Usage-in-Households-and-by-Individuals-2023-49407&dil=2 adresinden alındı

Torabi, M., Udzir, N. I., Abdullah, M. T., & Yaakob, R. (2021). A review on feature selection and ensemble techniques for intrusion detection system. *Science and Information (SAI) Organization Limited, 12*(5).

U.S. Department of Defense. (2023). *Cyber Strategy Summary*. https://www.defense.gov/News/Releases/Release/Article/3523199/dod-releases-2023-cyber-strategy-summary/ adresinden alındı

Umara, M. A., Chenb, Z., Shuaiba, K., & Liuc, Y. (2023). *Effects of Feature Selection and Normalization on Network Intrusion Detection.*

Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-L. A., Elkhatib, Y., . . . Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE access, 7*, 65579-65615.

Yu, K., Nguyen, K., & Park, Y. (2022). Flexible and robust real-time intrusion detection systems to network dynamics. *IEEE Access, 10*, 98959-98969.

Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer networks, 174*, 107247.

ÖZGEÇMİŞ

Berksu Ertuğrul

EĞİTİM

- Matematik Lisans Derecesi

İstanbul Üniversitesi, İstanbul / Mezuniyet: Haziran 2021

- Yönetim Bilişim Sistemleri Lisans Derecesi

İstanbul Üniversitesi, İstanbul / Beklenen Mezuniyet: Haziran 2024

- Hesaplamalı Bilim Yüksek Lisans Derecesi

Trakya Üniversitesi, Edirne / Beklenen Mezuniyet: Haziran 2024

Tez: "Saldırı Tespit Sistemlerinde Makine Öğrenme Tekniklerinin Analizi ve Kullanımı"

YAYINLAR

Ertuğrul, B., & Arda, D. (2024). *Modern siber güvenlik yaklaşımlarında saldırı tespit sistemleri*. Selcuk 10th international conference on applied sciences, Konya.

Ertuğrul, B., & Arda, D. (2024). *Makine öğrenimi ile DOS saldırılarının tespiti*. Selcuk 10th international conference on applied sciences, Konya.