T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

NESNELERİN İNTERNETİ AĞLARINDA DERİN ÖĞRENME TABANLI YENİ BİR SALDIRI TESPİT SİSTEMİ

KEMAL AKKANAT

DOKTORA TEZİ

HESAPLAMALI BİLİMLER ANABİLİM DALI

Tez Danışmanı: Prof. Dr. M. Tolga SAKALLI

EDİRNE-2025

Kemal AKKANAT'ın hazırladığı **"Nesnelerin İnterneti Ağlarında Derin Öğrenme Tabanlı Yeni Bir Saldırı Tespit Sistemi"** başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Hesaplamalı Bilimler Anabilim Dalında bir **Doktora tezi** olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. Tolga SAKALLI	……
Prof. Dr. Metin AYDOĞDU	……
Prof. Dr. Hacı Ali GÜLEÇ	……
Prof. Dr. Erdinç UZUN	……
Dr. Öğr. Üyesi Bora ASLAN	……

Tez Savunma Tarihi: 03/01/2024

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Prof. Dr. Tolga SAKALLI

Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

……

Doç. Dr. Filiz UMAROĞULLARI

Fen Bilimleri Enstitüsü Müdür

T. Ü. FEN BİLİMLERİ ENSTİTÜSÜ

HESAPLAMALI BİLİMLER ANABİLİM DALI DOKTORA 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.

03 / 01 / 2025

Kemal AKKANAT

İmza

Doktora Tezi

Nesnelerin İnterneti Ağlarında Derin Öğrenme Tabanlı Yeni Bir Saldırı Tespit Sistemi

T.Ü. Fen Bilimleri Enstitüsü

Hesaplamalı Bilimler Anabilim Dalı

ÖZET

Günümüzde, teknolojinin hızla ilerlemesiyle birlikte internet tabanlı hizmetlerin sunduğu hizmetler artmış ve bu durum günlük yaşamımızı önemli ölçüde kolaylaştırmaktadır. Özellikle Nesnelerin İnterneti (IoT) teknolojisinin gelişimi, birçok farklı cihazın internet üzerinden birbiriyle ve kullanıcılarla iletişim kurmasına olanak sağlamıştır, böylece farklı hizmetlere erişim daha da basitleşmektedir. Nesnelerin İnterneti, fiziksel nesnelerin sanal dünyayla entegrasyonunu ifade ederken, bu nesnelerin internet üzerinden birbirleriyle veya daha geniş ağlarla etkileşimde bulunabilmesini mümkün kılmaktadır. Bu teknolojinin yaygınlaşması, sağlık, tarım, enerji, ulaşım ve birçok diğer alanı dönüştürmüş, akıllı şehirlerden ev otomasyon sistemlerine kadar geniş bir kullanım alanı bulmaktadır.

Nesnelerin İnterneti ağlarına yönelik en yaygın saldırı türleri arasında hizmet reddi (DoS), veri sızdırma, Kimlik Hırsızlığı ve kötü amaçlı yazılım saldırıları bulunmaktadır. Saldırganlar genellikle yazılım güvenlik açıklarını, zayıf parola uygulamalarını ve ağ güvenlik eksikliklerini kullanarak bu tür saldırılar gerçekleştirmektedir. Örneğin, bir botnet (birden fazla cihazın ele geçirilerek oluşturduğu kötü amaçlı ağ), saldırganların ele geçirdiği çok sayıda Nesnelerin İnterneti cihazını uzaktan kontrol ederek bir ağ oluşturduğu yapıdır. Bu yapı sayesinde cihazlar, kötü amaçlı trafiğin bir parçası olarak kullanılabilmektedir ve geniş çaplı ağ kesintilerine yol açan koordineli saldırılar gerçekleştirebilmektedirler. Nesnelerin İnterneti ortamlarındaki bu güvenlik zafiyetleri, yetkisiz erişimler ve veri ihlalleri gibi ciddi tehditlere neden olabilmektedir. Özellikle özel verilerin korunması, Nesnelerin İnterneti cihazlarının ve ağlarının güvenliğinin sağlanması için kritik bir öneme sahiptir. Güvenlik uzmanları, Nesnelerin İnterneti

güvenliğini sağlamak için sürekli yeni yöntemler geliştirmektedir ve bu cihazları koruyacak daha etkili güvenlik protokollerini ve yazılımlarını tasarlamaktadır.

Nesnelerin İnterneti güvenlik çözümleri arasında, ağ trafiğini izleme, tehditleri tespit etme ve hızlı bir şekilde müdahale edebilme yeteneğine sahip sistemler yer almaktadır. Bu sistemler, genellikle yapay zekâ ve makine öğrenimi tekniklerinden faydalanarak, normalden sapma gösteren davranışları saptayabilmektedir ve olası saldırıları önceden tahmin edebilmektedir. Gelişmiş tespit sistemleri, Nesnelerin İnterneti cihazlarına yönelik saldırıları anında belirleyebilir ve bu sayede, cihazların ve ağların zarar görmesini önleyebilmektedir.

Nesnelerin İnterneti teknolojisinin getirdiği avantajlar neredeyse tartışılmazdır; ancak bu teknolojinin güvenli bir şekilde kullanılabilmesi için güvenlik alanında sürekli yenilikler yapılması ve güvenlik önlemlerinin artırılması gerekmektedir. Geliştirilen yeni güvenlik stratejileri ve teknolojiler, Nesnelerin İnterneti cihazlarına ve ağlarına yönelik tehditlerin azaltılmasında ve bu yeni teknolojik dönemin güvenle benimsenmesinde büyük bir rol oynamaktadır.

Bu tezde Nesnelerin İnterneti ağlarında çalışacak bir saldırı tespit sistemi tasarlanmıştır. Sistem modellenirken büyük ölçüde derin öğrenme algoritmalarından faydalanılmıştır. Uzun Kısa Vadeli Hafıza (LSTM) sinir ağları, Kapılı Tekrarlayan Hücre (GRU) Sinir Ağları, Konvolüsyonel Sinir Ağları (CNN), Oto Kodlayıcı (AE) Sinir Ağları kullanılan belli başlı derin öğrenme algoritmalarıdır. Bu algoritmalar model içinde hibrit bir şekilde kullanılmıştır. Veri seti olarak literatürde bu alanda sıklıkla kullanılan CIC-IDS-2018, CIC-IDS-2017, CIC-IOT-2023, UNSW-NB15, N-BAIOT, BOT-IOT, CIC-DDOS-2019, TON-IOT, IOT-23 gibi veri setleri kullanılmıştır. Veri setleri öncelikli olarak bir ön işlemden geçirilmiş, faydasız veriler çıkarılmış, gerekli görülen veriler nümerik hale dönüştürülmüş, hatalı veriler düzeltilmiş, eksik veriler çeşitli yöntemlerle tamamlanmıştır. Veriler belirli bir aralıkta normalizasyona tabi tutulmuştur.

Kullanılan bütün veri setleriyle uyumlu olabilecek ve iyi sonuçlar alabilecek bir model tasarlanmıştır. Model ile bütün veri setlerinde ortalama %98,7 başarı oranına ulaşmıştır.

Yıl : 2025

Sayfa Sayısı : 155

Anahtar Kelimeler : IDS, IoT, Derin Öğrenme, Saldırı Tespit Sistemi

PhD Thesis

A Novel Deep Learning-Based Intrusion Detection System for Internet of Things Networks

Trakya University Institute of Natural and Applied Sciences

Department of Computational Sciences

ABSTRACT

The rapid advancement of technology today has increased the opportunities provided by internet-based services is significantly simplifying our daily lives. The development of Internet of Things (IoT) technology has particularly enabled various devices to communicate with each other and with users via the internet, simplifying access to different services. While IoT refers to the integration of physical objects with the virtual world, it allows these objects to interact with each other or with larger networks via the internet. The widespread use of this technology has transformed areas such as healthcare, agriculture, energy, transportation, and many others, finding applications ranging from smart cities to home automation systems.

The most common types of attacks on IoT networks include denial of service (DoS), data leakage, identity theft, and malware attacks. Attackers often exploit software vulnerabilities, weak password practices, and network security gaps to carry out such attacks. For example, a botnet is a structure in which attackers remotely control a large number of compromised IoT devices to form a network. Through this structure, the devices can be used as part of malicious traffic and can carry out coordinated attacks that cause large-scale network disruptions. These security vulnerabilities in IoT environments can lead to serious threats, such as unauthorized access and data breaches. Particularly, the protection of personal data is critical to ensuring the security of IoT devices and networks. Security experts are constantly developing new methods to ensure IoT security and designing more effective security protocols and software to protect these devices.

IoT security solutions include systems capable of monitoring network traffic, detecting threats, and responding quickly. These systems often utilize artificial intelligence and machine learning techniques to detect behaviors deviating from the norm and predict potential attacks in advance. Advanced detection systems can instantly identify attacks targeting IoT devices, thereby preventing damage to devices and networks.

The advantages brought by IoT technology are nearly undeniable, however, to ensure the safe use of this technology, continuous innovations in the field of security and the strengthening of security measures are necessary. Newly developed security strategies and technologies play a crucial role in reducing threats to IoT devices and networks and ensuring that this new technological era is adopted safely.

In this thesis, an intrusion detection system designed to operate in IoT networks has been developed. While modeling the system, deep learning algorithms have been extensively utilized. Long Short-Term Memory (LSTM) neural networks, Gated Recurrent Unit (GRU) neural networks, Convolutional Neural Networks (CNN), and Autoencoder (AE) Neural Networks are the main deep learning algorithms used. These algorithms have been used in a hybrid manner within the model. Data sets such as CIC-IDS-2018, CIC-IDS-2017, CIC-IOT-2023, UNSW-NB15, N-BAIOT, BOT-IOT, CIC-DDOS-2019, TON-IOT and IOT-23 which are frequently used in the literature in this field, have been employed as data sets. The data sets were initially preprocessed. Irrelevant data were removed, necessary data were converted into numerical formats, erroneous data were corrected, and missing data were completed using various methods. The data were normalized within a specific range.

A model was designed that would be compatible with all data sets used and would yield good results. An average success rate of 98.7% has been achieved across all datasets with the model.

Year 2025

Number of Pages : 155

Keywords : IDS, IoT, Deep Learning, Intrusion Detection System

TEŞEKKÜR

Bu tezin hazırlanması sürecinde her türlü konuda bana yardımcı olan ve yol gösteren değerli hocam Prof. Dr. M. Tolga SAKALLI ’ya ve doktora tez çalışmamda destek olan, yol gösteren değerli hocalarım Prof. Dr. Metin AYDOĞDU’ya, Prof. Dr. Hacı Ali GÜLEÇ’e, Prof. Dr. Erdinç UZUN’a ve Dr. Öğr. Üyesi Bora ASLAN’a çok teşekkür ederim.

İÇİNDEKİLER

ÖZET iv

ABSTRACT vii

TEŞEKKÜR ix

İÇİNDEKİLER x

SİMGELER VE KISALTMALAR DİZİNİ xiv

ŞEKİLLER DİZİNİ xvii

ÇİZELGELER DİZİNİ xviii

BÖLÜM 1 1

GİRİŞ 1

1.1. Literatür Özeti 4

1.2. Tezin Amacı 9

1.3. Tezin Hipotezleri 10

1.4. Tezin Kapsamı 10

BÖLÜM 2 12

SİBER SALDIRI TÜRLERİ VE SALDIRI TESPİT SİSTEMLERİ 12

2.1. Sosyal Mühendislik Saldırıları 12

2.1.1. Oltalama 13

2.1.2. Dinleme 14

2.1.3. Sahtecilik 15

2.1.4. Kimlik Hırsızlığı 16

2.2. Kötü Amaçlı Yazılımlar 16

2.2.1. Bilgisayar Virüsleri 17
2.2.2. Casus Yazılımlar 19
2.2.3. Solucanlar 20
2.2.4. Reklam Yazılımları 21
2.2.5. Tuş Kaydediciler 23
2.2.6. Donanım Truva Atları 24
2.3. Kaba Kuvvet Saldırıları 25
2.4. Hizmet Reddi, Dağıtık Hizmet Reddi Saldırıları 26
2.4.1. Hizmet Reddi 26
2.4.2. Dağıtık Hizmet Reddi 27
2.5. Gelişmiş Sürekli Tehdit 28
2.6. Veri Tabanı, Yazılım, Tarayıcı Tabanlı Saldırılar 29
2.6.1. SQL Enjeksiyonu Saldırıları 29
2.6.2. Siteler Arası Komut Dosyası Çalıştırma Saldırıları 30
2.7. Ortadaki Adam Saldırısı 31
2.8. Saldırı Tespit Sistemleri 32
2.8.1. Saldırı Tespit Sistemleri Türleri 32
BÖLÜM 3 35
NESNELERİN İNTERNETİ AĞLARI 35
3.1. Algılama Katmanı 36
3.2. Ağ Katmanı 37
3.3. Uygulama Katmanı 38
3.4. Nesnelerin İnterneti Mimarisine Tehditler 39
3.4.1. Algılama Katmanı 39
3.4.2. Ağ Katmanı 40

3.4.3. Uygulama Katmanı 40
3.5. Kullanılan Protokoller 41
3.5.1. Zigbee 41
3.5.2. 6LoWPAN 42
3.5.3. BLE 43
3.5.4. LoraWAN 44
3.5.5. Mesaj Kuyruğu Telemetri Taşıması 45
3.5.6. Kısıtlı Uygulama Protokolü 45
BÖLÜM 4 47
KLASİK MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME ALGORİTMALARI 47
4.1. Klasik Makine Öğrenmesi Algoritmaları 47
4.1.1. Destek Vektör Makineleri 47
4.1.2. K-En Yakın Komşular Algoritması 49
4.1.3. Karar Ağaçları 50
4.1.4. Naive Bayes Algoritması 51
4.1.5. AdaBoost Algoritması 52
4.1.6. Rastgele Orman Algoritması 53
4.2. Derin Öğrenme Algoritmaları 54
4.2.1. Konvolüsyonel Sinir Ağları 55
4.2.2. Oto Kodlayıcılar 56
4.2.3. Tekrarlayan Sinir Ağları 57
4.2.4. Rekabetçi Üretici Ağlar 61
BÖLÜM 5 63
ÖNERİLEN SALDIRI TESPİT SİSTEMİ UYGULAMASI 63
5.1. Uygulama Planlaması ve Hazırlık Süreci 63

5.2. Uygulamada Kullanılan Araç ve Yöntemler 64
5.3. Uygulamada Kullanılan Veri Setleri 64
5.3.1. CIC-IDS-2018 Veri Seti 64
5.3.2. CIC-IDS-2017 Veri Seti 65
5.3.3. UNSW-NB15 Veri Seti 65
5.3.4. CIC-IOT-2023 Veri Seti 65
5.3.5. IOT-23 Veri Seti 66
5.3.6. N-BAIOT Veri Seti 67
5.3.7. TON-IOT Veri Seti 68
5.3.8. BOT-IOT Veri Seti 68
5.3.9. CIC-DDOS-2019 Veri Seti 68
5.4. Uygulama Detayları 69
5.4.1. Ölçeklendirme Türleri 69
5.4.2. Aktivasyon Fonksiyonları 72
5.4.3. Optimizasyon Algoritmaları 75
5.4.4. Veri Ön İşleme ve Model Oluşturma 79
5.5. Değerlendirme Metrikleri 93
BÖLÜM 6 95
SONUÇLAR VE TARTIŞMA 95
6.1. Önerilen Model Sonuçları 95
6.2. Sonuçların Değerlendirilmesi 104
6.3. Gelecek Çalışmalar İçin Öneriler 110
KAYNAKLAR 111
ÖZGEÇMİŞ 136

SİMGELER VE KISALTMALAR DİZİNİ

AE:	Oto Kodlayıcı
AMQP:	Gelişmiş Mesaj Kuyruğu Protokolü
API:	Uygulama Programlama Arayüzü
APT:	Gelişmiş Sürekli Tehdit
ARP:	Adres Çözümleme Protokolü
AWID :	Aegean Wi-Fi Intrusion Dataset
BI-LSTM:	İki Yönlü Uzun Kısa Vadeli Hafıza
CNN:	Konvolüsyonel Sinir Ağları
COAP:	Sınırlı Uygulama Protokolü
DAS:	Veri Toplama Sistemleri
DDOS:	Dağıtık Hizmet Reddi
DKNN:	Derin Kronecker Sinir Ağı
DL:	Derin Öğrenme
DNFN:	Derin Nöro-Bulanık Ağlar
DNN:	Derin Sinir Ağları
DNS:	Alan Adı Sistemi
DOM:	Doküman Nesne Modeli
DOS:	Hizmet Reddi
FHSS:	Frekans Atlamalı Yayılım Spektrumu
GA:	Genetik Algoritma
GAN:	Rekabetçi Üretci Ağlar
GBS:	Küresel Konumlandırma Sistemi

GMM:	Gauss Karışım Modelleri
GPS:	Küresel Konumlama Sistemi
GRU:	Kapı Kontrollü Tekrarlayan Birim Sinir Ağları
GWO:	Gri Kurt Optimizasyonu
HART:	Otoyola Bağlanabilir Uzaktan Sensör
HIDS:	Host Tabanlı Saldırı Tespit Sistemleri
ICMP:	İnternet Kontrol Mesaj Protokolü
IDC:	Uluslararası Veri Şirketi
IDS:	Saldırı Tespit sistemleri
IIOT:	Endüstriyel Nesnelerin İnterneti
IOT:	Nesnelerin İnterneti
IOV:	Araçların İnterneti
IP:	Internet Protokolü
KNN:	K-En Yakın Komşular Algoritması
KRACK:	Anahtar Yeniden Kurma Saldırısı
LSTM:	Uzun Kısa Vadeli Hafıza
MQTT:	Mesaj Kuyruğu Telemetri Taşıma Protokolü
MSE:	Ortalama Karesel Hata
MTD:	Hareketli Hedef Savunması
NFC:	Yakın Alan İletişimi
NIDS:	Ağ Tabanlı Saldırı Tespit Sistemleri
NTP:	Ağ Zaman Protokolü
PMT:	Koruma Motivasyon Teorisi
REST:	Temsili Durum Transferi
RFID:	Radyo Frekansı ile Tanımlama

RNN:	Tekrarlayan sinir ağları
RPL:	Güvenilir Olmayan Ağlar İçin Yönlendirme Protokolü
SCPP:	Spearman Korelasyon Tabanlı Olasılıksal Budama
SSSA:	Sosyal Sincap Arama Algoritması
SVM:	Destek Vektör Makineleri
SYN:	Eşzamanlama Bayrağı
TCP:	İletim Kontrol Protokolü
TLS:	Taşıma Katmanı Güvenliği
UBA:	Kullanıcı Davranış Analizi
UDP:	Kullanıcı Datagram Protokolü
WSN:	Kablosuz Sensör Ağı
XMPP:	Genişletilebilir Mesajlaşma ve Varlık Protokolü
XSS:	Siteler Arası Komut Dosyası

ŞEKİLLER DİZİNİ

Şekil 2.1. Siber Saldırı Türleri 12

Sekil 2.2. Saldırı Tespit Sistemleri Türleri 32

Şekil 3.1. Nesnelerin İnterneti Genel Mimarisi..... 35

Şekil 3.2. Nesnelerin İnterneti Katmanları..... 36

Şekil 4.1. CNN Genel Mimarisi 55

Şekil 4.2. AE Genel Mimarisi 56

Şekil 4.3. RNN Genel Mimarisi 58

Şekil 4.4. LSTM Genel Mimarisi..... 60

Şekil 4.5. GAN Genel Mimarisi..... 61

Şekil 5.1. Sigmoid Aktivasyon Fonksiyonu..... 73

Şekil 5.2. ReLU Aktivasyon Fonksiyonu 74

Şekil 5.3. Tanh Aktivasyon Fonksiyonu..... 75

Şekil 6.1. UNSW-NB15 Veri Seti Karmaşıklık Matrisi..... 96

Şekil 6.2. CIC-IDS-2018 Veri Seti Karmaşıklık Matrisi..... 97

Şekil 6.3. CIC-IDS-2017 Veri Seti Karmaşıklık Matrisi..... 98

Şekil 6.4. CIC-IOT-2023 Veri Seti Karmaşıklık Matrisi..... 99

Şekil 6.5. BOT-IOT Veri Seti Karmaşıklık Matrisi 100

Şekil 6.6. N-BAIOT Veri Seti Karmaşıklık Matrisi..... 101

Şekil 6.7. TON-IOT Veri Seti Karmaşıklık Matrisi 102

Şekil 6.8. IOT-23 Veri Seti Karmaşıklık Matrisi 103

Şekil 6.9. CIC-DDOS-2019 Veri Seti Karmaşıklık Matrisi..... 104

ÇİZELGELER DİZİNİ

Çizelge 5.1. CIC-IDS-2018 Veri Seti Saldırı Türleri 64

Çizelge 5.2. CIC-IDS-2017 Saldırı Türleri 65

Çizelge 5.3. UNSW-NB15 Veri Seti Saldırı Türleri 65

Çizelge 5.4. CIC-IOT-2023 Veri Seti Saldırı Türleri 66

Çizelge 5.5. IOT-23 Veri Seti Saldırı Türleri 67

Çizelge 5.6. N-BAIOT Veri Seti Saldırı Türleri 67

Çizelge 5.7. TON-IOT Veri Seti Saldırı Türleri 68

Çizelge 5.8. BOT-IOT Veri Seti Saldırı Türleri 68

Çizelge 5.9. CIC-DDOS-2019 Veri Seti Saldırı Türleri 69

Çizelge 5.10. Tanh ve Sigmoid Fonksiyonu Karşılaştırması 75

Çizelge 5.11. Karmaşıklık Matrisi 93

Çizelge 6.1. UNSW-NB15 Veri Seti İkili Sınıflandırma Sonuçları 95

Çizelge 6.2. CIC-IDS-2018 Veri Seti İkili Sınıflandırma Sonuçları 96

Çizelge 6.3. CIC-IDS-2017 Veri Seti İkili Sınıflandırma Sonuçları 97

Çizelge 6.4. CIC-IOT-2023 Veri Seti İkili Sınıflandırma Sonuçları 98

Çizelge 6.5. BOT-IOT Veri Seti İkili Sınıflandırma Sonuçları 99

Çizelge 6.6. N-BAIOT Veri Seti İkili Sınıflandırma Sonuçları 100

Çizelge 6.7. TON-IOT Veri Seti İkili Sınıflandırma Sonuçları 101

Çizelge 6.8. IOT-23 Veri Seti İkili Sınıflandırma Sonuçları 102

Çizelge 6.9. CIC-DDOS-2019 Veri Seti İkili Sınıflandırma Sonuçları 103

Çizelge 6.10. Benzer Çalışmalar İle Karşılaştırmalı Sonuçlar 1 107

Çizelge 6.11. Benzer Çalışmalar İle Karşılaştırmalı Sonuçlar 2 108

Çizelge 6.12. Benzer Çalışmalar İle Karşılaştırmalı Sonuçlar 3 109

BÖLÜM 1

GİRİŞ

Teknolojinin hızla ilerlemesiyle birlikte, bilgi işlem teknolojileri günlük yaşamımızın vazgeçilmez bir parçası haline gelmiştir. Bu ilerleme, özellikle Nesnelerin İnterneti (Internet of Things) gibi alanlarda kendini göstermektedir. Nesnelerin İnterneti, çeşitli akıllı cihazların birbirine bağlanarak tek bir ağ oluşturduğu ve bu cihazların hassas veriler toplayıp önemli işlevleri yerine getirebilmesini sağladığı bir yapıdır etmektedir (Saranya, Sridevi, Deisy, Chung & Khan, 2020). Bu yapı, farklı hizmetleri bir araya getirerek etkili bir iletişim ve veri işleme platformu oluşturmaktadır. Kullanıcılar, mobil uygulamalar veya web hizmetleri aracılığıyla cihazlara erişebilmektedir ve onları kontrol edebilmektedirler.

Nesnelerin İnterneti teknolojisinin gelişimi, birçok farklı cihazın internet üzerinden birbiriyle ve kullanıcılarla iletişim kurmasına olanak sağlamıştır, böylece farklı hizmetlere erişim daha da basitleşmiştir (Askar, 2019). Nesnelerin İnterneti, fiziksel nesnelerin sanal dünyayla entegrasyonunu ifade etmektedir. Bu sayede, fiziksel nesneler internet üzerinden birbirleriyle veya daha geniş ağlarla etkileşimde bulunabilmektedir. Bu teknolojinin yaygınlaşması, sağlık (Subramaniyaswamy vd., 2019), tarım (Khanna & Kaur, 2019), enerji (Liu vd., 2019), ulaşım (Hyman, Alisha, & Gordon, 2019) ve birçok diğer alanı dönüştürmüş, akıllı şehirlerden ev otomasyon sistemlerine kadar geniş bir kullanım alanı bulmuştur (Modiri vd., 2022).

1990'lı yıllarda geliştirilen Nesnelerin İnterneti teknolojisi, teknolojinin gelişimi ve üretim maliyetlerindeki azalma ile hızlı bir şekilde yaygınlaştığı gözlemlenmiştir. Bu yaygınlaşmanın bir sonucu olarak ise ağ trafiğinde yoğunluk ve tıkanıklık gibi yeni sorunların ortaya çıktığı tespit edilmiştir. Statista raporlarına göre toplam internet verisi 2023 yılında 123 zettabayta ulaşmıştır. Bu verinin 2025 yılında 182 zettabayta ulaşması beklenmektedir (Statista, 2024).

İstatistiksel tahminlere göre, 2025 yılına kadar dünya genelinde Nesnelerin İnterneti bağlantılı cihaz sayısının 30 milyarı aşması beklenmektedir (Cisco, 2023). Bu büyük artış, Nesnelerin İnterneti cihazlarının hayatımızın birçok noktasında aktif olarak kullanılacağı anlamına gelmektedir.

Nesnelerin İnterneti'nin sunduğu akıllı cihaz bağlantıları, farklı hizmetler aracılığıyla tek bir ağ üzerinden birleşerek çalışır ki, bu da cihazların daha hızlı ve etkin bir şekilde iletişim kurmasına yardım etmektedir (Saranya vd., 2020). Bu cihazlar, önemli işlevleri yerine getirirken hassas veriler toplar ve bulut hizmetlerini kullanarak bu verilerin işlenmesini ve bu verilere uzaktan erişimini mümkün kılmaktadır. Kullanıcılar, bu verilere mobil uygulamalar veya web hizmetleri üzerinden erişebilmektedirler.

Bilgi teknolojisi güvenliği, kullanılan her türlü elektronik bilginin korunmasını amaçlamaktadır (Barros, Basgalupp, de Carvalho & Freitas, 2012; Wang, Yang & Ren, 2011). Ayrıca, saldırı tespit sistemleri (IDS), ağ üzerinden gerçekleşebilecek her türlü kötüye kullanımı veya yetkisiz erişimi belirleyebilir; böylece veri iletim sisteminin sürekliliğini ve bireysel bilgilerin gizliliğini garanti altına almaktadır (Mienye & Jere, 2024).

Nesnelerin İnterneti sistemlerinin daha güvenilir ve daha verimli olmasını sağlayan yöntemlerden bir tanesi, sensörlerden toplanan verilerin yapay zekâ algoritmaları tarafından analiz edilmesidir. Bu analizler, akıllı cihazların daha doğru kararlar almasına olanak tanımaktadır ve sistem performansını artırmaktadır (Abdelhalim & Traore, 2009; Safavian & Landgrebe, 1991). Ayrıca, Saldırı Tespit Sistemleri (IDS), bilgi işlem sistemlerinin bütünlüğünü korumak ve kötü niyetli saldırılara karşı bu sistemleri savunmak için kullanılmaktadır. Bu sistemler, veri aktarım güvenliğini artırarak sistem sürekliliğine katkı vermektedir ve kullanıcı bilgilerinin gizliliğini korumaktadırlar (Mienye & Jere, 2024).

Nesnelerin İnterneti teknolojisinin getirdiği kolaylıklar, aynı zamanda yeni güvenlik zafiyetlerini de beraberinde getirmektedir. Nesnelerin İnterneti cihazları, özellikle kötü amaçlı kişiler tarafından hedef alınmaktadır. Çünkü bu cihazlar genellikle düşük güvenlik önlemleri ile korunmaktadır. Bu durum, siber saldırılar için potansiyel bir açık kapı bırakmaktadır (Quan, Kumar, Sundaresan & Sutaria, 2019). Nesnelerin İnterneti ağlarına yönelik en yaygın saldırı türleri arasında dağıtık hizmet reddi, veri sızdırma,

Kimlik Hırsızlığı ve kötü amaçlı yazılım saldırıları bulunmaktadır. Saldırganlar genellikle yazılım güvenlik açıklarını (Wan, Lai, Song, Imran & Jia, 2019), zayıf parola uygulamalarını (Jourdan & Stavrou, 2019) ve ağ güvenlik eksikliklerini (Msadek, Soua & Engel, 2019) kullanarak bu tür saldırılar gerçekleştirmektedir. Örneğin, bir botnet saldırısı, çok sayıda Nesnelerin İnterneti cihazını ele geçirerek, bu cihazları kötü amaçlı trafiğin bir parçası olarak kullanabilmektedir. Ele geçirilen cihazlar vasıtasıyla kuvvetli DDoS saldırıları gerçekleştirilerek geniş çaplı ağ kesintilerine yol açılabilmektedir (Aisopos, Litke, Kardara, Tserpes, Campo & Varvarigou, 2019).

2016 yılında gerçekleşen ve Nesnelerin İnterneti cihazlarındaki güvenlik açıklarını hedef alan büyük çaplı Dağıtık Hizmet Reddi saldırıları (Lee, Sim & Hong, 2024), bu cihazların ne kadar savunmasız olabileceğini tüm dünyaya göstermiştir. Nesnelerin İnterneti cihazları, kişisel ve hassas verileri toplayıp işleyebildiği için, bu cihazları etkin bir şekilde korumak, büyük bir zorunluluk haline gelmiştir.

Nesnelerin İnterneti ortamlarındaki bu güvenlik zafiyetleri, yetkisiz erişimler ve veri ihlalleri gibi ciddi tehditlere yol açabilmektedir. Özellikle kişisel ve hassas verilerin korunması, Nesnelerin İnterneti cihazlarının ve ağlarının güvenliğinin sağlanması için kritik bir öneme sahiptir (Da Costa, Papa, Lisboa, Munoz & Albuquerque, 2019). Güvenlik uzmanları, Nesnelerin İnterneti güvenliğini sağlamak için sürekli yeni yöntemler geliştirmektedir ve bu cihazları koruyacak daha etkili güvenlik protokollerini ve yazılımlarını tasarlamaktadırlar (Dizdarević, Carpio, Jukan & Masip-Bruin, 2019). Nesnelerin İnterneti güvenlik çözümleri arasında ağ trafiğini izleme, tehditleri tespit etme ve hızlı bir şekilde müdahale edebilme yeteneklerine sahip sistemler yer almaktadır. Bu sistemler, yapay zekâ ve makine öğrenimi tekniklerinden faydalanarak, normalden sapma gösteren davranışları saptayabilmektedir ve olası saldırıları önceden tahmin edebilmektedirler (Zhou & Chou, 2019). Gelişmiş tespit sistemleri, Nesnelerin İnterneti cihazlarına yönelik saldırıları kolayca belirleyebilir ve bu sayede, cihazların ve ağların zarar görmesini önleyebilmektedirler.

Siber güvenlik uygulamaları, bilgisayarlar, mobil cihazlar, sunucular ve ağlar gibi elektronik sistemleri kötü amaçlı saldırılardan koruyarak bilgi teknolojisi güvenliği kapsamında genel bir güvenlik altyapısının sağlanmasında hayati bir rol oynamaktadır. (Facchini, Giorgi, Saracino & Dini, 2020; Venkatraman vd., 2021). Siber güvenlik

uygulamalarında klasik makine öğrenmesi yöntemleri ve derin öğrenme yöntemlerinden de faydalanılmaktadır. Derin Öğrenme (DL) tabanlı yöntemler etiketli verileri (sınıf bilgisi olan, örneğin saldırı veya normal sınıfının belirli olduğu veri) denetimli sınıflandırmada kullanmayı amaçlamaktadır (Tsimenidis, Lagkas & Rantos, 2022). Bu sayede bilgi güvenliği, konuşma, ses ve görüntü işleme gibi alanlarda etkili sonuçlar elde edilmektedir ve böylece teknolojik gelişmelerin pratik uygulamalara dönüşümü hız kazanmaktadır.

İnternetin çok yoğun kullanıldığı günümüzde, siber güvenlikte yapay zekâ kullanımı, tehditleri daha hızlı ve etkin bir şekilde analiz etme ve önleme potansiyeli sunmaktadır. Yapay zekânın, büyük veri setlerini analiz edebilme ve öğrenebilme yeteneği, onu siber güvenlik araçları için vazgeçilmez bir bileşen yapmaktadır. Derin öğrenme, yapay zekâ teknolojileri arasında özellikle ön plana çıkmaktadır çünkü karmaşık veri yapılarını işleyebilme, otomatik özellik çıkarma ve örüntü tanıma yetenekleriyle, daha önce tespit edilmemiş tehditleri dahi tespit edebilmektedir (Capuano, Fenza, Loia & Stanzione, 2022).

1.1. Literatür Özeti

Davahli ve arkadaşları (2020), çalışmalarında, Nesnelerin İnterneti kablosuz ağları için GA-GWO (Genetik Algoritma - Gri Kurt Optimizasyonu) tabanlı bir saldırı tespit sistemi geliştirmişlerdir. Kullanılan veri seti, AWID olup, gerçek bir kablosuz ağdan toplanmış veriler içermektedir. Çalışmanın amacı, verilerin boyutunu azaltarak en anlamlı özellikleri seçmektir. GA-GWO algoritması, SVM ile birleştirilerek saldırı ve normal trafik ayrımı yapılmıştır. Doğruluk (accuracy), kesinlik (precision), F1 Skoru metriklerimde sırasıyla %99,10, %96,03, %97,64 başarı elde edilmiştir. GA-GWO algoritması, orijinal GA ve GWO optimizasyon algoritmalarına göre daha yüksek doğruluk oranı sağlamıştır (Davahli, Shamsi & Abaei, 2020).

Andrade-Arenas ve arkadaşları (2020), çalışmalarında, AD-IoT adı verilen hafif bir saldırı tespit sistemi geliştirmişlerdir. Bu sistem, Nesnelerin İnterneti altyapısında güvenlik açıklarını analiz eder ve potansiyel tehditleri algılayarak, saldırıları önlemeyi hedefler. AD-IoT, özellikle düşük enerji tüketimi ve hızlı tepki vermesi gereken Nesnelerin İnterneti cihazları için optimize edilmiş hafif bir modeldir. Bu sistemde, Rastgele Orman gibi karar ağacı tabanlı algoritmalar kullanılır. Bu algoritmalar, ağ

trafiğini analiz ederek anormal davranışları belirler. Bu modelde, UNSW-15 veri seti kullanılarak hassasiyet (recall), kesinlik, F1 Skoru metriklerinde %98 başarı sağlanmıştır (Andrade-Arenas & Ramos-Romero, 2020).

Lee ve arkadaşları (2020), çalışmalarında, IMPACT adı verilen, kaynak kısıtlı cihazlar üzerinde çalışabilecek hafif bir makine öğrenimi tabanlı saldırı tespit sistemi geliştirmişlerdir. Bu çalışmada, AWID veri seti kullanılarak AE algoritmasından yararlanılmıştır. Çeşitli özellik seçimi yöntemleri kullanılarak veri boyutu azaltılmış ve sınıflandırma performansını artırılmıştır. %98,22 doğruluk oranı elde edilmiştir (Lee vd., 2020).

Özer ve arkadaşları (2021), çalışmalarında, BOT-IOT veri setini kullanarak bir saldırı tespit sistemi geliştirmişlerdir. Araştırmada, veri setinin belirli özellik çiftleri üzerinde farklı makine öğrenmesi algoritmalarıyla çalışılarak yüksek doğruluk ve düşük enerji tüketimi hedeflenmiştir. BOT-IOT veri setinin 12 ana özelliği arasından 66 benzersiz özellik çifti oluşturulmuş ve 10 farklı algoritma ile değerlendirilmiştir. %86,88 ile %99 arasında doğruluk başarı oranları elde edilmiştir (Özer, İskefiyeli & Azimjonov, 2021).

Filus ve arkadaşları (2021), çalışmalarında, BOT-IOT veri setini kullanarak hafif (lightweight) bir saldırı tespit sistemi geliştirmişlerdir. Çalışmada Tekrarlayan Rastgele Sinir Ağı kullanılmıştır. %96,09 oranında doğruluk başarı oranı elde edilmiştir (Filus, Domańska & Gelenbe, 2021).

Kumar ve arkadaşları (2022), çalışmalarında, EDIMA adlı Nesnelerin İnterneti botnetlerini erken aşamalarda tespit etmek amacıyla iki aşamalı bir makine öğrenimi tabanlı algılama sistemi geliştirmişlerdir. Çalışmada, IoT-NSS-BPR, UNSW-NB15 ve IoT-23 veri setlerini kullanılarak gerçek Nesnelerin İnterneti zararlı yazılımları ve bot-CnC (komut ve kontrol) iletişim kalıpları analiz edilmiştir. Doğruluk, hassasiyet, F1 Skoru metriklerinde %91 ve üzerinde başarı sonuçları elde edilmiştir (Kumar, Shridhar, Swaminathan & Lim, 2022).

Dao ve Lee (2022), çalışmalarında, geliştirdikleri SCPP (Spearman Korelasyon Tabanlı Olasılıksal Budama) algoritmasını Oto Kodlayıcılara uygulayarak bir model geliştirmişlerdir. UNSW-NB15 veri seti kullanılarak %74,17, CIC-IDS-2017 veri seti kullanılarak %95,58 doğruluk oranı elde edilmiştir (Dao & Lee, 2022).

Altunay ve Albayrak (2023), çalışmalarında, Endüstriyel Nesnelerin İnterneti (IIoT) ağları için hibrit bir CNN+LSTM tabanlı saldırı tespit sistemi geliştirilmişlerdir. Bu model, özellikle çok sınıflı sınıflandırmada diğer yöntemlere göre daha yüksek doğruluk oranlarına ulaşmıştır. UNSW-NB15 veri setinde, ikili sınıflandırmada %93,21, çok sınıflı sınıflandırmada ise %92,90 doğruluk başarı oranı elde edilmiştir (Altunay & Albayrak, 2023).

Dina ve arkadaşları (2023), çalışmalarında, Nesnelerin İnterneti ortamlarında siber güvenlik tehditlerini tespit etmek için odak kaybı (focal loss) fonksiyonunun kullanıldığı bir derin öğrenme yaklaşımı önermişlerdir. Çalışma, veri dengesizliği sorununu çözmek için odak kaybı fonksiyonunun avantajlarını vurgulamıştır. Çalışmada BOT-IOT veri seti kullanılmıştır. Önerilen FNN-Focal yöntemi ile doğruluk, kesinlik, hassasiyet, F1 Skoru metriklerinde sırasıyla %92, %56, %64, %58 başarı oranları elde edilmiştir. Önerilen CNN-Focal yöntemi ile doğruluk, kesinlik, hassasiyet, F1 Skoru metriklerinde sırasıyla %87, %62, %63, %59 başarı oranları elde edilmiştir (Dina, Siddique & Manivannan, 2023).

Kasongo (2023), çalışmasında Özellik Seçimi için XGBoost algoritmasını ve Derin Öğrenme tekniklerinden olan RNN tabanlı modelleri önermiştir. Çalışmada, LSTM, GRU ve Basit RNN modelleri detaylandırılmıştır. Veri seti olarak NSL-KDD ve UNSW-NB15 veri seti kullanılmıştır. NSL-KDD veri setinde, ikili sınıflandırmada, XGBoost-LSTM modeli %88,13 doğruluk oranı, çoklu sınıflandırmada XGBoost-LSTM %86,93 doğruluk oranı elde edilmiştir. UNSW-NB15 veri setinde, ikili sınıflandırmada, XGBoost-Simple-RNN modeli ile %87,07 doğruluk oranı, çoklu sınıflandırmada XGBoost-GRU modeli ile %78,40 doğruluk oranı elde edilmiştir (Kasongo, 2023).

Mayuranathan ve arkadaşları (2022), çalışmalarında, bulut bilişim ortamında bir saldırı tespit sistemi olarak EOS-IDS (Etkili Optimal Saldırı Tespit Sistemi) modeli önermişlerdir. Veri boyutunu azaltmak için CRDO (Kaotik Kızıl Geyik Optimizasyonu) algoritması kullanılmıştır. Saldırı tespiti ve sınıflandırması için de DKNN (Derin Kronecker Sinir Ağı) kullanılmıştır. Veri seti olarak DARPA ve CIC-IDS-2018 veri seti kullanılmıştır. DARPA veri setinde, doğruluk, kesinlik, hassasiyet, F1 Skoru metriklerinde sırasıyla % 97,22, %95,99, %95,33, %95,66 başarı elde edilmiştir. CIC-IDS-2018 veri setinde, doğruluk, kesinlik, hassasiyet, F1 skoru metriklerinde sırasıyla

%97,12, %95,65, %98,18, %96,91 başarı elde edilmiştir (Mayuranathan, Saravanan, Muthusenthil & Samydurai, 2022).

Sharma ve arkadaşları (2023), çalışmalarında, Nesnelerin İnterneti ağlarında siber tehditleri tespit etmek için DNN (Derin Sinir Ağları) ve GAN tabanlı bir model önermişlerdir. Veri setinde gereksiz ve ilişkili özellikleri filtrelemek için korelasyon tabanlı bir yöntem kullanılmıştır. GAN'lar kullanılarak küçük örneklem sınıflarından sentetik veri üretilmiştir. UNSW-NB15 veri seti ile çoklu sınıflandırmada, %91 doğruluk değeri elde edilmiştir (Sharma, Sharma, Lal & Roy, 2023).

Saba ve arkadaşları (2022), çalışmalarında, özellikle Nesnelerin İnterneti ortamındaki trafik davranışını analiz etmek ve anormal trafik veya saldırıları tespit etmek için bir CNN modeli önermişlerdir. Çalışmada NID ve BOT-IOT veri seti kullanılmıştır. Doğruluk metriğinde NID veri setinde %99,51, BOT-IOT veri setinde %95,55 başarı elde edilmiştir (Saba, Rehman, Sadad, Kolivand & Bahaj, 2022).

Ren ve arkadaşları (2023), çalışmalarında, saldırıları tespit etmek için CANET (Hiyerarşik CNN-Hierarchical CNN) modelini önermişlerdir. Mevcut yöntemlerin yüksek yanlış pozitif oranı ve sınıf dengesizliği gibi sorunların çözümü amaçlanmıştır. CANET, yerel ve çok seviyeli uzay-zamansal (spatio-temporal) özellikleri çıkarabilen bir CNN-Attention blok yapısı ile donatılmıştır. Ayrıca, dengesiz sınıflara yönelik öğrenme sürecini dengelemek için Eşitleme Kayıp v2 (Equalization Loss v2) yöntemi kullanılmıştır. UNSW-NB15, NSL-KDD, CIC-IDS-2017, CIC-DDOS-2019 veri setlerinde sırasıyla %89,39, %99,77, %99,88, %99,58 doğruluk başarı oranları elde edilmiştir (Ren, Yuan, Zhang, Shi & Huang, 2023).

Akgun ve Cavusoglu (2022), çalışmalarında, DNN, CNN ve LSTM gibi mimarileri kullanmışlardır. Önerilen model, özellik seçimi, veri temizleme, normalizasyon gibi veri ön işleme teknikleriyle desteklenen bir derin öğrenme mimarisine dayanmıştır. Veri seti olarak CIC-DDOS-2019 kullanmışlardır. CNN tabanlı model ile çok sınıflı sınıflandırmada %99,30 doğruluk ve ikili sınıflandırmada %99,99 doğruluk oranı elde edilmiştir (Akgun, Hizal & Cavusoglu, 2022).

Hnamte ve Hussain (2023), çalışmalarında, ağ tabanlı siber saldırıların tespiti için derin öğrenmeye dayalı bir hibrit bir model (DCNNBiLSTM) sunmuşlardır. Bu modelde, Konvolüsyonel Sinir Ağları ve İki Yönlü Uzun Kısa Vadeli Hafıza (BiLSTM)

modellerinin güçlü yönlerini bir araya getirilerek dikkat mekanizmasıyla desteklenmiştir. Veri seti olarak CIC-IDS-2018 ve Edge_IIoT kullanılmıştır. CIC-IDS-2018 veri setinde doğruluk metriği olarak %100, Edge_IIoT veri setinde doğruluk metriği olarak %99,64 başarı oranları elde edilmiştir (Hnamte & Hussain, 2023).

Selvan ve arkadaşları (2022), çalışmalarında, iki aşamalı bir tespit mekanizması sunmuşlardır. Birinci aşamada, RideSOA (Sürücü Sosyal Optimizasyon) algoritmasından faydalanılarak ikili bir sınıflandırma yapılmıştır. İkinci aşamada, DNFN (Derin Nöro-Bulanık Ağlar) ve SSSA (Sosyal Sincap Arama Algoritması) kullanılarak saldırı türleri sınıflandırılmıştır. Veri seti olarak Bot-IoT kullanılmıştır. İkili sınıflandırmada F1 Skoru, kesinlik ve hassasiyet metriklerinde sırasıyla %86,76, %90,14, %83,62, çoklu sınıflandırma da ise aynı metrikler için sırasıyla %87,18, %92,54, %82,41 başarı oranları elde edilmiştir (Selvan, Azees, Vinodkumar & Parthasarathy, 2022).

Pooja ve Shrinivasacharya (2021), çalışmalarında İki Yönlü Uzun Kısa Vadeli Hafıza kullanmışlardır. Veri seti olarak KDDCUP-99 ve UNSW-NB15 kullanılmıştır. Aktivasyon fonksiyonları arasında Softmax ile en iyi sonuçlar elde edilmiştir. İkili sınıflandırma olarak, KDDCUP-99 veri setinde %99,73, UNSW-NB15 veri setinde %99,67 doğruluk metriği başarı sonuçları elde edilmiştir (Pooja & Shrinivasacharya, 2021).

Bouke ve arkadaşları (2023), çalışmalarında, Gini indeksi özellik seçimi ile geliştirilmiş bir karar ağacı modeli kullanmışlardır. Gini indeksi ile yapılan özellik seçimi sonucunda, 45 özellikten sadece 13'ü kullanılmıştır. Veri seti olarak UNSW-NB15 kullanılmıştır. Çoklu sınıflandırmada kesinlik, hassasiyet, F1 Skoru, doğruluk metriklerinde %98 başarı oranları elde edilmiştir. Bu oranlar hesaplanırken ağırlıklı ortalama yöntemi kullanıldığı için sonuçlar yanıltıcı olabilir (Bouke, Abdullah, Alshatebi, Abdullah & El Atigh, 2023).

Ali ve arkadaşları (2024), çalışmalarında, LSTM, Oto Kodlayıcı ve MLP (Çok Katmanlı Algılayıcı) kombinasyonuna dayalı bir model önermişlerdir. Çalışmada N-BAIoT2018 veri seti ve UNSW-NB15 veri seti kullanılmıştır. N-BAIoT2018 veri setinde ikili sınıflandırmada, doğruluk, kesinlik, hassasiyet, F1 Skoru metriklerinde sırasıyla %99,77, %99,92, %99,83, %99,87, çoklu sınıflandırmada sırasıyla en düşük %99,61, %99,50, %99,18, %99,34 başarı değerleri elde edilmiştir. UNSW-NB15 veri setinde ikili

sınıflandırmada, doğruluk, kesinlik, hassasiyet, F1 Skoru metriklerinde sırasıyla %99,67, %99,62, %99,78, %99,70, çoklu sınıflandırmada sırasıyla en düşük %98,30, %96,60, %97,60, %97,10 başarı değerleri elde edilmiştir (Ali vd., 2024a).

Sonuç olarak yukarıdaki çalışmalara bakılacak olursa nesnelerin interneti ağlarına yapılan saldırıları tespit etmek için LSTM, AE, CNN, DNN ve GAN gibi derin öğrenme algoritmaları hibrit olarak yoğun bir şekilde kullanılmaktadır. Bu çalışmalarda özellik çıkarımı işlemlerine önem verilmekte ve bu işlemler için genellikle, XGBoost ve Rastgele Orman gibi karar ağacı temelli algoritmalar, Gini indeksi, Fisher Skoru ve Korelasyon temelli gibi istatistiki özellik çıkarımı yöntemleri ve CNN, AE gibi derin öğrenme temelli algoritmalar yoğun bir şekilde kullanılmaktadır. IOT-23, UNSW-NB15, BOT-IOT, CIC-DDOS-2019, CIC-IDS-2018 veri setleri, bu çalışmalarda sıklıkla kullanılmaktadır.

1.2. Tezin Amacı

Bu tez, Nesnelerin İnterneti ağlarında meydana gelen güvenlik tehditlerini etkili bir şekilde tespit edebilecek, yüksek doğruluk oranına sahip ve farklı senaryolara uyum sağlayabilen bir saldırı tespit sistemi geliştirmeyi amaçlamaktadır.

Tez kapsamında, Nesnelerin İnterneti ağlarına yönelik saldırıların tespiti için modern derin öğrenme algoritmalarının (LSTM, GRU, CNN, AE) hibrit bir yapıda bir araya getirildiği bir model tasarlanmıştır. Bu modelin temel amacı, farklı Nesnelerin İnterneti veri setlerinde yüksek başarı oranıyla çalışarak saldırıları hızlı ve doğru bir şekilde tanımlamaktır.

Bu hedef doğrultusunda aşağıdaki adımların gerçekleştirilmesi planlanmaktadır:

- Nesnelerin İnterneti ortamlarında sıkça karşılaşılan saldırı türlerini analiz etmek ve bu saldırıların neden olduğu etkileri detaylı bir şekilde Farklı derin öğrenme algoritmalarının güçlü yönlerini bir araya getirerek hibrit bir model tasarlamak,
- Modeli değerlendirmek,
- IoT ağlarına özgü veri setlerini işleyerek saldırı tespit sisteminin doğruluğunu ve performansını optimize etmek,
- Dengesiz veri setleri üzerinde sentetik veri üretimi ve ön işleme tekniklerini kullanarak veri kalitesini artırmak ve daha güvenilir sonuçlar elde etmek.

Bu tez çalışması, Nesnelerin İnterneti ağlarının güvenliğini sağlamak için yenilikçi bir yaklaşım sunmayı, güvenlik tehditlerini en aza indirmeyi ve Nesnelerin İnterneti teknolojilerinin daha güvenilir bir şekilde kullanılmasına katkıda bulunmayı amaçlamaktadır. Aynı zamanda, geliştirilen modelin literatürdeki yöntemlere göre daha genelleştirilebilir ve etkili bir çözüm sunması hedeflenmektedir.

1.3. Tezin Hipotezleri

Nesnelerin İnterneti ağlarında saldırı tespit sistemleri, derin öğrenme algoritmalarının (LSTM, GRU, CNN, AE) hibrit bir şekilde kullanımı ile desteklendiğinde, klasik makine öğrenmesi yöntemlerine göre daha yüksek doğruluk oranı ile saldırıları tespit edilebilir.

Nesnelerin İnterneti ağlarındaki güvenlik zafiyetlerinin tespiti ve önlenmesindeki başarı, eksik verilerin tamamlanması, hatalı verilerin düzeltilmesi, veri normalizasyonu gibi veri ön işleme ve veri hazırlama süreçlerinin doğru ve etkin kullanılmasıyla birlikte önemli ölçüde artırılabilir.

Yapay zekâ ve makine öğrenimi tabanlı saldırı tespit sistemleri, Nesnelerin İnterneti ağlarında normalden sapma gösteren davranışları gerçek zamanlı olarak saptayabilir ve bu sayede ağ güvenliğini proaktif bir şekilde sağlayabilir.

CIC-IOT-2023, IOT-23, BOT-IOT, CIC-DDOS-2019, N-BAIOT ve UNSW-NB15 gibi veri setleri, IoT saldırı tespit sistemleri için yeterli çeşitlilik ve zenginlikte veri sağlayarak yüksek performanslı modellerin geliştirilmesine olanak tanıyabilir.

1.4. Tezin Kapsamı

Bu tezde, Nesnelerin İnterneti ağlarında karşılaşılan güvenlik tehditlerini ele alarak, bu tehditleri tespit etmek için kullanılan yöntemleri detaylı bir şekilde incelemektedir ve yeni bir saldırı tespit sistemi tasarlamayı amaçlamaktadır. Tez çalışması, Nesnelerin İnterneti teknolojilerinin güvenlik zafiyetlerinden yola çıkarak modern yapay zekâ ve makine öğrenmesi algoritmalarının Nesnelerin İnterneti güvenliğindeki rolüne kadar geniş bir alanı kapsamaktadır.

Tezin kapsamı şu şekilde belirlenmiştir:

1. Nesnelerin İnterneti Güvenlik Tehditlerinin İncelenmesi

Tez çalışmasında, Nesnelerin İnterneti ağlarına yönelik tehditler detaylandırılmış, sosyal mühendislik saldırıları, kötü amaçlı yazılımlar, kaba kuvvet saldırıları ve hizmet reddi saldırıları gibi çeşitli tehdit türleri ele alınmıştır. Bu saldırıların Nesnelerin İnterneti ağlarına olan etkileri incelenmiş ve mevcut literatür ışığında Nesnelerin İnterneti güvenlik sorunları ortaya konmuştur.

2. Nesnelerin İnterneti Mimarisi ve Güvenlik Açıkları

Nesnelerin İnterneti ağlarının üç katmanlı mimarisi, kullanılan protokoller ve bu protokollere yönelik tehditler analiz edilmiştir. Ayrıca saldırı tespit sistemlerinin Nesnelerin İnterneti ortamlarındaki önemi ve bu sistemlerin mevcut yapıları kapsamlı bir şekilde ele alınmıştır.

3. Klasik ve Derin Öğrenme Algoritmaları

Nesnelerin İnterneti güvenliği için kullanılan klasik makine öğrenmesi yöntemleri (SVM, KNN, Karar Ağaçları, Rastgele Orman, Naive Bayes vb.) ve modern derin öğrenme algoritmaları (CNN, LSTM, AE, GRU vb.) detaylandırılmıştır. Bu algoritmaların avantajları, kısıtlamaları ve Nesnelerin İnterneti ağlarına uygunluğu kapsam içinde değerlendirilmiştir.

4. Önerilen Yöntemlerin Tanıtımı

Hibrit derin öğrenme modelleri (CNN ve LSTM; AE ve LSTM) kullanılarak bir saldırı tespit sistemi geliştirilmiştir. Önerilen sistemin temel bileşenleri, veri ön işleme yöntemleri, modelin mimarisi ve Nesnelerin İnterneti ağlarındaki uygulanabilirliği tartışılmıştır.

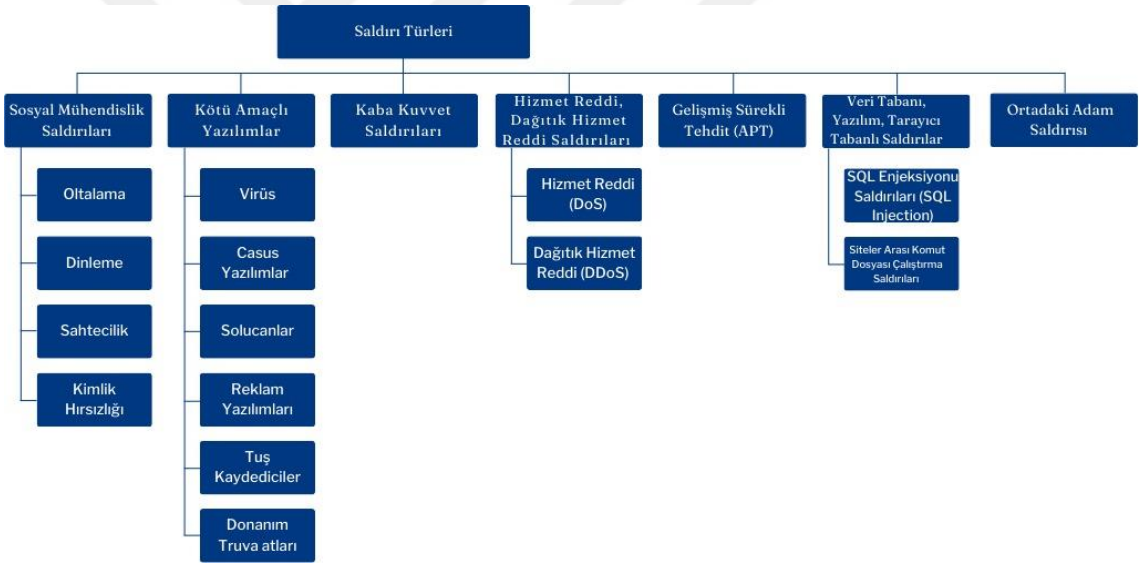
5. Deneysel Sonuçlar ve Performans Analizi

Geliştirilen sistemin performansı, farklı Nesnelerin İnterneti veri setleri (IOT-23, CIC-IOT-2023, UNSW-NB15, N-BAIOT, BOT-IOT, CIC-DDOS-2019, TON-IOT, CIC-IDS-2018, CIC-IDS-2017) üzerinde test edilmiş ve doğruluk oranı, kesinlik oranı, hassasiyet oranı, F1 Skoru, hata oranı gibi metrikler ile değerlendirilmiştir.

BÖLÜM 2

SİBER SALDIRI TÜRLERİ VE SALDIRI TESPİT SİSTEMLERİ

Siber saldırı, bir kişiye, ağa veya sisteme yönelik olarak doğrudan ya da ağ üzerinden gerçekleştirilen, bilgi çalma, izleme veya hizmet kesintisine yol açma gibi amaçları taşıyan eylemlerdir. Siber saldırılar, Şekil 2.1'de görüldüğü üzere farklı kategorilere ayrılmakta olup, bu kategoriler alt başlıklarda detaylı bir şekilde açıklanmaktadır.



Şekil 2.1. Siber Saldırı Türleri

2.1. Sosyal Mühendislik Saldırıları

Sosyal mühendislik saldırıları, insan zaaflarını ve psikolojisini manipüle ederek bilgiye veya sistemlere yetkisiz erişim sağlamayı hedefleyen bir saldırı stratejisi olarak tanımlanmıştır. Bu saldırılar, teknolojik güvenlik önlemlerini aşmayı denemek yerine, bireylerin güvenini kötüye kullanmayı veya dikkatlerini dağıtarak bilgi paylaşmalarını sağlamayı amaçlamıştır (Süzen, 2023).

Bu saldırı stratejisinin temelinde, saldırganların insanların duygusal tepkilerini ve davranışlarını analiz ederek bu zayıflıkları kendi lehlerine kullanmaları yatmaktadır. Bunun yanı sıra, sosyal mühendislik saldırılarının yalnızca psikolojik değil, aynı zamanda teknolojik araç ve yöntemlerle desteklendiği de gözlemlenmiştir (Aldawood & Skinner, 2020).

Sosyal mühendislik Saldırılarından Oltalama, Dinleme, Sahtecilik ve Kimlik Hırsızlığı saldırıları aşağıda detaylandırılmaktadır.

2.1.1. Oltalama

Oltalama (Phishing), bireylerden veya kuruluşlardan hassas bilgileri elde etmek için aldatıcı tekniklerin kullanıldığı yaygın bir siber tehdit türüdür. Bu saldırı türü, teknik aldatmaca ve sosyal mühendislik unsurlarını birleştirerek, giriş bilgileri, finansal detaylar veya kişisel bilgiler gibi gizli verilerin çalınmasını hedeflemektedir (Asiri, Xiao, Alzahrani, Li, & Li, 2023). Oltalama saldırıları genellikle meşru kaynaklardan gelen iletişimleri taklit eden sahte e-postalar veya web siteleri aracılığıyla gerçekleştirilmektedir. Bu yöntemle saldırganlar, kullanıcıların kimlik bilgilerini veya diğer kritik bilgilerini paylaşmalarını sağlamayı hedeflemektedirler.

Saldırganlar, mağdurları kandırmak için meşru gibi görünen sahte bağlantılar paylaşabilmektedirler. Örneğin, saldırganlar bir e-posta ya da sosyal medya paylaşımı yoluyla kullanıcıları sahte bir web sitesine yönlendirebilir ve burada kullanıcı giriş bilgilerini çalabilirler (Lee, Lee, Lee, Kwon, & Shin, 2021). Oltalama yöntemleri zamanla daha karmaşık hale gelmektedir. Günümüzde saldırganlar, özellikle sosyal mühendislik ve kötü amaçlı yazılımlar kullanarak hedeflerini daha kolay bir şekilde kandırabilmektedirler. Bu bağlamda, mobil sosyal ağlardaki kimlik avı saldırıları gibi daha yeni hedefler de ortaya çıkmaktadır (Gupta, Singhal & Kapoor, 2016).

Güncel araştırmalar, oltalama saldırılarını tespit etmek ve önlemek için derin öğrenme gibi ileri düzey tekniklerin uygulanmasını içermektedir. Ancak, bu sistemlerin veri işleme gereksinimleri ve maliyetleri gibi bazı kısıtlamaları bulunmaktadır. Oltalama, özellikle kurumsal ortamlarda önemli güvenlik açıklarına yol açabilmektedir. Endüstriyel kontrol sistemleri gibi alanlarda, Hedefli Oltalama (Spear Phishing) gibi hedef odaklı saldırılar, hassas verilerin çalınması ve sistemlerin işleyişinin kesintiye uğramasına neden olabilir (Abeywardana, Pfluegel & Tunnicliffe, 2016).

2.1.2. Dinleme

Dinleme (Eavesdropping), bilgiye yetkisiz erişim sağlayarak iletişim sistemlerinde ciddi güvenlik tehditleri oluşturan bir saldırı türüdür. Bu saldırılar hem pasif hem de aktif yollarla gerçekleşebilmektedirler. Pasif dinleme, saldırganların sadece veri topladığı ve sisteme herhangi bir müdahalede bulunmadığı durumlardır. Aktif dinleme ise saldırganların veri iletimini manipüle ettiği veya iletişim süreçlerine müdahale ettiği durumları kapsamaktadır (Huo, Mitran & Gong, 2016).

RFID (Radyo Frekansı ile Tanımlama) sistemlerinde, dinleme saldırıları, etiketlerden ve okuyuculardan geçen bilgilerin izlenmesiyle gerçekleşebilmektedirler. Bu sistemlerde kullanılan Frekans Atlamalı Yayılım Spektrumu (FHSS), dinlemeye karşı bazı korumalar sunsa da aktif saldırganlar bu korumaları aşabilmektedirler. Özellikle, RFID sistemlerinde aktif dinleme, saldırganların kendi sinyallerini kullanarak sistemden veri sızdırmasına olanak tanımaktadır (Huo, Mitran & Gong, 2016).

WiNoC (Çip Üzerinde Kablosuz Ağlar, Wireless Network-on-Chip) teknolojileri, çip üzerindeki veri iletimi için kablosuz bağlantıları kullanarak geleneksel kablolu sistemlere kıyasla önemli avantajlar sunmaktadır. Bu yenilikçi teknoloji, daha yüksek enerji verimliliği sağlarken, veri transferinde yaşanan gecikmeleri de azaltmaktadır. Ancak, kablosuz iletişim yöntemlerinin kullanılması, veri transferlerinin hava yoluyla gerçekleşmesi nedeniyle güvenlik açısından bazı riskler taşımaktadır. Bu durum, iletişim sinyallerinin saldırganlar tarafından kolayca dinlenebileceği bir ortam yaratmaktadır (Vashist, Keats, Pudukotai Dinakarrao & Ganguly, 2019).

Otonom araçlar ve akıllı taşıma sistemleri gibi modern uygulamalarda, Araçların İnterneti (Internet of Vehicles (IoV)) teknolojisi, araçların birbirleriyle ve altyapı ile iletişim kurmasını sağlamaktadır. Bu sistemlerde, dinleme, özellikle trafik bilgileri gibi kritik verilerin güvenliğini tehlikeye atabilmektedir. Araçlar arası iletişimde güvenlik, bu yüzden büyük önem taşır ve güvenli iletişim protokolleri, verilerin korunması için önemlidir (Bajracharya, 2022).

Dinleme'ye karşı korunma yöntemleri arasında şifreleme, dinamik frekans seçimi ve çoklu yol iletimi bulunmaktadır. Bu teknikler, verilerin izlenmesini zorlaştırır ve böylelikle iletişim güvenliğini artırmaktadır. Ayrıca, makine öğrenimi algoritmalarını

kullanarak anormal ağ aktivitelerini tespit etmek ve dinleme saldırılarına karşı proaktif önlemler almak mümkündür (Balakrishnan, 2019).

2.1.3. Sahtecilik

Sahtecilik (Spoofing), modern iletişim ve kimlik doğrulama sistemlerinde ciddi bir güvenlik tehdidi oluşturmaktadır. Bu saldırılar, yanıltıcı bilgiler sağlayarak veya sistemleri yanıltarak cihazların veya sistemlerin yanlış davranışlar sergilemesine neden olabilmektedir. Sahtecilik, genellikle bilgisayar ağları, GPS, yüz tanıma ve ses tanıma sistemleri gibi farklı teknolojilerde meydana gelebilmektedir (Naqvi vd., 2023).

Yüz tanıma sistemlerinde, kullanıcıların fotoğraf veya video gibi sahte görseller kullanarak sistemi aldatma girişimleri sıkça görülmektedir. Bu tür saldırıları önlemek ve tespit etmek amacıyla, doku analizi, görüntü kalitesi değerlendirmeleri ve hareket analizi gibi yöntemler kullanılmaktadır (Anthony, Ay & Aydin, 2021). Örneğin, yüz tanıma sistemleri, bir kişinin yüzünün canlılığını doğrulamak için doku düzeni ve hareket gibi özellikleri detaylı bir şekilde analiz edebilmektedir (Yang, Li, Li & Wang, 2019).

Sesli komut sistemlerinde, ses sahteciliği, sistemi yanıltıcı veya aldatıcı ses komutları ile manipüle etmeyi içermektedir. Bu tür saldırılara karşı önlemler geliştirilmiştir. Bu önlemler arasında ses özelliklerini analiz ederek gerçek kullanıcıların sesini taklit eden sahte sesleri tespit etmek sıralanabilmektedir (Wang & Yamagishi, 2023).

Ayrıca, siber-fiziksel sistemler üzerindeki GPS sahteciliği saldırıları da önemli bir tehdit oluşturmaktadır. GPS sahteciliği, bir GPS alıcısının yanlış konum bilgilerini kabul etmesine neden olabilmektedir. Bu tür saldırıları tespit etmek için, GPS ölçümlerinin diğer sensörlerle tutarlılığını analiz eden çoklu sensör entegrasyon teknikleri kullanılabilmektedir (Wagner & Fröhlich, 2024). Bu teknikler, GPS verilerinin diğer verilerle karşılaştırılması suretiyle herhangi bir tutarsızlık durumunda alarm verebilmektedir (Chen, Lu, Weng & Lin, 2023).

Yukarıda bahsedilen teknolojilerin sürekli geliştirilmesi, sahtecilik tehditlerine karşı etkili bir savunma sağlamaktadır ve bu tür saldırılara karşı korumayı artırmaktadır.

2.1.4. Kimlik Hırsızlığı

Kimlik Hırsızlığı (Identity Theft), bir kişinin kişisel bilgilerini yasa dışı yollarla kullanarak dolandırıcılık yapma veya maddi ya da diğer türden çıkarlar elde etme işlemidir. Teknolojinin gelişmesiyle birlikte, kimliklerin çalınma yöntemleri giderek daha karmaşık bir hale gelmiş, oltalama, bilgisayar korsanlığı, veri sızıntıları gibi taktikler devreye girmektedir. Bu durum, mağdurlar için önemli mali kayıplara ve kişisel zorluklara yol açmaktadır (Cheng, 2023).

Kimlik Hırsızlığı yalnızca bireysel bir sorun değildir, aynı zamanda kuruluşları ve işletmeleri de etkilemektedir. Şirketler, hassas verilerin çalınmasında Kimlik Hırsızlığının merkezi bir rol oynadığı dijital dolandırıcılık girişimlerine giderek daha fazla hedef olmaktadır. Pandemi sırasında dijital trafiğin artması, bu tür suç faaliyetlerinde bir artışa yol açmıştır (Li, Pan, Chen, Hoang & Wang, 2023).

Kimlik Hırsızlığına karşı koruma, antivirüs yazılımı gibi teknolojik çözümleri ve kişisel bilgilerin çevrimiçi paylaşımı konusunda daha bilinçli bireysel uygulamaları içermektedir. Bununla birlikte, mevcut teknolojilere ve bilgilere rağmen, koruyucu önlemlerin yeterince etkin bir şekilde uygulanamaması, bilgilerin dinlenmesi ve ele geçirilmesi güvenlik risklerinin ortaya çıkmasına neden olmaktadır (Cheng, 2023).

Bireylerin kendilerini Kimlik Hırsızlığına karşı koruma motivasyonunu anlamak için, PMT (Koruma Motivasyon Teorisi) gibi teoriler uygulanmıştır. Bu teori, bir tehdidin algılanması ve başa çıkma stratejilerinin değerlendirilmesinin koruyucu davranışları nasıl etkilediğini araştırmaktadır. Teori, bir bireyin bir tehdide yanıtının, algılanan tehdidin şiddeti ve potansiyel yanıtın etkinliği tarafından motive edildiğini; kişisel yeterlilik ile tehditten kaynaklanan korku veya endişe tarafından şekillendirildiğini öne sürmektedir (Cheng, 2023).

2.2. Kötü Amaçlı Yazılımlar

Kötü Amaçlı Yazılımlar (Malware), bir sistemde hasar oluşturmak veya kullanıcı verilerini izlemek, değiştirmek, çalmak amacıyla geliştirilen yazılımları ifade etmektedir. Bu tür yazılımlar, bilgisayar sistemlerine zarar verme, bilgileri ele geçirme ya da kullanıcının güvenliğini tehlikeye atma gibi amaçlarla çalışmaktadırlar. Kötü Amaçlı

Yazılımlar hem özel yazılımlardaki hem de açık kaynak kodlu yazılımlardaki güvenlik açıklarını kullanarak hedef sistemlerde veri ihlali oluşturmaktadırlar. Genellikle meşru programların içine gizlenen bu zararlı yazılımlar, gelişmiş güvenlik tedbirleri olmadan kolaylıkla fark edilememektedirler. Aşağıda sıklıkla karşılaşılan Kötü Amaçlı Yazılım türleri detaylandırılmaktadırlar.

Kötü Amaçlı Yazılımlardan Bilgisayar Virüsleri, Casus Yazılımlar, Solucanlar, Reklam Yazılımları, Tuş Kaydediciler, Donanım Truva Atları aşağıda detaylandırılmaktadır.

2.2.1. Bilgisayar Virüsleri

Bilgisayar Virüsleri (Computer Viruses), bilgisayar sistemlerine istem dışı olarak bulaşan, sistemin normal işleyişini bozarak veri ve programlara zarar verebilen zararlı yazılımlar olarak tanımlanmaktadır. Bilgisayar virüsleri, genellikle bir program veya dosya aracılığıyla sisteme sızar ve kendilerini çoğaltarak diğer sistem bileşenlerine bulaşmaktadırlar. Bu virüsler, yayılmak ve bulaşmak için sistemdeki dosyalara eklenir ya da onları manipüle etmektedirler. Bilgisayar virüsleri, çalışma sırasında sistem kaynaklarını tüketerek sistem performansını düşürebilir ve veri kaybına sebep olabilmektedirler (Subramanya & Lakshminarasimhan, 2001).

Bilgisayar virüsleri farklı türlerde olabilmektedirler. Örneğin, Dosya Bulaşıcı Virüsler (File Infectors), belirli program dosyalarına eklenir ve o program çalıştığında virüs etkinleşmektedir. Sistem veya Başlatma Kaydı Bulaşıcıları (Boot-record Infectors), doğrudan disk başlatma kayıtlarına yerleşmektedir ve bu şekilde çalışarak sistemde yayılmaktadırlar. Bunun yanında, Çok Parçalı Virüsler (Multi-partite Viruses) hem dosya hem de başlatma kaydı üzerinde etkili olup daha geniş kapsamda bulaşma yeteneğine sahiptirler. Makro virüsleri ise özellikle Microsoft Word ve Excel gibi uygulamalardaki makroları kullanarak yayılmaktadırlar. Bu virüsler, açılan bir belge veya çalışma kitabı ile etkinleşip diğer dosyalara bulaşabilmektedirler. Ayrıca, Polimorfik Virüsler (Polymorphic Viruses) gibi daha karmaşık yapıda olanlar, her bulaşma sürecinde yapılarını değiştirerek tespit edilmelerini zorlaştırmaktadırlar (Subramanya & Lakshminarasimhan, 2001).

Bilgisayar virüslerinin tespiti için kullanılan yöntemler, gelişen teknoloji ile sürekli olarak yenilenmektedir. Geleneksel virüs tespit yöntemleri, çoğunlukla imza tabanlı olup, belirli bir virüs imzasını içeren dosyaları tanımlamaya yöneliktirler. Ancak bu yöntemler yalnızca veri tabanında kayıtlı olan, yani bilinen virüsleri tanımlayabilmektedirler. Bu nedenle, günümüzde makine öğrenimi tabanlı yöntemler gibi daha ileri tespit tekniklerine ihtiyaç duyulmaktadır (Qingyu & Dawei, 2010). Makine öğrenimi tabanlı virüs tespiti, özellik vektörleri kullanarak dosyaları eğitilen sınıflandırıcılara sunmaktadır ve böylece yeni dosyaların virüs veya normal dosya olarak sınıflandırılmasına yardım etmektedir. Özellikle aktif öğrenme teknikleri, daha az sayıda etiketli veri ile yüksek doğruluk oranına sahip modellerin geliştirilmesine imkân tanımaktadır (Qingyu & Dawei, 2010). Bu teknikler, sürekli güncellenen veri setleri ile eğitilerek yeni virüs türlerinin de tespit edilmesini sağlamaktadır.

Dinamik analiz ve davranışsal analiz gibi yöntemler, virüslerin sistemdeki davranışlarını izleyerek zararlı faaliyetleri tespit etmeye çalışmaktadırlar. Dinamik analiz, şüpheli dosyaları sanal bir ortamda çalıştırarak, virüsün sistem üzerindeki etkilerini inceleyerek zararlı davranışları tespit etmeye odaklanmaktadır (Yangchun, Zhao & Yang, 2020). Bu tür analizler, şifreleme veya gizleme teknikleri kullanılarak kendini gizlemeye çalışan virüslerin tespit edilmesine olanak tanımaktadırlar. Dinamik analiz yöntemleri genellikle sanal makineler veya Korumalı Alan (Sandbox) ortamları kullanılarak gerçekleştirilebilmektedir. Korumalı Alan ortamında şüpheli dosyalar izole edilmiş bir sanal ortamda çalıştırılarak, dosyanın sistem üzerinde yaptığı değişiklikler incelenir. Böylece gerçek sistem etkilenmeden zararlı yazılımın davranışları gözlemlenebilmektedir (Yangchun vd., 2020).

Yeni nesil güvenlik çözümleri arasında, virüslerin bulaşma tekniklerini inceleyerek sistemlere nasıl sızdıklarını tespit etmek önemli bir yer tutmaktadır. Özellikle polimorfik ve metamorfik virüsler, her bulaşma sürecinde kendilerini yeniden yapılandırarak farklı bir imza ile yayılabilmektedirler. Bu tür virüslerin tespit edilmesi, yalnızca imza tabanlı sistemlerle mümkün değildir. Bu yüzden, gelişmiş davranışsal analiz ve makine öğrenimi yöntemleri gibi yeni teknolojilere ihtiyaç duyulmaktadır (Lang & Zilong, 2022). Polimorfik virüslerin tespitinde kullanılan algoritmalar, virüsün kod yapısını değiştirerek tespit edilmesini engelleme girişimlerine karşı sürekli olarak güncellenmelidirler.

2.2.2. Casus Yazılımlar

Casus Yazılımlar (Spyware), kullanıcıların izni olmaksızın bilgisayar sistemlerine sızarak kişisel bilgileri toplayan ve bu bilgileri üçüncü taraflara aktaran zararlı bir yazılımdır (Wazid vd., 2013; Kilic & Yaman, 2024a; Sheta, El Hadad, M & Zaki, 2016). Casus yazılımlar, internet üzerindeki etkinlikleri izleyebilmekte, hassas verileri çalabilmektedir ve reklam amaçlı içerikleri yönlendirebilmektedirler. Casus yazılımlar, genellikle kullanıcının bilgisi dışında, indirilen yazılımlarla birlikte gizlice yüklenirler ve sistem kaynaklarını kullanarak cihazın performansını düşürebilmektedirler.

Casus Yazılımlar'ın tehlikeli olmasının diğer bir sebebi, bu yazılımların çeşitli formlarda bulunabilmesidir. Tuş Kaydediciler (Keyloggers), Reklam Yazılımları (Adware), sistem izleyicileri gibi çeşitli türleri vardır. Örneğin, Tuş Kaydediciler, kullanıcıların klavye ile girdiği bilgileri kaydeder ve bu bilgileri dışarıdaki bir saldırgana göndermektedir. Bu, çok hassas bilgilerin çalınmasına yol açabilmektedir (Wazid vd., 2013).

Casus Yazılım tehditlerinin tanınması ve engellenmesi için çeşitli yöntemler geliştirilmiştir. Bu yöntemlerden biri olan davranış bazlı tespit yöntemleri, yazılımın normal dışı davranışlar sergilemesi durumunda alarm vermektedir. Örneğin, bir yazılımın beklenmedik şekilde klavye girdilerini kaydetmeye başlaması bu tür davranışlardan biridir (Kilic & Yaman, 2024a).

Bunun yanı sıra, bazı Casus Yazılım türleri, sistemdeki güvenlik yazılımlarını devre dışı bırakarak kendilerini gizleyebilir ve bu, onları tespit etmeyi zorlaştırmaktadır. Sheta ve arkadaşları (2016) tarafından önerilen güvenlik tasarım desenleri, Casus Yazılım gibi zararlı yazılımların tespit edilmesi ve izole edilmesi konusunda yenilikçi çözümler sunmaktadır (Sheta, El Hadad, M & Zaki, 2016). Bu çözümler, veri madenciliği teknikleri ve tasarım desenlerini kullanarak daha etkili bir koruma sağlamayı hedeflemektedirler. Gelişmiş koruma yöntemlerinden biri olan dinamik analiz, yazılımın çalışma zamanı sırasında davranışlarını izler ve anormal etkileri tespit etmektedir. Bu yöntem, daha önce belirtilen çalışmalarda bahsedilen statik analiz yöntemlerine kıyasla daha güçlü olabilmektedir ve zararlı yazılımın sisteme verdiği zararı en aza indirgemek için kullanılmaktadır (Kilic & Yaman, 2024b).

2.2.3. Solucanlar

Solucanlar (Worms), siber güvenlik alanında ciddi bir tehdit olarak karşımıza çıkmaktadır. Bu saldırı türü, ağlardaki cihazlara hızla yayılan ve kontrol edilemeyen zararlı yazılımlar aracılığıyla gerçekleşmektedir. Solucanlar kendilerini çoğaltabilen ve bir kez yayılmaya başladığında kullanıcı müdahalesi olmadan milyonlarca cihazı enfekte edebilen zararlı yazılımlardır. Bu saldırı türü, internet bağlantılı her cihaz için büyük bir risk oluşturur ve özellikle zayıf güvenlik önlemleri olan cihazlar Solucanlar için kolay hedeflerdir (Zhou vd., 2020).

Solucan Saldırıları, ilk kez 1988 yılında ortaya çıkan Morris Worm ile dikkat çekmiştir. O zamandan bu yana, Code Red, SQL Slammer, Conficker gibi birçok yüksek profilli solucan saldırısı meydana gelmiştir. Günümüzde ise Mirai ve Hajime gibi modern solucanlar, özellikle Nesnelerin İnterneti cihazlarını hedef alarak yayılmaya devam etmektedirler (Pham Anh & Nakamura, 2024; Li, Sisodia & Stafford, 2023).

Solucanlar, genellikle hedef cihazların güvenlik açıklarından yararlanarak saldırı başlatmaktadır. Bir solucan, bir cihaza bulaştıktan sonra ağ üzerindeki diğer cihazlara yayılmak için zayıf parolaları veya açık portları taramaktadır. Bu tür zararlı yazılımlar, hedef cihazlara zarar verebilmekte, onları uzaktan kontrol edilebilir hale getirebilmekte veya büyük çaplı saldırılara, örneğin Dağıtık Hizmet Reddi saldırılarına zemin hazırlamaktadır (Shaashua & Shaashua, 2019).

Geleneksel solucanların yanı sıra, günümüzde daha akıllı olarak adlandırılan solucanlar da mevcuttur. Bu tür solucanlar, tespit edilmemek için çeşitli kaçınma teknikleri kullanmaktadırlar. Örneğin, bu solucanlar tespit sistemlerine karşı çeşitli karşı önlemler geliştirebilir ve ağda normal trafiğe benzer şekilde davranarak fark edilmeden yayılmaya çalışmaktadırlar (Li, Sisodia & Stafford, 2023).

Solucan Saldırıları'nın tespiti zorlu bir süreçtir, çünkü bu saldırılar çok hızlı bir şekilde gerçekleşir ve birçok ağ cihazını aynı anda etkileyebilmektedirler. Özellikle geleneksel imza tabanlı saldırı tespit sistemleri, bu tür dinamik ve sürekli gelişen tehditlerle başa çıkmakta yetersiz kalabilmektedirler Modern solucanların tespiti için davranış temelli tespit yöntemleri kullanılmaktadır. Bu yöntemler, ağdaki olağandışı bağlantı ve trafik aktivitelerini izleyerek saldırıyı fark etmeye çalışmaktadır (Li, Sisodia & Stafford, 2023). Ayrıca, bazı çalışmalar, solucan tespitinde ağın farklı noktalarından

gelen veri akışlarını karşılaştırmak yerine, ağdaki trafik akışlarının zaman içindeki değişikliklerini izleyen sistemler geliştirmişlerdir (Hua, Zhou & Zhong, 2021).

Solucan Saldırıları'na karşı mücadele etmek için çeşitli savunma mekanizmaları geliştirilmiştir. Bu mekanizmalar arasında Hareketli Hedef Savunması (Moving Target Defense - MTD) öne çıkmaktadır. MTD, ağ yapısını ve sistem konfigürasyonlarını sürekli olarak değiştirerek saldırganların hedefe odaklanmasını zorlaştırmaktadır (Tian, Tan, Guan, Xu & Liu, 2020). Bunun yanı sıra, yapay zekâ ve makine öğrenimi teknikleri kullanılarak oluşturulan saldırı tespit sistemleri de solucanlara karşı etkin bir savunma aracı olarak kullanılmaktadır (Ali vd., 2024b).

Solucan saldırıları, internet bağlantılı cihazların hızla yayıldığı günümüzde ciddi bir tehdit olarak varlığını sürdürmektedir. Mirai, Hajime ve diğer modern solucanlar, özellikle Nesnelerin İnterneti cihazları hedef alarak büyük ölçekli siber saldırılara neden olmaktadırlar. Bu saldırıları tespit etmek ve engellemek için geliştirilen yeni nesil savunma yöntemleri, saldırganların sürekli olarak gelişen taktiklerine karşı önemli bir savunma hattı oluşturmaktadır. Ancak, bu saldırılarla başa çıkmak için daha ileri düzey güvenlik stratejilerine ve proaktif (önleyici ve sorunlar ortaya çıkmadan önce harekete geçmeyi amaçlayan) yaklaşımlara ihtiyaç duyulmaktadır (Bailey, Cooke, Jahanian & Watson, 2005).

2.2.4. Reklam Yazılımları

Reklam Yazılımları (Adware), genellikle kullanıcıların cihazlarına onların bilgisi dışında yüklenen ve reklam gösterme amacı taşıyan bir yazılım türüdür. Reklam Yazılımları, kullanıcıların internette gezinme davranışlarını izleyerek, bu verilere dayalı reklamlar göstermektedir. Reklam Yazılımları, kullanıcıya çevrimiçi olduğu sırada, çoğunlukla istenmeyen içerikler taşıyan reklam materyalleri izletmektedir veya sisteme indirmektedir (Aboosh & Aldabbagh, 2021). Reklam Yazılımları, çoğunlukla ücretsiz yazılımlarla birlikte gelir ve kullanıcı farkında olmadan cihazına yüklenmektedir. Ayrıca, bu tür yazılımlar, kullanıcıların kişisel verilerini toplamaktadır ve bu bilgileri reklam verenlerle paylaşarak kullanıcıların daha hedef odaklı reklamlara maruz kalmalarına sebep olmaktadırlar (Gao, Li, Kong, Bissyande & Klein, 2019).

Reklam Yazılımları'nın sadece reklam gösterme amacı taşımasının ötesinde, zamanla daha karmaşık hale geldiği ve kullanıcıların gizliliğini ihlal eden bir tehdit

oluşturduğu görülmektedir. Reklam Yazılımları, kullanıcı makinesine reklam göstermek amacıyla yüklenen bir yazılım olarak tanımlanabilmektedir. Bazı Reklam Yazılımı türleri, kullanıcıların kişisel verilerini toplar, bu bilgileri üçüncü taraflarla paylaşır ve hatta daha fazla zararlı yazılımın cihazlara yüklenmesine olanak tanımaktadır (Shahzad, Lavesson & Johnson, 2011). Bu durum, Reklam Yazılımı'nın yalnızca bir reklam yazılımı olmanın ötesine geçtiğini ve zararlı yazılım olarak değerlendirilmesi gerektiğini göstermektedir. Yapılan bazı araştırmalar, Reklam Yazılımı'nın %50'ye varan oranlarda zararlı davranışlar sergileyebileceğini ortaya koymaktadırlar (Gao vd., 2019).

Reklam Yazılımları, genellikle açılır pencere (pop-up) reklamlar, afiş reklamlar veya internet tarayıcıları üzerinde yerleştirilen reklamlardan oluşmaktadır. Bu tür reklamlar, kullanıcı deneyimini olumsuz yönde etkilerken, sistem performansını da düşürebilmektedirler. Ayrıca, Reklam Yazılımları, kullanıcıların kişisel bilgilerini toplayarak gizliliği ihlal eder ve bu bilgiler, reklam hedefleme amacıyla kullanılmaktadır (Shahzad, Lavesson & Johnson, 2011). Bazı durumlarda, Reklam Yazılımları kullanıcının cihazını daha savunmasız hale getirerek, diğer zararlı yazılımların da yüklenmesine yol açabilmektedir (Aboosh & Aldabbagh, 2021).

Reklam Yazılımları tespiti konusunda geleneksel imza tabanlı yaklaşımlar yetersiz kalabilmektedir, bu nedenle makine öğrenimi ve yapay zekâ tabanlı yöntemler Reklam Yazılımı tespitinde daha etkili olmaktadırlar. Statik ve dinamik analiz yöntemlerini kullanan bu sistemler, Reklam Yazılımları'nın davranışlarını inceleyerek yüksek doğruluk oranları ile tespit yapabilmektedirler. Bazı araştırmalar, bu tür sistemlerin %97 gibi yüksek tespit oranlarına sahip olduğunu göstermektedirler (Ideses & Neuberger, 2014). Reklam Yazılımları'nın hem kullanıcı gizliliğini ihlal etmesi hem de zararlı yazılım davranışları sergilemesi, bu yazılımın ciddi bir güvenlik riski taşıdığını ortaya koymaktadır (Gao, Li, Kong, Bissyandé & Klein, 2019).

Reklam Yazılımları, kullanıcıların cihazlarına genellikle ücretsiz yazılımlar aracılığıyla ve kullanıcı farkında olmadan bulaşmaktadır. Reklam Yazılımları kütüphaneleri, mobil uygulamalarda sıklıkla kullanılır ve bu kütüphaneler, kullanıcıların kişisel bilgilerini toplayarak üçüncü taraflarla paylaşabilmektedir. Bu durum, kullanıcı gizliliğini tehlikeye atmaktadır ve Reklam Yazılımı'nı sadece bir reklam gösterme

yazılımı olmaktan çıkararak, daha karmaşık bir güvenlik tehdidi haline getirmektedir (Gao vd, 2019).

2.2.5. Tuş Kaydediciler

Tuş Kaydediciler (Keyloggers), kullanıcının klavye tuş vuruşlarını kaydederek gizli bilgiler toplamak amacıyla kullanılan bir casus yazılım türüdür (Wazid, Katal, Goudar, Singh & Tyagi, 2012). Bu tür yazılımlar, tuş vuruşlarını kaydederek, şifreler, banka hesap bilgileri ve diğer hassas bilgilerin üçüncü taraflara iletilmesine yardım etmektedir (Bhat, Namratha & Mohana, 2020). Tuş Kaydediciler hem donanım tabanlı hem de yazılım tabanlı olarak sınıflandırılabilmektedirler. Donanım tabanlı Tuş Kaydediciler, klavye ile bilgisayar arasına yerleştirilen fiziksel cihazlarken; yazılım tabanlı olanlar ise işletim sistemine entegre edilerek çalışan ve çoğu zaman kullanıcı tarafından fark edilmeyen yazılımlardır (Victoria, Adebiyi & Igbekele, 2021).

Yazılım tabanlı Tuş Kaydediciler, iki ana türde bulunur: Kullanıcı alanında (User-Space) çalışan Tuş Kaydediciler ve çekirdek alanında (Kernel-Space) çalışan Tuş Kaydediciler olmak üzere ikiye ayrılabilmektedir. Kullanıcı alanında çalışan Tuş Kaydediciler, işletim sisteminin sağladığı belirli API'leri kullanarak her tuş vuruşunu kaydetmektedir. Bu tür Tuş Kaydediciler, kolayca dağıtılabilmesi ve düşük seviyede sistem izinleri gerektirmemesi nedeniyle yaygındır. Diğer yandan, çekirdek alanında çalışan Tuş Kaydediciler daha derin erişimle işletim sisteminin çekirdek seviyesine yerleşir, bu durum da onların tespit edilmesini zorlaştırmaktadır (Safavian & Landgrebe, 1991).

Tuş Kaydediciler'in tespiti için çeşitli yöntemler geliştirilmektedir. Kanca Tabanlı (Anti-Hook) teknik, sistemde çalışan tüm işlemleri tarayarak, Tuş Kaydedicilerin kullandığı API çağrılarını tespit etmektedir. Bu yöntem, Tuş Kaydedicilerin işletim sistemindeki Kanca mekanizmasını kullanmalarına dayanarak çalışan programları belirler (Facchini, Giorgi, Saracino & Dini, 2020). Diğer bir tespit yöntemi ise Bal Küpü (Honeypot) gibi tuzak teknikleriyle casus yazılımı belirlemek ve tuş vuruşlarını saptamaktır (Da Costa, Papa, Lisboa, Munoz & de Albuquerque, 2019).

Yazılım tabanlı Tuş Kaydedicilerden korunmak için, sanal klavyeler veya rastgele klavye düzeni kullanan yöntemler tercih edilebilmektedir. Sanal klavye kullanımı, fiziksel tuşlara basılmadığı için tuş vuruşlarının kaydedilmesini zorlaştırmaktadır.

Rastgele klavye düzeni ise, her tuşa basıldığında klavye düzenini değiştirerek Tuş Kaydedicilerin anlamlı veri kaydetmesini engellemektedir. Ayrıca, Tuş Kaydedicilere karşı koruma sağlayan bazı yazılımlar, kullanıcıdan alınan tuş vuruşlarını izlemek yerine davranışsal özellikleri kullanarak casus yazılımı algılamaya çalışmaktadır (Safavian & Landgrebe, 1991).

Günümüzde, Tuş Kaydedicilere karşı gelişmiş güvenlik önlemlerinin alınması gerekmektedir. Tuş Kaydediciler, özellikle kimlik doğrulama ve finansal işlemler sırasında ciddi güvenlik riskleri doğurmaktadır. Kullanıcıların güvenliğini artırmak için, iki faktörlü kimlik doğrulama ve OTP (Tek Kullanımlık Şifre) gibi yöntemler kullanılarak Tuş Kaydediciler tehditleri azaltılabilmektedir (Tsimenidis, Lagkas & Rantos, 2022). Tuş kaydedicilerin saptanması ve etkili bir şekilde yönetilmesine yönelik yöntemlerin güncellenmesi, güvenlik açıklarını en aza indirmek açısından büyük öneme sahiptir (Venkatraman vd., 2021).

2.2.6. Donanım Truva Atları

Donanım Truva Atları (Hardware Trojans, entegre devreler (IC)) gibi elektronik donanımlara, tasarım veya üretim süreçlerinde kötü niyetli değişiklikler yapılması olarak tanımlanmaktadır. Bu tür değişiklikler, cihazın işlevselliğini etkileyebilmektedir, hassas bilgilerin sızdırılmasına yol açabilir veya sistem arızalarına neden olabilmektedir (Liu, Zhao & Chen, 2019).

Günümüzde, IC'lerin küresel olarak tasarlanması ve üretilmesi, Donanım Truva Atları'nın ortaya çıkışını daha da mümkün kılmaktadır. Bu durum, birçok güvenilmez tarafın süreç içerisine dahil olmasıyla güvenlik risklerini arttırmaktadır (Bhunia, Hsiao, Banga & Narasimhan, 2014).

Donanım Truva Atları'nı tespit etmek için çeşitli yöntemler mevcuttur. Yapısal özelliklerin analizi ve güç, zamanlama ve elektromanyetik emisyonlar gibi yan kanal işaretlerinin analizi, Donanım Truva Atı barındıran IC'lerin tespit edilmesinde kullanılan başlıca yöntemler arasındadır. Bu işaretler, Donanım Truva Atı içermeyen çeşitli modellerle karşılaştırılarak değerlendirilmektedir (Salmani, Tehranipoor & Plusquellic, 2012).

Tersine mühendislik, entegre devrelerin (IC) katmanlarının ve bileşenlerinin sökülüp incelenmesiyle, tasarım özelliklerine uymayan farklılıkların belirlenmesi için

kullanılmaktadır. Bu yöntem, Donanım Truva Atları'nın tespitinde oldukça etkili bir araçtır (Bao, Forte & Srivastava, 2016).

Gelişmiş algılama araçları ve yöntemlerinin yanı sıra, makine öğrenimi tekniklerinin kullanımı da Donanım Truva Atı bulaşmış ve bulaşmamış IC'ler arasında ayrım yapma kapasitesini artırmaktadır (Vakil, Mirzaeian, Homayoun, Karimi & Sasan,2021).

Sonuç olarak, Donanım Truva Atları'na karşı etkili bir savunma geliştirilmesi hem fiziksel hem de işlevsel testlerin birleşimini gerektirmektedir (Shabani & Alizadeh, 2021).

2.3. Kaba Kuvvet Saldırıları

Kaba Kuvvet Saldırıları (Brute Force), bilgi güvenliği alanında yaygın olarak karşılaşılan ve saldırganların sistemlere yetkisiz erişim sağlamak için şifre olarak tüm olası kombinasyonları denediği bir yöntemdir. Bu saldırılar, özellikle şifreli verileri çözmek veya erişim sağlamak için kullanılır ve zayıf kimlik doğrulama sistemlerine karşı özellikle etkilidir.

Kaba Kuvvet saldırılarının başarılı olmasının temel nedenlerinden biri, insanların tahmin edilmesi kolay ve sık kullanılan şifreleri tercih etmeleridir. Bu tür saldırılar, genellikle otomatik araçlar kullanılarak gerçekleştirilir ve kısa sürede binlerce veya hatta milyonlarca kombinasyon denenebilmektedir (Zhang, Xie & Xu, 2020). Özellikle SSH gibi uzaktan erişim protokolleri, Kaba Kuvvet saldırıları için sıklıkla tercih edilen hedeflerdir. Bu tür saldırılar, ağ güvenliği açısından ciddi tehditler oluşturmaktadırlar (Tiwari & Hubballi, 2023).

Kaba Kuvvet saldırılarının etkisini azaltmanın yollarından biri, karmaşık şifre politikaları uygulamaktır. Bu politikalar, kullanıcıların tahmin edilmesi zor şifreler kullanmasını zorunlu kılmaktadır. Ancak, sadece güçlü şifre politikaları uygulaması yeterli olmayabilir. Çok katmanlı güvenlik önlemleri, saldırı tespit sistemleri ve davranışsal analiz teknikleri de önem taşımaktadır (Patil, Laturkar, Athawale, Takale & Tathawade, 2017)

Günümüzde, Kaba Kuvvet saldırılarını tespit etmek ve önlemek için gelişmiş teknolojiler ve yöntemler kullanılmaktadır. Örneğin, steganografi araçları kullanılarak

saklanmış veriler üzerinde Kaba Kuvvet saldırıları gerçekleştirilebilir ve bu saldırılar, paralel hesaplama teknikleri kullanılarak daha etkin hale getirilebilir (Bhowal, Dutta & Mitra, 2017). Ayrıca, yapay zekâ teknolojileri, saldırıları tespit etmede ve güvenlik sistemlerinin savunmasını artırmada kullanılmaktadır (Kumari, Gupta & Uppal, 2024).

Kaba Kuvvet saldırılarına karşı koymak için kullanılan stratejiler arasında, kullanıcıların şifre güvenliğini artırmaları, sistem yöneticilerinin düzenli olarak güvenlik kontrolleri yapmaları ve güvenlik duvarları ile erişim kontrol sistemlerini güçlendirmeleri yer almaktadır. Ayrıca, merkezi olmayan ve senkronize olmayan sistemlerde Asenkron Kaba Kuvvet saldırılarına karşı savunma mekanizmaları geliştirmek, birden fazla saldırı vektörüne karşı koruma sağlamaktadır (Merhav & Cohen, 2020)

Sonuç olarak, Kaba Kuvvet saldırılarına karşı etkili savunma stratejileri geliştirmek, günümüzün dijital dünyasında hayati önem taşımaktadır. Bu saldırılar sadece bireysel kullanıcıları değil, büyük ölçekli kurumsal sistemleri de tehdit edebilir ve geniş çaplı veri ihlallerine yol açabilmektedirler. Bu nedenle, sürekli gelişen siber tehditlere karşı proaktif ve çok katmanlı güvenlik yaklaşımlarını benimsemek gerekmektedir.

2.4. Hizmet Reddi, Dağıtık Hizmet Reddi Saldırıları

Hizmet Reddi (DoS) ve Dağıtık Hizmet Reddi (DDoS) saldırıları, hedef sistemin meşru kullanıcılar için erişilemez hale gelmesine neden olan siber saldırı türleridir. Bu saldırılar aşağıda detaylandırılmaktadır.

2.4.1. Hizmet Reddi

DoS saldırısı, tek bir kaynağın veya birkaç kaynağın kullanılmasıyla hedef sistemin kaynaklarının aşırı yüklenip normal işlevlerini yerine getiremez hale gelmesine neden olan bir saldırı türüdür. Bu saldırıda, saldırganlar bir bilgisayar veya az sayıda cihaz ile sisteme sürekli olarak trafik göndererek sunucu, ağ veya uygulama kaynaklarını tüketir ve meşru kullanıcıların hizmetlere erişimini engellemektedir. Bu saldırı tipik olarak, aşırı sayıda istek göndererek hedef sistemin yanıt veremez hale gelmesini sağlamayı amaçlamaktadır. SYN Flood gibi saldırılar, TCP'nin üç yönlü el sıkışma sürecini istismar eder ve sistem kaynaklarını hızla tüketmektedir (Dong & Sarem, 2020).

2.4.2. Dağıtık Hizmet Reddi

Dağıtı Hizmet Reddi Saldırıları (DDoS Saldırıları), daha geniş kapsamlı ve etkili bir hizmet engelleme saldırısı türüdür. Saldırganlar, kontrol ettikleri ve çoğunlukla zararlı yazılım ile ele geçirilmiş çok sayıda cihazı (botnet olarak bilinen bir ağ) kullanarak, hedef sisteme dağıtık bir saldırı gerçekleştirmektedir. Botnet içerisindeki her cihaz, aynı anda hedefe yüksek miktarda trafik gönderir ve kaynakların tükenmesine yol açmaktadır. DNS Yansıma ve NTP Yansıma saldırıları, mevcut protokollerin zafiyetlerini kullanarak hedefe büyük yanıtlar gönderilmesine olanak tanımaktadır (Pérez-Díaz, Valdovinos, Choo & Zhu, 2020; El Sayed, Abdelaziz, Hussein & Elbayoumy, 2024)

DoS ve DDoS saldırıları, güvenlik zafiyetlerinden yararlanarak hizmetlerin erişilemez hale getirilmesine yol açmaktadır. DDoS saldırıları, özellikle dağıtık yapısı sayesinde daha yıkıcı sonuçlar doğurabilir ve savunma mekanizmalarının aşılmasını kolaylaştırabilmektedir. Bu saldırılara karşı geliştirilen savunma mekanizmaları, ağ izleme, saldırı tespiti ve saldırı trafiğini filtreleme üzerine yoğunlaşmaktadır (Kumar, Somani & Agarwal, 2024; Almaraz-Rivera, Pérez-Díaz, Cantoral-Ceballos, Botero & Trejo, 2022).

Dağıtık Hizmet Reddi Saldırıları, aşağıdaki gibi detaylandırılabilmektedir.

Hacim Tabanlı Saldırılar

Bu saldırılar, hedefin ağ bant genişliğini doldurarak hizmetlerin aksamasına neden olmaktadırlar. Genellikle UDP Flood ve ICMP Flood gibi saldırılar bu kategoride yer almaktadır. UDP Flood saldırılarında, saldırgan bağlantısız UDP protokolünü kullanarak hedefe büyük miktarda veri göndermektedir. ICMP Flood ise hedefe çok sayıda ping isteği göndererek kaynak tüketimini hızlandırmaktadır (Bhayo, Hameed & Shah, 2020).

Protokol Tabanlı Saldırılar

Bu saldırılar, ağ protokollerindeki zayıflıkları istismar etmektedirler. SYN Flood, SYN-ACK Flood ve Smurf Saldırısı gibi saldırılar, OSI modelinin 3. ve 4. katmanlarına (Ağ ve İletim katmanları) yönelik gerçekleştirmektedirler. Örneğin, Smurf saldırısında, ICMP protokolü kullanılarak ağın her cihazının hedefe yanıt göndermesi sağlanmaktadır (Pérez-Díaz, Valdovinos, Choo & Zhu, 2020).

Uygulama Katmanı Saldırıları

Bu saldırılar, belirli uygulama protokollerini hedef almaktadır. HTTP Flood gibi saldırılarda, web sunucularına çok sayıda sahte HTTP isteği gönderilerek sunucu kaynakları tüketilmektedir (Dong & Sarem, 2020). Slowloris gibi saldırılar ise yavaş HTTP istekleri göndererek sunucunun tüm bağlantılarını meşgul ederek sistemi yeni gelen isteklere yanıt veremez hale getirmektedir (Yungaicela-Naula, Vargas-Rosales, Pérez-Díaz, Jacob & Martinez-Cagnazzo, 2023).

Tampon Taşması Saldırıları

Bu saldırılarda, bellekteki tamponlar aşırı veri ile doldurulmaktadır. Bunun sonucu olarak sistem çalışamaz hale gelmektedir. Bellek taşması saldırıları, genellikle sisteme yetkisiz erişim sağlamak veya zararlı kod enjekte etmek amacıyla kullanılmaktadır (Butt, Ajmal, Khan, Idrees & Javed, 2022).

2.5. Gelişmiş Sürekli Tehdit

Gelişmiş Sürekli Tehdit (APT Saldırıları), hedefe yönelik, uzun vadeli ve gizli kalan siber saldırılar olarak tanımlanmaktadır. Bu saldırılar, genellikle devlet destekli gruplar veya büyük suç örgütleri tarafından planlanır ve karmaşık saldırı yöntemleri kullanılarak gerçekleştirilmektedir (He, Gu, Zhu, Chan & Guizani, 2024). APT Saldırılarının en belirgin özellikleri, hedef sistemlere sızma, orada uzun süre kalmak ve önemli verileri çalmaktır (Mei vd., 2024).

APT Saldırılarının başlangıç noktası genellikle bir oltalama saldırısı veya sosyal mühendislik teknikleri kullanılarak elde edilen erişimlerdir. Bu ilk erişim, daha sonra iç ağdaki diğer sistemlere yatay geçiş yapılmasına olanak tanımaktadır (Tang, Xu, Fu & Huang, 2018). Saldırı süresince, saldırganlar genellikle tehlike içermeyen davranışlar sergileyerek tespit edilmemeye büyük özen göstermektedirler (Alshamrani, Myneni, Chowdhary & Huang, 2019).

Geleneksel güvenlik çözümleri genellikle bu tür saldırıları tespit etmekte yetersiz kalır çünkü APT Saldırıları sürekli gelişmektedir ve yeni yöntemler geliştirilmektedir. Bu nedenle, bu tip saldırıların engellenmesinde sürekli güncellenen, karmaşık davranış analizi ve anomali tespiti yapan sistemler kullanılması önerilmektedir (Khosravi & Ladani, 2020). Bu sistemler, ağ trafiğini sürekli olarak izleyerek ve analiz ederek anormal davranışları saptamaya çalışmaktadır (Khosravi & Ladani, 2020).

Kurumlar, APT tehditlerine karşı korunmak için sadece teknolojik çözümlere bağlı kalmamalı, aynı zamanda çalışanlarını bu tür saldırılara karşı bilinçlendirmeli ve güvenlik protokollerini sürekli güncellemelidir. Çalışan eğitimi ve güvenlik farkındalığı, bu tür ileri düzey tehditlere karşı savunmada kritik bir öneme sahiptir (Alshamrani, Myneni, Chowdhary & Huang, 2019).

2.6. Veri Tabanı, Yazılım, Tarayıcı Tabanlı Saldırılar

Veri tabanı saldırıları, saldırganların veri tabanlarına yetkisiz erişim sağlayarak kullanıcıdan bilgi çalma, değiştirme veya silme gibi eylemler gerçekleştirdiği siber saldırı türleridir. Yazılım tabanlı saldırılar, bir yazılımın güvenlik açıklarını hedef alan saldırılardır. Bu tür saldırılar, kötü amaçlı yazılımların sisteme yüklenmesi, yazılım hatalarının suistimal edilmesi veya güvenlik açıklarının kullanılması yoluyla gerçekleştirilebilmektedirler. Tarayıcı tabanlı saldırılar, kullanıcıların web tarayıcılarını hedef alarak hassas verilerin ele geçirilmesini amaçlayan saldırılardır.

Bu gruba giren saldırı türlerinden SQL Enjeksiyonu Saldırıları, Siteler Arası Komut Dosyası Çalıştırma Saldırıları aşağıda detaylandırılmaktadır (Iqbal, Khan & Niazi, 2024).

2.6.1. SQL Enjeksiyonu Saldırıları

SQL Enjeksiyon Saldırıları (SQL Injection), web uygulamalarının güvenliğini tehdit eden ve kullanıcı giriş alanları üzerinden kötü niyetli SQL sorgularının eklenmesiyle gerçekleşen bir saldırı yöntemidir. Bu saldırı, veri tabanı sorgularının yapısını değiştirerek veri çalma, manipülasyon veya sistemin işleyişini bozmaya neden olmaktadır. SQL Enjeksiyon Saldırıları, özellikle kullanıcı giriş doğrulamasındaki eksikliklerden kaynaklanmaktadırlar. Giriş formuna yerleştirilen bir SQL sorgusuyla veri tabanındaki bilgileri görüntülenebilir veya değiştirilebilir. Bu yöntem, SQL ifadelerinin, kullanıcı girişinden doğrudan veri tabanına iletilmesi sırasında yapılan hatalardan kaynaklanmaktadır (Kareem vd., 2021).

Bu tür saldırılara karşı alınabilecek önlemlerden biri, parametreli sorguların kullanılmasıdır. Parametreli sorgular, SQL kodunun kullanıcı girdisinden ayrı tutulmasını sağlayarak kötü niyetli girişlerin etkisini ortadan kaldırabilirler. Web uygulamaları için en etkili savunma yöntemlerinden biri de kullanıcı girdisinin doğru bir şekilde doğrulanması ve filtrelenmesidir. Bu yöntemler, SQL sorgularına zararlı içerik

eklenmesini engelleyerek uygulamanın güvenliğini artırmaktadır (Okesola, Ogunbanwo, Owoade, Olorunnisola & Okokpuji, 2023).

2.6.2. Siteler Arası Komut Dosyası Çalıştırma Saldırıları

Siteler Arası Komut Dosyası Çalıştırma Saldırıları (Cross Site Scripting (XSS)), modern web uygulamalarında yaygın olarak görülen ve kullanıcıların tarayıcılarında kötü amaçlı kod çalıştırmak için kullanılan bir güvenlik açığı türüdür. XSS Saldırıları, genellikle kullanıcı girdilerinin yetersiz filtrelenmesi nedeniyle ortaya çıkar ve saldırganların, kurbanların oturum çerezleri, kişisel bilgileri ve diğer hassas verilerini çalmalarına sebep olmaktadırlar. XSS Saldırıları, özellikle internetin geniş çapta kullanımıyla birlikte önem kazanmış ve yıllar boyunca OWASP Top 10 güvenlik açıkları listesinde yüksek sıralarda yer almıştır (Alenzi & Abbas, 2022).

XSS Saldırılarının en yaygın türleri arasında Kalıcı (Stored), Yansıtılmış (Reflected) ve DOM (Doküman Nesne Modeli) Tabanlı XSS Saldırıları bulunmaktadır. Kalıcı XSS saldırısı, saldırganların kötü amaçlı kodu bir web uygulamasının veri tabanına kaydetmesi ve bu kodun, veri tabanını kullanan her ziyaretçiye otomatik olarak iletilmesiyle gerçekleşmektedir. Bu tür saldırılar, özellikle sosyal medya platformları ve forum gibi kullanıcı etkileşiminin yoğun olduğu ortamlarda risk oluşturmaktadırlar (Liu, Zhang, Chen & Zhang, 2019). Yansıtılmış XSS saldırısı, genellikle bir URL'ye gizlenmiş kötü amaçlı kodun kullanıcı tarafından tıklanmasıyla başlamaktadır. Daha sonra kötü amaçlı içerik tarayıcıda çalıştırılır ve saldırgan hedeflediği bilgileri ele geçirmeye başlamaktadır (Abhishek, Ravindran, A. T. & S. V., 2023).

DOM Tabanlı XSS saldırısında ise kod sadece istemci tarafında çalışır ve saldırı, sunucu üzerinden herhangi bir veri akışı olmaksızın JavaScript etkileşimleri ile gerçekleştirilmektedir. Bu durum, sunucu tarafı güvenlik çözümlerini baypas ederek saldırganın tarayıcıdaki mevcut verileri manipüle etmesini sağlamaktadır (En & Selvarajah, 2022).

XSS Saldırıları, e-ticaret siteleri gibi kullanıcı verilerinin yoğun olarak işlendiği platformlarda ciddi sonuçlar doğurabilmektedirler. Bu tür saldırılar, kullanıcıların hesap bilgilerinin ele geçirilmesine, oturumlarının çalınmasına ve sahte içeriklerle yanıltılmalarına neden olabilmektedir (Blancaflor, Araullo, Corcuera, Rivera & Velarde, 2023).

XSS gibi güvenlik açıklarına karşı etkin bir koruma için çok katmanlı savunma ve yapay zekâ tabanlı tespit sistemlerinin entegre edilmesi önerilmektedir (Lu & Liu, 2023).

2.7. Ortadaki Adam Saldırısı

Ortadaki Adam (Man in the Middle-MITM) Saldırıları, iletişim halindeki iki taraf arasındaki trafiğe müdahale eden bir saldırganın, iletişimin gizliliğini ve bütünlüğünü tehlikeye atarak veri çalabileceği ya da manipüle edebileceği bir siber saldırı türüdür. Bu saldırı türü, özellikle ağ güvenliği açısından ciddi tehditler oluşturur, çünkü saldırganlar araya girerek veri paketlerini gizlice izleyebilir, değiştirebilir ya da iki tarafın kimliğine bürünebilmektedir (Bruschi, Di Pasquale, Ghilardi, Lanzi & Pagani, 2022).

MITM Saldırıları, çeşitli ağ katmanlarında farklı tekniklerle gerçekleştirilmektedirler. Örneğin, Adres Çözümleme Protokolü (Address Resolution Protocol, ARP) zehirlenmesi veri bağlantı katmanında, IP (Internet Protokolü) sahteciliği ağ katmanında ve tarayıcıda gerçekleşen MITM Saldırıları, uygulama katmanında görülen örneklerdir (Thankappan, Rifà-Pous & Garrigues, 2024).

Wi-Fi ağlarında gerçekleştirilen Anahtar Yeniden Kırma Saldırısı (Key Reinstallation Attack, KRACK) ve Parçalama Saldırıları (Fragmentation Attacks, FragAttacks) gibi saldırılar, şifreli çerçeveleri bile manipüle ederek özellikle Nesnelerin İnterneti cihazlarına yönelik büyük bir güvenlik açığı oluşturmaktadırlar (Kosugi, Teranishi & Kogiso, 2024).

Kontrol sistemlerine yönelik MITM Saldırıları'nın etkileri de oldukça kritiktir. Bu tür sistemlerde saldırganlar, sahte veriler ekleyerek ya da paketleri değiştirerek sistemin güvenilirliğini bozabilmektedirler. Örneğin, uzaktan kumandalı endüstriyel kontrol sistemlerinde bu tür saldırılar, sistemin yanlış tepki vermesine neden olabilmektedir. Bu durum, özellikle altyapı hizmetlerinin güvenliğini tehlikeye sokar ve ciddi hasarlara yol açabilmektedir (Letafati, Behroozi, Khalaj & Jorswieck, 2022).

Nesnelerin İnterneti ortamlarında MitM saldırıları, saldırganın Düşük Güçlü ve Güvenilir Olmayan Ağlar İçin Yönlendirme Protokolü (Routing Protocol for Low-Power and Lossy Networks, RPL) kullanarak ağdaki belirli cihazları hedef alabilmesine olanak vermektedir. SPRINKLER adı verilen bir sistem, bu tür saldırılara karşı Nesnelerin İnterneti ağlarında saldırganı tespit edebilmek için geliştirilmiştir. Bu sistem, mevcut

protokolleri kullanarak ve ek kaynak harcamadan saldırganı tespit etme olanağı sunmaktadır (Zilberman, Dvir & Stulman, 2024).

Web uygulamaları üzerinden gerçekleştirilen MITM Saldırıları ise, ağ trafiğini izleyerek kullanıcıların oturum bilgilerini ele geçirme amacı taşımaktadır. Bu tür saldırılar özellikle banka gibi hassas verilerin bulunduğu uygulamalarda büyük risk oluşturabilirler ve kullanıcıların özel bilgilerini açığa çıkarabilmektedirler (Muzammil, Bilal, Ajmal, Shongwe & Ghadi, 2024).

2.8. Saldırı Tespit Sistemleri

Saldırı Tespit Sistemleri (Intrusion Detection Systems-IDS), bilgi güvenliği alanında kritik bir rol oynayan ve kötü amaçlı etkinlikleri tespit etmek için kullanılan bir teknolojidir. IDS'ler, ağ trafiğini analiz ederek, yetkisiz erişim ve güvenlik ihlalleri gibi kötü niyetli aktiviteleri saptamak amacıyla tasarlanmaktadırlar. Kurumlar, kötü niyetli aktörlerin sistemlerinde herhangi bir faaliyet gösterip göstermediğini anlamak için IDS'leri güvenlik stratejilerinin vazgeçilmez bir unsuru olarak kullanmaktadırlar. Bu sistemler, olası tehditleri erkenden tespit ederek, kurumların güvenlik önlemlerini proaktif bir şekilde yönetmelerine olanak tanımaktadır ve böylece ciddi zararların önüne geçilebilmektedir (Halimaa & Sundarakantham, 2019). Saldırı Tespit Sistemleri, şekil 2.2'de görülebileceği gibi bölümlere ayrılabilmektedir.



Sekil 2.2. Saldırı Tespit Sistemleri Türleri

2.8.1. Saldırı Tespit Sistemleri Türleri

Saldırı Tespit Sistemleri, İmza Tabanlı, Anomali Tabanlı, Durumsal Tabanlı, Davranış Tabanlı, Ağ Tabanlı, Host Tabanlı Saldırı Tespit sistemleri olarak aşağıda detaylandırılmaktadır.

- **İmza Tabanlı Saldırı Tespit Sistemleri:** İmza tabanlı tespit, güvenlik tehditlerini önceden tanımlanmış imzalarla tespit eden bir yaklaşımdır. Bu imzalar, kötü

amaçlı yazılımlar, saldırı desenleri ve anormal davranışlar gibi tehditleri belirten kalıplardır. Antivirüs yazılımları, IDS'ler ve güvenlik duvarları bu yöntemi kullanmaktadır. Antivirüsler yazılımları, veri tabanlarındaki imzaları kullanarak virüs ve zararlı yazılımları tespit etmektedirler. IDS'ler, ağ trafiği ve sistem aktivitelerini izlerken bilinen saldırı imzalarını kontrol etmektedirler. Güvenlik duvarları ise belirli trafik kalıplarını engellemek için imzaları kullanılmaktadırlar. İmza tabanlı tespitin avantajları arasında, hızlı ve etkili tespit sağlaması ve düşük yanlış alarm oranı (normal veri iken saldırı olarak sınıflandırılan veri oranı) gibi özelikler ön plana çıkmaktadır. Ancak, bu yöntem yeni veya bilinmeyen tehditleri tespit edemez ve düzenli imza güncellemeleri gerektirmektedir.

- **Anomali Tabanlı Saldırı Tespit Sistemleri:** Anomali Tabanlı Saldırı Tespit Sistemleri, sistemin normal davranışından sapmaları tespit eden bir yöntemdir. Sistem davranışları modellenir ve bu modelden sapmalar potansiyel tehdit olarak değerlendirilmektedir. Makine öğrenimi modelleri, istatistiksel analiz ve zaman serisi analizi bu yöntemin bileşenleridir. Makine öğrenimi, normal davranışları öğrenip sapmaları tespit ederken, istatistiksel analiz, normal dağılımdan sapmaları değerlendirmektedir. Zaman serisi analizi ise zamana bağlı değişiklikleri izleyip anormallikleri belirlemektedir. Bu yöntemin avantajları, bilinmeyen tehditleri tespit edebilmesi ve sürekli öğrenme yeteneğidir. Bununla beraber yüksek yanlış pozitif oranına sahip olabilirler ve kurulum ve bakım aşamasında zorluklarla karşılaşılabilmektedir.
- **Durumsal Tabanlı Saldırı Tespit Sistemleri:** Durumsal Tabanlı Saldırı Tespit Sistemleri, sistemin mevcut durumu veya bağlamına göre tehditleri değerlendiren bir yöntemdir. Bu yöntem, sistem veya ağın mevcut durumu temel alınarak tehditleri belirlemektedir. Durumsal tabanlı tespit, durum farkındalığı, bağlam analizi ve kural tabanlı sistemleri içermektedir. Durum farkındalığı, sistemin mevcut durumunu analiz ederek tehditleri tespit ederken, bağlam analizi, kullanıcı davranışları ve çevresel faktörleri değerlendirmektedir. Kural tabanlı sistemler ise belirli koşullar altında tehdit tespiti yapmaktadır. Avantajları arasında bağlamsal bilgilerle daha doğru tespit yapabilmesi ve tehditleri geniş bir bakış açısıyla değerlendirebilmesi bulunmaktadır. Bununla beraber bağlamın doğru

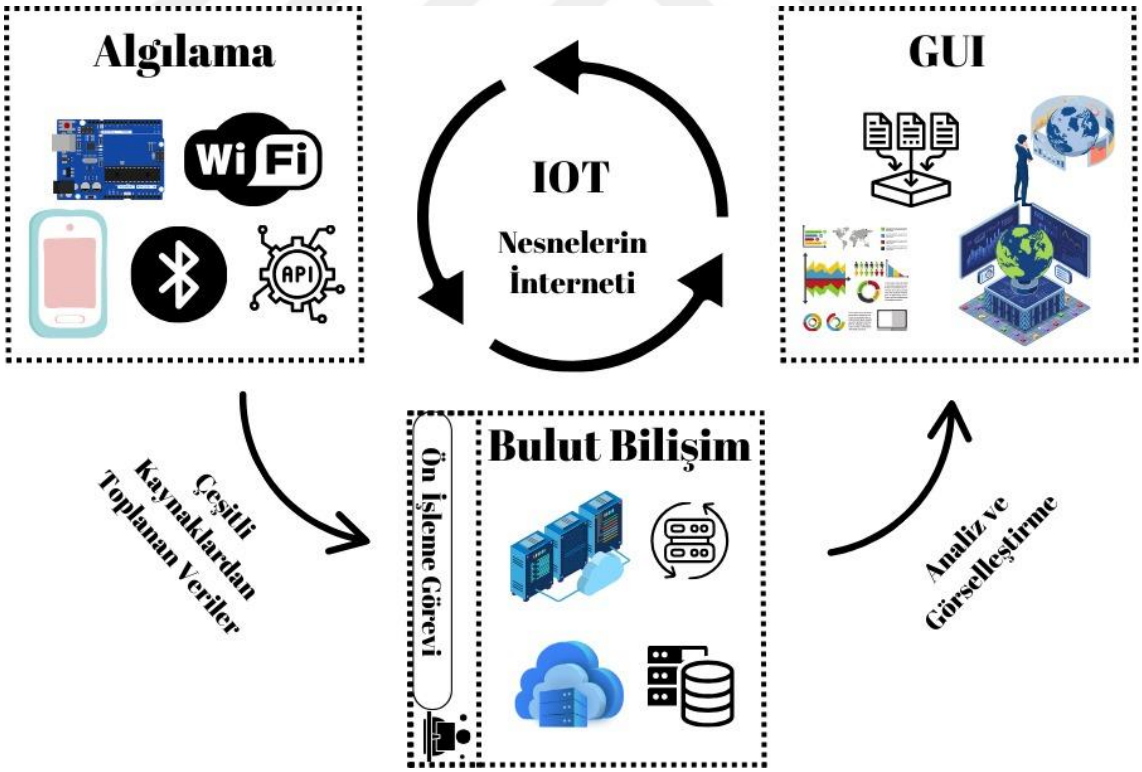
tanımlanması zor olabilir ve çeşitli senaryolar için özel ayarlamalar gerektirebilmektedir.

- **Davranış Tabanlı Saldırı Tespit Sistemleri:** Davranış Tabanlı Saldırı Tespit Sistemleri, sistemde veya kullanıcıdaki anormal davranışları belirleyerek tehditleri tespit eden bir yöntemdir. Bu yöntemde normal davranış profilleri oluşturulur ve sapmalar potansiyel tehdit olarak değerlendirilmektedir. Kullanıcı Davranış Analizi (UBA), makine öğrenimi algoritmaları ve anomali algılama bu yöntemin temel bileşenleridir. UBA, kullanıcıların normal davranışlarını öğrenip anormallikleri tespit ederken, makine öğrenimi algoritmaları normal davranış modelleri oluşturmaktadır. Anomali algılama ise sistem aktivitelerini izleyerek sapmaları değerlendirmektedir. Bu yöntemin avantajları, yeni ve bilinmeyen tehditleri tespit edebilmesi ve sürekli güncelleme gerektirmemesidir. Ancak, yüksek yanlış pozitif oranına sahip olabilir ve karmaşık davranış modelleri oluşturulması gerekebilmektedir.
- **Ağ Tabanlı Saldırı Tespit Sistemleri (NIDS):** Ağ Tabanlı Saldırı Tespit Sistemleri, ağ trafiğini izleyerek anormal faaliyetleri tespit etmek ve bildirmek amacıyla tasarlanmaktadır. Genellikle ağ geçitleri veya yönlendiriciler gibi kritik noktalara yerleştirilmektedir ve ağ trafiğini detaylı bir şekilde incelemektedirler. NIDS, imza tabanlı veya davranış tabanlı yaklaşımlar kullanarak, olası tehditleri ve saldırıları saptayabilmektedirler. Bu sistemler, ağ güvenliğini sağlamak için kritik bir rol oynar ve tehditlerin hızlı bir şekilde tespit edilmesine olanak tanımaktadır.
- **Host Tabanlı Saldırı Tespit Sistemleri (HIDS):** Host Tabanlı Saldırı Tespit Sistemleri, bireysel cihazlar veya sunucular üzerinde meydana gelen anormal faaliyetleri tespit etmek ve bildirmek amacıyla tasarlanmış önemli bir güvenlik bileşenidir. HIDS, sistemin çalışma ortamını detaylı bir şekilde izleyerek, sistem kayıtları (log), dosya bütünlüğü, ağ trafiği, işlem etkinlikleri ve kullanıcı davranışlarını analiz etmektedir. Bu analizler sonucunda, cihaz üzerinde yapılan yetkisiz değişiklikler, şüpheli dosya değişiklikleri, izinsiz erişim denemeleri ve diğer potansiyel tehditler tespit edilmektedir.

BÖLÜM 3

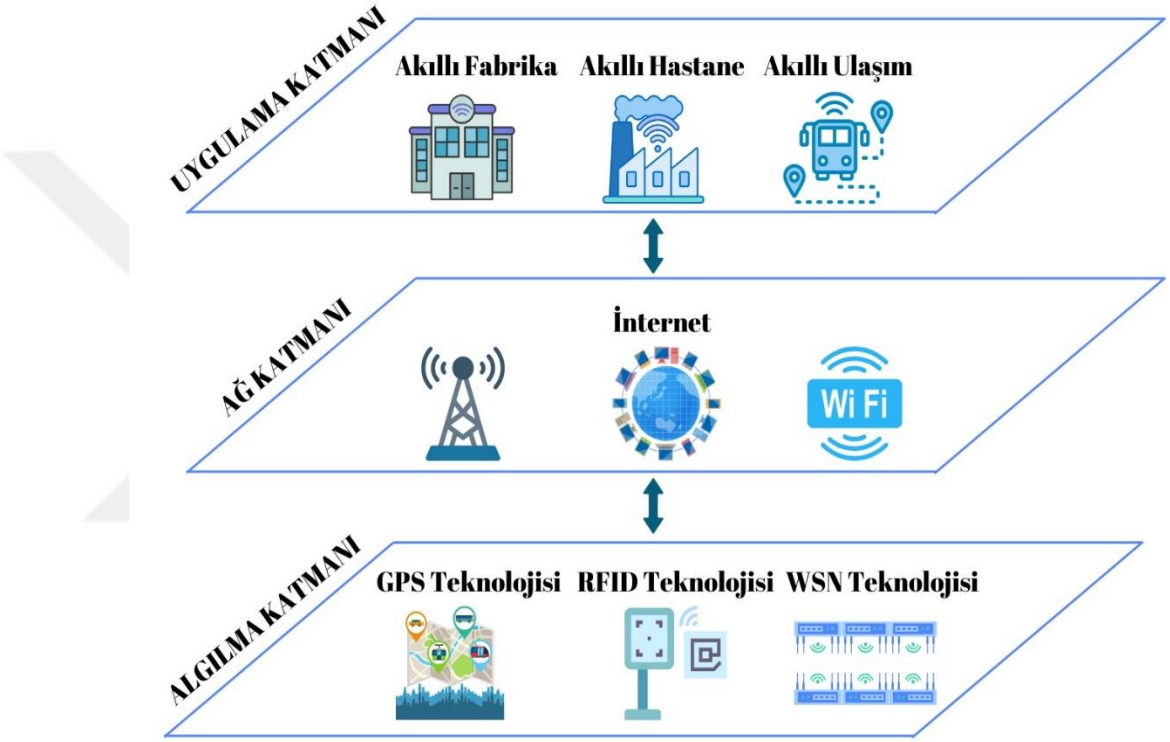
NESNELERİN İNTERNETİ AĞLARI

Nesnelerin İnterneti Ağları, çeşitli akıllı cihazların birbirine bağlanarak tek bir ağ oluşturduğu ve bu cihazların hassas veriler toplayıp önemli işlevleri yerine getirebilmesini sağladığı bir yapıdır (Saranya vd., 2020). Bu yapı, farklı hizmetleri bir araya getirerek etkili bir iletişim ve veri işleme platformu oluşturmaktadır. Kullanıcılar, mobil uygulamalar veya web hizmetleri aracılığıyla cihazlara ulaşarak bu cihazları kontrol edebilmektedirler. Şekil 3.1'de Nesnelerin İnterneti Mimarisi genel olarak gösterilmektedir.



Şekil 3.1. Nesnelerin İnterneti Genel Mimarisi

Nesnelerin İnterneti mimarisi için bütün otoriteler tarafından kabul edilmiş, tek tip bir mimari yoktur. Farklı sayıda katmanlardan oluşan mimari yapılar kullanılmaktadır. Üç katmanlı, dört katmanlı, beş katmanlı şekilde ifade edilen mimari yapılar vardır. Bunların hepsi temel olarak üç katmanlı mimarideki üç temel katmanı kullanılmaktadır ve ilave olarak bu katmanların arasına bazı ilave katmanlar getirilmektedir. Üç katmanlı mimari katmanları, Şekil 3.2'de bu gösterilmektedir. Aşağıda bu katmanlar detaylandırılmaktadır.



Şekil 3.2. Nesnelerin İnterneti Katmanları

3.1. Algılama Katmanı

Nesnelerin İnterneti mimarisinin en temel unsurlarından biri olan algılama katmanı, fiziksel çevreden veri toplama işlevini yerine getirmektedir. Bu katman, sensörler ve aktüatörler aracılığıyla çalışarak, çevresel değişkenleri sürekli olarak izlemektedir ve bu verileri daha yüksek katmanlara iletmeye hazır hale getirmektedir. Algılama katmanının birincil görevleri arasında veri toplama, gerçek zamanlı izleme, veri ön işleme ve aktüatörlerle etkileşim yer almaktadır. Sensörler, ortamda meydana gelen değişiklikleri kaydederken, bu verileri anlamlı hale getirmek için gerekli ön işlemleri de yapmaktadır. Örneğin, sinyal gürültüsünü azaltmak ve veri birleştirmek gibi işlemler bu

katmanın işlevleri arasındadır. Veri toplamak için RFID (Radio Frequency Identification-Radyo Frekansı ile Tanımlama), WSN (Wireless Sensor Network-Kablosuz Sensör Ağı), GPS (Global Positioning System - Küresel Konumlandırma Sistemi) gibi çeşitli teknolojiler kullanılmaktadır (Xu & Yuan, 2020).

Bu katman, Nesnelerin İnterneti sistemlerinin sağlıklı bir şekilde çalışabilmesi için kritik bir öneme sahiptir. Eğer algılama katmanı işlevini yerine getiremezse, sistemin diğer bileşenleri de çevreyle etkileşim kuramaz, bu da veriye dayalı karar alma süreçlerini etkilemektedir. Dolayısıyla, üst katmanların etkinliği ve genel sistem performansı, algılama katmanının verimliliğine bağlıdır (Mukhopadhyay vd., 2021).

Algılama Katmanı, Nesnelerin İnterneti sistemlerinin temelini oluşturmaktadır ve veri toplama ve işlemeyi gerçekleştirerek daha üst düzey işlevlerin etkili bir şekilde çalışmasına olanak tanımaktadır. Bu durum, Nesnelerin İnterneti uygulamalarının genel verimliliğini ve faydalarını artırmaktadır (Mukhopadhyay vd., 2021).

3.2. Ağ Katmanı

Ağ katmanı, cihaz katmanı olarak da bilinir ve Nesnelerin İnterneti yapısındaki tüm bileşenler arasında dijital verilerin hareketini sağlamak için büyük bir öneme sahiptir. Veri iletişimi, Zigbee, 6LoWPAN gibi çeşitli ağ iletişim teknolojileri aracılığıyla gerçekleştirilmektedir. Bu teknolojiler, LAN (Yerel Alan Ağları) gibi yerel ağlarda, kablosuz ve kablolu ağlarda kullanılabilmektedirler (Das vd., 2023).

Ağ katmanının çeşitli bileşenleri vardır. Bu yapıda öncelikle, yerel sistemlerin dışındaki iletişimi sağlayan internet geçitleri bulunmaktadır. Yerel sistemlerde internet geçidi ve erişim noktaları sağlayan intranet portları, ağ geçitleri aracılığıyla birden fazla cihazın köprüleme işleviyle bağlantı kurmasına imkân tanımaktadır. Ayrıca, DAS (Veri Toplama Sistemleri), bir sistem içerisindeki uygulamalar arasında sensör okumalarını toplamak için kullanılmaktadır. Bu bileşenler, Nesnelerin İnterneti sistemlerinin etkinliğini artırmak ve veri iletim süreçlerini geliştirmek için kritik bir rol oynamaktadır. Veri iletimi için kullanılan ana iletişim seçenekleri arasında 3G, 4G ve 5G mobil iletişim teknolojileri, Wi-Fi, Bluetooth ve Zigbee gibi kablosuz iletişim standartları bulunmaktadır. Kızılötesi teknoloji ve optik fiber iletişim ağları da veri aktarımında önemli roller üstlenmektedir. Ayrıca, geniş bant televizyon ağları ve NFC (Yakın Alan İletişimi) gibi daha spesifik teknolojiler de çeşitli uygulama senaryolarında

kullanılmaktadır. Kablosuz HART (Otoyola Bağlanabilir Uzaktan Sensör-Highway Addressable Remote Transducer) gibi endüstriyel iletişim protokolleri, mobil iletişim sistemleriyle bir araya gelerek veri iletimini daha güvenilir ve etkili yapmaktadır (Xu & Yuan, 2020).

Veri iletişim kanallarındaki bu çeşitlilik, Nesnelerin İnterneti uygulamalarının daha geniş bir alanda etkin bir şekilde kullanılabilmesine olanak tanımaktadır. Akıllı ev sistemlerinden akıllı tarım uygulamalarına, lojistik yönetiminden sağlık hizmetlerine kadar birçok alanda bu iletişim teknolojileri, verilerin güvenli bir şekilde toplanmasını ve işlenmesini sağlamaktadır. Bu sayede, kullanıcıların ihtiyaçlarına göre özelleştirilmiş hizmetler sunulabilmektedir ve sistemin genel verimliliği artırılabilmektedir (Abdul Hafeez, Singh, Singh, Prabha & Verma, 2022).

3.3. Uygulama Katmanı

Uygulama katmanı, insanlarla Nesnelerin İnterneti ekosistemi arasında önemli bir etkileşim noktası işlevi görmektedir. Bu katman, kullanıcıların akıllı cihazlar üzerinden çeşitli hizmetlere erişimini sağlayarak, mobil uygulamalar veya bilgisayar ekranındaki merkezi kontrol panelleri aracılığıyla rahat bir kullanım sunmaktadır. Örneğin, akıllı ev sistemleri için özel olarak geliştirilmiş bir uygulama sayesinde, kullanıcılar kahve makinelerini basit bir dokunma ile çalıştırabilmektedirler. Bu uygulamalar, kullanıcıların Nesnelerin İnterneti sistemlerinden en iyi şekilde yararlanabilmelerini temin etmek için verimli bir etkileşim imkânı sunmaktadırlar (Yassein, Shatnawi & Al-zoubi, 2016).

Bu katman, ağ katmanından gelen verileri aldıktan sonra bu verilerin formatını ayarlayarak sunmaktadır. Ayrıca, son kullanıcıların talep ettiği çeşitli hizmetleri vermektedir. Akıllı ulaşım, lojistik yönetimi, akıllı sağlık uygulamaları, akıllı tarım sistemleri, akıllı ev otomasyonu, akıllı şehir uygulamaları, trafik yönetimi, bina otomasyonu, akıllı ölçüm sistemleri ve otomatik taşıtlar gibi pek çok alanda Nesnelerin İnterneti uygulamaları bulunmaktadır (Yassein, Shatnawi & Al-zoubi, 2016).

Bu uygulamalara bilgisayarlar, dizüstü bilgisayarlar ve akıllı telefonlar gibi çeşitli akıllı cihazlar üzerinden erişim sağlanabilmektedir. Uygulama katmanında, veri iletiminde kullanılan çeşitli mesajlaşma protokolleri büyük bir öneme sahiptir. AMQP (Gelişmiş Mesaj Kuyruğu Protokolü), COAP (Sınırlı Uygulama Protokolü), XMPP (Genişletilebilir Mesajlaşma ve Varlık Protokolü) ve MQTT (Mesaj Kuyruğu Telemetri

Taşıma Protokolü) gibi protokoller, verilerin düzgün bir biçimde iletilmesini sağlamaktadır (Xu & Yuan, 2020).

Uygulama katmanı, kullanıcıların sistemle etkileşimlerini kolaylaştırarak, çok çeşitli endüstriyel uygulamalar için geniş bir potansiyel sunmaktadır. Kullanıcı deneyimini geliştiren bu katman hem etkileşimli hem de erişilebilir uygulamalar geliştirilmesine olanak tanımaktadır (Yassein, Shatnawi & Al-zoubi, 2016).

3.4. Nesnelerin İnterneti Mimarisine Tehditler

Nesnelerin İnterneti ağlarında, düşük güç ve kaynak kullanımı gibi çeşitli sebeplerden dolayı güvenlik ikinci plana itilmiştir. Bu sebeple bu ağlar birçok tehdide maruz kalmaktadır. Bu saldırılar mimarinin farklı katmanlarında kendini gösterebilmektedir (Ahemd, Shah & Wahid, 2017). Algılama Katmanı, Ağ Katmanı ve Uygulama Katmanındaki tehditler aşağıda detaylandırılmaktadır.

3.4.1. Algılama Katmanı

Algılama katmanı sensörlerin çevreden bilgileri topladıkları, kendi aralarında veya merkezi ara birimlerle iletişim kurdukları bir katmandır. Bu katmanın karşılaştığı tehditlerden bazıları aşağıda detaylandırılmaktadır.

- Sahtecilik (Spoofing) Saldırısı: Saldırganlar, Nesnelerin İnterneti sistemine tam erişim elde etmek için sensör cihazlara sahte mesajlar iletmektedir. Bu tür bir saldırı, RFID kimlik sahteciliği ile gerçekleştirilmektedir (Naqvi vd., 2023).
- DDoS Saldırısı: Bu saldırılar, geniş kapsamlı etkili bir hizmet engelleme saldırısı türüdür. Saldırganlar, kontrol ettikleri ve çoğunlukla zararlı yazılım ile ele geçirdikleri cihazlara, yüksek miktarda trafik göndermektedir ve kaynakların tükenmesine yol açmaktadır (Pérez-Díaz, Valdovinos, Choo & Zhu, 2020).
- Sinyal Bozma (Signal Jamming) Saldırısı: Bir tür Dos saldırısıdır. Saldırgan sensör düğümleri ile diğer düğümler arasındaki iletişimi bozmaktadır (Guan & Ge, 2017).
- Düğüm Ele Geçirme (Node Capturing) Saldırısı: Saldırganlar, Nesnelerin İnterneti ağlarında ağ geçidi düğümü gibi düğümleri fiziksel olarak ele geçirip talimatlar verebilmektedirler. Bu durum, eşleşme anahtarı, radyo anahtarı gibi

önemli bilgilerin sızmasına neden olabilmektedir (Tahir & McDonald-Maier, 2012).

- Tekrarlama (Replay) Saldırısı: Saldırganlar veri alışverişi sırasında çeşitli verileri elde ederek kaydedebilmektedirler. Bunları başka bir iletişimde kendilerini yasal kullanıcı olarak kanıtlamak için kullanmaktadırlar (Taher, Tariq, Behnamfar & Sarwat, 2023).
- Sahte Düğüm (Fake Node): Saldırganlar sahte düğümler oluşturmaktadır. Bu düğümler vasıtasıyla sahte kimlikler üzerinden veri transferi gerçekleştirerek meşru düğümlerin enerjisini tüketmeyi ve iletişimi sekteye uğratmayı hedeflemektedirler (Tahir & McDonald-Maier, 2012).

3.4.2. Ağ Katmanı

Ağ katmanı, Nesnelerin İnterneti cihazları arasındaki veri aktarımını sağlamakta ve bu nedenle güvenli veri iletimi kritik bir önem taşımaktadır. Bu katmanda çeşitli tehditler meydana gelmektedir:

- Ortadaki Adam (MITM) saldırıları: Saldırganlar, iki taraf arasındaki iletişimi ele geçirerek verileri manipüle etmektedir. Bu durum, veri gizliliğini ve bütünlüğünü tehdit etmektedir (Mosenia & Jha, 2016).
- Veri dinleme (Eavesdropping): Şifrelenmemiş ağ trafiği, saldırganların Nesnelerin İnterneti cihazlarından veri ele geçirmesine olanak tanımaktadır (Islam vd., 2015).
- Hizmet Reddi (DoS/DDoS) saldırıları: Nesnelerin İnterneti cihazlarına aşırı trafik gönderilerek sistemlerin hizmet veremez duruma getirilmesi amaçlanmaktadır (Alaba, Othman, Hashem & Alotaibi, 2017).

3.4.3. Uygulama Katmanı

Uygulama katmanı, Nesnelerin İnterneti sistemlerinin kullanıcılarla etkileşim kurmasını sağlayan uygulama tabanlı hizmetleri içermektedir. Ancak bu katmanda yazılım güvenlik açıkları ve yetkisiz erişim riskleri ortaya çıkmaktadır.

- Kimlik doğrulama zafiyetleri: Nesnelerin İnterneti cihazlarında yeterli kimlik doğrulama mekanizmalarının bulunmaması, yetkisiz erişimlere yol açmaktadır (Sicari, Rizzardi, Grieco & Coen-Porisini, 2015).

- Yazılım güvenlik açıkları: Nesnelerin İnterneti sistemlerinin yazılımlarındaki güvenlik açıkları, saldırganların sistem üzerinde kontrol sağlamasına neden olmaktadır (Obaidat, Nicopolitidis & Zarai, 2020).

3.5. Kullanılan Protokoller

Nesnelerin İnternetinde veri iletimini sağlamak için birçok protokolden faydalanılmaktadır. Bunlardan bazıları aşağıda detaylandırılmaktadır.

3.5.1. Zigbee

Zigbee, IEEE 802.15.4 standardına dayanarak geliştirilen, düşük güç tüketimine sahip dijital radyo iletişim teknolojilerini kullanan bir kablosuz iletişim protokolüdür. Bu teknoloji, özellikle düşük veri hızı gerektiren ev otomasyonu, endüstriyel kontrol sistemleri ve sensör ağları gibi uygulamalar için tasarlanmıştır. Zigbee'nin temel amacı, düşük maliyet, düşük güç tüketimi ve yüksek güvenlik sağlayarak geniş bir uygulama alanında kablosuz bağlantı sunmaktır (A Simulation Platform, 2013).

Zigbee ağlarının tasarımı, enerji verimliliği ve maliyet etkinliği ile dikkat çeker. Bu ağlar, çok az enerji tükettikleri ve uzun ömürlü oldukları için batarya ile çalışan cihazlar için kullanışlıdır. Zigbee protokolü, cihazların ağa kolayca eklenebilmesi ve ağ yapılandırmalarının esnek bir şekilde yönetilmesi için gerekli araçları vermektedir. Ayrıca, Zigbee'nin güvenlik protokolleri, veri bütünlüğünü ve iletişimin gizliliğini sağlamak için gelişmiş şifreleme teknikleri içermektedir (Krishnan & Binu, 2017).

Zigbee'nin ağ yapılandırması, dinamik ortamlara hızlı bir şekilde uyum sağlayabilmektedir ve değişen koşullara göre kendini otomatik olarak yeniden yapılandırabilmektedir. Bu özellik, Zigbee'yi akıllı ev otomasyon sistemleri ve endüstriyel izleme uygulamaları için son derece uygun kılmaktadır. Ağın bu adaptif yapısı, yeni cihazların ağa dâhil edilmesini kolaylaştırmaktadır ve sistem yöneticilerine kapsamlı kontrol ve yönetim imkânı sunmaktadır (Zhang, Zhou & Zhou, 2009).

Zigbee, esnekliği ve güvenliği ile ön plana çıkmaktadır. Bu teknoloji, dünya çapında birçok farklı uygulamada kullanılmaktadır ve Nesnelerin İnterneti ekosisteminde önemli bir yer tutmaktadır. Kapsamlı bir kullanım alanına sahip olan Zigbee, giderek daha fazla tercih edilen bir teknoloji haline gelmiştir (Pereira, De Morais, Nascimento, Alsina, & Santos, 2020).

3.5.2. 6LoWPAN

Düşük Güçlü Kablosuz Kişisel Alan Ağları Üzerinde (Over Low-Power Wireless Personal Area Networks-6LoWPAN) IPv6 (IPv6-İnternet Protokolü Sürüm 6) düşük güçlü ve sınırlı kaynaklara sahip kablosuz cihazların internet protokolü (IP) ile uyumlu hale gelmesini sağlayan bir protokoldür. Bu teknoloji, özellikle enerji kısıtlamaları nedeniyle düşük güç tüketimine odaklanmış cihazlar için tasarlanmıştır. IEEE 802.15.4 standardı üzerine kurulu olup, Nesnelerin İnterneti uygulamaları ile büyük ağlarla entegrasyon sağlamaktadır. 6LoWPAN'in sunduğu başlık sıkıştırma, veri parçalama ve yeniden birleştirme özellikleri sayesinde, düşük güçlü cihazlar büyük veri ağlarına etkin bir şekilde bağlanabilmektedir (Gomes, Salgado, Pinto, Cabral & Tavares, 2018).

6LoWPAN, Nesnelerin İnterneti cihazlarının geniş çapta erişim sağladığı kablosuz sensör ağları (WSN) için de ideal bir çözümdür. Özellikle, IEEE 802.15.4 tabanlı ağlarda düşük veri hızları ve düşük enerji tüketimi sağlayan protokol yapısı sayesinde sınırlı enerji kaynakları olan cihazlar için veri iletim maliyetlerini önemli ölçüde azaltmaktadır. Bu etkinlik, Nesnelerin İnterneti cihazlarının ağ içinde daha uzun süre aktif kalmalarını ve veri iletiminde yüksek performans sunmalarına yardımcı olmaktadır. Bunun yanı sıra, tıkanıklık kontrol mekanizmaları ağın genel verimliliğini artırmada kritik rol oynamaktadır (Hou, He, Jiang, Chen & Yang, 2023).

Mobilite desteği, 6LoWPAN protokolünün Nesnelerin İnterneti uygulamalarında önemli bir rol oynamasını sağlayan bir başka özelliktir. Sağlık sistemleri, akıllı ulaşım ve lojistik gibi uygulamalarda düğümlerin hareketliliği sırasında veri iletimi sürekliliğinin korunması önemlidir. Bu tür mobilite yönetim çözümleri, Nesnelerin İnterneti cihazlarının dinamik koşullar altında dahi etkin bir şekilde çalışmasını sağlamaktadır (Qiu & Ma, 2017).

Güvenlik açısından ise 6LoWPAN, Nesnelerin İnterneti cihazlarının sınırlı işlem gücü ve enerji kaynakları göz önünde bulundurularak özelleştirilmiş çözümler içermektedir. Bu güvenlik önlemleri, genellikle şifreleme ile veri bütünlüğünü ve gizliliği korumaktadır. Ayrıca, anomali tespiti yapılarak potansiyel güvenlik tehditlerine karşı koruma sağlanmaktadır (Al-Amiedy vd., 2022).

6LoWPAN, veri iletiminde çeşitli optimizasyon stratejileri sunarak ağ performansını iyileştirmektedir. Bu optimizasyonlar, ağdaki veri iletim maliyetini ve

gecikme süresini düşürerek geniş çaplı veri dağıtımını daha verimli hale getirmektedir (Al-Amiedy vd., 2022).

3.5.3. BLE

BLE (Bluetooth Low Energy-Bluetooth Düşük Enerji) düşük enerji tüketimi ile çalışan kablosuz cihazların bağlantı ihtiyaçlarını karşılamak amacıyla geliştirilmiş bir teknolojidir. BLE, Bluetooth 4.0 spesifikasyonuna dâhil edilmiş olup, düşük enerji tüketimi ve uzun pil ömrü sağlamak için tasarlanmıştır. 2.400-2.4835 GHz frekans bandında çalışmaktadır ve bu sayede diğer birçok kablosuz cihazla aynı radyo spektrumunu paylaşabilmektedir. BLE'nin en büyük avantajlarından biri, veri transferi sırasında enerji tüketimini minimumda tutarak uzun süre dayanabilmesidir. Bu özelliği sayesinde, BLE özellikle düşük güç tüketimi gereksinimi olan uygulamalar için çok kullanışlıdır (Bulić, Kojek & Biasizzo, 2019).

BLE'nin düşük enerji tüketimi, veri gönderim sıklığının azaltılması ve bağlantı kurma işlemlerinin basitleştirilmesiyle sağlanmaktadır. BLE, bağlantı kurarken ya da veri transferi sırasında enerji tüketimini optimize etmektedir ve yalnızca veri transferi gereken durumlarda enerji harcamaktadır. Bu özellik, BLE'yi giyilebilir cihazlar, sağlık izleme sistemleri ve akıllı ev otomasyonu gibi sürekli veri aktarımına ihtiyaç duymayan uygulamalar için oldukça cazip kılmaktadır. Özellikle bu alanlarda, BLE'nin basit ve hızlı bağlantı kurma özellikleriyle sağladığı enerji verimliliği büyük önem taşımaktadır (Bulić, Kojek & Biasizzo, 2019).

BLE'nin popüler uygulama alanlarından biri de akıllı ev teknolojileridir. Akıllı ev sistemlerinde kullanılan BLE özellikli cihazlar, düşük enerji tüketimi sayesinde uzun pil ömrü sunmaktadır ve bu cihazlar merkezi bir kontrol sistemi üzerinden yönetilebilmektedir. Bu durum, evdeki aydınlatma, ısıtma ve güvenlik gibi sistemlerin enerji verimli bir şekilde kullanılmasına yardımcı olmaktadır. BLE ile donatılmış akıllı ev cihazları, kullanıcıların enerji tüketimini daha etkili bir şekilde yönetmelerini sağlamaktadır (Researcher, 2020).

Perakende sektöründe ise BLE, müşterilere mağaza içindeki belirli alanlarda özel teklifler sunmak için kullanılmaktadır. BLE İşaret Teknolojisi sayesinde müşteriler, mağazanın belirli bölümlerine geldiklerinde özel indirim ve promosyonlarla karşılaşabilmektedirler. Bu teknoloji, alışveriş deneyimini kişiselleştirerek müşteri

memnuniyetini artırırken, perakendecilere de hedeflenen promosyonlar yapma imkânı tanımaktadır. Örneğin, bir müşteri mağazada belirli bir ürüne yaklaştığında, BLE İşaret Teknolojisi sayesinde bu ürüne özel bir indirim teklifi alabilmektedir. Bu tür kişiselleştirilmiş deneyimler, müşterilerin mağaza içindeki hareketlerinin izlenmesi ve analiz edilmesi ile gerçekleştirilebilmektedir (Spachos & Plataniotis, 2020).

Enerjiyi verimli kullanan BLE, düşük maliyetli ve kullanıcı dostu bir teknoloji olarak modern kablosuz iletişim sistemlerinde önemli bir yer edinmiştir. Bu avantajları sayesinde Nesnelerin İnterneti ekosisteminde de geniş bir uygulama alanı bulmaktadır. BLE'nin sunduğu düşük enerji tüketimi ve uzun pil ömrü gibi özellikler, onu sağlık izleme cihazlarından akıllı ev sistemlerine, giyilebilir teknolojilerden perakende sektörüne kadar geniş bir yelpazede vazgeçilmez bir çözüm haline getirmiştir (Spachos & Plataniotis, 2020).

3.5.4. LoraWAN

LoRa, düşük güç tüketimi ile uzun mesafe kablosuz iletişim sağlayan bir teknolojidir. Gelişmiş modülasyon teknikleri sayesinde çok düşük veri hızlarında bile kilometrelerce mesafeye veri iletimi yapabilen Lora, özellikle Nesnelerin Internet uygulamalarında kullanılmak üzere tasarlanmıştır. Bu özellik, LoRa'yı akıllı şehirler, çevresel izleme ve tarım gibi alanlarda ideal bir çözüm haline getirilmektedir (Petäjäjärvi, Mikhaylov, Hämäläinen & Iinatti, 2017).

LoRa'nın teknik yapısı, düşük güç tüketimi ve geniş kapsama alanı ile dikkat çekmektedir. Bu iki özellik, sensörler ve diğer Nesnelerin İnterneti cihazlarının yıllarca herhangi bir pil değişimi gerektirmeden çalışabilmesini desteklemektedir. Özellikle kırsal ve uzak alanlarda, bu cihazlar uzun süreli izleme görevlerini yerine getirebilmektedirler (Centenaro & Vangelista, 2016).

Güvenlik, LoRa teknolojisinin önemli bir başka yönüdür. Veri iletimi sırasında kullanılan şifreleme teknikleri, iletilen verilerin güvenliğini maksimize etmektedir. Bu, özellikle enerji santralleri gibi kritik altyapıların izlenmesi durumlarında büyük önem taşımaktadır (Petäjäjärvi, Mikhaylov, Hämäläinen & Iinatti, 2017).

Akıllı şehirlerde LoRa'nın kullanımı, enerji yönetimi, trafik kontrolü ve halka açık alanların bakımı gibi çeşitli işlevler için ideal bir çözümdür. LoRa'nın sağladığı uzun

menzilli ve düşük güç tüketimli iletişim, bu tür uygulamalar için gerekli olan temel gereksinimleri karşılamaktadır (Seo & Singh, 2018).

3.5.5. Mesaj Kuyruğu Telemetri Taşıması

Mesaj Kuyruğu Telemetri Taşıması (Message Queuing Telemetry Transport-MQTT), düşük bant genişliği sunan ve cihaz kaynaklarıyla etkili iletişim sağlayan bir mesaj protokolüdür ve özellikle Nesnelerin İnterneti uygulamaları için elverişlidir. MQTT, yayıncı/abone modelini kullanarak çalışmaktadır. Bu modelde yayıncılar, abonelerin bilgi almasını sağlayan bir aracı (broker) üzerinden mesajlarını göndermektedir (Kim & Kyung, 2024). Bu, yayıncıların ve abonelerin doğrudan iletişim kurma ihtiyacını ortadan kaldırmaktadır böylece ağ trafiğini ve gecikme sürelerini azaltmaktadır. MQTT'nin bu esnek ve ölçeklenebilir yapısı, akıllı ev uygulamalarından sanayi otomasyon sistemlerine kadar geniş bir kullanım alanı sunmaktadır. (Cornel-Cristian, Gabriel, Arhip-Calin & Zamfirescu, 2019).

MQTT, farklı hizmet kalitesi seviyeleriyle mesaj teslimini garanti eder. Bu, uygulamanın gereksinimlerine göre esneklik sunmaktadır ve ağ kaynaklarını verimli bir şekilde kullanmayı desteklemektedir. (Cornel-Cristian, Gabriel, Arhip-Calin & Zamfirescu, 2019)

Güvenlik konusunda MQTT, bazı temel zayıflıklara sahiptir çünkü protokol varsayılan olarak şifreleme veya kapsamlı kimlik doğrulama mekanizmaları içermemektedir. Bu nedenle, hassas verilerin güvenliğini sağlamak amacıyla genellikle Taşıma Katmanı Güvenliği (TLS) ile birlikte kullanılması önerilmektedir**.** TLS, verilerin şifrelenmesini ve ağ üzerinden güvenli bir şekilde iletilmesini sağlayarak iletişim kanalını korumaktadır. MQTT'nin esnekliği ve düşük maliyetli yapısı, Nesnelerin İnterneti çözümleri için çok elverişlidir. Ancak güvenlik için ek önlemler alınmalıdır. Güvenliği artırmak ve potansiyel zafiyetleri gidermek için, uygun şifreleme protokolleri ve kimlik doğrulama yöntemleri kullanılmalıdır (Alkhafajee, Al-Muqarm, Alwan & Mohammed, 2021)

3.5.6. Kısıtlı Uygulama Protokolü

Kısıtlı Uygulama Protokolü (Constrained Application Protocol-CoAP), düşük güç ve kapasiteye sahip cihazlar arasında verimli veri iletişimini sağlamak amacıyla tasarlanmış bir ağ protokolüdür (Naik & Joshi 2017), CoAP, standartlaştırılmış, basit,

düşük bant genişliği gerektiren ve minimal veri değişimini destekleyen bir protokoldür (Batmaz & Doğan, 2021). Yoğun olarak Nesnelerin İnterneti uygulamalarında kullanılan bu protokol, HTTP'ye benzer bir yapıya sahiptir. CoAP, UDP üzerinden çalışır ve güvenilmez ağ ortamlarında bile güvenilir veri aktarımı sağlamak için mesaj tekrarlama gibi mekanizmaları içermektedir.

CoAP, genellikle kablosuz sensör ağları, akıllı ev otomasyon sistemleri ve diğer Nesnelerin İnterneti uygulamaları gibi kaynak kısıtlı ortamlar için uygun şartlar sunmaktadır. CoAP, düşük güç tüketimi ve basit işlem gereksinimleri nedeniyle bu tür ortamlarda etkili bir şekilde çalışabilmektedir. Batarya ile çalışan veya sınırlı işlem kapasitesine sahip cihazlar için uygun olan CoAP, ayrıca hafif şifreleme ve güvenlik standartları sunarak veri iletişimini güvence altına almaktadır (Coetzee, Oosthuizen & Mkhize, 2018).

CoAP, enerji verimliliğini optimize etmek ve ağ üzerindeki yükü azaltmak amacıyla tasarlanmıştır. Protokol, düşük maliyetli uygulamalar için geliştirilen yöntemlerle, ağ üzerindeki veri paketlerinin boyutunu ve sayısını minimize ederek iletişim kurmaktadır (Coetzee, Oosthuizen & Mkhize, 2018).

BÖLÜM 4

KLASİK MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME ALGORİTMALARI

Klasik makine algoritmaları ve derin öğrenme algoritmaları aşağıdaki gibi gruplandırılıp, detaylandırabilmektedir.

4.1. Klasik Makine Öğrenmesi Algoritmaları

Klasik makine öğrenmesi algoritmaları, verilerdeki örüntüleri ve ilişkileri öğrenerek tahmin yapmayı veya verileri sınıflandırmayı hedeflemektedir. Bu algoritmalar, belirli bir veri kümesi üzerinden genellemeler yaparak yeni verilere uygulandığında doğru sonuçlar üretmeyi amaçlamaktadır. Öğrenme süreci sırasında, algoritmalar verideki örüntüleri keşfederek karar verme kuralları geliştirmektedir. Hem doğrusal hem de doğrusal olmayan ilişkileri modelleyebilen bu yöntemler, farklı türdeki problemlerde geniş bir kullanım alanına sahiptir (Ray, 2019).

Klasik Makine Öğrenmesi Algoritmalarından Destek Vektör Makineleri, K-En Yakın Komşular Algoritması, Karar Ağaçları, Naive Bayes Algoritması, AdaBoost Algoritması ve Rastgele Orman Algoritması aşağıda detaylandırılmaktadır.

4.1.1. Destek Vektör Makineleri

Destek Vektör Makineleri (SVM) kavramı ilk olarak 1964'te Vladimir Vapnik ve Alexey Chervonenkis tarafından tanıtılmıştır ve o zamandan beri bu algoritma birçok disiplinde yaygın bir popülerlik kazanmıştır. SVM'ler, protein sıralamadan metin sınıflandırmasına, yüz tanımadan otonom araçlar ve robotik sistemlere kadar çeşitli işlemlerde kullanılmaktadır (Vapnik & Chervonenkis, 1964).

Destek Vektör Makineleri (SVM), sınıflandırma, regresyon ve aykırı değer tespiti içeren karmaşık sorunları çözmek için tasarlanmış güçlü bir makine öğrenme algoritmasıdır. SVM'nin temel amacı, veri noktalarını etiketlerine veya çıktılarına göre farklı sınıflara ayıran optimum ayırıcı düzlemi bulmaktır. Bu algoritma, karmaşık veri

dönüşümlerini ele alma becerisi nedeniyle sağlık, doğal dil işleme, sinyal işleme, konuşma tanıma ve görüntü analizi gibi çeşitli alanlarda yaygın olarak kullanılmaktadır. (Hearst, Dumais, Osuna, Platt & Scholkopf, 1998)

Bir SVM'nin merkezinde, farklı sınıflar arasında bir karar sınırı görevi gören ayırıcı düzlem bulunmaktadır. Algoritma, farklı sınıfların veri noktaları arasındaki marjı en üst düzeye çıkaran ayırıcı düzlemi belirleyerek çalışmaktadır. Bu marj, sınırlar içindeki veri noktalarından hiçbirini kesmeden sınıflar arasında korunabilen en büyük mesafedir. Doğrusal olarak ayrılabilir sınıflandırma problemleri için böyle bir ayırıcı düzlemi tanımlamak basittir, ancak gerçek dünya uygulamalarında veriler genellikle doğrusal değildir ve verileri doğru şekilde sınıflandırmak için daha karmaşık teknikler gerekmektedir. SVM'ler başlangıçta, amacın iki sınıf arasında ayrım yapmak olduğu ikili sınıflandırma sorunları için geliştirilmiştir. Bununla beraber makine öğrenimindeki gelişmelerle birlikte SVM'ler, birden fazla ikili sınıflandırıcıyı birleştirerek çok sınıflı sınıflandırma problemlerini ele almak üzere geliştirilmiştir. Bu, SVM'lerin daha karmaşık sınıflandırma görevlerini verimli bir şekilde ele almasına yardım etmektedir (Chamasemani & Singh, 2011)

SVM'lerin temel bir özelliği, çekirdek yöntemlerinin kullanılmasıdır. Çekirdek işlevleri, doğrusal olmayan veri noktalarını ayırmanın daha kolay hale geldiği daha yüksek boyutlu alanlara dönüştürmeye yardımcı olmaktadır. "Çekirdek hilesi" olarak bilinen bu teknik, SVM'lerin önemli bir hesaplama yükü oluşturmadan karmaşık veri kümeleriyle çalışmasına olanak vermektir. Çekirdek işlevi, esasen giriş verilerini, veri noktalarının daha kolay ayrılabileceği yeni bir alana eşleştirmektedir ve SVM'nin daha net bir karar sınırı oluşturmasına yardımcı olmaktadır (Kivinen, Smola & Williamson, 2004).

SVM'lerin temel avantajlarından birisi hem doğrusal hem de doğrusal olmayan verileri işleyebilme yeteneğidir. Veriler doğrusal olarak ayrılabilir olduğunda, verileri sınıflandırmak için basit bir doğrusal SVM kullanılabilmektedir. Ancak, gerçek dünya senaryolarının çoğunda veriler doğrusal değildir ve doğrusal olmayan SVM'ler burada devreye girmektedir. Doğrusal olmayan SVM'ler, verileri daha yüksek boyutlu bir alana eşleştirmek için çekirdek fonksiyonlarından yararlanmaktadır ve bu da optimum bir ayırıcı düzlemi bulmayı kolaylaştırmaktadır (Suykens, 2001).

En iyi ayırıcı düzlemi belirlemek için olasılık temelli bir yaklaşım kullanan lojistik regresyonun aksine, SVM optimum karar sınırını belirlemek için istatistiksel yöntemlere dayanmaktadır. Her iki algoritma da sınıfları ayıran ayırıcı düzlemi bulmayı amaçlasa da SVM'ler özellikle küçük ve karmaşık veri kümeleriyle çalışırken etkilidir. Lojistik regresyonun yüksek doğruluk elde edemediği durumlarda, çekirdeği olmayan SVM daha verimli bir alternatif olarak hizmet edebilmektedir (Purnama, Ginantra, Darma & Udayana 2023; Wang & Hu 2005).

4.1.2. K-En Yakın Komşular Algoritması

İlk olarak 1951'de Fix ve Hodges tarafından desen tanıma için parametrik olmayan bir yöntem olarak tanıtılan K-En Yakın Komşular (KNN) Algoritması, 1967'de Cover ve Hart tarafından daha da geliştirilmiştir (Fix & Hodges, 1951; Cover & Hart 1967). Günümüzde KNN Algoritması, öneri motorları, metin sınıflandırması, tıbbi teşhis ve borsa analizi dâhil olmak üzere çok çeşitli uygulamalarda kullanılmaktadır.

KNN Algoritması hem sınıflandırma hem de regresyon problemlerinde kullanılan, basit ama güçlü bir denetimli öğrenme yöntemidir. Bu algoritma, yeni bir veri noktasının sınıfını veya değerini, veri kümesindeki en yakın komşularına dayanarak belirlemektedir. KNN Algoritması, özellikle veri noktaları arasındaki mesafenin tahminleri doğrudan etkilediği durumlarda tercih edilmektedir. Bir diğer dikkat çekici özelliği, algoritmanın eğitim sürecinde özel bir model oluşturmaması, bunun yerine tüm verileri saklayıp sadece tahmin aşamasında hesaplamalar yapmasıdır. Bu yüzden KNN Algoritması, "tembel öğrenme" algoritmaları kategorisine girmektedir.

KNN Algoritmasının işleyişi oldukça basittir. Yeni bir veri noktası geldiğinde, algoritma en yakın K veri noktasını bulmaktadır ve sınıflandırma görevlerinde bu komşuların çoğunluğunun sınıfını yeni noktaya atamaktadır. Regresyon problemlerinde ise, en yakın K komşunun değerlerinin ortalaması alınarak tahmin yapılmaktadır. Ancak, K değerinin doğru seçilmesi büyük önem taşımaktadır. Küçük bir K değeri modelin gürültüye duyarlı hale gelmesine neden olabilirken, çok büyük bir K değeri de modelin genel performansını zayıflatabilmektedir.

Bu algoritma, veri kümesinin yapısı hakkında herhangi bir varsayımda bulunmadığından, farklı veri türlerine uyum sağlayabilme esnekliği sunmaktadır. Mesafe ölçümü olarak genellikle Öklid mesafesi kullanılmaktadır, ancak kategorik veriler için

Hamming mesafesi gibi farklı metrikler de uygulanabilmektedir. Bunun yanı sıra, verilerin farklı ölçü birimlerinde olması durumunda, özelliklerin standardizasyonu önem kazanmaktadır.

KNN Algoritmasının temel avantajı, anlaşılması ve uygulanması kolay bir algoritma olmasıdır. Ancak, büyük veri kümelerinde her yeni tahmin için tüm veri noktalarıyla mesafe hesaplanması gerektiğinden, hesaplama yükü artabilmektedir. Bu nedenle KNN Algoritması, daha küçük veya orta büyüklükteki veri setleri için daha uygun bir yöntemdir. Performansı büyük ölçüde K değerine ve seçilen mesafe ölçüm yöntemine bağlı olarak değişmektedir, bu nedenle parametrelerin dikkatli bir şekilde ayarlanması optimum sonuçlar elde etmek için kritik bir faktördür

4.1.3. Karar Ağaçları

Karar ağaçları, sınıflandırma ve regresyon görevleri için kullanılan popüler bir makine öğrenimi algoritmasıdır. Karar ağaçları, veri içerisindeki örüntüleri ve yapıları ortaya çıkarmak için karar kuralları seti oluşturmaktadır. Temelde, karar ağaçları bir dizi evet/hayır kararlarına dayalı sorular serisi ile modeli oluşturmaktadır. Her düğüm bir soruyu göstermektedir ve her dal bu sorunun bir yanıtını temsil etmektedir. Yaprak düğümler ise modelin tahmin veya sınıflandırma sonucunu vermektedir. Bu yapısal özellikler, karar ağaçlarının hem teorik hem de pratik uygulamalar açısından çekici hale gelmesini sağlamaktadır (Abdelhalim & Traore, 2009).

Karar ağaçlarının geliştirilmesi ve kullanımı, veri madenciliği ve yapay zekâ alanlarında önemli bir yere sahiptir. Örneğin, ID3, C4.5, CART ve CHAID en bilinen karar ağacı algoritmalarından bazılarıdır. Bu algoritmalar, veri setlerinden bilgi çıkarmak için çeşitli yöntemler kullanmaktadır. ID3 ve C4.5, bilgi kazancına dayalı kriterler kullanırken, CART Gini indeksini kullanır ve CHAID ise ki-kare testine dayanmaktadır (Safavian & Landgrebe, 1991; Mienye & Jere, 2024).

Karar ağaçları, özellikle büyük ve karmaşık veri setleri üzerinde çalışırken, anlaşılması ve yorumlanması kolay modeller oluşturması nedeniyle tercih edilmektedir. Bu modeller, hangi özelliklerin hedef değişkeni en iyi şekilde tahmin ettiğini belirlemede kullanıcıya büyük bir öngörü vermektedir. Ayrıca, karar ağaçları eksik verilerle çalışabilmektedir ve veri ön işleme gereksinimini azaltmaktadır, bu da onları diğer

makine öğrenimi tekniklerine göre daha kullanışlı hale getirmektedir (Barros, Basgalupp, de Carvalho & Freitas, 2012; Wang, Yang & Ren, 2011).

Karar ağaçlarının başka bir önemli avantajı da hem kategorik hem de sürekli değişkenlerle çalışabilmesidir. Bu özellik, farklı türde veriler içeren gerçek dünya uygulamaları için çok kullanışlıdır. Örneğin, tıbbi tanı koyma, müşteri tercih analizi ve kredi risk değerlendirme gibi alanlarda etkin bir şekilde kullanılabilmektedir (Lee, Sim & Hong, 2024; Wenhua & Zheng, 2012)

Bununla birlikte, karar ağaçlarının aşırı öğrenme gibi bazı dezavantajları da vardır. Aşırı öğrenme, modelin eğitim verilerine çok iyi uyum sağlaması ancak genelleme yapamaması durumudur. Bu sorunu çözmek için çeşitli budama teknikleri geliştirilmiştir. Budama, karar ağacının aşırı büyümesini önleyerek daha genel bir model oluşturmayı amaçlamaktadır (Nandgaonkar, Attar & Sinha, 2014; El Zein, Lemay & Huguenin, 2024).

Karar ağaçları ayrıca, toplu öğrenme yöntemleriyle birleştirildiğinde performanslarını artırabilmektedir. Rastgele ormanlar gibi teknikler, birden fazla karar ağacını birleştirerek tahmin doğruluğunu artırmaktadır ve aşırı öğrenmeyi azaltmaktadır. Bu yöntemler, karar ağaçlarının zayıf yönlerini dengelemektedir ve daha sağlam modeller oluşturmaktadır (Ahmad & Brown, 2014; Author, 2022).

4.1.4. Naive Bayes Algoritması

Naive Bayes Algoritması, olasılık temelli bir model öğrenen algoritmalar grubuna dâhildir. Bu algoritma, Bayes teoremine dayanarak, hedef sınıfın değeri bilindiğinde her bir özellik çiftinin birbirinden bağımsız olduğunu varsaymaktadır. Naive Bayes Algoritması, verilen bir özellik seti göz önüne alındığında, her sınıfın koşullu olasılığını hesaplamaktadır ve en yüksek olasılığa sahip sınıfı tahmin olarak sunmaktadır (Aridas, Karlos, Kanas, Fazakis & Kotsiantis, 2020). Bu yöntem, özellikle metin sınıflandırma ve spam tespiti gibi alanlarda sıklıkla kullanılmaktadır.

Naive Bayes algoritmasının basitliği, onu çeşitli uygulamalar için cazip bir seçenek yapmaktadır. Ancak, bazı durumlarda, özellikler arası bağımsızlık varsayımı gerçek dünya verileri için çok kısıtlayıcı olabilmektedir. Bu nedenle, araştırmacılar bu

varsayımı hafifletmek için çeşitli yöntemler önermektedirler. Bunlardan biri, özellik ağırlıklandırma yöntemidir ki bu yöntem, her bir özelliğin model üzerindeki etkisini yeniden değerlendirerek sınıflandırıcının performansını artırabilmektedir.

Bazı çalışmalarda, Naive Bayes algoritması karmaşık veri kümeleri üzerindeki performansını artırmak için özellik seçimi ve özellik dönüşüm teknikleri kullanılmıştır. Bu teknikler, modelin daha fazla bilgi içeren ve daha az gürültülü özelliklere odaklanmasına yardımcı olmaktadır. Özellikle dengesiz veri kümeleri üzerinde çalışırken, Naive Bayes algoritmasının azınlık sınıfına ait örnekleri yeterince temsil etmediği gözlemlenmektedir. Bu sorunu çözmek için, çeşitli yeniden örnekleme teknikleri ve maliyet duyarlı öğrenme yöntemleri önerilmektedir. Bu yöntemler, modelin azınlık sınıfına daha fazla önem vermesini sağlayarak daha dengeli ve adil tahminler yapmasına olanak tanımaktadır (Aridas, Karlos, Kanas, Fazakis & Kotsiantis, 2020).

4.1.5. AdaBoost Algoritması

AdaBoost algoritması, Freund ve Schapire tarafından geliştirilen bir toplu öğrenme algoritmasıdır. AdaBoost, sınıflandırma görevlerinde yüksek doğruluk elde etmek için birçok orta dereceli sınıflandırıcıyı bir araya getirerek güçlü bir sınıflandırıcı oluşturmaktadır (Xu & Yuan, 2020). Ortalama derecedeki sınıflandırıcılar, bağımsız olarak çoğunlukla ortalama derecede performans gösteren ancak tamamen başarılı olmayan basit sınıflandırıcılardır. AdaBoost algoritması, her iterasyonda zayıf sınıflandırıcıların hata oranını optimize etmektedir ve böylece sınıflandırıcıların ağırlıklarını güncelleyerek daha iyi bir genel sınıflandırıcı elde etmektedir (Wang vd., 2019).

AdaBoost algoritmasının temel çalışma prensibi, yanlış sınıflandırılmış örneklerin ağırlıklarını artırarak bu örneklere daha fazla dikkat edilmesini sağlamak ve sonraki iterasyonlarda bu örneklerin doğru sınıflandırılması olasılığını yükseltmektir (Wu & Nagahashi, 2014). Böylelikle, AdaBoost algoritması her iterasyonda hataları minimize etmek için yeniden düzenlenmiş bir veri kümesiyle çalışmaktadır. AdaBoost, bu süreci daha da ileri götürerek, yanlış sınıflandırılan örneklerin yanı sıra doğru sınıflandırılmış örneklerin ağırlıklarını da optimize ederek daha kararlı bir sınıflandırıcı oluşturmaktadır (Xu, Baraldi, Lu & Zio, 2022).

AdaBoost'un özellikle veri sınıflandırma, nesne algılama ve ağ izleme gibi alanlarda oldukça başarılı olduğu kanıtlanmıştır. Örneğin, yüksek hızlı trenlerin otomatik kapılarında anomali tespitinde kullanılan AdaBoost algoritması, genellikle sınırlı anomalilik verisi ile çalışılan durumlarda, tek bir anormal veri sınıfıyla eğitilmektedir (Hu, Gao, Wang, Wu & Maybank, 2014). Aynı zamanda, AdaBoost Algoritmasının uyarlamalı yapısı sayesinde, çok boyutlu veri analizlerinde örneğin LED ömrünün tahmininde veya ağ izleme sistemlerinde saldırı tespitinde de kullanılmaktadır. Bu uygulamalarda, zayıf sınıflandırıcılar olarak kullanılan GMM (Gauss Karışım Modelleri) veya Karar Ağaçları gibi modeller, ağırlıklandırma mekanizması sayesinde daha yüksek tespit oranları sağlamıştır (Lu, Zhang & Sun, 2017).

AdaBoost algoritmasının gücü, hataları minimize ederek sınıflandırma doğruluğunu artırmasından gelmektedir. Algoritma, her zayıf sınıflandırıcının hata oranını optimize ederken, aynı zamanda sınıflandırıcıların ağırlıklarını güncelleyerek tüm sınıflandırma sürecini geliştirmektedir. Bu özellik, AdaBoost algoritmasının yüksek doğruluğa sahip, kararlı ve genel hatalardan bağımsız modeller üretmesine olanak tanımaktadır. Bu nedenle, AdaBoost çok çeşitli uygulamalarda etkili bir sınıflandırıcı olarak yerini almıştır (Ma & Ding, 2005).

4.1.6. Rastgele Orman Algoritması

Rastgele Orman (Random Forest) algoritması, sınıflandırma ve regresyon gibi görevlerde kullanılan, karar ağaçlarına dayalı popüler bir topluluk öğrenme modelidir (Ghosh & Cabrera, 2022). Karar ağaçları, eğitim verileri kullanılarak inşa edilmektedir. Bu öğrenme süreci, ağacı adım adım oluşturmaktadır ve uygulama bağlamında girdilerin önem sırasına göre kararlar verilmektedir. Tüm eğitim verilerini kullanarak ilk olarak en önemli özellik belirlenmektedir ve veriler bu özelliğin sonucuna göre bölünmektedir Daha sonra bu işlem her bir veri alt kümesi için tekrar edilmektedir ve her seferinde bir sonraki en önemli özellik ile yeni bir bölme yapılmaktadır. Bu işlem, ağacın tüm dallarında yaprak düğümleri bulunana kadar devam etmektedir. İşlemler sırasında, orijinal veri kümesindeki gözlemlerden bir kısmı tekrar edilerek yeni bir veri kümesi oluşturulmaktadır. Bu yeni örnekler üzerinde her bir karar ağacı eğitilmektedir ve sonuçta tüm ağaçlar çoğunluk oyu prensibi ile bir araya getirilmektedir. Yeni bir gözlem

sınıflandırılırken, tüm karar ağaçlarının tahminleri değerlendirilmektedir ve en sık verilen yanıt son karar olarak alınmaktadır.

Bazı sınıflandırmalarda, tüm karar ağaçları yüksek korelasyon gösterme eğilimindedir. Ancak, Rastgele Orman Algoritması bu duruma farklı bir çözüm getirmektedir ve her bölmede yalnızca rastgele bir özellik alt kümesi dikkate alınmaktadır. Böylece, karar ağaçlarının birbirleriyle korelasyonu azaltılmaktadır ve daha çeşitli ağaçlar oluşturulmaktadır. Bu yöntem, Rastgele Orman algoritmasının güçlü ve bağımsız bir model olmasını sağlamaktadır. Rastgele Orman Algoritması, sınıflandırma ve regresyon gibi farklı görevlerde etkili sonuçlar sunarken, veri dağılımı veya girdi özelliklerinin standartlaştırılması gibi özel gereksinimlere ihtiyaç duymamaktadır. Rastgele Orman Algoritması, makine öğreniminin denetimli öğrenme kategorisine ait olup, etiketlenmiş verilerle eğitilerek yeni veriler üzerinde doğru tahminler yapmak amacını taşımaktadır.

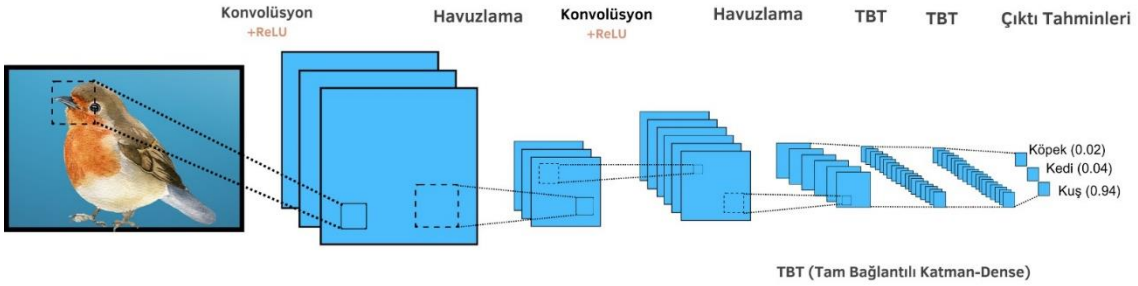
4.2. Derin Öğrenme Algoritmaları

Derin öğrenme algoritmaları, büyük miktarda veriyi kullanarak karmaşık görevleri çözebilen, çok katmanlı yapay sinir ağlarına dayalı makine öğrenmesi yöntemleridir. Geleneksel makine öğrenmesi algoritmalarının aksine, derin öğrenme verilerdeki çok boyutlu ve karmaşık desenleri otomatik olarak öğrenmektedir ve sonuç çıkarmaktadır. Bu sayede, özellikle görüntü tanıma, doğal dil işleme, ses tanıma ve otonom sistemler gibi alanlarda, oldukça yüksek doğruluk oranlarına ulaşmaktadır. Derin öğrenmenin en büyük avantajlarından biri, manuel olarak özellik çıkarma ihtiyacını azaltmasıdır. Ağ, veriyi katmanlar arasında işleyerek kendi başına özellikleri öğrenebilmektedir. Yüksek performansına rağmen, derin öğrenme algoritmaları büyük veri kümeleri ve güçlü hesaplama kaynakları gerektirmektedir. Bu algoritmaların uygulanmasında GPU'lar ve TPU'lar gibi donanımlar, hesaplamaların hızlandırılmasına önemli katkı vermektedir. Derin öğrenme, yapay zekânın hızlı gelişimini sağlayan en önemli teknolojilerden biri haline gelmiştir.

Derin Öğrenme algoritmalarından Konvolüsyonel Sinir Ağları, Oto Kodlayıcılar, Tekrarlayan Sinir Ağları ve Rekabetçi Üretici Ağlar aşağıda detaylandırılmaktadır.

4.2.1. Konvolüsyonel Sinir Ağları

Konvolüsyonel Sinir Ağları (CNN), derin öğrenmenin en etkili yöntemlerinden biri olarak öne çıkmaktadır. Özellikle bilgisayarla görme alanında, görüntülerin tanımlanması, belirlenmesi ve ayrıştırılması gibi görevlerde üstün bir başarı gösteren CNN'ler, dijital görüntülerin içindeki karmaşık desenleri çözmede kritik bir rol oynamaktadırlar. CNN'ler, verileri ızgara benzeri bir yapıda işleyen özel bir yapay sinir ağı türüdür. Dijital bir görüntü, her biri belli bir parlaklık ve renk değerine sahip piksellerden oluşmaktadır. Bu pikseller, bir araya gelerek görüntünün bütününü oluşturmaktadır. CNN'ler ise bu ızgara yapısını kullanarak pikselleri anlamlı özelliklere dönüştürmektedirler. Şekil 4.1'de CNN'in genel mimarisi gösterilmektedir.



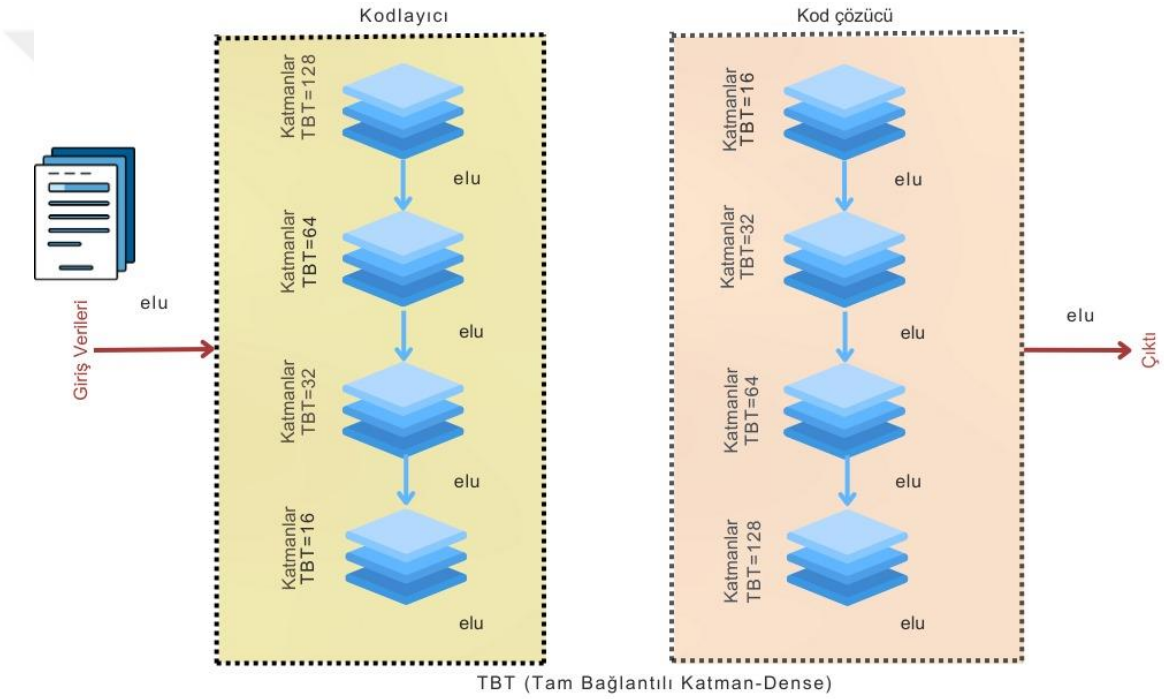
Şekil 4.1. CNN Genel Mimarisi

CNN'lerin en önemli özelliklerinden biri, konvolüsyon adı verilen bir yöntemle giriş verisine bir filtre uygulayarak aktivasyonlar oluşturmasıdır. Bu aktivasyonlar, giriş verileri üzerindeki belirli özelliklerin yerini ve şiddetini gösteren özellik haritalarına dönüşmektedir. Görüntü sınıflandırma gibi görevlerde, CNN'ler aynı anda birçok filtre öğrenme yeteneğine sahiptir. Bu filtreler, eğitim verilerine özgü olarak optimize edilmektedir ve modelin doğruluğunu artırmaktadır (Géron, 2019).

CNN'ler, günümüzde otonom araçlar, güvenlik kamerası sistemleri gibi birçok alanda yaygın olarak kullanılmaktadırlar. CNN'ler görüntü verilerinin karmaşıklığını anlama ve işleme becerileri sayesinde, bilgisayarla görme alanında vazgeçilmez bir teknoloji haline gelmiştir. Saldırı tespit sistemleri gibi birçok uygulamada ara katman olarak özellik çıkarım işlevinde yoğun olarak kullanılmaktadır.

4.2.2. Oto Kodlayıcılar

Oto Kodlayıcılar (AE), gözetimsiz öğrenmede kullanılan, açık etiketler gerektirmeden verileri işlemek üzere tasarlanmış ilgi çekici bir sinir ağı sınıfıdır. Giriş verilerini kompakt, daha düşük boyutlu bir gösterime sıkıştırarak ve ardından orijinal verilere yakın olacak şekilde yeniden yapılandırarak çalışmaktadırlar. Verileri sıkıştırma ve ardından yeniden oluşturma yeteneği, AE'leri özellikle veri azaltma, anormallik tespiti ve hatta sentetik veri oluşturmayı içeren görevler için kullanışlı hale getirmektedir. Şekil 4.2'de AE'nin genel mimarisi gösterilmektedir.



Şekil 4.2. AE Genel Mimarisi

Bir AE'nin mimarisi iki ana bölümden oluşur: kodlayıcı ve kod çözücü. Kodlayıcı, yüksek boyutlu giriş verilerini gizli alan olarak bilinen daha düşük boyutlu bir forma dönüştürmekten sorumludur. Bu dönüşüm, daha az önemli bilgileri atarken giriş verilerinin en önemli özelliklerini yakalamaktadır. Gizli alan, esasen orijinal verilerin sıkıştırılmış bir sürümüdür ve temel özelliklerinin bir özeti olarak düşünülebilmektedir. Kod çözücü daha sonra bu sıkıştırılmış gösterimi almaktadır ve daha sonra orijinal verileri mümkün olduğunca doğru bir şekilde yeniden oluşturmaya çalışmaktadır. Sıkıştırma sürecini tersine çevirerek gizli alanı girişin orijinal boyutuna genişletmektedir. Bu

yeniden yapılandırmanın başarısı, orijinal girdi ile yeniden yapılandırılan çıktı arasındaki farkı ölçen bir kayıp fonksiyonu kullanılarak değerlendirilmektedir. Kullanılan yaygın kayıp fonksiyonları arasında sürekli veriler için ortalama MSE (karesel hata) ve ikili veriler için ikili çapraz entropi bulunmaktadır (Géron, 2019).

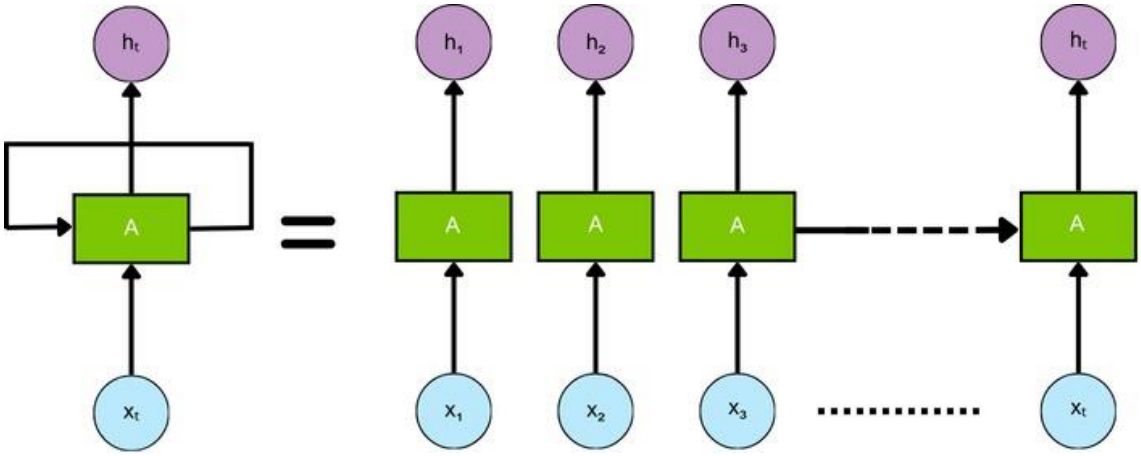
AE'lerin temel özelliklerinden biri, kaybı en aza indirecek şekilde verileri kodlamayı ve kod çözmeyi öğrenme yetenekleridir ve bu sayede yeni, görülmemiş verilere genelleme yapma yeteneklerini geliştirmektedirler. Kodlama, kod çözme ve parametreleri geri yayılım yoluyla ayarlamanın bu yinelemeli süreci, AE'nin performansını zaman içinde detaylıca ayarlamasına olanak tanımaktadır. Bu durum, özellikle verilerin gürültülü olduğu veya verimli bir gösterimin gerekli olduğu uygulamalar için faydalıdır.

AE'ler yalnızca boyut azaltma için değil, aynı zamanda diğer karmaşık görevler için de kullanılmaktadırlar. Örneğin, yeniden yapılandırma hatası bu anormallikleri vurgulayabileceğinden, anormallik tespitinde normalden önemli ölçüde sapan alışılmadık desenleri belirlemek için AE'ler kullanılabilmektedirler. Görüntü işlemede, AE'ler önemli özellikleri korurken gürültüyü gidermeyi öğrenerek gürültüleri görüntüden arındırmak için kullanılabilmektedirler (Géron, 2019).

AE'ler denetimli öğrenmede çok yönlü, güçlü bir araçtırlar ve etiketli örneklere ihtiyaç duymadan verileri sıkıştırmak, yeniden yapılandırmak ve anlamak için bir imkân sunmaktadırlar. Karmaşık veri dönüşümlerini işleme yetenekleri ve çeşitli görevlere uyarlanabilirlikleri, AE'leri birçok makine öğrenimi uygulamasında değerli bir algoritma haline getirmektedir.

4.2.3. Tekrarlayan Sinir Ağları

Tekrarlayan Sinir Ağları (RNN), her seviyede veri işleyebilen ve önceki seviyelerdeki hesaplama sonuçlarını dikkate alarak modelleri eğiten bir yapay sinir ağı türüdür. RNN'leri diğer ağlardan ayıran en önemli özelliklerinden biri, "hafıza" yetenekleridir. Bu hafıza, ağın önceki verileri kullanarak mevcut veri ve sonuç üzerinde etki yaratmasına olanak tanımaktadır. Şekil 4.3'te RNN genel mimarisi gösterilmektedir.



Şekil 4.3. RNN Genel Mimarisi

RNN'ler, ardışık verilerin kritik olduğu doğal dil işleme, konuşma tanıma ve zaman serisi tahmini gibi pek çok uygulama alanında yaygın olarak kullanılmaktadırlar. Ayrıca, bu ağlar, ağ üzerindeki tehdit ve saldırıların tespiti için geliştirilen Saldırı Tespit Sistemleri programlanmasında rol oynamaktadırlar. RNN'ler, geçmiş verileri değerlendirerek anomalileri daha hassas bir şekilde belirleyebilmektedirler ve siber güvenliğe, önemli bir katkı sağlamaktadırlar (Géron, 2019).

4.2.3.1. Basit Tekrarlayan Sinir Ağları

Bu ağlar, önceki zaman adımlarının çıktıları ve gizli durumlarını bir sonraki adımlara aktararak, gelecekteki sonuçların etkilenmesine olanak tanımaktadır. Dolayısıyla, t zaman adımı, t−1 ve daha önceki adımlardan etkilenebilir bir özellik olarak RNN'lerin zaman içindeki bağımlılıkları öğrenmelerine yardım etmektedir. RNN'ler, dil modelleme, ses tanıma ve video analizi gibi çeşitli alanlarda yaygın olarak kullanılmaktadırlar. Ancak, RNN'ler uzun vadeli bağımlılıkları öğrenirken gradyan kaybolması ve patlaması sorunlarıyla karşılaşabilmektedirler. Bu sorunlar, ağın geçmiş bilgileri etkili bir şekilde hatırlamasını zorlaştırabilmektedirler. Bu nedenle, LSTM (Uzun Kısa Süreli Bellek-Long Short Term Memory) ve GRU (Kapı Kontrollü Tekrarlayan Hücre-Gated Recurrent Unit) gibi gelişmiş RNN türleri, bu sorunları hafifletmek amacıyla geliştirilmiştir (Géron, 2019).

Basit tekrarlayan sinir ağlarında bazı problem vardır. Kaybolan gradyan sorunu, derin sinir ağlarının eğitimi esnasında sıkça karşılaşılmaktadır ve özellikle geri yayılım algoritması kullanılırken gradyanların ağın katmanları boyunca taşınması sırasında

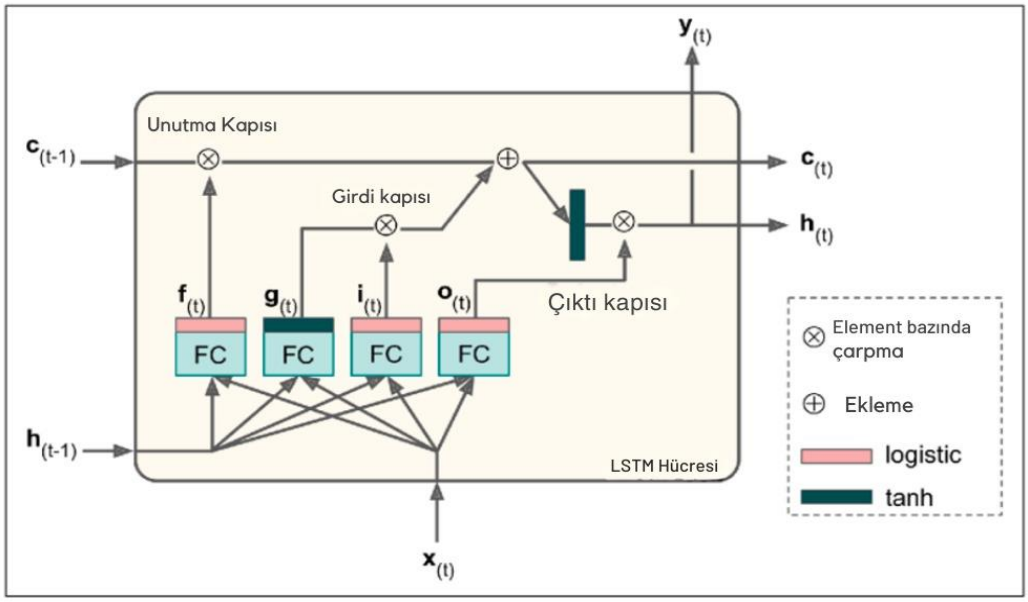
meydana gelen küçülmelerle kendini göstermektedir. Bu sorun, gradyanların küçülmesi nedeniyle ağın özellikle giriş katmanlarına yakın olan katmanlarının yeterince güncellenememesiyle sonuçlanmaktadır. Böylece, bu katmanlarda etkili bir öğrenme gerçekleştirilememektedir. Gradyanların küçülmesi, ardışık küçük sayıların sürekli çarpılması ile gradyanların sıfıra yakın bir değere ulaşmasına ve nihayetinde kaybolmasına neden olmaktadır. Bu durum, derin sinir ağlarının eğitiminin etkinliğini ve verimliliğini önemli ölçüde olumsuz etkileyebilmektedir (Géron, 2019).

Patlayan gradyan sorunu, özellikle geri yayılım algoritması sırasında gradyanların hesaplanması ve katmanlar boyunca taşınması sürecinde ortaya çıkmaktadır. Bu problem, gradyanların ardışık büyük sayılarla çarpılması sonucunda aşırı büyümesiyle kendini göstermektedir. Gradyanların aşırı büyümesi, ağın ağırlıklarının çok büyük değerlere güncellenmesine yol açarak modelin kararsız hale gelmesine neden olmaktadır. Bu durum, ağın öğrenme sürecinin dengesini bozabilmektedir ve bu durum ağırlıkların sonsuz değerler almasına kadar gidebilmektedir. Patlayan gradyan sorunu, derin sinir ağlarının eğitiminin etkinliğini ve kararlılığını ciddi şekilde olumsuz etkileyebilmektedir ve eğitim sürecini zorlaştırabilmektedir.

4.2.3.2. Uzun Kısa Vadeli Hafıza Sinir Ağları

RNN'lerin bir çeşidi olan Uzun Kısa Süreli Bellek (LSTM) Ağları, uzun süreli bağımlılıkları öğrenme kapasitesi ile dikkat çekmektedir. Sepp Hochreiter ve Jürgen Schmidhuber tarafından 1997'de önerilen LSTM'ler, zamana bağlı bilgileri daha uzun süre depolayabilen ve unutma kapıları içeren benzersiz bir yapıya sahiptirler. LSTM'ler, geleneksel RNN'lerden farklı olarak, karmaşık zaman serileri ve sekans verilerinde üstün performans göstermektedir (Géron, 2019). Şekil 4.4'te LSTM genel mimarisi gösterilmektedir.

Geleneksel RNN'ler, ardışık veri dizilerini eğitirken iyi performans sergilemektedirler ancak uzun dizilerdeki bağımlılıkları öğrenmekte zorlanabilmektedirler. Bu RNN türleri, geçmiş bilgileri hızlı bir şekilde unutmaya eğilimlidir. Buna karşın, LSTM'ler, hücre durumu denilen bir mekanizma ile bilgiyi daha uzun süre koruyabilir ve unutma, giriş, çıkış kapıları sayesinde hangi verilerin saklanacağı veya unutulacağına dinamik olarak karar verebilmektedirler.



Şekil 4.4. LSTM Genel Mimarisi

LSTM'nin yapısal özellikleri, dil modelleme, konuşma tanıma ve zaman serisi tahmini gibi alanlarda geleneksel RNN'lere nazaran daha iyi performans göstermesine katkıda bulunmaktadır. LSTM ağları, karmaşık yapılarıyla dört gizli katmanlı geleneksel RNN'lere göre daha gelişmiştir ve bu sayede daha karmaşık öğrenme görevlerini üstlenebilmektedirler.

4.2.3.3. Kapı Kontrollü Tekrarlayan Birim Sinir Ağları

RNN'lerin bir başka türü olan Kapı Kontrollü Tekrarlayan Birim Sinir Ağları (Gated Recurrent Unit-GRU), LSTM Ağlarına benzer şekilde uzun süreli bağımlılıkları öğrenme kapasitesine sahiptir. GRU'nun daha basit bir yapısı vardır. LSTM'ye kıyasla daha az kapı ve parametre içermesi sayesinde hesaplama maliyetlerini düşürürken performanstan ödün vermemektedir (Géron, 2019).

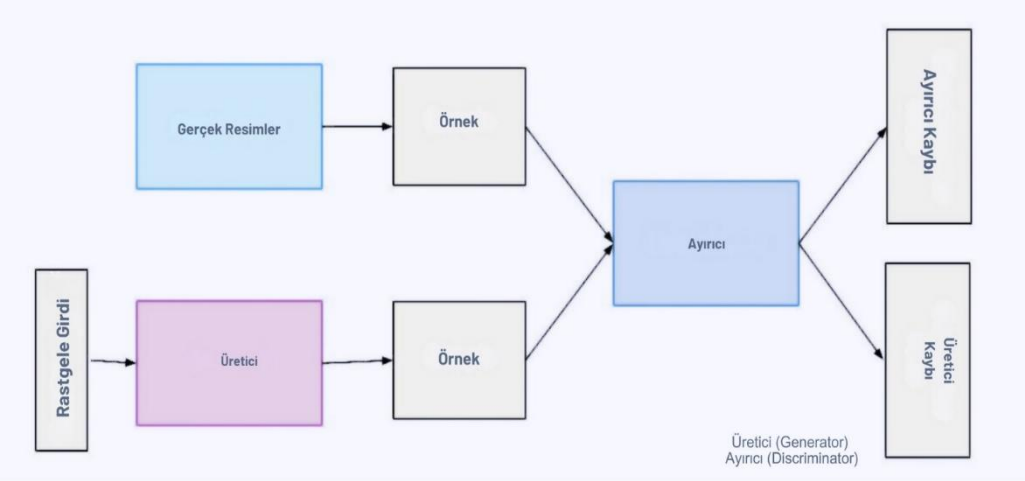
GRU Ağları, geleneksel RNN'lerin zaman serileri ve sekans verilerindeki kısıtlamalarını aşmak için geliştirilmiştir. Güncelleme kapısı ve resetleme kapısı olmak üzere iki temel kapı mekanizması vardır. Güncelleme Kapısı, hangi bilgilerin hatırlanacağını belirlerken, Resetleme Kapısı ise hücre durumunun ne kadarının unutulacağına karar vermektedir. Bu iki kapı, GRU'nun bilgiyi daha etkin bir şekilde işleyebilmesini yardımcı olmaktadır (Géron, 2019).

GRU Ağları, LSTM ağlarına kıyasla daha basit bir yapı sunmaktadır ve benzer görevlerde çoğu zaman benzer veya daha iyi performans gösterebilmektedir. GRU'nun yapısal özellikleri, özellikle dil modelleme, konuşma tanıma ve zaman serisi tahmini gibi alanlarda hızlı ve etkili sonuçlar elde edilmesini sağlamaktadır.

4.2.4. Rekabetçi Üretici Ağlar

Rekabetçi Üretici Ağlar (Generative Adversial Networks-GAN) derin öğrenme alanında önemli bir yenilik olarak kabul edilmektedir. İlk olarak Ian Goodfellow ve çalışma arkadaşları tarafından 2014 yılında tanıtılan GAN'lar, iki sinir ağının üretici (generator) ve ayırt edici (discriminator) birbirleriyle rekabet ettiği bir çerçevede çalışmaktadır (Goodfellow vd., 2014).

Üretici ağ, gerçek veriye benzer sahte örnekler oluşturmayı amaçlarken, ayırt edici ağ ise gerçek ve sahte verileri ayırt etmeye çalışmaktadır. Bu rekabetçi süreç, üretici ağın zamanla daha gerçekçi veriler üretmesini sağlamaktadır. GAN'lar, özellikle görüntü oluşturma, veri artırma ve model öğrenme süreçlerinde geniş bir uygulama yelpazesine sahiptir. Şekil 4.5'te GAN genel mimarisi gösterilmektedir.



Şekil 4.5. GAN Genel Mimarisi

GAN'ların başarısı, çeşitli varyantların geliştirilmesine yol açmıştır. Örneğin, Koşullu GAN (Conditional GAN), belirli koşullara bağlı olarak veri üretimini mümkün kılmaktadır. Ayrıca, Döngüsel GAN (CycleGAN), çiftler halinde olmayan veri kümeleri

arasında dönüşüm yapabilmektedir (Zhu, Park, Isola & Efros, 2017). Bu çeşitlilik, GAN'ların farklı alanlarda uyarlanabilirliğini artırmaktadır.

Bununla birlikte, GAN'ların eğitim süreçleri bazı zorluklar içermektedir. Mod çökmesi (mode collapse) gibi sorunlar, üretilen verilerin çeşitliliğini sınırlayabilmektedir. Bu tür zorlukların üstesinden gelmek için çeşitli iyileştirme yöntemleri önerilmiştir (Arjovsky, Chintala & Bottou, 2017).

BÖLÜM 5

ÖNERİLEN SALDIRI TESPİT SİSTEMİ UYGULAMASI

Nesnelerin İnterneti araçları son yıllarda yaygın bir kullanım alanı bularak yaşamın birçok alanına girmiştir. Bu araçlara yapılacak saldırılar sadece finansal kayıplara, kişisel bilgilerin çalınmasına sebep olmamakta hatta hayati tehlikelere yol açmaktadır. Bu nedenle yapılabilecek saldırıların tespit edilerek önlenmesi büyük önem arz etmektedir. Literatürde bu alanda birçok çalışma yapılmıştır. Önerilen sistemin amacı sadece birkaç veri seti eğitilerek bu veri setlerinde çok yüksek başarı oranında bir sistem geliştirmekten daha çok literatürde bu alanda sıklıkla kullanılan veri setlerinin çoğu kullanılarak daha genel olarak veri setlerinde başarı sonuçlar alabilecek bir sistem geliştirmektir. Uygulama planlaması, hazırlık süreci, uygulamada kullanılan yöntemler ve programın ayrıntıları aşağıda detaylandırılmaktadır.

5.1. Uygulama Planlaması ve Hazırlık Süreci

Bu tez çalışmasında ilk olarak Saldırı Tespit Sistemleri hakkında derinlemesine bir literatür çalışması yapılmıştır. Nesnelerin İnterneti ağlarına yapılan saldırılar derin öğrenme vb. yapay zeka uygulamalarının da yardımıyla her geçen gün büyük ölçekte güçlenerek artmaktadır. Bunun paralelinde bunlara karşı geliştirilen savunmalar da yine derin öğrenme başta olmak üzere birçok yapay zekâ vb. uygulama kullanılarak gittikçe kuvvetlenmektedir. Hem saldırı hem savunma sürekli olarak birbirini teşvik ettiği için bu alandaki çalışmalar sürekli devam etmektedir. Bu tezin hazırlanma sürecinde de bu alanda çok fazla yeni çalışma yapılmıştır. Literatür takibi bu tez süreci boyunca devam etmiştir. 2023 ve 2024 yılında yeni oluşturulan veri setleri tezin kapsamına dâhil edilmiş ve onlar üzerinde de model geliştirme ve eğitim süreçlerine devam edilmiştir.

5.2. Uygulamada Kullanılan Araç ve Yöntemler

Bu tezde yapılan çalışmalarda programlama dili olarak Python dili kullanılmaktadır. Python dili kolay okunabilirliği ve sunduğu zengin kütüphanelerle son zamanlarda birçok alanda yaygın olarak kullanılmaktadır. Uygulama geliştirme platformu olarak Google Colab Platformu kullanılmıştır. Colab Platformu ile uygulama çalışmalarına İnternet üzerinden ihtiyaç duyulan her yerden ulaşılabilmiş ve model eğitim süreçleri düzenli bir zaman kaybına uğramadan uygulanmıştır. Uygulamalarda AE, CNN, LSTM vb. derin öğrenme algoritmaları yoğun olarak kullanılmaktadır

5.3. Uygulamada Kullanılan Veri Setleri

Uygulamada, Saldırı Tespit Sistemleri'ndeki çalışmalarında sıklıkla kullanılan birçok veri setine yer verilmiştir. Kullanılan veri setleri, IOT-23, CIC-IOT-2023, UNSW-NB15, N-BAIOT, BOT-IOT, CIC-DDOS-2019, TON-IOT, CIC-IDS-2018, CIC-IDS-2017'dir. Bunlar aşağıda detaylandırılmaktadır.

5.3.1. CIC-IDS-2018 Veri Seti

Veri seti 80 özellikten oluşmaktadır. Veri seti 19.231.336 veriden oluşmaktadır. Verilerin %70'i normal verilerden %30'u saldırı verilerinden oluşmaktadır. DOS Saldırı türüne ait veriler ağırlıklıdır. CIC-IDS-2018 Veri Seti'nde yer alan saldırı türlerine ait detaylar Çizelge 5.1'de verilmektedir.

Çizelge 5.1. CIC-IDS-2018 Veri Seti Saldırı Türleri

Türü	Toplam Miktarı
Benign	13.484.708
DoS Attack-HOIC	1.686.012
DoS Attack-LOIC http	1.576.191
DoS Attack-Hulk	1.461.912
Bot	286.191
FTP-Brute Force	193.360
SSH-Brute Force	187.589
Infiltration	161.934
SlowHTTPTest	139.890
DoS Attack-GoldenEye	41.508
DoS Attack-Slowloris	9.908
DoS Attack-LOIC UDP	1.384
Brute Force Web	440
SQL Injection	82
Brute Force XSS	227
Toplam	19.231.336

5.3.2. CIC-IDS-2017 Veri Seti

Veri seti 79 özellikten oluşmaktadır. Veri seti toplam 2.830.743 veriden oluşmaktadır. Verilerin %80'ini normal veriler, %20'sini saldırı verileri oluşturmaktadır. Saldırı verilerinin ağırlığı DOS/DDOS Saldırıları oluşturmaktadır. Bu veri setinde yer alan saldırı türlerine ait detaylar Çizelge 5.2'de verilmektedir.

Çizelge 5.2. CIC-IDS-2017 Saldırı Türleri

Türü	Toplam Miktarı
Normal	2.259.087
Botnet ARES	1.966
Brute Force	13.835
Dos/DDos	294.506
Infiltration	36
Port Scan	158.930
Web Attack	2.180
Toplam	2.830.743

5.3.3. UNSW-NB15 Veri Seti

UNSW-NB15 Veri Seti, 2.539.044 veriden oluşmaktadır. Veri setinde normal verilerin ağırlığı %87, saldırı verilerinin ağırlığı %13'tür. Veri setinde 47 tane özellik vardır. Bu veri setinde yer alan saldırı türlerine ait detaylar Çizelge 5.3'te verilmektedir.

Çizelge 5.3. UNSW-NB15 Veri Seti Saldırı Türleri

Türü	Toplam Miktarı
Normal	2.218.761
Fuzzers	24.246
Analysis	2.677
Backdoors	2.329
DoS	16.353
Exploits	44.525
Generic	215.481
Reconnaissance	13.987
Shellcode	1.511
Worms	174
Toplam	2.539.044

5.3.4. CIC-IOT-2023 Veri Seti

CIC-IOT-2023 Veri Seti, 46.686.579 veriden oluşmaktadır. Verilerin %24'ü normal verilerden, %76'sı saldırı verilerinden oluşmaktadır. 33 farklı alt saldırı türünü

içererek zengin bir veri setidir. DDOS Saldırı Türü ağırlıklı olan ana saldırı türüdür. Veri setinde bulunan saldırı türlerinin detayları Çizelge 5.4'te yer almaktadır. Veri setinde 47 tane özellik vardır.

Çizelge 5.4. CIC-IOT-2023 Veri Seti Saldırı Türleri

Saldırı	Türü
DDoS-ICMP_Flood	7.200.504
DDoS-UDP_Flood	5.412.287
DDoS-TCP_Flood	4.497.667
DDoS-PSHACK_Flood	4.094.755
DDoS-SYN_Flood	4.059.190
DDoS-RSTFINFlood	4.045.285
DDoS-SynonymousIP_Flood	3.598.138
DoS-UDP_Flood	3.318.595
DoS-TCP_Flood	2.671.445
DoS-SYN_Flood	2.028.834
BenignTraffic	1.098.195
Mirai-greeth_flood	991.866
Mirai-udpplain	890.576
Mirai-greip_flood	751.682
DDoS-ICMP_Fragmentation	452.489
MITM-ArpSpoofing	307.593
DDoS-UDP_Fragmentation	286.925
DDoS-ACK_Fragmentation	285.104
DNS_Spoofing	178.911
Recon-HostDiscovery	134.378
Recon-OSScan	98.259
Recon-PortScan	82.284
DoS-HTTP_Flood	71.864
VulnerabilityScan	37.382
DDoS-HTTP_Flood	28.790
DDoS-SlowLoris	23.426
DictionaryBruteForce	13.064
BrowserHijacking	5.859
CommandInjection	5.409
SqlInjection	5.245
XSS	3.846
Backdoor_Malware	3.218
Recon-PingSweep	2.262
Uploading_Attack	1.252
Toplam	46.686.579

5.3.5. IOT-23 Veri Seti

IOT-23 Veri Seti, 325.309.946 veriden oluşmaktadır. Verilerin %9,5'i normal verilerden, %90,5'i saldırı verilerinden oluşmaktadır. Bu veri seti 15 farklı alt saldırı

türünü içermektedir. Veri setinde bulunan saldırı türlerinin detayları Çizelge 5.5'te yer almaktadır. Veri setinde 23 tane özellik vardır.

Çizelge 5.5. IOT-23 Veri Seti Saldırı Türleri

Saldırı	Türü
PartOfAHorizontalPortScan	213.852.924
Okiru	60.990.708
Normal	30.860.691
DDoS	19.538.713
C&C-HeartBeat	33.673
C&C	21.995
Attack	9.398
C&C-PartOfAHorizontalPortScan	888
C&C-HeartBeat-Attack	834
C&C-FileDownload	53
C&C-Torii	30
FileDownload	18
C&C-HeartBeat-FileDownload	11
PartOfAHorizontalPortScan-Attack	5
Okiru-Attack	3
C&C-Mirai	2
Toplam	325.309.946

5.3.6. N-BAIOT Veri Seti

N-BAIOT Veri Seti, 7.062.606 veriden oluşmaktadır. Verilerin %8'i normal verilerden, %92'si saldırı verilerinden oluşmaktadır. Bu veri seti 10 farklı alt saldırı türünü içermektedir. Veri setinde bulunan saldırı türlerinin detayları Çizelge 5.6'da yer almaktadır. Veri setinde 116 tane özellik vardır.

Çizelge 5.6. N-BAIOT Veri Seti Saldırı Türleri

Saldırı	Türü
Mirai.udp	1.229.999
Gafgyt.udp	946.366
Gafgyt.tcp	859.850
Mirai.syn	733.299
Mirai.ack	643.821
Benign	555.932
Mirai.scan	537.979
Mirai.udpplain	523.304
Gafgyt.combo	515.156
Gafgyt.junk	261.789
Gafgyt.scan	255.111
Toplam	7.062.606

5.3.7. TON-IOT Veri Seti

TON-IOT Veri Seti, 461.043 veriden oluşmaktadır. Verilerin %65'i normal verilerden, %35'i saldırı verilerinden oluşmaktadır. Bu veri seti 9 farklı alt saldırı türünü içermektedir. Veri setinde bulunan saldırı türlerinin detayları Çizelge 5.7'de yer almaktadır. Veri setinde 45 tane özellik vardır.

Çizelge 5.7. TON-IOT Veri Seti Saldırı Türleri

Saldırı	Türü
Normal	300.000
Scanning	20.000
Dos	20.000
Injection	20.000
Ddos	20.000
Password	20.000
Xss	20.000
Ransomware	20.000
Backdoor	20.000
Mitm	1.043
Toplam	461.043

5.3.8. BOT-IOT Veri Seti

BOT-IOT Veri Seti, 73.370.443 veriden oluşmaktadır. Verilerin %99.99'u saldırı verilerden, %0,01'i normal verilerinden oluşmaktadır. Bu veri seti 5 farklı alt saldırı türünü içermektedir. Veri setinde bulunan saldırı türlerinin detayları Çizelge 5.8'de yer almaktadır. Veri setinde 35 tane özellik vardır.

Çizelge 5.8. BOT-IOT Veri Seti Saldırı Türleri

Saldırı	Türü
DDoS	38.532.480
DoS	33.005.194
Reconnaissance	1.821.639
Normal	9.543
Theft	1.587
Toplam	73.370.443

5.3.9. CIC-DDOS-2019 Veri Seti

CIC-DDOS-2019 Veri Seti, 50.063.112 veriden oluşmaktadır. Verilerin %0,1'i normal verilerden, %99,9'si saldırı verilerinden oluşmaktadır. Bu veri seti 12 farklı alt

saldırı türünü içermektedir. Veri setinde bulunan saldırı türlerinin detayları Çizelge 5.9'da yer almaktadır. Veri setinde 88 tane özellik vardır.

Çizelge 5.9. CIC-DDOS-2019 Veri Seti Saldırı Türleri

Saldırı Türü	Toplam
TFTP	20.082.580
DrDoS_SNMP	5.159.870
DrDoS_DNS	5.071.011
DrDoS_MSSQL	4.522.492
DrDoS_NetBIOS	4.093.279
DrDoS_UDP	3.134.645
DrDoS_SSDP	2.610.611
DrDoS_LDAP	2.179.930
Syn	1.582.289
DrDoS_NTP	1.202.642
UDP-lag	366.461
BENIGN	56.863
WebDDoS	439
Toplam	50.063.112

5.4. Uygulama Detayları

Tezdeki uygulama Python dilinde, TensorFlow kütüphanesi kullanılarak yazılmıştır. Uygulamanın detayları aşağıda verilecektir.

5.4.1. Ölçeklendirme Türleri

Min-Max Normalizasyonu, Z-Skoru Normalizasyonu, Logaritmik Dönüşüm, L1 ve L2 Normalizasyonu, Dayanaklı Ölçeklendirme (Robust Scaling) gibi ölçeklendirme türleri vardır. Aşağıda ölçeklendirme türleri detaylandırılmaktadır (Kantardzic, M. 2011).

Min-Max Normalizasyonu

Min-Max Normalizasyonu, verileri belirli bir aralığa (genellikle [0, 1] veya [-1, 1]) ölçeklemek için kullanılan bir yöntemdir. Bu yöntem, tüm değerlerin aynı ölçeğe getirilmesini sağlayarak öğrenme algoritmalarının performansını artırmaktadır. Özellikle derin öğrenme gibi sinir ağları kullanan modellerde, aktivasyon fonksiyonlarının daha stabil çalışmasını sağlamaktadır. Bu dönüşüm, aşağıdaki şekilde formülüze edilebilmektedir.

$$X' = \frac{X - X_{Min}}{X_{Max} - X_{Min}} \quad (5.1)$$

X : $Normalizasyon$ yapılacak orijinal veri değeri

X' : $Normalizasyon\ sonrası$ ölçeklenmiş değer

Özelliklerin ölçek farklarından kaynaklanan dengesizliği ortadan kaldırmaktadır. Özellikle gradyan inişi gibi optimizasyon algoritmalarında daha hızlı yakınsamayı desteklemektedir. Hesaplama açısından çok basittir. Bu avantajları yanında bazı dezavantajları da vardır. Aykırı (outlier) değerler X_Min ve X_Max değerlerini ciddi şekilde etkileyebilmektedir. Min-Max normalizasyonu, yalnızca sabit bir veri dağılımı için etkili çalışmaktadır. Yeni veri eklendiğinde yeniden hesaplama gerekmektedir (Haykin, S. 2008).

Z-Skoru Normalizasyonu

Z-Skoru (Z-Score) Normalizasyonu, bir veri kümesindeki değerleri standart bir ölçeğe (ortalama 0 ve standart sapma 1 olacak şekilde) dönüştürmek için kullanılan yaygın bir veri ön işleme yöntemidir. Bu yöntem, özellikle farklı ölçeklere sahip özelliklerin yer aldığı büyük veri setlerinde faydalı olmaktadır. Z-Skoru, bir veri noktasının o veri kümesindeki dağılıma göre ne kadar uzak olduğunu ölçmektedir. Bu yöntemle, veriler ortalama sıfır ve standart sapma bir olacak şekilde yeniden ölçeklenmektedir. Bu dönüşüm, aşağıdaki şekilde formülüze edilebilmektedir (Cabello-Solorzano, Ortigosa de Araujo, Peña, Correia, & Tallón-Ballesteros, 2023)

$$Z = \frac{\mathrm{X} - \mu}{\sigma} \quad (5.2)$$

Z : Z Skoru

X : Standartlaştırılacak Orijinal Değer

μ : Veri sütununun ortalama değeri

σ : Veri Sütununun standart sapması

Logaritmik Dönüşüm

Logaritmik dönüşüm, verinin her bir değerine logaritma işlemi uygulanarak daha küçük bir ölçeğe getirilmesidir. Bu dönüşüm, aşağıdaki şekilde formülüze edilebilmektedir (Feng, Wang, Lu, Chen, He, Lu & Tu, 2014).

X'= log(X) (5.3)

X : Normalize edilecek orijinal değer

X': Normalize edilmiş değer

n : Özellik sayısı

L1 ve L2 Normalizasyonu

L1 normalizasyonu, bir veri noktasının bileşenlerini toplam mutlak değerleri (L1 normu) kullanarak yeniden ölçeklenmektedir. Bu yöntem, her bir özelliği [0,1] veya [-1,1] aralığında olacak şekilde normalize etmektedir. Bu dönüşüm, aşağıdaki şekilde formülüze edilebilmektedir.

$$X_i' = \frac{X_i}{\sum_{j=1}^{n} |X_j|} \quad (5.4)$$

X : Normalize edilecek orijinal değer

X': Normalize edilmiş değer

n : Özellik sayısı

Seyrek (Sparse) özelliği yoğun şekilde taşıyan veri setlerinde kullanılabilmektedir.

L2 normalizasyonu, bir veri noktasının bileşenlerini toplam kareler toplamının karekökü̈ne bölerek ölçeklemektedir. Bu yöntem, özellikle büyük değerlere sahip özelliklerin etkisini azaltmaktadır. Aykırı değerlere daha dayanıklıdır. Bu dönüşüm, aşağıdaki şekilde formülüze edilebilmektedir (Cortes, Mohri & Rostamizadeh, 2012).

$$X_i' = \frac{X_i}{\sqrt{\sum_{j=1}^{n} X_j^2}} \quad (5.5)$$

X : Normalize edilecek orijinal değer

X': Normalize edilmiş değer

n : özellik sayısı

Özellikler, birim uzunluğa sahip olacak şekilde yeniden ölçeklenmektedir. Bu yöntem, sürekli ve yoğun (dense) özelliği fazla miktarda taşıyan veri setleri için uygundur. Aykırı değerlerden daha fazla etkilenmektedir.

Dayanıklı Ölçeklendirme

Dayanıklı Ölçeklendirme (Robust Scaling), verilerin ölçeklendirilmesi sırasında aykırı değerlerin etkisini en aza indirmek amacıyla kullanılmaktadır. Medyan ve IQR (Interquartile Range - Çeyrekler Arası açıklık) gibi istatistiksel değerleri kullanmaktadır. Verinin her bir özelliğinden Medyan değer çıkartılarak veriler merkeze yaklaştırılmakta ve sonrasında IQR değerine bölerek ölçeklenmektedir. IQR, veri setinin %25 ile %75 arasındaki değer aralığını ifade etmektedir. IQR kullanılarak verideki aykırı değerlerin etkisi minimize edilmiş olmaktadır. Bu yöntem aykırı verilerin yoğun olduğu veri setlerinde başarılıdır. Bu yöntemin veri setinin fazla aykırı değerlere sahip olmadığı durumlarda fazla bir avantajı yoktur. Büyük veri setlerinde medyan ve IQR değerlerini hesaplamak zaman alabilmektedir (Al-jabery, Obafemi-Ajayi, Olbricht & Wunsch II, 2020a).

5.4.2. Aktivasyon Fonksiyonları

Makine Öğrenmesi ve Derin Öğrenme uygulamalarında Softmax, Sigmoid, ReLU, Tanh gibi aktivasyon fonksiyonları kullanılmaktadır. Bu fonksiyonlar aşağıda detaylandırılmaktadır.

Sigmoid Fonksiyonu

Sigmoid fonksiyonu, S şeklinde bir eğriye sahip, doğrusal olmayan bir fonksiyondur. Bu fonksiyon, herhangi bir gerçek sayıyı 0 ile 1 arasında bir değere dönüştürmektedir. Bu özelliği sayesinde, sinir ağlarında olasılık değerlerini temsil etmek için sıkça kullanılmaktadır. Şekil 5.1.'de Sigmoid Aktivasyon fonksiyonuna verilen girdiler ve çıktılar grafikte görselleştirilmektedir.

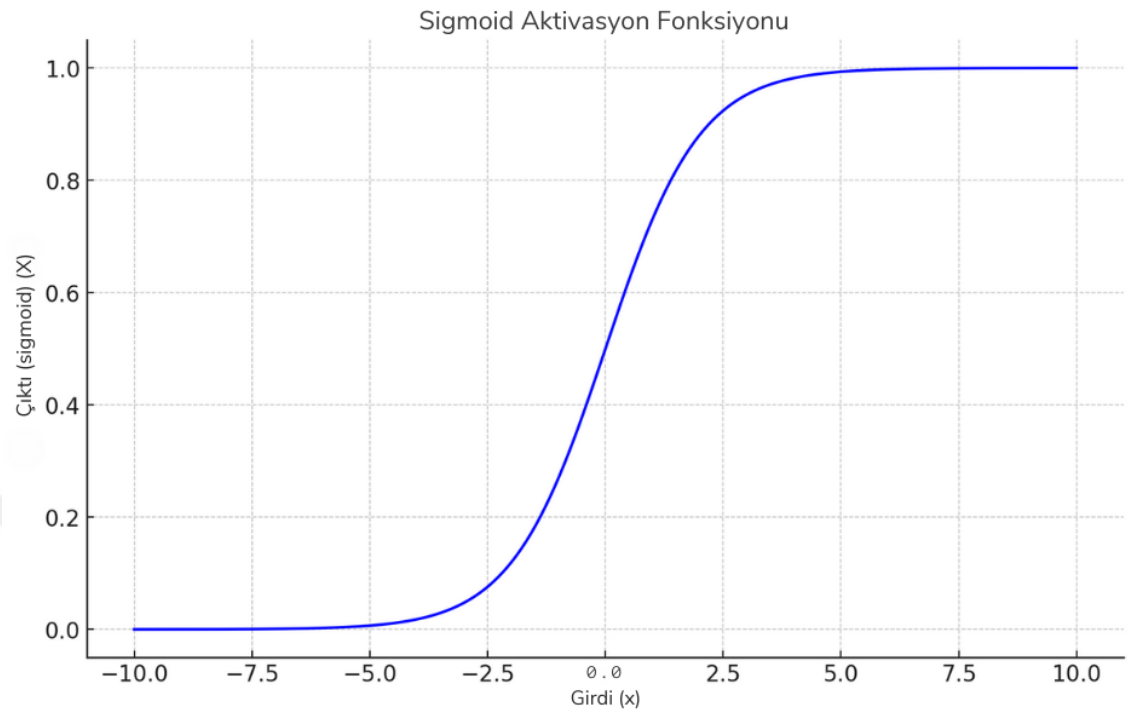
Matematiksel Formülü ve Grafiği:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{5.6}$$

x:Fonksiyona giren giriş değeri (gerçek bir sayı)

e:Doğal logaritmanın tabanı ($\approx 2{,}718$)

Sigmoid fonksiyonu grafiği Şekil 5.1'de gösterilmektedir.



Şekil 5.1. Sigmoid Aktivasyon Fonksiyonu

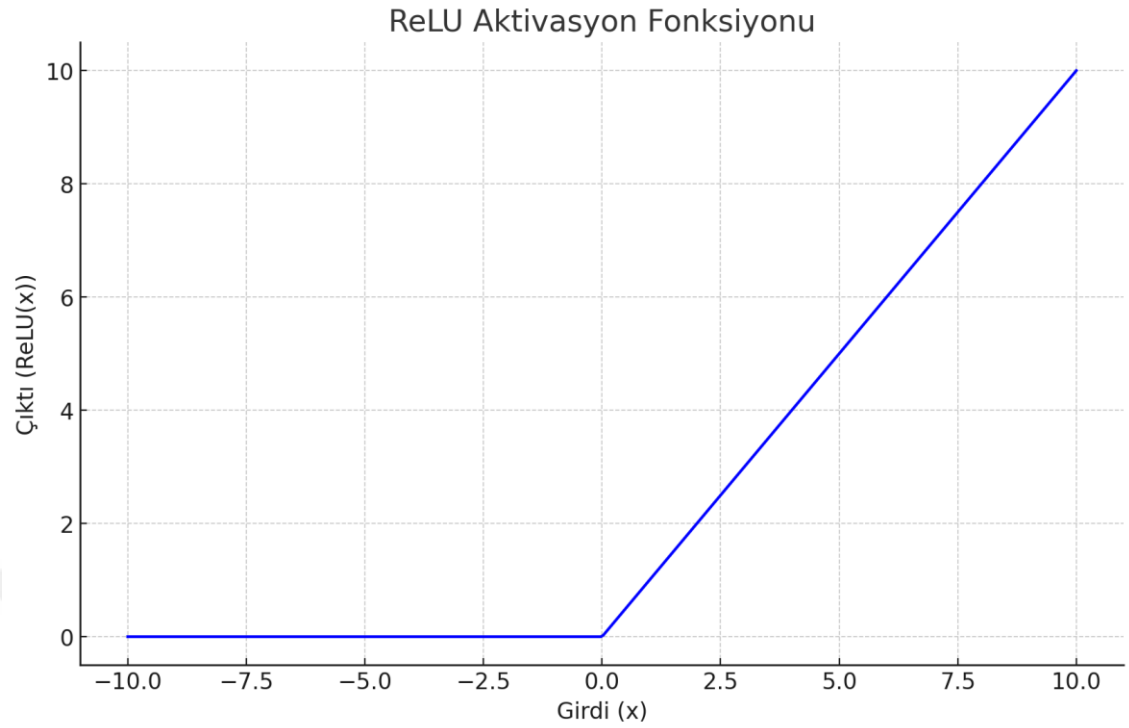
ReLU Fonksiyonu

ReLU (Rectified Linear Unit), sinir ağlarında en sık kullanılan aktivasyon fonksiyonlarından biridir. İsmi, negatif değerleri "düzeltmesi" (rectify) ve doğrusal (linear) bir çıktı üretmesinden gelmektedir. ReLU, bir giriş değerini alır ve bu değer sıfırdan büyükse aynı değeri, sıfırdan küçükse sıfırı döndürmektedir. Şekil 5.2.'de ReLU Aktivasyon fonksiyonuna verilen girdiler ve çıktılar grafikte görselleştirilmektedir.

Matematiksel Formülü ve Grafiği:

- $(x) = \max(0, x)$ (5.7)
- Eğer $x > 0$ ise $f(x) = \mathrm{x}$
- Eğer $x \leq 0$ ise $f(x) = 0$

ReLU'nun fonksiyonu Şekil 5.2'de gösterilmektedir.



Şekil 5.2. ReLU Aktivasyon Fonksiyonu

Tanh Fonksiyonu

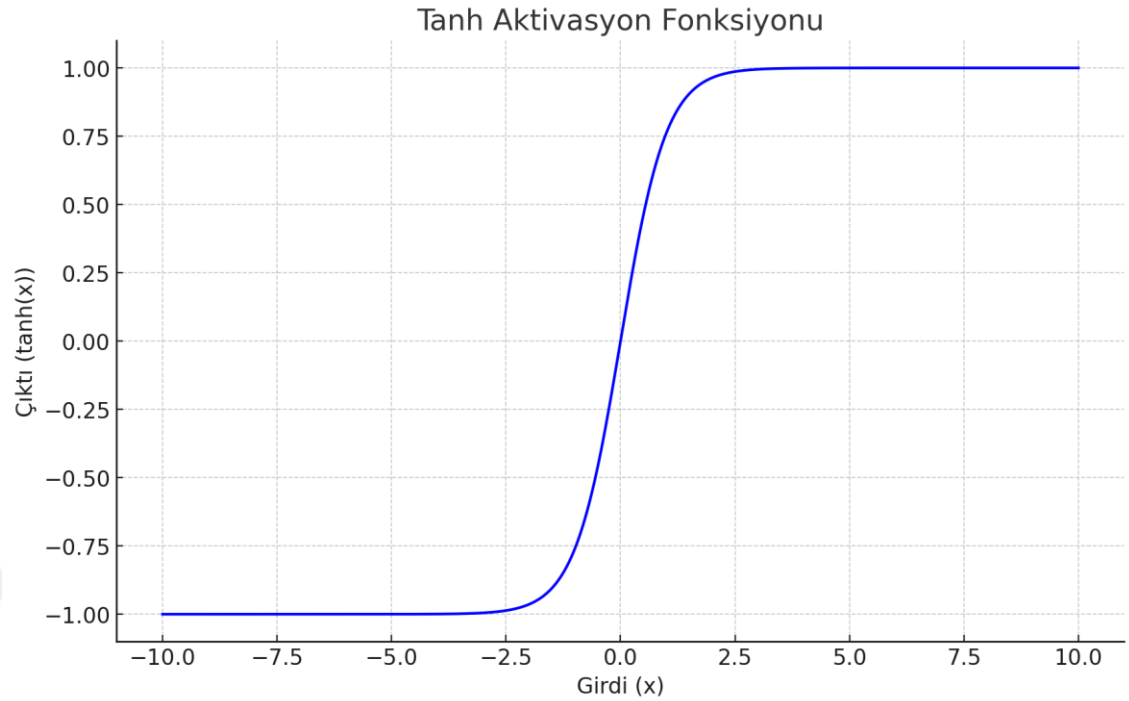
Tanh (Hiperbolik Tanjant) fonksiyonu, bir giriş değerini -1 ile 1 arasında bir değere dönüştüren, S şeklindeki bir aktivasyon fonksiyonudur. Şekil 5.3'te Tanh Aktivasyon fonksiyonuna verilen girdiler ve çıktılar grafikte görselleştirilmektedir.

Matematiksel Formülü ve Grafiği:

$$\tan h(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{5.8}$$

Bu formül, hiperbolik sinüs ($\sin h(x)$) ve hiperbolik kosinüs ($\cos h(x)$) fonksiyonları kullanılarak da ifade edilebilmektedir:

$$\tan h(x) = \frac{\sin h(x)}{\cos h(x)}$$



Şekil 5.3. Tanh Aktivasyon Fonksiyonu

Tanh ve Sigmoid fonksiyonun karşılaştırması Çizelge 5.10'da yapılmaktadır.

Çizelge 5.10. Tanh ve Sigmoid Fonksiyonu Karşılaştırması

Özellik	**Sigmoid**	**Tanh**
Çıktı Aralığı	0 ile 1 arasında	-1 ile 1 arasında
Ortalama Çıktı	Sıfırdan farklı	Sıfıra yakın
Kaybolan Gradyan Problemi	Evet	Daha az

5.4.3. Optimizasyon Algoritmaları

Makine Öğrenmesi ve Derin Öğrenme uygulamalarında ADAM, SGD ve RMSprop gibi optimizasyon algoritmaları kullanılmaktadır. Bu algoritmalar aşağıda detaylandırılmaktadır.

ADAM Optimizasyon Algoritması

Bu algoritma, her bir parametre için adaptif bir öğrenme oranı belirleyerek, geleneksel stokastik gradyan iniş algoritmasına göre daha hızlı ve daha iyi sonuçlar vermektedir. Temel olarak, aşağıdaki iki moment değeri hesaplanmaktadır.

- Birinci Moment (Bias-corrected Momentum): Gradyanın hareketli ortalamasıdır ve parametrelerin genel yönünü belirlemektedir.
- İkinci Moment (Bias-corrected Variance): Gradyanın karelerinin hareketli ortalamasıdır ve parametrelerin değişim hızını belirlemektedir.

Bu iki moment değeri, her bir parametre için ayrı ayrı hesaplanır ve öğrenme oranını belirlemek için kullanılmaktadır. Böylece, sık sık güncellenen parametreler için daha küçük, nadiren güncellenen parametreler için ise daha büyük öğrenme oranları kullanılmaktadır (Manjula & Vijaya, 2020).

Adam algoritması, genellikle diğer optimizasyon algoritmalarına göre daha hızlı yakınsamaktadır. Adaptif öğrenme oranı sayesinde, modelin eğitimi sırasında daha kararlı bir ilerleme sağlamaktadır. Adam algoritması, genellikle varsayılan hiper parametreleriyle iyi çalışmakta ve fazla ayarlama gerektirmemektedir. Birçok farklı derin öğrenme modeli ve problem için etkilidir (Kingma & Ba, 2014).

SGD Optimizasyon Algoritması

SGD (Olasılıksal Gradyan İnişi - Stochastic Gradient Descent), sinir ağlarını ve makine öğrenimi modellerini eğitmek için yaygın olarak kullanılan bir optimizasyon algoritmasıdır. Temel amacı, bir kayıp fonksiyonunu (örneğin, doğruluk ya da hata) minimize etmek için modelin parametrelerini (ağırlıklar ve sapmalar) güncellemektir (Robbins & Monro, 1951).

SGD, modelin parametrelerini, kayıp fonksiyonunun negatif gradyanını takip ederek küçük adımlarla güncellemektedir (Robbins & Monro, 1951). Standart Gradyan İnişi (Gradient Descent) algoritmasından farklı olarak, her adımdaki güncellemeyi tüm veri kümesi yerine veri kümesinden rastgele seçilmiş bir örnekle veya küçük bir örnek grubu (mini-batch) ile yapmaktadır. Bu özellik, hesaplamaları hızlandırır ve daha büyük veri kümeleri üzerinde çalışmayı mümkün kılmaktadır (Ruder, 2016).

Matematiksel Formül

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta J(\theta; x^{(i)}, y^{(i)}) \quad (5.9)$$

- θ_t: Güncellenen parametre vektörüdür (örneğin ağırlıklar).
- η: Öğrenme oranıdır, modelin adım büyüklüğünü belirlemektedir.

- $\nabla_\theta J(\theta; x^{(i)}, y^{(i)})$: Parametrelerin mevcut durumundaki kayıp fonksiyonunun gradyanıdır.
- θ_{t+1}: Sonraki adımda parametrenin güncellenmiş değeri

SGD, her iterasyonda tüm veri kümesi yerine sadece bir örneği $(x^{(i)}, y^{(i)})$ kullanarak gradyanı hesaplamaktadır. Bu, daha hızlı iterasyon sağlar ancak bazı durumlarda gürültü nedeniyle doğruluktan sapmalara neden olabilmektedir. SGD'nin rastgele örneklerle çalışması, onu yerel minimumdan çıkarabilmekte, böylece genel minimuma ulaşma şansını artırmaktadır. Genellikle, SGD'nin performansı artırmak için mini-batch adı verilen küçük veri grupları kullanılmaktadır. Bu yönteme Küçük Grup Gradyan İnişi (Mini-Batch Gradient Descent) denmektedir (Khirirat, Feyzmahdavian & Johansson, 2017). SGD, büyük veri kümeleriyle çalışırken daha hızlı sonuç verir, çünkü her iterasyonda tüm veri kümesini işlememektedir. Veri örneklerinin rastgele kullanılması modelin aşırı uyum yapma olasılığını azaltmaktadır. Rastgele örnekleme, modelin yerel minimumlardan kurtulmasına yardımcı olabilmektedir. Her örnekteki sapma nedeniyle parametre güncellemeleri gürültülüdür, bu yüzden her adımda tam olarak doğru yöne gitmemektedir. Yanlış bir öğrenme oranı seçilirse, model ya çok yavaş öğrenmekte ya da optimuma ulaşamadan sapmaktadır (Khirirat, Feyzmahdavian & Johansson, 2017).

RMSprop Optimizasyon Algoritması

RMSprop (Kök Ortalama Kare Yayılımı-Root Mean Square Propagation) algoritması, SGD algoritmasındaki öğrenme oranı problemine bir çözüm olarak geliştirilmiştir (Wilson, Roelofs, Stern, Srebro & Recht, 2017). Rastgele bir öğrenme oranı kullanmak yerine, RMSprop algoritması her bir parametreye özgü farklı öğrenme oranları kullanır ve bu oranları dinamik olarak gradyanların karesinin üstel hareketli ortalamasına göre güncellemektedir. Bu yaklaşım, önceki gradyanları tamamen unutmayarak, sürekli güncellenen ve parametrelere göre uyarlanabilen bir öğrenme oranı sunmaktadır. Ayrıca algoritma, sıfıra bölünme sorunlarını önlemek için küçük bir epsilon değeri kullanmaktadır.

Bu algoritma, kayıp fonksiyonun çok değiştiği durumlarda çok kullanışlıdır. Büyük verileri, derin sinir ağlarında işlerken öğrenme sürecini hızlandırabilmektedir. Kullanılan parametrelerin ayarlanmasın yapılabilecek hatalar algoritmanın daha düşük

performansta çalışmasına sebep olabilmektedir. Dinamik olarak öğrenme oranının çok düşük değerlere ulaşmasına neden olup, modelin eğitilme sürecini uzatabilmektedir (Ruder, 2016).

Özellikle derin ve karmaşık sinir ağlarında, RMSprop sıklıkla tercih edilmektedir. Zaman serisi verileri üzerinde çalışan RNN'lerde, RMSprop iyi sonuçlar vermektedir. Görüntü tanıma gibi görevlerde kullanılan CNN'lerde de RMSprop kullanılabilmektedir (Ruder, 2016).

Matematiksel Formül

Gradyan Hesaplama: $g_t = \nabla_\theta L(\theta_t)$ (5.10)

g_t: t. iterasyondaki gradyan vektörü

∇_θ: Parametrelerin (θ) türevini göstermektedir.

$L(\theta_t)$: t. iterasyonda kayıp fonksiyonu

Gradyanların Karesinin Hareketli Ortalaması:

$E[g^2]_t = \beta \cdot E[g^2]_{t-1} + (1-\beta) \cdot g_t^2$ (5.11)

$E[g^2]_t$: t. iterasyonda gradyanların karesinin üstel hareketli ortalaması

β: Momentum katsayısı (örneğin 0,9)

$E[g^2]_{t-1}$: Bir önceki iterasyonda hesaplanan üstel hareketli ortalama

g_t^2: Gradyanların her bir bileşeninin karesi

Parametre Güncelleme: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot g_t$ (5.12)

θ_{t+1}: Güncellenmiş parametre değeri

η: Öğrenme oranı

ϵ: Çok küçük bir sabit

5.4.4. Veri Ön İşleme ve Model Oluşturma

Verilerin işleneceği Colab Platformuna veri seti sıkıştırılmış olarak indirilmiştir. Daha sonra sıkıştırılmış dosyalar açılmıştır. Veri seti kodda işlemek için bir Veri Çerçevesi'ne (Dataframe) kaydedilmiştir.

'df.sample(frac=1).reset_index(drop=True)'

Yukarıdaki kodda, '(frac=1)' parametresi ile veri setinin tamamını kapsayacak bir örnek alınmıştır. Örneğin, '(frac=0.5)' yazılırsa veri setinin yarsından bir örnek oluşturulmuştur. 'Reset_index (drop=True)' metodu ile örneklem sonrasında oluşan yeni Veri Çerçevesi'nin indeksi sıfırlanmıştır. Bu sayede, satırlar karıştırılmış olsa bile, yeni, baştan başlayan, sıralı bir indeks elde edilmiştir. Bu iki işlemin birleşimiyle Veri Çerçevesi'nin satırlarının yeri rastgele değiştirilerek yeni bir indeksli Veri Çerçevesi oluşturulmuştur. Veri analizlerinde, özellikle makine öğrenmesi modellerini eğitirken verinin rastgele karıştırılması önemlidir. Böylelikle model, veri setindeki herhangi bir sıralamaya özel öğrenme eğilimi göstermemektedir. (Zhang, Bengio, Hardt, Recht & Vinyals, 2016).

Daha sonra veri setindeki özellikleri barındıran kolonlar incelenmiştir. Bu tezde kullanılan veri setleri her bir verinin hangi sınıfa ait olduğu bilgileri tutan özelliklere sahiplerdir. Örneğin kullanılan UNSW-NB15 veri seti 'Label' ve 'Attack_Cat' olarak isimlendirilen iki özelliğe sahiptir. Bu özellikler veri setinin aktarıldığı Veri Çerçevesi'nde aynı isimde iki kolon olarak aktarılmaktadırlar. 'Label' kolonunda, veri saldırı verisi ise 1 değilse 0 değerini tutmaktadır. 'Attack_Cat' kolonu, DoS, Brute_Force vb. 18 farklı saldırı türü içermektedir. Veri hangi saldırı türüne ait ise bu kolonda o saldırı ismi yer almaktadır (Lashkari, A. H. 2018).

'X = df.drop('Label', axis=1)'

'y = df['Label']'

Yukarıdaki koddaki 'drop' fonksiyonu kullanılarak sınıflandırma türlerini barındıran 'Label' kolonu Veri Çerçevesi'nden çıkarılarak 'y' adındaki ayrı bir listeye aktarılmıştır. Böylelikle veri setine ait özelliklerin tümü bir listede, sınıflandırma türüne ait 'Label' özelliğini barındıran kolondaki veriler de ayrı bir listede tutulmaktadır. Model eğitimi sırasında 'y' bağımlı değişken (hedef) olarak kullanılacaktır. Bu ayrımın

yapılmasının nedeni, Makine öğrenmesi modellerinin, özellikleri ‘X’ kullanarak hedef değişkeni ‘y’ tahmin etmek üzere eğitmesidir. Bu şekilde, model hangi verileri kullanarak eğitim yapması gerektiğini öğrenmektedir. Eğitilen modelin performansını değerlendirmek için, model görmediği yeni veriler üzerinde test edilmektedir. Bu test verisinde de aynı şekilde ‘X’ ve ‘y’ olarak ayırma yapılmasını gerektirmektedir (Wu, 2013).

‘rows_all_zeros = X[(X == 0).all(axis=1)]’

Yukarıdaki kod kullanılarak tüm özellikleri 0 olan veriler bulunmuştur. Bu tür veriler genellikle eksik verileri temsi edeceğinden modelin eğitilmesi sırasında bir fayda sağlamamaktadır. Bu nedenle eğitime sokulacak verilerden çıkartılabilmektedirler (Zhou, X.-H., Zhou, C., Lui, D., & Ding, X. 2014).

‘print((X == float('inf')).sum()) ve print((X == float('-inf')).sum())’

Yukarıdaki kod kullanılarak, her bir sütunda kaç tane sonsuz değer olduğu bulunmuştur. Sonsuz değerler genellikle hesaplama hataları veya veri girişindeki problemlerden kaynaklanabilmektedir. Bu veriler, modelin eğitilmesinde sorun yaratabileceği için çeşitli tekniklerle tamamlanmıştır ve Veri Çerçevesi’nden çıkartılmıştır (Zhou, X.-H., Zhou, C., Lui, D., & Ding, X. 2014).

‘print(X.isna().sum())’

Yukarıdaki kod ile Veri Çerçevesi'ndeki her bir hücreyi kontrol edilmiştir ve eksik değer (NaN) olup olmadığını belirlenmiştir. Eksik değerler varsa ‘True’, yoksa ‘False’ dönmektedir. Veri Çerçevesi'nde True değerlerinin sayısı sütun bazında toplanmıştır. Sonuç olarak her bir sütunda kaç tane eksik değer olduğu görülmüştür. Bu kod ile Veri Çerçevesi'nde kaç tane eksik değer olduğu sayılmaktadır. Eksik değerler, veri analizinde dikkat edilmesi gereken önemli bir konudur ve genellikle doldurulması veya çıkarılması gereken değerlerdir. Bu kodlar, veri setindeki potansiyel sorunları tespit etmeye yardımcı olmaktadır. Sıfır değerli satırlar, sonsuz değerler ve eksik değerler, model eğitimi ve sonuçların doğruluğunu etkileyebilmektedir. Bu işlemler, veri temizleme sürecinin ilk adımını oluşturmaktadır. Tespit edilen sorunlu veriler, daha sonraki analizler veya model eğitimi öncesinde düzeltilmelidir. Yapılan bu işlemler, veri setinde olabilecek daha önemli sorunları ortaya çıkarmaya yardımcı olabilmektedir (Zhou, X.-H., Zhou, C., Lui, D., & Ding, X. 2014).

‘kf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)’

Yukarıdaki kod ile 10 katlı çapraz doğrulama ile model eğitilmiştir. Veri seti eşit büyüklükte 10 parçaya bölünmüştür. Her döngüde bir parça test verisi olarak kullanılmaktadır ve diğer 9 parça eğitim verisi olarak kullanılmaktadır. Bu işlem 10 defa tekrarlanarak verilerin tamamının hem eğitim veri setinde hem de test veri setinde kullanılması sağlanmaktadır. Özellik çıkarımı yöntemi olarak korelasyon tabanlı Spearman ve Kandall Tau algoritmalarıyla, makine öğrenmesi tabanlı Rastgele Orman algoritması hibrit olarak uygulanmıştır.

‘scaler = StandardScaler()’

‘X_train = scaler.fit_transform(X_train)’

‘X_test = scaler.transform(X_test)’

Yukarıdaki kodlar ile öncelikle bir ‘StandardScaler’ nesnesi oluşturulmuştur. Örneğin, tüm değerleri sıfır olan satırlar, veri toplama sürecinde bir hata olduğunu gösterebilmektedir. Eğitim verisi üzerinde dönüşüm ve ölçekleme işlemleri bir arada yapılmaktadır. Bu sayede eğitim verileri, ortalaması 0 ve standart sapması 1 olan yeni bir veri kümesine dönüştürülmüştür. Daha sonra test verisi, eğitim verisinde hesaplanan istatistiksel ölçülere göre düzenlenmiştir. Bu sayede, eğitim ve test verileri aynı ölçekte ayarlanmıştır. Özetle, standartlaştırma, makine öğrenmesi modellerinde önemli bir ön işleme adımıdır. Bu işlem, verilerin farklı ölçeklerde olmasından kaynaklanan sorunları çözerek, modelin performansını artırmaktadır.

Uygulamada ölçeklendirme işlemlerinden sonra kategorik verilerin dönüşüm işlemlerine geçilmiştir.

‘label_encoder = LabelEncoder()’

‘y_train_encoded = label_encoder.fit_transform(y_train)’

‘y_test_encoded = label_encoder.transform(y_test)’

Yukarıdaki kodda LabelEncoder Kütüphanesi kullanılarak, metin veri türü özelliği taşıyan kategorik veriler, sayısal verilere dönüştürülmüştür. İlgili özellikteki benzersiz isimler için sayısal veri ataması yapılmıştır. Dönüştürme işlemleri hem eğitim verilerine hem de test verilerine uygulanmaktadır (Shah, Xue & Aamodt, 2022).

Çoğu makine öğrenmesi ve derin öğrenme algoritması, sayısal verilerle çalışmak üzere tasarlanmıştır. Bu nedenle, kategorik verileri sayısal bir formata dönüştürmek gerekmektedir. Sayısal veriler üzerinde yapılan hesaplamalar, metinsel verilere göre daha hızlı ve verimlidir. Doğru bir şekilde yapıldığında, sayısallaştırma modelin performansını artırabilmektedir.

Daha sonra bu sayısal etiketler tek-sıcak kodlama (one-hot encoding) formatına dönüştürülmüştür. Tek-sıcak kodlama ile her etiket ikili vektöre (binary vector) dönüştürülmüştür. Örneğin, 3 sınıf varsa: sınıf 0 için [1, 0, 0], sınıf 1 için [0, 1, 0], sınıf 2 için [0, 0, 1] şeklinde bir vektör elde edilmiştir. Bu format, özellikle softmax kullanan çok sınıflı sınıflandırma modellerinde gereklidir, çünkü softmax her sınıfın olasılığını bir vektör olarak döndürmektedir ve bunu tek-sıcak kodlama ile elde edilen vektörle karşılaştırarak doğru tahmini yapmaktadır. Bu dönüşümlerden sonra model oluşturulma aşamasına geçilmiştir (Al-jabery vd., 2020a).

‘model = Sequential()’

Yukarıdaki kod ile bu model, katmanların sıralı bir şekilde eklenmesine izin vermektedir. Bu sıralı yapı, birçok sinir ağı mimarisinde kullanılmaktadır Sıralı (Sequential) modelde, yeni katmanlar ‘add’ metodu kullanılarak eklenmektedir. Her eklenen katman, önceki katmanın çıkışını almakta ve kendi işlemlerini uygulamaktadır. Veri, ağa girdi olarak verildiğinde, sırayla her bir katmandan geçmektedir. Her katman, veriyi dönüştürmekte ve bir sonraki katmana aktarmaktadır. Son katman, genellikle modelin tahminini yapmaktadır. Örneğin, bir sınıflandırma probleminde bu katman softmax aktivasyon fonksiyonu ile birlikte kullanılarak her bir sınıf için bir olasılık değeri üretmektedir (Manjula & Vijaya, 2020).

Birçok sinir ağı mimarisi, Sequential model kullanılarak kolayca oluşturulabilmektedir. Keras Kütüphanesi, farklı katman türlerini (Dense, Conv2D, LSTM vb.) desteklemektedir. Bu sayede çeşitli problemler için model oluşturulması kolaylıkla yapılabilmektedir (Manjula & Vijaya, 2020).

Modele ilk olarak 1D Konvolüsyonel Katman (Convolution1D) eklenmiştir. Bu katman, zaman serisi veya sıralı veriler üzerinde özellik çıkarmak için kullanılmaktadır. Bu katman eklenirken çeşitli parametreler kullanılmaktadır. ‘Filters’ parametresi uygulanacak filtre (nöron) sayısını belirtmektedir. Her bir filtre, giriş verisinden farklı bir

özellik çıkarmaktadır. Böylelikle model, filtre sayısı kadar farklı özellik öğrenebilmektedir. ‘Kernel_size’ parametresi, konvolüsyonel katmanda kullanılan çekirdek boyutunu belirtmektedir. Örnek olarak 6 değeri verilirse, her bir filtre 6 zaman adımını veya verinin 6 ardışık değerini işlemekte kullanılmıştır. Çekirdek boyutu, modelin ne kadar geçmiş veriyi göz önünde bulundurarak özellik çıkaracağını belirlemektedir. ‘Padding="same"’ şeklinde bu parametre kullanılarak, çıktı boyutunun giriş boyutuyla aynı kalması sağlanmıştır. Bu parametre, girişe kenarlarında sıfır ekleyerek işlemi gerçekleştirmektedir. Böylece çıkış boyutu giriş boyutuna eşit olmaktadır. ‘Activation="relu"’ ifadesi, katmanın aktivasyon fonksiyonunu belirtmektedir. ReLU (Rectified Linear Unit - Doğrultulmuş Doğrusal Birim), yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Negatif değerleri sıfıra çevirir ve pozitif değerleri olduğu gibi bırakmaktadır. Bu sayede modelin daha hızlı öğrenmesine yardımcı olmaktadır. ‘input_shape=(80, 1)’ ifadesi, ilk katmana girecek verinin boyutunu tanımlamaktadır (Kiranyaz, Avci, Abdeljaber, Ince, Gabbouj & Inman, 2021).

‘model.add(MaxPooling1D(pool_size=2))’

Yukarıdaki kod ile MaxPooling1D (Havuzlama) katmanı eklenmiştir. Bu katman, bir girdi dizisindeki belirli bir pencere boyutunda maksimum değeri alarak boyut indirgeme işlemi gerçekleştiren bir katmandır. Bu işlem, Havuzlama (Pooling) olarak da adlandırılmaktadır. MaxPooling1D katmanına, genellikle bir Convolution1D katmanından gelen bir özellik haritası (feature map) verilmektedir. Bu özellik haritası, bir dizi sayıdan oluşmaktadır (Ait Skourt, El Hassani & Majda, 2022).

‘Pool_size’ parametresi ile belirlenen bir pencere, girdi dizisi üzerinde uygulanır kaydırılmaktadır. Her bir pencere için en büyük değer seçilmektedir. Seçilen maksimum değerler, yeni bir özellik haritası oluşturmak için kullanılmaktadır. Bu yeni özellik haritası, girdi haritasına göre daha küçük boyutludur. Büyük boyutlu veriler, modelin karmaşıklığını artırabilmektedir ve hesaplama maliyetini yükseltebilmektedir. Havuzlama, verinin boyutunu küçülterek bu sorunu çözmektedir. Havuzlama sırasında, maksimum değerleri alınarak, verideki en önemli özelliklerin korunmasını sağlanmaktadır. Bu, modelin daha iyi genelleme yapabilmesine yardımcı olmaktadır. Bu uygulama verinin, küçük kaymalara veya deformasyonlara karşı daha dayanıklı hale gelmesine yardımcı olmaktadır. Örneğin, bir görüntüde bir nesnenin konumunda küçük

bir değişiklik, maksimum değerin konumunu çok etkilememektedir. Modelin eğitim verisine aşırı uyumunu (overfitting) önleyerek genelleme yeteneğini artırmaktadır. ‘Pool_size’ parametresi, havuzlama penceresinin boyutunu belirtmektedir. Genellikle 2 veya 3 gibi pencere boyutları kullanılarak havuzlama yapılmaktadır. ‘Strides’ parametresi, pencerenin her adımda ne kadar kaydırılacağını belirtmektedir (Ait Skourt vd., 2022).

MaxPooling1D genellikle Convolution1D katmanından sonra kullanılmaktadır. Convolution1D katmanı, verideki yerel özelliklerin çıkarılmasını sağlarken, MaxPooling1D bu özelliklerin en önemlilerini seçerek boyut indirgeme ve özetleme yapmaktadır (Ait Skourt vd., 2022).

‘model.add(Convolution1D(filters=256, kernel_size=6, padding="same", activation="relu"))’

‘model.add(MaxPooling1D(pool_size=2))’

Yukarıdaki kod ile bu katmana ek olarak, yeni bir konvolüsyon karmanı eklenmiştir. Filtre sayısını 128'den 256'ya çıkararak modelin öğrenme kapasitesini artırılmıştır. Bu sayede, model veri içerisindeki daha fazla detayı ve daha karmaşık özellikleri yakalayabilmektedir. Daha fazla filtre, modelin daha önceki katmanlardan elde ettiği bilgiyi kullanarak daha soyut kavramları öğrenmesine olanak tanımaktadır.

Sonrasında, aşırı öğrenme (overfitting) sorununu önlemek için bir havuzlama (MaxPooling1D) katmanı eklenmiştir. Bu katman, özellik haritalarının boyutunu küçülterek modelin karmaşıklığını azaltmaktadır ve böylece modelin eğitim verisine aşırı uyum yapmasını engellemektedir (Al-jabery vd., 2020b).

‘model.add(Flatten())’

Yukarıdaki kod ile bir düzleştirme (flatten) katmanı eklenmiştir. Bu katman, bir sinir ağında, özellikle de konvolüsyonel sinir ağlarında, çok boyutlu bir girdiyi tek boyutlu bir vektöre dönüştüren bir katmandır. Bu dönüşüm, modelin sonraki katmanlarında daha kolay işlenebilmesi için gereklidir. Konvolüsyonel katmanlar, girdi verisindeki yerel özellikleri öğrenirken tam bağlantılı (dense) katmanlar, bu özellikleri birleştirerek daha global bir karar verebilmektedir. Düzleştirme (flatten) katmanı, bu iki katman türü arasındaki köprü görevi görmektedir. Konvolüsyonel katmanların çıkışı,

genellikle çok boyutlu bir tensör (örneğin, yükseklik, genişlik, kanal sayısı) şeklindedir. Tam bağlantılı katmanlar ise düz bir vektör kullanmaktadırlar. Düzleştirme katmanı bu dönüşümü yapmaktadır. Düzleştirme katmanına, genellikle bir konvolüsyon veya havuzlama katmanından gelen çok boyutlu bir tensör verilmektedir. Bu tensör, uzun bir satıra (vektör) dönüştürülmektedir. Bu satırın uzunluğu, girdi tensöründeki tüm elemanların sayısına eşittir. Elde edilen vektör, sonraki tam bağlantılı katmana gönderilmektedir. Örneğin, (h, w, c) bir girdi tensörü olarak kabul edilirse, bu tensör, (h,w,c) uzunluğunda bir vektöre dönüştürülmektedir (Chumachenko, Iosifidis & Gabbouj, 2022).

Düzleştirme katmanı sayesinde, konvolüsyonel ve tam bağlantılı katmanlar aynı modelde kullanılabilmektedir. Tüm özellikler, tam bağlantılı katmanlara aktarılmaktadır ve böylece modelin daha fazla bilgiyi kullanmasına olanak tanımaktadır. Özetle Flatten katmanı, konvolüsyonel sinir ağlarında çok önemli bir rol oynamaktadır. Bu katman sayesinde, konvolüsyonel katmanların yerel özellikleri öğrenme yeteneği ile tam bağlantılı katmanların global karar verme yeteneği birleştirilmektedir.

‘model.add(Dense(16, activation='relu'))’

Yukarıdaki kod ile modele bir tam bağlantılı (Dense) katman eklenmektedir ve bu katman, sinir ağında tüm nöronların birbiriyle tam bağlantılı olduğu bir yapıyı temsil etmektedir. Bu katman, düzleştirilmiş özellikleri işlemekte ve modelin son katmanlarda daha karmaşık özellikler öğrenmesine olanak vermektedir. Bu, katmanda 16 nöron (birim) kullanılmaktadır. Her bir nöron, kendisine gelen girdiyi bir ağırlık matrisi ile çarparak kendi aktivasyon fonksiyonuna göndermektedir. Bu 16 nöron, verideki özelliklerden farklı soyut temsiller öğrenebilmektedir. Relu aktivasyon fonksiyonu, negatif girdileri sıfıra çevirirken pozitif girdileri olduğu gibi bırakmaktadır. Bu, modelin doğrusal olmayan ilişkiyi öğrenmesine ve daha derin katmanlarda gradyan azalma sorununu çözmesine yardımcı olmaktadır. Bu katman, Düzleştirme Katmanı’ndan gelen girdiyi işler ve modelin daha karmaşık, yoğun bağlantılı yapılar üzerinden sınıflandırma veya regresyon gibi görevler için hazırlık yapmasını sağlamaktadır. Modelin daha yüksek seviyeli özellikleri öğrenmesine olanak tanımaktadır ve çıktıyı bir sonraki katmana aktarmaktadır (Manjula & Vijaya, 2020).

Modelin eğitim verisini ezberlemesi ve yeni, görülmemiş verilere genelleme yeteneğini kaybetmesi durumuna Aşırı Uyum (Overfitting) denmektedir. Bu durum, modelin eğitim verisindeki gürültüleri veya rastgele varyasyonları da öğrenmesine neden olur (Al-jabery vd., 2020b).

Aşırı uyum problemini azaltmak için kullanılan yöntemlerden bir tanesi seyreltme (dropout) işlemidir. Seyreltme işlemi sırasında bir katmanda bulunan nöronların belli bir orandaki kısmı pasif hale getirilerek modelin daha çok zorlanarak daha fazla öğrenme yapması sağlanır. Bu sayede, model daha genelleyici özellikler öğrenmektedir. Seyreltme, bir tür Düzenlileştirme (Regularization) tekniğidir. Düzenleme, modelin karmaşıklığını azaltarak aşırı uyumu önlemeyi amaçlamaktadır. Seyreltme, L1 ve L2 düzenleme gibi diğer düzenleme tekniklerine ek olarak veya onların yerine kullanılabilir. Seyreltme, modelin yeni verilere daha iyi genelleme yapmasına yardımcı olmaktadır. Modelin eğitim verisine aşırı uyum yapmasını engellemektedir. Seyreltme, bazı nöronları devre dışı bırakarak hesaplama maliyetini azaltabilmektedir. Seyreltme, bir model yerine birden fazla model eğitimi yapmaya benzer bir etki yaratmaktadır. Bu sayede, modelin farklı alt ağları öğrenmesine yardımcı olmaktadır. Seyreltme oranının belirlenmesi önemlidir. Seyreltme oranını düşük belirlemek, aşırı uyumu önlemeye yetmeyebilirken, çok yüksek bir oran belirlemek ise modelin öğrenme kapasitesini azaltabilmekte ve eğitim süresini uzatabilmektedir (Wager, Wang & Liang, 2013).

Seyreltme, derin sinir ağlarında, sıklıkla kullanılmaktadır. Görüntü Sınıflandırma yapan Konvolüsyonel Sinir Ağları'nda aşırı uyumu önlemek için etkilidir. Saldırı tespit sistemlerinde yoğun olarak kullanılan Tekrarlayan Sinir Ağları ve Uzun Kısa Vadeli Hafıza Sinir Ağları gibi algoritmalarda da sıklıkla kullanılabilmektedir. Seyreltme sırasında, ileri yayılım sırasında rastgele seçilen nöronların çıktıları sıfıra eşitlenmektedir Geri yayılım sırasında, devre dışı bırakılan nöronlara ait ağırlıklar güncellenmektedir. Her eğitim adımında, hangi nöronların devre dışı bırakılacağını belirlemek için bir seyreltme maskesi oluşturulmaktadır (Wager, Wang & Liang, 2013)

'model.add(Dense(2,activation='softmax'))'

Yukarıdaki kod ile 2 nöronlu aktivasyon fonksiyonu olarak 'softmax' kullanan tam bağlı bir sinir katmanı (Dense) eklenmektedir. Softmax, bir vektördeki tüm değerleri 0 ile 1 arasında normalize etmekte ve toplamlarının 1 olmasını sağlamaktadır. Bu sayede,

her bir değer bir olasılık olarak yorumlanabilmektedir. Çoklu sınıflandırma problemlerinde, her bir sınıf için bir olasılık değeri üretmek için çok caziptir. En yüksek olasılığa sahip sınıf, modelin tahmini olarak kabul edilmektedir (Wei, Wu, Wu, Chen, Barrett & Farchi, 2023).

İkili sınıflandırmada çıkış katmanında iki nöron olduğu için, model iki farklı sınıf için birer olasılık değeri üretmektedir. Bu iki değer, softmax fonksiyonu ile 0 ile 1 arasında normalize edilmekte ve toplamları 1 olmaktadır. Örneğin, [0.7, 0.3] çıktısı, modelin ilk sınıfı %70, ikinci sınıfı %30 olasılıkla tahmin ettiği anlamına gelmektedir. İki sınıflı sınıflandırma problemlerinde, çapraz entropi kaybı (cross-entropy loss) sıklıkla kullanılmaktadır. Bu kayıp fonksiyonu, modelin tahminleri ile gerçek etiketler arasındaki farkı ölçmektedir. Kayıp fonksiyonunu minimize etmek için geri yayılım algoritması kullanılmaktadır. Bu algoritma, ağırlık ve sapma değerlerinin güncellenmesini sağlamaktadır. Saldırı veya normal şeklinde ikili yapılmak istenen sınıflandırmalarda, pozitif veya negatif olduğunu tespitine yönelik hastalık teşhisinde, olumlu veya olumsuz gibi iki sınıflı duygu sınıflandırmalarında yoğun bir şekilde kullanılmaktadır (De Boer, Kroese, Mannor & Rubinstein, 2005).

Sigmoid Fonksiyonu, çıktısı 0 ile 1 arasında olduğu için, doğrudan bir olasılık olarak yorumlanabilmektedir. Bu özellik, özellikle ikili sınıflandırma problemlerinde (örneğin, evet/hayır, saldırı/saldırı değil) oldukça kullanışlıdır. Fonksiyon sürekli ve türevi alınabilir olduğu için, optimizasyon algoritmalarında kolaylıkla kullanılabilmektedir. Çok küçük veya çok büyük giriş değerleri için, fonksiyonun gradyanı sıfıra yaklaşmaktadır. Bu durum, derin ağlarda öğrenme hızını yavaşlatan ve ağırlıkların güncellenmesini zorlaştıran kaybolan gradyan (vanishing gradient) problemine neden olmaktadır. Çıktılar 0 ile 1 arasında olduğu için, çıktıların ortalaması sıfır değildir. Bu durum, ağırlıkların güncellenmesini zorlaştırabilmektedir. Çıktısı 0 ile 1 arasında olduğu için, doğrudan bir sınıfın olma olasılığını temsil etmektedir. Lojistik Regresyon gibi ikili sınıflandırma problemlerinde yaygın olarak kullanılmaktadır. Daha eski sinir ağlarında ilk katmanlarda kullanılmaktaydı ancak günümüzde ReLU gibi fonksiyonlar daha çok tercih edilmektedir.

ReLU, sadece bir karşılaştırma ve bir aritmetik işlem gerektirir, bu da onu hesaplama açısından oldukça verimli kılmaktadır. Sigmoid gibi fonksiyonlarda görülen

kaybolan gradyan (vanishing gradient) problemi, derin ağlarda öğrenmeyi zorlaştırmaktadır. ReLU, pozitif girdiler için sabit bir gradyan ürettiği için bu sorunu büyük ölçüde azaltmaktadır. ReLU, ağdaki nöronların bir kısmının sıfır çıktı vermesine neden olabilmektedir. Bu, ağın daha seyrek (sparse) olmasına neden olmakta ve aşırı uyumu azaltmaya yardımcı olabilmektedir.

ReLU'nun bazı dezavantajları da vardır. Eğer bir nöronun tüm girdileri negatif olursa, bu nöron sürekli olarak sıfır çıktı verecek ve ağırlıkları güncellenmeyecektir. Bu duruma "ölü ReLU" denir. ReLU, negatif değerler için sıfir çıktı verdiği için çıktıların ortalamasının verdiği değer anlamsızlaşabilmektedir. Bu durum, ağırlıkların güncellenmesini biraz zorlaştırabilmektedir.

ReLU, özellikle Konvolüsyonel Sinir Ağlarında ve Derin Sinir ağlarında yaygın olarak kullanılmaktadır. Genellikle gizli katmanlarda kullanılmaktadır. Regresyon problemlerinde olduğu gibi çıkış katmanında da kullanılabilmektedir.

ReLU'nun bazı dezavantajlarını gidermek için farklı varyantları geliştirilmiştir. Leaky ReLU, negatif girdiler için küçük bir değerle doğrusal bir çıktı verir. Bu sayede ‘ölü ReLU’ problemi azaltılabilmektedir. Parametric ReLU (PReLU), Leaky ReLU'nun bir genellemesidir. Negatif eğim, her nöron için öğrenilebilir bir parametredir. ELU (Exponential Linear Unit), negatif girdiler için üstel bir fonksiyon kullanılmaktadır. ReLU'ya göre daha iyi genelleme performansı gösterdiği durumlar olabilmektedir.

Tanh, çıktıların ortalaması sıfır olduğu için, ağırlıkların güncellenmesi sigmoid fonksiyonuna göre daha kolaydır. Bu, öğrenme hızını artırabilmektedir. -1 ile 1 arasında daha geniş bir çıktı aralığına sahip olması, modelin daha karmaşık ilişkileri öğrenmesine olanak tanımaktadır. Çok büyük veya çok küçük girdi değerleri için gradyan sıfıra yaklaşabilmektedir. Bu durum, derin ağlarda öğrenme hızını yavaşlatabilmektedir.

Özellikle LSTM ve GRU gibi RNN mimarilerinde sıklıkla kullanılmaktadır. Sigmoid yerine gizli katmanlarda kullanılarak daha iyi sonuçlar alınabilmektedir. Tanh fonksiyonu, sinir ağlarında kullanılan güçlü bir aktivasyon fonksiyonudur. Özellikle RNN'ler ve DNN’lerin gizli katmanlarında sigmoid'e göre daha iyi performans göstermektedir. Ancak, kaybolan gradyan problemi gibi bazı dezavantajları da vardır.

Keras'taki ‘model.summary’ fonksiyonu, bir sinir ağının yapısını özetleyen, oldukça kullanışlı bir araçtır. Bu özet, modelin katmanlarını, her bir katmandaki nöron sayısını,

kullanılan aktivasyon fonksiyonlarını, çıkış boyutlarını ve en önemlisi modeldeki toplam parametre sayısını göstermektedir. Oluşturulan modelin beklenen mimariye uygun olup olmadığını kontrol etmek için kullanılmaktadır. Modelin karmaşıklığını ve hesaplama maliyetini tahmin etmek için kullanılmaktadır. Çok fazla parametreye sahip bir model, aşırı uyuma eğilimli olabilir ve eğitim süresini uzatabilmektedir. Modelde beklenmedik bir yapı veya hatalı bir tanımlama varsa, 'model.summary' fonksiyonu, bu durumu görselleştirerek tespit edilmesine yardımcı olmaktadır (Manjula & Vijaya, 2020).

'keras.optimizers.Adam(learning_rate=0.0001)'

Yukarıdaki kod ile optimizasyon algoritması olarak Adam Optimizasyon Algoritması seçilmektedir. Adam algoritması, genellikle diğer optimizasyon algoritmalarına göre daha hızlı yakınsamaktadır. Adaptif öğrenme oranı sayesinde, modelin eğitimi sırasında daha kararlı bir ilerleme sağlamaktadır. Adam algoritması, genellikle varsayılan hiper parametreleriyle iyi çalışmakta ve fazla ayarlama gerektirmemektedir. Birçok farklı derin öğrenme modeli ve problem için etkilidir.

'model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])'

Yukarıdaki kod ile model derlenmektedir. Model derleme, bir makine öğrenmesi modelinin eğitim aşamasına hazır hale getirilmesi sürecidir. Bu süreçte, modelin hangi kayıp fonksiyonunu kullanacağı, hangi optimizasyon algoritması ile eğitileceği ve performansı nasıl ölçüleceği gibi kritik parametreler belirlenmektedir (Heaton, 2018). Yukarıdaki kodda kayıp fonksiyonu olarak 'categorical_crossentropy' kullanılmaktadır. Kayıp Fonksiyonu, bir modelin tahminlerinin gerçek değerlerden ne kadar uzak olduğunu ölçen bir matematiksel fonksiyondur. Bu fonksiyon, modelin eğitimi sırasında ağırlıkların güncellenmesi için kullanılan bir sinyaldir. Kayıp değeri ne kadar düşükse, modelin tahminleri o kadar doğru demektir. 'Categorical Crossentropy', çok sınıflı sınıflandırma problemlerinde sıklıkla kullanılan bir kayıp fonksiyonudur. Her bir sınıf için bir olasılık değeri tahmin eden modellerde kullanılmaktadır. Bu fonksiyon, modelin tahmin ettiği olasılık dağılımı ile gerçek sınıfın olasılık dağılımı arasındaki farkı ölçmektedir. Fark ne kadar küçükse, modelin tahminleri o kadar doğrudur. Optimizasyon Algoritması, modelin ağırlıklarını güncelleyerek kayıp fonksiyonunu minimize etmeye çalışan bir algoritmadır. 'Adam', günümüzde en popüler optimizasyon algoritmalarından biridir. Adaptif Moment Tahmini (Adaptive Moment Estimation) anlamına gelmektedir. Adam hem momentumun

hem de RMSprop'un avantajlarını birleştirmektedir. Bu sayede, öğrenme hızını otomatik olarak ayarlayarak daha hızlı ve daha iyi sonuçlar elde etmeye yardımcı olmaktadır. Metrikler, modelin performansını değerlendirmek için kullanılan ölçütlerdir. 'Accuracy' (Doğruluk), modelin doğru tahmin ettiği örneklerin toplam örnek sayısına oranıdır. Sınıflandırma problemlerinde sıklıkla kullanılan temel bir performans ölçütüdür (Kingma & Ba, 2014).

'early_stop_callback = keras.callbacks.EarlyStopping(monitor='loss', patience=5)'

Yukarıdaki kodda Erken Durma (EarlyStopping) özelliği kullanılmaktadır, Keras Kütüphanesi'nde model eğitimi sırasında aşırı öğrenmeyi önlemek için kullanılan bir mekanizmadır. Model, eğitim verisi üzerinde çok iyi öğrenir ancak yeni, görmemiş veri üzerinde genelleme yeteneğini kaybetmesi durumuna Aşırı Uyum denmektedir. Erken Durma, bu durumu tespit ederek eğitimi erken durdurmakta ve böylece modelin genelleme yeteneğini korumasına yardımcı olmaktadır. 'monitor='loss'' parametre kullanımı, eğitim sırasında hangi metrik değerine bakılarak karar verileceğini belirtmektedir. Modelin tahminlerinin gerçek değerlerden ne kadar uzak olduğunu gösteren Kayıp Değeri izlenmektedir. Kayıp değeri düştükçe model daha iyi öğrenmiş demektir. 'patience=5' parametre kullanımı, izlenen metrikte (bu durumda kayıp) iyileşme görülmediği takdirde kaç eğitim dönemi (epoch) beklenileceğini belirtmektedir. patience=5 olduğunda, modelin kaybı 5 eğitim dönemi boyunca iyileşmezse eğitim durdurulmaktadır. 5 eğitim dönemi boyunca kayıp değeri düşmezse modelin daha fazla eğitimle önemli ölçüde iyileşmeyeceği varsayılmaktadır (Prechelt, 1998).

Eğitim sırasında belirtilen kayıp (loss) metriği sürekli olarak takip edilmektedir. Belirlenen bekleme süresi (patience) değeri kadar epoch boyunca metrikte iyileşme olmazsa bir sayaç başlatılmaktadır. Sayaç, belirlenen bekleme süresi değerine ulaştığında eğitim durdurulmaktadır. Modelin eğitim verisine Aşırı Uyum yapmasını ve yeni verilere genelleme yeteneğini kaybetmesini engellemektedir. Bu metot, gereksiz yere uzun süreli eğitimleri engelleyebilmekte ve hesaplama kaynaklarının daha verimli kullanılmasını yol açmaktadır.

'reduce_lr = keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=0.001)'

Yukarıdaki kodda, ‘ReduceLROnPlateau’, Keras'ta model eğitimi sırasında öğrenme oranını dinamik olarak ayarlayan bir geri çağırmadır (callback). Özellikle, modelin öğrenme hızı yavaşladığında öğrenme oranını düşürerek, modelin daha iyi yerel minimumlara ulaşmasına yardımcı olmaktadır. Bu sayede, modelin hem daha hızlı hem de daha iyi öğrenmesi sağlanmaktadır. ‘monitor='val_loss'’ parametre kullanımı, hangi metrik değerine bakılarak öğrenme oranı kararının verileceğini belirtmektedir. 'val_loss' seçildiğinde, doğrulama setindeki kayıp değeri izlenmektedir. Bu, modelin eğitim setine aşırı uyum (overfitting) yapmadan genel performansını değerlendirmek için önemlidir. ‘factor=0.2’ parametre kullanımı ile öğrenme oranının ne kadar azaltılacağı belirtilmektedir. ‘factor=0.2’ olduğunda, öğrenme oranı her azaltıldığında 0.2 ile çarpılmaktadır. Yeni öğrenme oranı eski öğrenme oranının %20'si olmaktadır. 'patience=3’ parametre kullanımı ile izlenen doğrulama kaybı metriğinde iyileşme görülmediği takdirde kaç eğitim dönemi beklenileceğini belirtmektedir. ‘patience=3’ olduğunda, doğrulama kaybı 3 eğitim dönemi boyunca iyileşmezse öğrenme oranı azaltılmaktadır. ‘min_lr=0.001’ parametre kullanımı ile öğrenme oranının düşürülmesine izin verilen en düşük değer belirtilmektedir. Öğrenme oranının 0,001'in altına düşmemesi faydalı olmaktadır.

Her eğitim dönemi sonunda doğrulama setindeki kayıp değeri hesaplanmaktadır. Eğer doğrulama kaybı belirlenen bekleme süresi boyunca iyileşmezse bir sayaç başlatılmaktadır. Sayaç belirlenen bekleme süresi değerine ulaştığında öğrenme oranı belirlenen faktör ile çarpılarak azaltılmaktadır. Öğrenme oranı, belirlenen minimum öğrenme oranının altına düşerse daha fazla azaltılamamaktadır.

Modelin başlangıçta daha büyük adımlarla öğrenmesine ve daha sonra daha küçük adımlarla ince ayar yapmasına olanak tanıyarak öğrenme sürecinin optimize edilmesine yardım etmektedir. Doğrulama kaybı üzerinde durarak, modelin eğitim setine aşırı uyum yapmasını engellemektedir. Bu geri çağırma (callback) , öğrenme oranını elle ayarlamak yerine, otomatik olarak ayarlayarak kullanım kolaylığı ve zaman tasarrufu sağlamaktadır (Smith, 2015).

‘model.fit(X_train, y_train_cat, batch_size=128, epochs=30, verbose=1, validation_data=(X_test, y_test_cat), callbacks=[early_stop_callback, reduce_lr])’

Yukarıdaki kod ile model eğitilmektedir. ‘model.fit’ yöntemi, modeli eğitim verisi üzerinde eğitmek için kullanılmaktadır. Eğitim sürecinde, modelin ağırlıkları güncellenmekte ve belirlenen kayıp fonksiyonuna göre modelin performansı izlenmektedir. ‘X_train’, eğitim için kullanılan girdi verilerinin (özelliklerin) dizisidir. Genellikle bir NumPy dizisi veya bir TensorFlow Tensörü biçimindedir. ‘y_train_cat’, eğitim sırasında modelin tahmin etmesi beklenen hedef etiketlerin dizisidir. ‘y_train_cat’, genellikle kategorik verilere sahipse tek sıcak kodlama (one-hot encoding) dönüşümü yapılır. ‘batch_size’ (toplu işlem boyutu) parametresi, her güncellemede kullanılacak örnek sayısını belirlemektedir. 128 olarak seçildiğinde model her 128 örnek için ağırlıkları güncellemektedir. Küçük Toplu İşlem Boyutu genellikle daha fazla gürültü içerebilirken, büyük Toplu İşlem Boyutu daha az güncelleme ile daha az gürültü içermektedir. ‘epochs=30’, parametre kullanımı, modelin eğitim verisi üzerinde kaç eğitim dönemi (epoch) eğitileceğini belirtmektedir. Model, her eğitim döneminde tüm eğitim verisini bir kez görerek öğrenmektedir. Bu kodda 30 eğitim dönemi boyunca eğitim yapılması planlanmıştır. ‘verbose=1’ parametre kullanımı, eğitim sürecinin ne kadar ayrıntılı bir şekilde rapor edileceğini kontrol etmektedir. ‘verbose=1’, eğitim sürecinin ilerlemesini (eğitim dönemi başına kayıp ve metrik değerlerini) gösterir. validation_’data=(X_test, y_test_cat)’ parametre kullanımı, doğrulama verisini tanımlamaktadır. Model eğitim sırasında, doğrulama kaybını ve doğruluğunu izlemek için bu veriyi kullanmaktadır. ‘X_test’, doğrulama girdi verileri, ‘y_test_cat’ ise doğrulama etiketleridir. ‘callbacks=[early_stop_callback, reduce_lr]’ parametre kullanımı, eğitim sırasında kullanılacak geri çağırmaları (callback) tanımlamaktadır. Burada iki geri çağırma kullanılmıştır. ‘early_stop_callback’ geri çağırması, eğitim sürecinin, kayıp belirli bir süre (5 eğitim dönemi) boyunca iyileşmediğinde durdurulmasını sağlamaktadır. reduce_lr geri çağırması, doğrulama kaybı belirli bir süre (3 eğitim dönemi) boyunca iyileşmediğinde öğrenme oranını azaltmaktadır (Prechelt, 1998).

Özetle bu kod parçası, modelin ‘X_train’ ve ‘y_train_cat’ verileri üzerinde 30 eğitim dönemi boyunca, 128 örneklik toplu işlem boyutuyla eğitimi başlatmıştır. Eğitim sürecinde her eğitim döneminde kayıp ve doğruluk gibi metrikler izlenmiş ve test seti için doğrulama ile performans değerlendirilmiştir. Ayrıca, modelin aşırı öğrenmesini önlemek için erken durdurma ve öğrenme oranını azaltma mekanizmaları kullanılmıştır. Eğitim

süreci, sonuçları history (Geçmiş) değişkeninde saklanmıştır. Bu değişken kullanılarak modelin eğitimi süresince gerçekleşen metrik değerleri görüntülenmektedir.

5.5. Değerlendirme Metrikleri

Makine Öğrenmesi ve Derin Öğrenme 'de eğitilen modellerin başarısını ölçmek için çeşitli metrikleri geliştirilmiştir. Bunlardan en çok kullanılanları Doğruluk (Accuracy), Kesinlik (Precision), Hassasiyet (Recall), F1 Skoru'dur. Sınıflandırma sırasında eğitilen model test ettiği verinin sonucunu tahmin etmektedir. Tahmin edilen değerlerle gerçek değerler bu metriklerde hesaplamalarda kullanılır. Çizelge 5.11'de bu değerleri anlatan Karışıklık (Confusion) Matrisi detaylandırmaktadır.

Çizelge 5.11. Karmaşıklık Matrisi

	Hesaplama Sonucu (Tahmin)	Gerçek Değer
TP (True Positive)	Pozitif	Pozitif
TN (True Negative)	Negatif	Negatif
FP (False Positive)	Pozitif	Negatif
FN (False Negative)	Negatif	Pozitif

Çizelge 5.10'daki Pozitif (P) denilen değerler üzerinde inceleme yapılan hedef sınıf değerleridir. Negatif değerler ise hedef değerin olmadığı diğer durumu ifade eder. İkili sınıflandırma için örneklendirilecek olursa pozitif durum incelen durumun var olma durumu, negatif durum incelenen durumun olmaması olarak düşünülebilir. Örneğin, bir Saldırı Tespit Sistemi modelleniyorsa pozitif (Saldırı), negatif (Normal) veya kanser olma durumu araştırılıyorsa pozitif (kanser var), negatif (kanser yok) olarak Karışıklık Matrisi düzenlenebilir.

Saldırı Tespit Sınıflandırması özelinde değerlendirme metrikleri şu şekilde detaylandırılabilmektedir.

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

Doğruluk modelin test sonucu doğru olarak tespit ettiği saldırı (TP) ve normal (TN) verilerinin toplamının bütün test edilen verilere oranını vermektedir. Modelin genel olarak ne kadar doğru oranda tahmin yaptığını gösterir.

$$\text{Kesinlik} = \frac{TP}{TP + FP}$$

Kesinlik modelin test sonucu saldırı olarak tahmin ettiği verilerin (TP+FP) ne kadarının gerçekte saldırı olduğunu göstermektedir.

$$\text{Hassasiyet} = \frac{TP}{TP + FN}$$

Hassasiyet modelin test sonucu gerçekte saldırı olan verilerin (TP+FN) ne kadarını doğru tahmin ettiğini göstermektedir.

$$F1 = 2 \cdot \frac{\text{Kesinlik} \cdot \text{Hassasiyet}}{\text{Kesinlik} + \text{Hassasiyet}}$$

F1 Skoru ise kesinlik ve doğruluk değerlerinin harmonik ortalamasını vermektedir.

$$FPR = \frac{FP}{FP + TN}$$

FPR, modelin negatif sınıfları doğru bir şekilde tahmin edememe oranını göstermektedir. Saldırı Tespit Sistemi özelinde normal verilerin ne kadarının saldırı olarak tahmin edildiğini göstermektedir. FPR, Yanlış Alarm olarak da isimlendirilmektedir.

BÖLÜM 6

SONUÇLAR VE TARTIŞMA

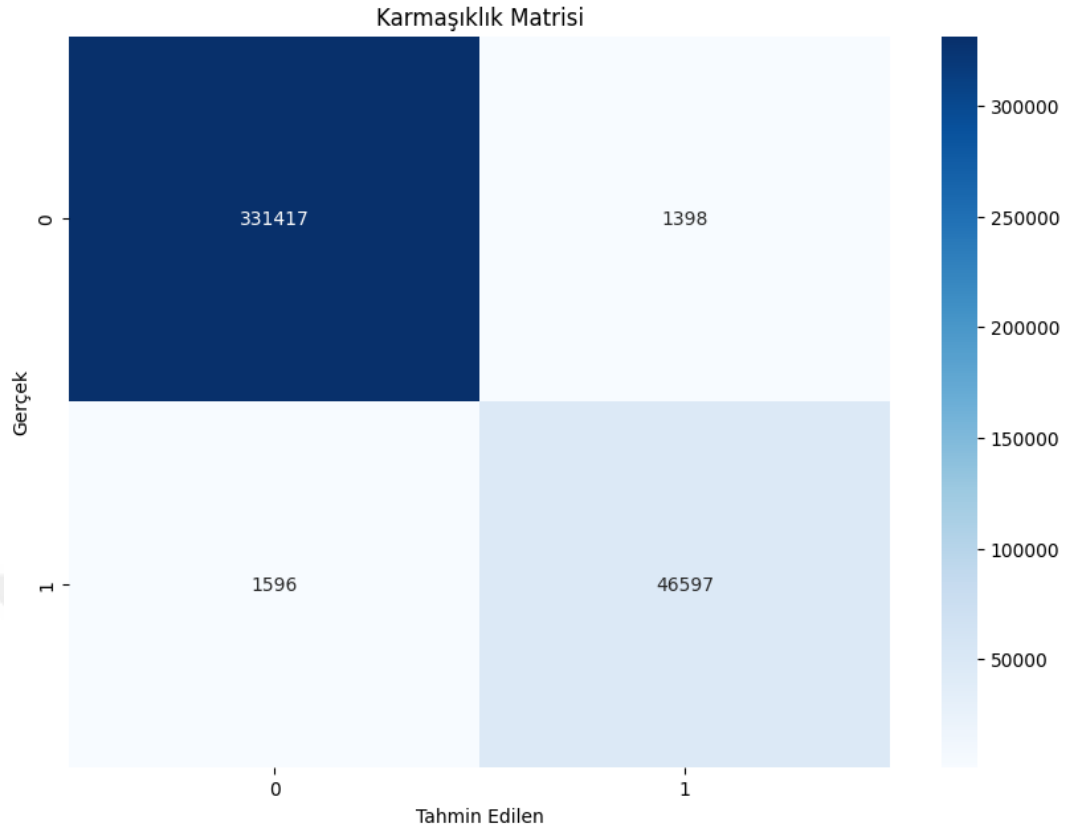
Bölüm 6.1'de tezde önerilem saldırı tespit sisteminin literatürde sıklıkla kullanılan veri setlerinde denenerek elde edilen sonuçlar gösterilmektedir. Bölüm 6.2'de bu sonuçlar literatürdeki diğer çalışmalarla karşılaştırmalı olarak değerlendirilmektedir. Bölüm 6.3'te sonraki çalışmalar için öneriler verilmektedir.

6.1. Önerilen Model Sonuçları

Bu bölümde tezde önerilen saldırı tespit sistemiyle Saldırı Tespit Alanı'nda literatürde sıklıkla kullanılan IOT-23, CIC-IOT-2023, UNSW-NB15, N-BAIOT, BOT-IOT, CIC-DDOS-2019, TON-IOT, CIC-IDS-2018, CIC-IDS-2017 veri setleri denenmiştir. UNSW-NB15 veri setine ilişkin sonuçlar Çizelge 6.1'de listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.1'de verilmiştir.

Çizelge 6.1. UNSW-NB15 Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,995	0,996	0,996	0,996	0,033
Saldırı	0,971	0,967	0,969	0,967	0,004
Test Ortalama	0,983	0,982	0,982	0,982	0,019
Test Ağırlıklı Ortalama	0,992	0,992	0,993	0,992	0,029
Doğrulama Ortalama	0,982	0,980	0,982	0,980	0,020
Doğrulama Ağ. Ortalama	0,992	0,992	0,992	0,992	0,031
Model Hata Oranı	0,016				

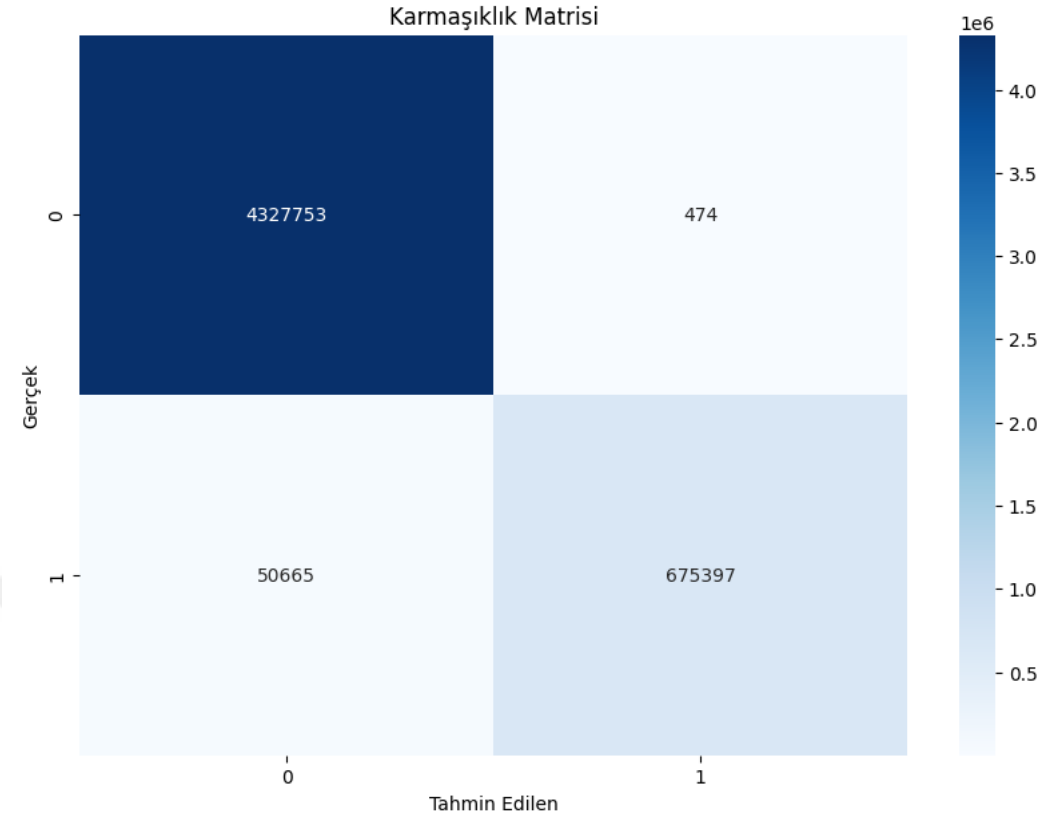


Şekil 6.1. UNSW-NB15 Veri Seti Karmaşıklık Matrisi

CIC-IDS-2018 veri setine ilişkin sonuçlar Çizelge 6.2'de listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.2'de verilmiştir.

Çizelge 6.2. CIC-IDS-2018 Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,988	1,000	0,994	1,000	0,070
Saldırı	0,999	0,930	0,964	0,930	0,000
Test Ortalama	0,994	0,965	0,979	0,965	0,035
Test Ağırlıklı Ortalama	0,990	0,990	0,990	0,990	0,060
Doğrulama Ortalama	0,994	0,966	0,979	0,966	0,035
Doğrulama Ağ. Ortalama	0,990	0,990	0,990	0,990	0,059
Model Hata Oranı	0,046				

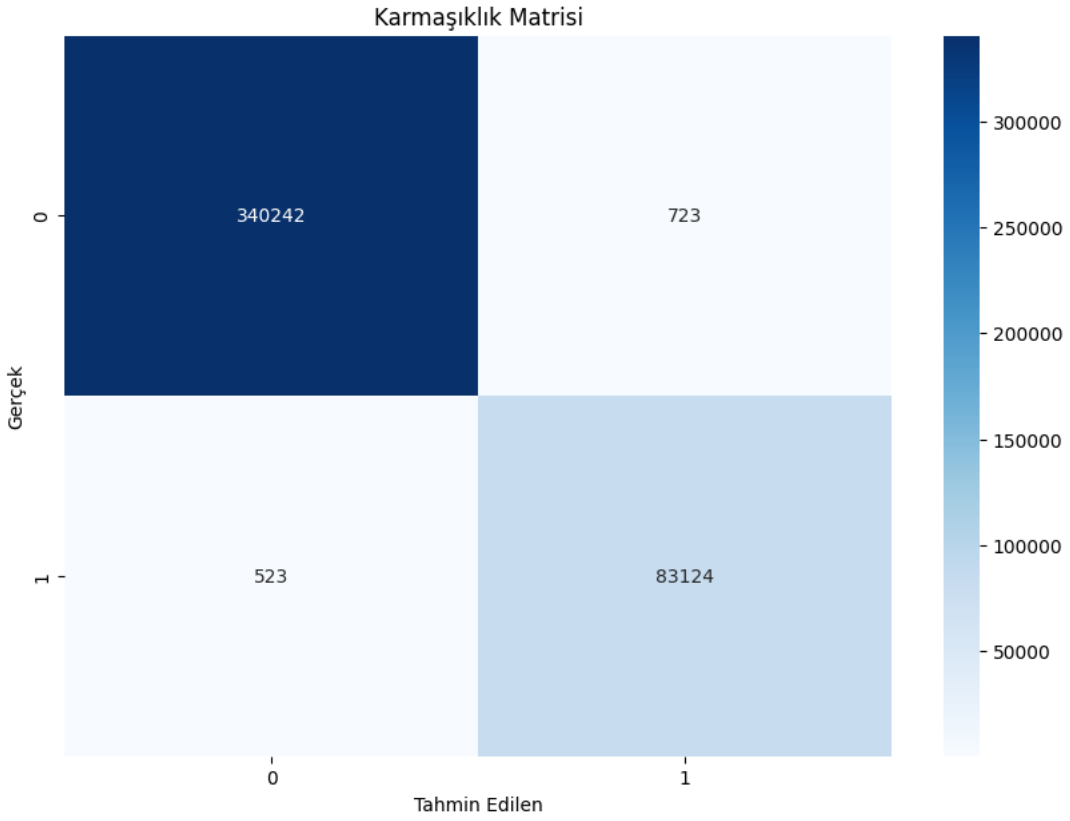


Şekil 6.2. CIC-IDS-2018 Veri Seti Karmaşıklık Matrisi

CIC-IDS-2017 veri setine ilişkin sonuçlar Çizelge 6.3'te listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.3'te verilmiştir.

Çizelge 6.3. CIC-IDS-2017 Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,998	0,998	0,998	0,998	0,006
Saldırı	0,991	0,994	0,993	0,994	0,002
Test Ortalama	0,994	0,996	0,996	0,996	0,004
Test Ağırlıklı Ortalama	0,997	0,997	0,997	0,997	0,005
Doğrulama Ortalama	0,994	0,996	0,996	0,996	0,004
Doğrulama Ağ. Ortalama	0,997	0,997	0,997	0,997	0,005
Model Hata Oranı	0,012				

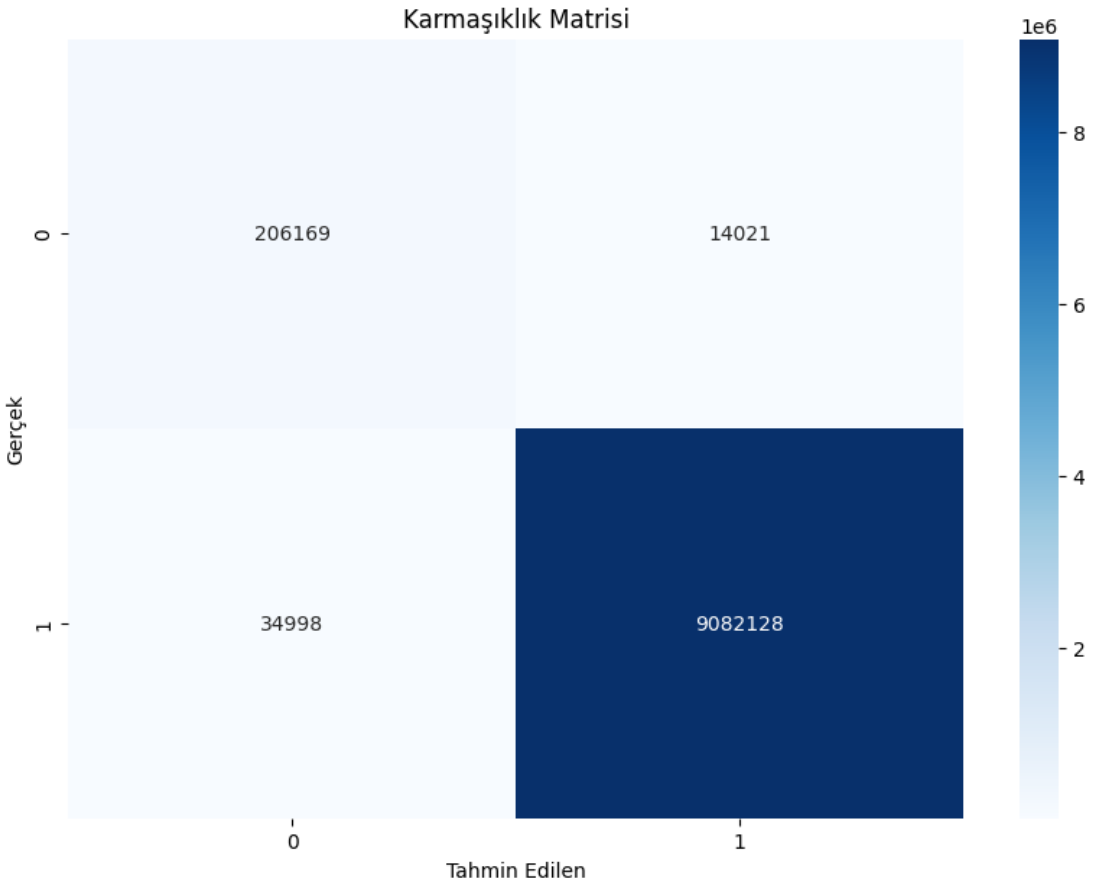


Şekil 6.3. CIC-IDS-2017 Veri Seti Karmaşıklık Matrisi

CIC-IOT-2023 veri setine ilişkin sonuçlar Çizelge 6.4'te listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.4'te verilmiştir.

Çizelge 6.4. CIC-IOT-2023 Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,855	0,936	0,894	0,936	0,004
Saldırı	0,998	0,996	0,997	0,996	0,064
Test Ortalama	0,926	0,966	0,946	0,966	0,034
Test Ağırlıklı Ortalama	0,995	0,995	0,995	0,995	0,063
Doğrulama Ortalama	0,926	0,967	0,947	0,966	0,034
Doğrulama Ağ. Ortalama	0,995	0,994	0,994	0,997	0,062
Model Hata Oranı	0,012				

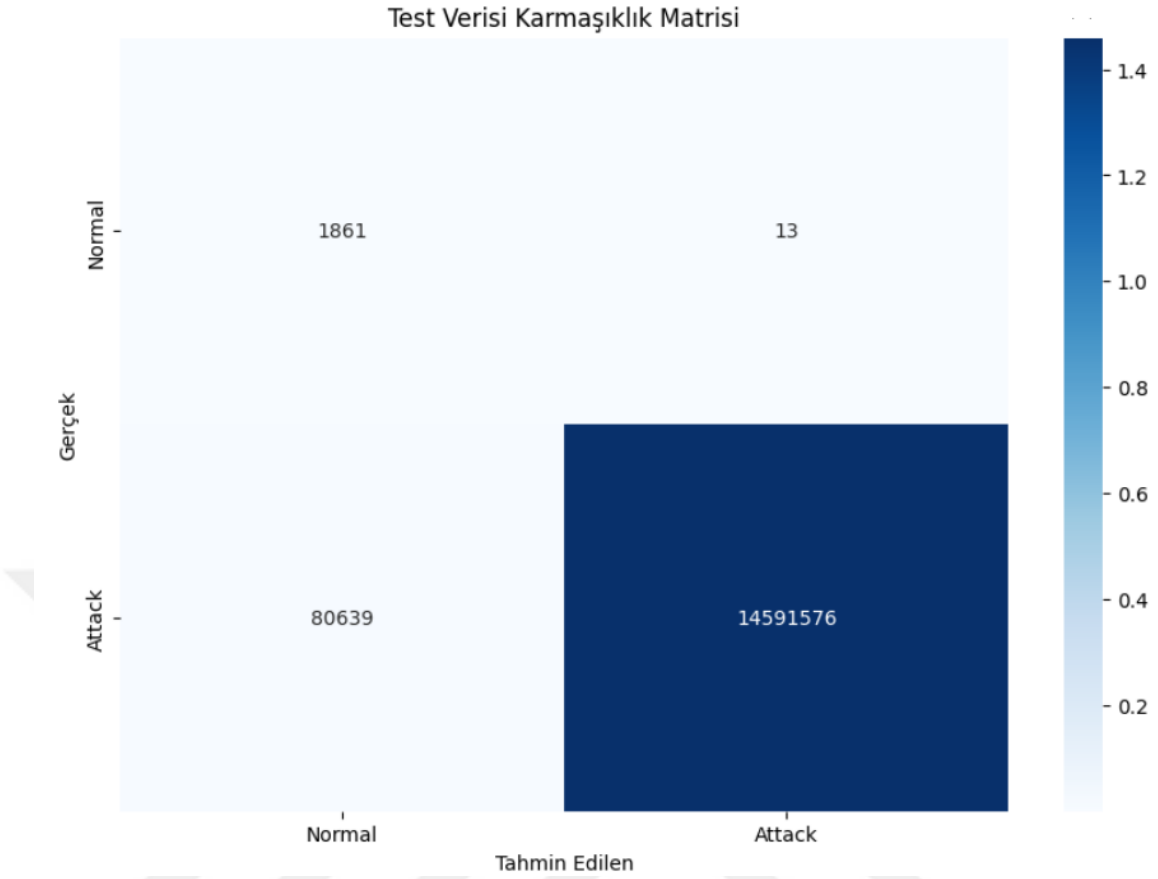


Şekil 6.4. CIC-IOT-2023 Veri Seti Karmaşıklık Matrisi

BOT-IOT veri setine ilişkin sonuçlar Çizelge 6.5'te listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.5'te verilmiştir.

Çizelge 6.5. BOT-IOT Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,0226	0,9931	0,0441	0,9931	0,0055
Saldırı	1,0000	0,9945	0,9972	0,9945	0,0069
Test Ortalama	0,5113	0,9938	0,5207	0,9938	0,0062
Test Ağırlıklı Ortalama	0,9999	0,9945	0,9971	0,9945	0,0069
Doğrulama Ortalama	0,9969	0,9968	0,9968	0,9968	0,0032
Doğrulama Ağ. Ortalama	0,9969	0,9968	0,9968	0,9968	0,0032
Model Hata Oranı	0,018				

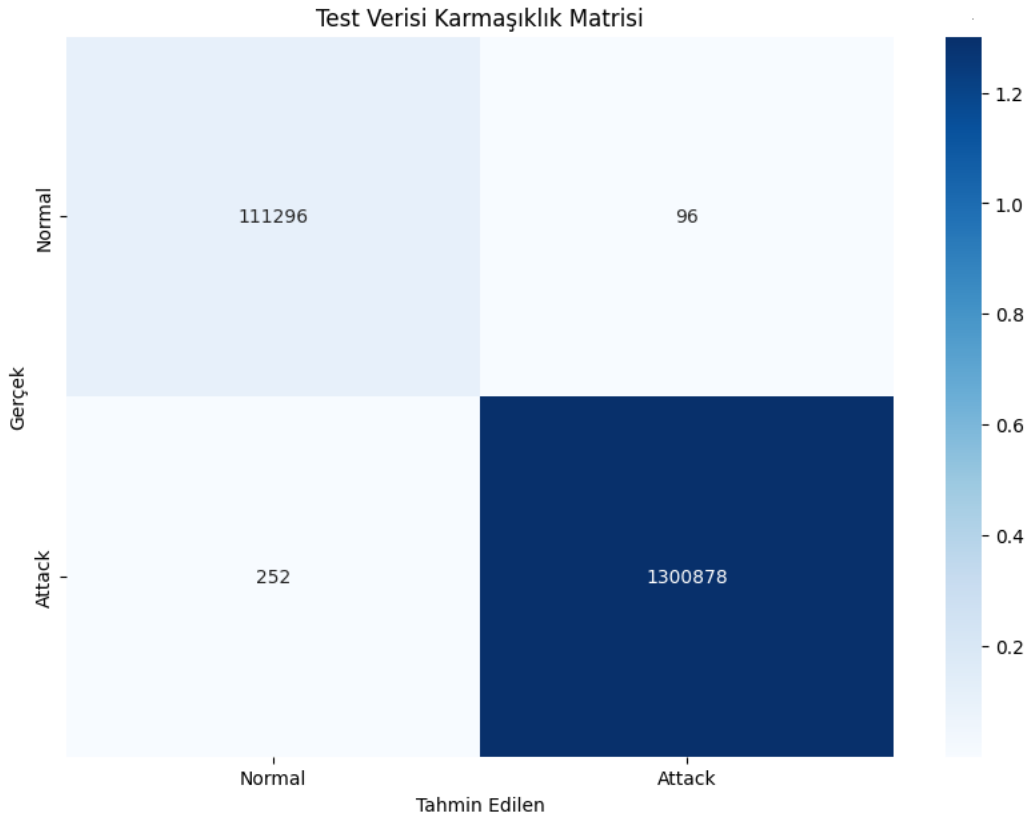


Şekil 6.5. BOT-IOT Veri Seti Karmaşıklık Matrisi

N-BAIOT veri setine ilişkin sonuçlar Çizelge 6.6'da listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.6'da verilmiştir.

Çizelge 6.6. N-BAIOT Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,9977	0,9991	0,9984	0,9991	0,0002
Saldırı	0,9999	0,9998	0,9999	0,9998	0,0009
Test Ortalama	0,9988	0,9995	0,9992	0,9995	0,0005
Test Ağırlıklı Ortalama	0,9998	0,9998	0,9998	0,9998	0,0008
Doğrulama Ortalama	0,9987	0,9994	0,9991	0,9994	0,0006
Doğrulama Ağ. Ortalama	0,9997	0,9997	0,9997	0,9997	0,0008
Model Hata Oranı	0,0031				

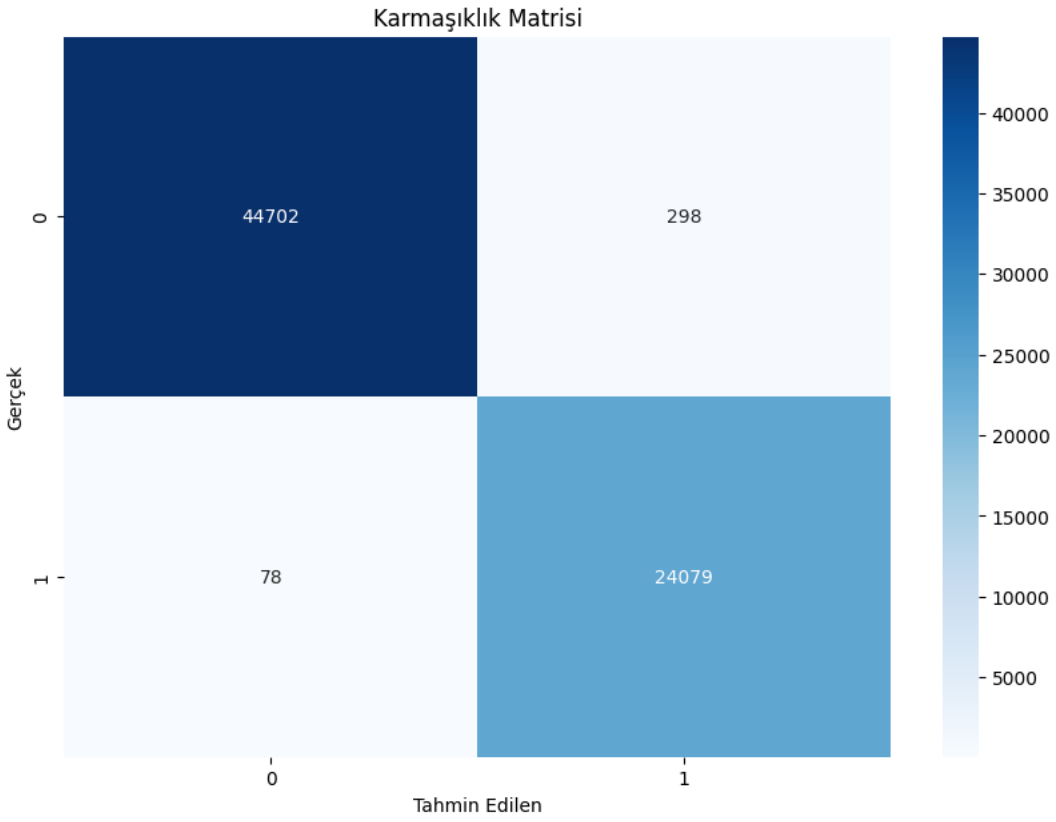


Şekil 6.6. N-BAIOT Veri Seti Karmaşıklık Matrisi

TON-IOT veri setine ilişkin sonuçlar Çizelge 6.7'de listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.7'de verilmiştir.

Çizelge 6.7. TON-IOT Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,998	0,993	0,996	0,993	0,003
Saldırı	0,988	0,997	0,992	0,997	0,007
Test Ortalama	0,993	0,995	0,994	0,995	0,005
Test Ağırlıklı Ortalama	0,995	0,994	0,995	0,994	0,004
Doğrulama Ortalama	0,993	0,995	0,994	0,995	0,005
Doğrulama Ağ. Ortalama	0,995	0,995	0,995	0,995	0,005
Model Hata Oranı	0,02				

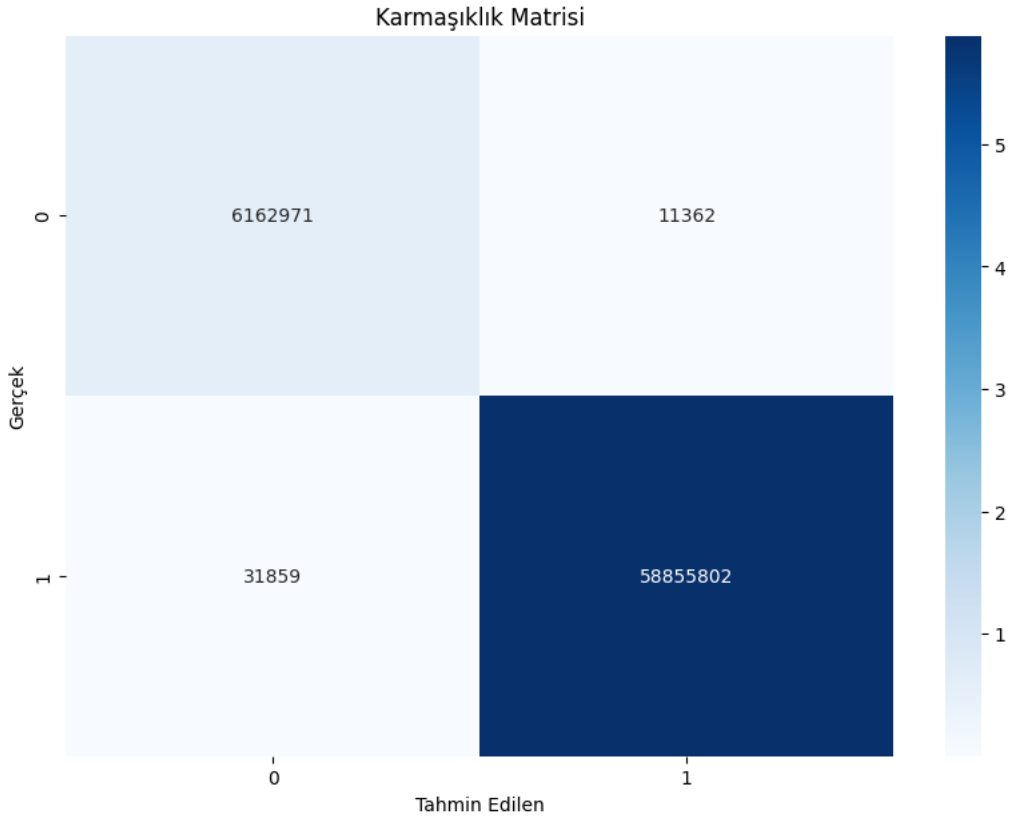


Şekil 6.7. TON-IOT Veri Seti Karmaşıklık Matrisi

IOT-23 veri setine ilişkin sonuçlar Çizelge 6.8'de listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.8'de verilmiştir.

Çizelge 6.8. IOT-23 Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,995	0,998	0,997	0,998	0,001
Saldırı	1,000	0,999	1,000	0,999	0,002
Test Ortalama	0,998	0,998	0,998	0,998	0,002
Test Ağırlıklı Ortalama	1,000	0,999	1,000	0,999	0,002
Doğrulama Ortalama	0,998	0,998	0,998	0,998	0,002
Doğrulama Ağ. Ortalama	1,000	0,997	1,000	0,997	0,002
Model Hata Oranı	0,023				

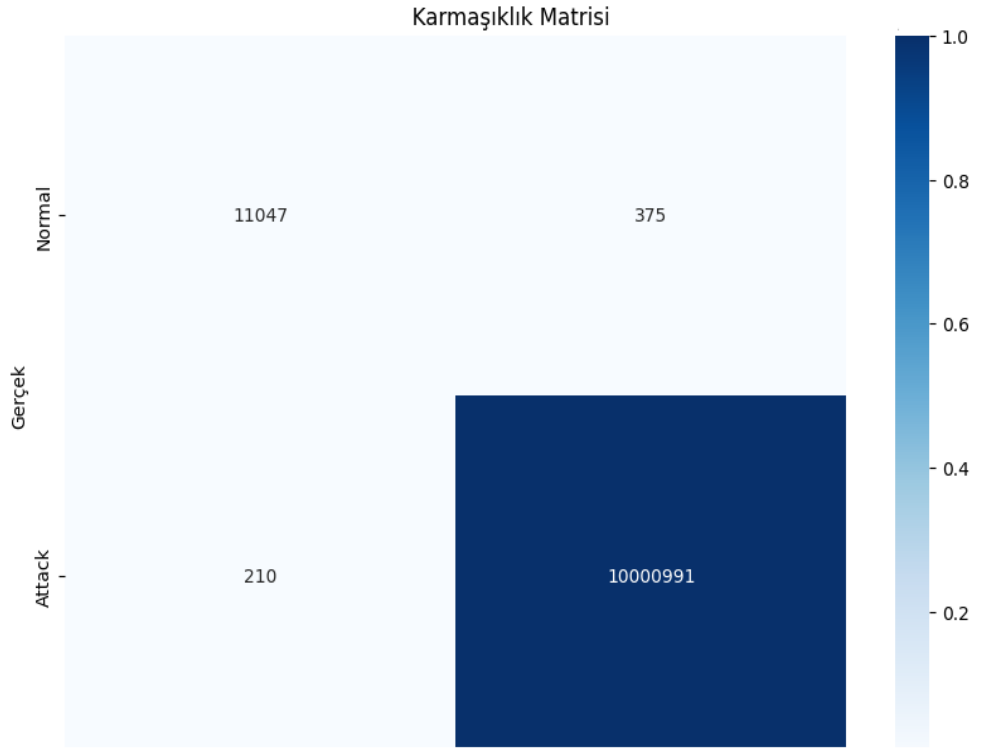


Şekil 6.8. IOT-23 Veri Seti Karmaşıklık Matrisi

CIC-DDOS-2019 veri setine ilişkin sonuçlar Çizelge 6.9'da listelenmiştir. Bu veri setine ait Karmaşıklık Matrisi Şekil 6.9'da verilmiştir.

Çizelge 6.9. CIC-DDOS-2019 Veri Seti İkili Sınıflandırma Sonuçları

	Kesinlik	Hassasiyet	F1-Skoru	Doğruluk	FPR
Normal	0,981	0,967	0,974	0,9672	0,000
Saldırı	0,9999	0,9999	0,9999	0,9999	0,033
Test Ortalama	0,991	0,984	0,987	0,9836	0,016
Test Ağırlıklı Ortalama	0,9999	0,9999	0,9999	0,9999	0,033
Doğrulama Ortalama	0,992	0,985	0,988	0,9850	0,015
Doğrulama Ağ. Ortalama	0,9999	0,9999	0,9999	0,9999	0,030
Model Hata Oranı	0,0007				



Şekil 6.9. CIC-DDOS-2019 Veri Seti Karmaşıklık Matrisi

6.2. Sonuçların Değerlendirilmesi

Bu bölümde, bölüm 6.1'de elde edilen sonuçlar literatürdeki diğer çalışmalarla karşılaştırılarak değerlendirilmektedir. Önerilen sistem, ön işleme, özellik seçimi ve model eğitim aşamalarından oluşmaktadır. Özellik seçimi önerilen saldırı tespit sisteminin en önemli adımlarından biri olmuştur. Özellik seçiminde Spearman ve Kendall Tau gibi korelasyon tabanlı yöntemler ile Rastgele Orman makine öğrenmesi algoritması hibrit bir şekilde kullanılmıştır. Bu yöntemlerin birleştirilmesiyle hibrit bir yaklaşım geliştirilmiş ve performans artışı sağlanmıştır. Hedef, gereksiz verilerin elenmesi ve önemli özelliklerin en etkili şekilde öne çıkarılmasıdır.

Spearman korelasyonu, iki değişken arasındaki sıralı ilişkiyi ölçmektedir ve doğrusal olmayan ilişkilere de duyarlıdır. Özellikle, veri setinde doğrusal olmayan ancak monoton ilişkilerin olduğu durumlarda etkili bir yöntemdir. Kendall Tau, iki değişken arasındaki monoton ilişkiyi inceleyen bir sıra tabanlı korelasyon yöntemidir. Doğrusal

olmayan ilişkilere duyarlıdır. Rastgele Orman, karar ağaçlarına dayanan bir topluluk yöntemidir ve veri setinde bulunan özelliklerin önem sırasını belirlemektedir. Ağaç yapılarında özelliklerin dallanma sıklığı üzerinden özellik önem puanları hesaplanmaktadır. Bu yöntem, gereksiz özelliklerin otomatik olarak elenmesini sağlamaktadır ve önemli özellikleri vurgulamaktadır. Spearman ve Kendall Tau'nun sağladığı korelasyon analiziyle, verideki ilişkiler detaylı şekilde ortaya çıkartılmaktadır. Rastgele Orman ise bu analizlerin ardından anlamlı özellikleri seçerek model performansını artırmaktadır.

Davahli ve arkadaşları (2020), çalışmalarında, Genetik Algoritma ve Gri Kurt Optimizasyonu hibrit modeliyle özellik seçimi yapmışlardır. (Davahli, Shamsi & Abaei, 2020). Andrade-Arenas ve arkadaşları (2020), çalışmalarında, Rastgele Orman tabanlı özellik sıralama ve önem puanlama yöntemi kullanmışlardır (Andrade-Arenas & Ramos-Romero, 2020). Lee ve arkadaşları (2020), çalışmalarında, karşılılı bilgi çıkarımı yöntemini kullanarak özellik seçimi yapmışlardır (Lee vd., 2020). Filus ve arkadaşları (2021), çalışmalarında, Rastgele Sinir Ağlarını kullanarak özellik çıkarımı yapmışlardır (Filus, Domańska & Gelenbe, 2021). Dao ve Lee (2022), çalışmalarında, Oto Kodlayıcı ve Spearman Korelasyon Yöntemi tabanlı bir özellik çıkarımı uygulamışlardır (Dao & Lee, 2022). Altunay ve Albayrak (2023), çalışmalarında, CNN tabanlı özellik çıkarımı yapmışlardır (Altunay & Albayrak, 2023). Kasongo (2023), çalışmasında Özellik Seçimi için XGBoost algoritmasını kullanmıştır (Kasongo, 2023). Mayuranathan ve arkadaşları (2022), çalışmalarında, özellik seçimi için CRDO algoritmasını kullanmışlardır. (Mayuranathan, Saravanan, Muthusenthil & Samydurai, 2022). Selvan ve arkadaşları (2022), çalışmalarında, Fisher Skoru yöntemini kullanarak özellik çıkarımı yapmışlardır (Selvan, Azees, Vinodkumar & Parthasarathy, 2022). Bouke ve arkadaşları (2023), çalışmalarında, Gini indeksi özellik seçimi yöntemi kullanmışlardır (Bouke, Abdullah, Alshatebi, Abdullah & El Atigh, 2023). Ali ve arkadaşları (2024), çalışmalarında, Oto Kodlayıcı tabanlı bir özellik çıkarımı yöntemi uygulamışlardır (Ali vd., 2024a).

Yukarıdaki çalışmaların bazılarında, birden fazla özellik çıkarımı yöntemi birleştirilerek daha iyi sonuçlar alınmıştır. Önerilen sistemde diğerlerinden farklı olarak Spearman, Kendall Tau korelasyon yöntemleri ile Rastgele Orman özellik çıkarımı yöntemi birleştirilerek modelin başarı oranı artırılmıştır.

Önerilen sistemde, saldırı tespit sistemi geliştirilirken literatürde sıklıkla kullanılan AE, LSTM, GRU algoritmaları kullanılmış, diğer çalışmalardan farklı olarak bu algoritmalar Kalıntı (Residual) katmanı ile birleştirilmiştir. Kalıntı katmanı, eğitim sırasında kullandığı ham veriyi, işlemeden sonraki çıktıya eklemektedir. Bu sayede öğrenme kabiliyeti artmaktadır ve gereksiz detaylar yerine daha hızlı bir şekilde hedef noktalara göre eğitim yapmaktadır.

Bu tezde tasarlanan model ile Saldırı Tespit Alanı'nda literatürde sıklıkla kullanılan CIC-IDS-2018, CIC-IDS-2017, CIC-IOT-2023, UNSW-NB15, N-BAIOT, BOT-IOT, CIC-DDOS-2019, TON-IOT, IOT-23 veri setleri denenmiştir. Bu alandaki yaygın veri setlerinin büyük çoğunluğu tez çalışmasına dâhil edilmiştir. Bütün veri setlerinde iyi çalışacak bir model tasarlanmaya çalışılmıştır. Bütün veri setlerinde doğruluk, kesinlik, hassasiyet, F1 Skoru metriklerinde %93'ün üzerinde başarı sağlanmıştır. Tek veri seti bazlı ölçümlerde %99,99 a yakın değerler de alınmıştır ama asıl amaç çok iyi bir başarı sonucu almak değil genelleştirilmiş, yeni veri setlerinde bile iyi sonuçlar verebilecek bir model tasarlamak olmuştur. Bu hedef de gerçekleştirilmiştir. Elde edilen başarı sonuçlarının literatürdeki diğer çalışmalar ile karşılaştırmaları Çizelge 6.10, 6.11 ve 6.12'de verilmiştir.

60.000.000'un üzerinde veriye sahip veri setleri ile de çalışılmıştır. Literatürdeki birçok çalışmanın aksine bu veri setlerinin bazı parçaları üzerinde değil, daha objektif sonuçlar çıkması adına veri setinin tamamı üzerinde çalışılmıştır. Dengesiz veri setlerinde başarılar nispeten düşük çıkmıştır. Örneğin BOT-IOT Veri Seti'nde normal veri oranı binde 1'den azdır. Bu durum nedeniyle saldırı verilerinde çok başarılı sonuçlar alınırken, normal verilerin sınıflandırılmasında daha az başarılı sonuçlar elde edilmiştir. Ağırlıklı olarak başarı oranına bakıldığı zaman bu veri setinde bile %99 üstü başarı sağlanmıştır. Veri setlerindeki normal ve saldırı verilerinin daha dengeli bir şekilde dağılması, yapılan çalışmaların sonucunun daha kabul edilebilir olmasını sağlayabilmektedir.

Çizelge 6.10. Benzer Çalışmalar İle Karşılaştırmalı Sonuçlar 1

Çalışma	Veri Seti	Kullanılan Yöntemler	Doğruluk	Kesinlik	Hassasiyet	F1	FPR
(Davahli, Shamsi & Abaei, 2020)	AWID	GA, GWO, SVM	99,10	96,03	97,64	-	-
(Lee vd., 2020)	AWID	IMPACT, AE	98,22	-	-	-	-
ÖNERİLEN SİSTEM	**BOT-IOT**	**Korelasyon Hesaplama, Smote, LSTM,**	**99,38**	**51,13**	**99,38**	**52,07**	**0,62**
(Özer, İskefiyeli & Azimjonov, 2021)	BOT-IOT	Rastgele Orman, Adaboost	99	-	-	-	-
(Filus, Domańska & Gelenbe, 2021)	BOT-IOT	Rastgele Sinir Ağları	96,09	-	-	-	-
(Saba, Rehman, Sadad, Kolivand & Bahaj, 2022)	BOT-IOT	CNN	95,55	-	-	-	-
(Selvan, Azees, Vinodkumar & Parthasarathy, 2022)	BOT-IOT	RIDESOA, DNFN, SSSA	-	86,76	90,14	86,76	-
(Akgun, Hizal & Cavusoglu, 2022)	CIC-DDOS-2019	DNN, CNN, LSTM	99,99	-	-	-	-
(Ren, Yuan, Zhang, Shi & Huang, 2023)	CIC-DDOS-2019	CANET, EŞİTLEME	99,58	-	-	-	-

Çizelge 6.11. Benzer Çalışmalar İle Karşılaştırmalı Sonuçlar 2

Çalışma	Veri Seti	Kullanılan Yöntemler	Doğruluk	Kesinlik	Hassasiyet	F1	FPR
ÖNERİLEN SİSTEM	**CIC-DDOS-2019**	**Korelasyon Hesaplama, Smote, LSTM, GRU, CNN**	**98,40**	**99,10**	**98,40**	**98,70**	**1,60**
(Ren, Yuan, Zhang, Shi & Huang, 2023)	CIC-IDS-2017	CANET, EŞİTLEME	99,88	-	-	-	-
ÖNERİLEN SİSTEM	**CIC-IDS-2017**	**Korelasyon Hesaplama, Smote, LSTM, GRU, CNN**	**99,60**	**99,40**	**99,60**	**99,60**	**0,40**
(Dao & Lee, 2022)	CIC-IDS-2017	SCPP	95,58	-	-	-	-
(Hnamte & Hussain, 2023)	CIC-IDS-2018	CNN, BILSTM	100	-	-	-	-
(Mayuranathan, Saravanan, Muthusenthil & Samydurai, 2022)	CIC-IDS-2018	EOS-IDS, CRDO, DKNN	97,12	95,65	98,18	96,91	-
ÖNERİLEN SİSTEM	**CIC-IDS-2018**	**Korelasyon Hesaplama, Smote, LSTM, GRU, CNN**	**96,50**	**99,40**	**96,50**	**97,90**	**0,20**

Çizelge 6.12. Benzer Çalışmalar İle Karşılaştırmalı Sonuçlar 3

Çalışma	Veri Seti	Kullanılan Yöntemler	Doğruluk	Kesinlik	Hassasiyet	F1	FPR
ÖNERİLEN SİSTEM	**IOT-23**	**Korelasyon Hesaplama, Smote, LSTM, GRU, CNN**	**99,80**	**99,80**	**99,80**	**99,80**	**0,20**
(Kumar, Shridhar, Swaminathan & Lim, 2022)	IOT-23	Rastgele Orman, SVM	99,00	-	99,00	99,00	-
ÖNERİLEN SİSTEM	**N-BAIOT**	**Korelasyon Hesaplama, Smote, LSTM, GRU, CNN**	99,95	99,88	99,95	99,92	0,05
(Ali vd., 2024a)	N-BAIOT	LSTM, AE, MLP	99,77	99,92	99,83	99,87	-
(Pooja & Shrinivasacharya, 2021)	UNSW-NB15	BILSTM	99,67				-
(Ali vd., 2024a)	UNSW-NB15	LSTM, AE, MLP	99,67	99,62	99,78	99,70	-
ÖNERİLEN SİSTEM	**UNSW-NB15**	**Korelasyon Hesaplama, Smote, LSTM, GRU, CNN**	**98,20**	**98,30**	**98,20**	**98,20**	**1,90**
(Andrade-Arenas & Ramos-Romero, 2020)	UNSW-NB15	AD-IoT, Rastgele Orman	-	98,00	98,00	98,00	-
(Altunay & Albayrak, 2023)	UNSW-NB15	CNN+LSTM	93,21	-	-	-	-

6.3. Gelecek Çalışmalar İçin Öneriler

Bu bölümde ileride geliştirilecek saldırı tespit sistemlerinin daha başarılı olması konusunda öneriler yer almaktadır.

Saldırı tespit sistemlerinde, daha fazla çeşitlilik sağlayacak, farklı ağ protokollerini ve saldırı türlerini içeren yeni veri setlerinin kullanılması faydalı olabileceği düşünülmektedir. Özellikle, gerçek zamanlı veri setlerinin saldırı tespit sistemlerine dâhil edilmesinin, modelin pratik uygulamalardaki başarısını test etmek açısından önemli olduğu değerlendirilmektedir.

Saldırı tespit sistemleri geliştirirken veri setlerinin ön işleme adımlarına daha fazla ağırlık verilerek veriler arasındaki korelasyonlara göre veya bilgi kazancına göre işlenecek özelliklerin belirlenmesinin başarıyı artırabileceği öngörülmektedir.

Modellerin bilinmeyen saldırıları tespit etmek için denetimsiz öğrenme veya yarı denetimli öğrenme yöntemleriyle geliştirilmesinin faydalı olabileceği düşünülmektedir. Bu sayede, yeni ve tanımlanmamış saldırılara karşı modelin daha başarılı olabileceği değerlendirilmektedir.

Nesnelerin İnterneti cihaz sayısı her geçen gün arttığı ve her geçen gün daha fazla hayati faaliyetlerin Nesnelerin İnterneti cihazlarıyla gerçekleştirildiği için bu alana saldırılar sürekli olarak artmaktadır. Bu nedenle modelin de sık sık güncellenmesinin yararlı olabileceği düşünülmektedir.

KAYNAKLAR

A Simulation Platform. IEEE Communications Letters. (2013). A Simulation Platform for ZigBee-UMTS Hybrid Networks. IEEE Communications Letters, 17(2).

Abdelhalim, A. & Traore, I. (2009). A new method for learning decision trees from rules. In Proceedings of the 2009 International Conference on Machine Learning and Applications (ss. 25–30).

Abdul Hafeez, P., Singh, G., Singh, J., Prabha, C. & Verma, A. (2022). IoT in agriculture and healthcare: Applications and challenges. 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, ss. 446-450, doi: 10.1109/ICOSEC54921.2022.9952061.

Abeywardana, K. Y., Pfluegel, E. & Tunnicliffe, M. J. (2016). A layered defense mechanism for a social engineering aware perimeter. 2016 SAI Computing Conference (SAI), London, UK, ss. 1054-1062. doi:10.1109/SAI.2016.7556108.

Abhishek, S., Ravindran, R., A.T. & S.V. (2023). AI-Driven Deep Structured Learning for Cross-Site Scripting Attacks. In Proceedings of the 2023 IEEE International Conference on Innovative Data Communication Technologies and Application (ICIDCA) (ss. 701–706).

Aboosh, O.Sh.A. & Aldabbagh, O.A.I. (2021). Android Adware Detection Model Based on Machine Learning Techniques. In Proceedings of the 2021 International Conference on Computing and Communications Applications and Technologies (I3CAT) (ss. 1–7).

Ahemd, M. M., Shah, M. A. & Wahid, A. (2017). IoT security: A layered approach for attacks & defenses. 2017 International Conference on Communication Technologies (ComTech), Rawalpindi, Pakistan, ss. 104-110, doi: 10.1109/COMTECH.2017.8065757.

Ahmad, A. & Brown, G. (2014). Random Projection Random Discretization Ensembles—Ensembles of Linear Multivariate Decision Trees. IEEE Transactions on Knowledge and Data Engineering, 26(5), 1225–1234.

Aisopos, F., Litke, A., Kardara, M., Tserpes, K., Campo, P.M. & Varvarigou, T. (2019). Social network services for innovative smart cities: the RADICAL platform approach. Journal of Smart Cities, 2, 1–15.

Ait Skourt, B., El Hassani, A., & Majda, A. (2022). Mixed-pooling-dropout for convolutional neural network regularization. Journal of King Saud University - Computer and Information Sciences, 34(8, Part A), 4756–4762. https://doi.org/10.1016/j.jksuci.2021.05.001

Akgun, D., Hizal, S. & Cavusoglu, U. (2022). A new DDoS attacks intrusion detection model based on deep learning for cybersecurity. Computers & Security, 118, 102748. https://doi.org/10.1016/j.cose.2022.102748.

Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. Journal of Network and Computer Applications, 88, 10-28. https://doi.org/10.1016/j.jnca.2017.04.002

Al-Amiedy, T. A., Anbar, M., Belaton, B., Kabla, A. H. H., Hasbullah, I. H. & Alashhab, Z. R. (2022). A systematic literature review on machine and deep learning approaches for detecting attacks in RPL-based 6LoWPAN of Internet of Things. Sensors, 22(9), 3400, doi:10.3390/s22093400.

Aldawood, H. & Skinner, G. (2020). An advanced taxonomy for social engineering attacks. International Journal of Computer Applications, 177(30), 1-11. doi:10.5120/ijca2020919744.

Alenzi, K.F. & Abbas, O.A.B. (2022). A Defensive Framework for Reflected XSS in Client-Side Applications. Journal of Web Engineering, 21(7), 2209–2230.

Ali, M., Shahroz, M., Mushtaq, M.F., Alfarhood, S., Safran, M. & Ashraf, I. (2024b). Hybrid machine learning model for efficient botnet attack detection in IoT environment. IEEE Access, 12. doi:10.1109/ACCESS.2024.3376400

Ali, S., Ghazal, R., Qadeer, N., Saidani, O., Alhayan, F., Masood, A., Saleem, R., Khan, M.A. & Gupta, D. (2024a). A novel approach of botnet detection using hybrid

deep learning for enhancing security in IoT networks. Alexandria Engineering Journal, 103(1), 88–97. https://doi.org/10.1016/j.aej.2024.05.113

Ali, T.O.M., Awadelseed, O.S.A. & Eldewahi, A.E.W. (2016). Random Multiple Layouts: Keylogger Prevention Technique. In SGCAC Conference Proceedings.

Al-jabery, K. K., Obafemi-Ajayi, T., Olbricht, G. R., & Wunsch, D. C., II. (2020a). Clustering algorithms. In K. K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, & D. C. Wunsch II (Eds.), Computational learning approaches to data analytics in biomedical applications (ss. 29–100). Academic Press. https://doi.org/10.1016/B978-0-12-814482-4.00003-6

Al-jabery, K. K., Obafemi-Ajayi, T., Olbricht, G. R., & Wunsch, D. C., II. (2020b). Selected approaches to supervised learning. In K. K. Al-jabery, T. Obafemi-Ajayi, G. R. Olbricht, & D. C. Wunsch II (Eds.), Computational learning approaches to data analytics in biomedical applications (ss. 101–123). Academic Press. https://doi.org/10.1016/B978-0-12-814482-4.00004-8

Alkhafajee, A. R., Al-Muqarm, A. M. A., Alwan, A. H. & Mohammed, Z. R. (2021). Security and performance analysis of MQTT protocol with TLS in IoT networks. 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA), Necef, Irak, ss. 206-211. https://doi.org/10.1109/IICETA51758.2021.9717495

Almaraz-Rivera, J.G., Pérez-Díaz, J.A., Cantoral-Ceballos, J.A., Botero, J.F. & Trejo, L.A. (2022). Toward the protection of IoT networks: Introducing the LATAM-DDoS-IoT dataset. IEEE Access, 10, 106909–106922.

Alshamrani, A., Myneni, S., Chowdhary, A. & Huang, D. (2019). A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. IEEE Communications Surveys & Tutorials, 21(2), 1851–1877. doi:10.1109/COMST.2019.2891891

Altunay, H. & Albayrak, Z. (2023). A hybrid CNN + LSTM-based intrusion detection system for industrial IoT networks. Engineering Science and Technology, an International Journal, 38. https://doi.org/10.1016/j.jestch.2022.101322

Andrade-Arenas, L. & Ramos-Romero, J.A. (2020). Analysis and prevention of IoT vulnerabilities by implementing a lightweight AD-IoT intrusion detection system model. In Proceedings of the 2020 IEEE Biennial Congress of Argentina (ARGENCON). doi:10.1109/ARGENCON49523.2020.9505497

Anthony, P., Ay, B. & Aydin, G. (2021). A Review of Face Anti-spoofing Methods for Face Recognition Systems. In Proceedings of the IEEE International Conference on Innovations in Intelligent Systems and Applications (INISTA).

Aridas, C.K., Karlos, S., Kanas, V.G., Fazakis, N. & Kotsiantis, S.B. (2020). Uncertainty Based Under-Sampling for Learning Naive Bayes Classifiers Under Imbalanced Data Sets. IEEE Access, 8, 2122–2133. doi:10.1109/ACCESS.2019.2961784

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. arXiv preprint arXiv:1701.07875. https://arxiv.org/abs/1701.07875

Asiri, S., Xiao, Y., Alzahrani, S., Li, S., & Li, T., "A Survey of Intelligent Detection Designs of HTML URL Phishing Attacks," IEEE Access, vol. 11, ss. 6421–6442, Jan. 2023. DOI: 10.1109/ACCESS.2023.3237798.

Askar, A. (2019). Internet of Things (IoT) Device Registration. Google Patents.

Author, K. (2022). Limitations and Prospects in Decision Tree Learning. Artificial Intelligence Review, 39(1), 42–55.

Bailey, M., Cooke, E., Jahanian, F. & Watson, D. (2005). The Blaster worm: Then and now. IEEE Security & Privacy, 3(4), 26-31. doi:10.1109/MSP.2005.106.

Bajracharya, C. (2022). Performance Evaluation for Secure Communications in Mobile Internet of Vehicles With Joint Reactive Jamming and Eavesdropping Attacks. IEEE Transactions on Intelligent Transportation Systems, 23(11).

Balakrishnan, S., Wang, P., Bhuyan, A., & Sun, Z. (2019). Modeling and analysis of eavesdropping attack in 802.11ad mmWave wireless networks. IEEE Access, PP, 1-1. https://doi.org/10.1109/ACCESS.2019.2919674

Bao, C., Forte, D. & Srivastava, A. (2016). On Reverse Engineering-Based Hardware Trojan Detection. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 35(1), 49–59.

Barros, R.C., Basgalupp, M.P., de Carvalho, A.C.P.L.F. & Freitas, A.A. (2012). A survey of evolutionary algorithms for decision-tree induction. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 42(3), 291–312.

Batmaz, B. & Dogan, A. (2021). CoAP Acceleration on FPSoC for Resource Constrained Internet of Things Devices. IEEE Internet of Things Journal, 8(24), 17790–17801.

Bhat, P., Namratha, H.J. & Mohana. (2020). Cyber Security Testbed: Keyloggers and Data Visualization on Keyloggers- A Case Study. International Conference Proceedings.

Bhayo, J., Hameed, S. & Shah, S.A. (2020). An efficient counter-based DDoS attack detection framework leveraging software defined IoT (SD-IoT). IEEE Access, 8, 221612–221623.

Bhowal, S., Dutta, S. R. & Mitra, S. (2017). An efficient reduced set brute force attack technique for a particular steganographic tool using vername algorithm. 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 1-4, doi:10.1109/ICIIP.2017.8313698.

Bhunia, S., Hsiao, M. S., Banga, M. & Narasimhan, S. (2014). Hardware Trojan Attacks: Threat Analysis and Countermeasures. Proceedings of the IEEE, 102(8), 1229-1247, doi:10.1109/JPROC.2014.2334493.

Blancaflor, E., Araullo, E.E., Corcuera, J.A., Rivera, J.R. & Velarde, L.N. (2023). Vulnerability Assessment on Cross-site Scripting Attack in a Simulated E-commerce Platform Using BeEF and XSStrike. In Proceedings of the 2023 13th International Conference on Software Technology and Engineering (ICSTE) (ss. 1–6).

Bouke, M.A., Abdullah, A., Alshatebi, S.H., Abdullah, M.T. & El Atigh, H. (2023). An intelligent DDoS attack detection tree-based model using Gini index feature selection method. Microprocessors and Microsystems, 98, 104823. https://doi.org/10.1016/j.micpro.2023.104823

Bruschi, D., Di Pasquale, A., Ghilardi, S., Lanzi, A. & Pagani, E. (2022). A Formal Verification of ArpON – A Tool for Avoiding Man-in-the-Middle Attacks in

Ethernet Networks. IEEE Transactions on Dependable and Secure Computing, 19(6), 4082–4094. doi:10.1109/TDSC.2021.3118448.

Bulić, P., Kojek, G. & Biasizzo, A. (2019). Data transmission efficiency in Bluetooth Low Energy versions. Sensors, 19(17), 3746, doi:10.3390/s19173746.

Butt, M. A., Ajmal, Z., Khan, Z. I., Idrees, M. & Javed, Y. (2022). An in-depth survey of bypassing buffer overflow mitigation techniques. Applied Sciences, 12(13), 6702. doi:10.3390/app12136702.

Cabello-Solorzano, K., Ortigosa de Araujo, I., Peña, M., Correia, L., & Tallón-Ballesteros, A. J. (2023). The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis. In P. García Bringas et al. (Eds.), 18th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2023) (Lecture Notes in Networks and Systems, Vol. 750). Springer, Cham. https://doi.org/10.1007/978-3-031-42536-3_33

Capuano, N., Fenza, G., Loia, V. & Stanzione, C. (2022). Explainable artificial intelligence in cybersecurity: A survey. IEEE Access, 10, 93575-93600. https://doi.org/10.1109/ACCESS.2022.3204171

Centenaro, M. & Vangelista, L. (2016). Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios. IEEE Wireless Communications, 23(5), 60–67.

Chamasemani, F. F. & Singh, Y. P. (2011). Multi-class Support Vector Machine (SVM) classifiers -- An application in hypothyroid detection and classification. 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications, Penang, Malezya, ss. 351-356. https://doi.org/10.1109/BIC-TA.2011.51

Chen, H. H., Lu, H. H. S., Weng, W. H. & Lin, Y. H. (2023). Developing a machine learning algorithm to predict the probability of medical staff work mode using human-smartphone interaction patterns: Algorithm development and validation study. Journal of Medical Internet Research, 25, e48834. doi:10.2196/48834.

Cheng, H.-H. (2023). Understanding Protection Motivation for Identity Theft in Information Security: An Extension of The Protection Motivation Theory with Anxiety. In Proceedings of the 18th Iberian Conference on Information Systems and Technologies (CISTI).

Chumachenko, K., Iosifidis, A., & Gabbouj, M. (2022). Feedforward neural networks initialization based on discriminant learning. Neural Networks, 146, 220–229. https://doi.org/10.1016/j.neunet.2021.11.020

Cisco. (2023). Cisco annual internet report (2018–2023) white paper. Cisco. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

Coetzee, L., Oosthuizen, D. & Mkhize, B. (2018). An analysis of CoAP as transport in an Internet of Things environment. 2018 IST-Africa Week Conference (IST-Africa), Gaborone, Botswana, ss. 1-7.

Cornel - Cristian, A., Gabriel, T., Arhip-Calin, M. & Zamfirescu, A. (2019). Smart home automation with MQTT. 2019 54th International Universities Power Engineering Conference (UPEC), Bükreş, Romanya, ss. 1-5. https://doi.org/10.1109/UPEC.2019.8893617

Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). L2 regularization for learning kernels. arXiv. https://doi.org/10.48550/arXiv.1205.2653

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), ss. 21-27. https://doi.org/10.1109/TIT.1967.1053964.

Da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R. & de Albuquerque, V.H.C. (2019). Internet of Things: a survey on machine learning-based intrusion detection approaches. Computer Networks, 151, 147–157.

Dao, T.-N. & Lee, H.J. (2022). Stacked Autoencoder-Based Probabilistic Feature Extraction for On-Device Network Intrusion Detection. IEEE Internet of Things Journal, 9(16), 14438–14450. https://doi.org/10.1109/JIOT.2021.3078292

Das, L., Raman Chandan, R., Kaur, P., Singh, A., Rana, A. & Shivhare, B. D. (2023). Advancements in wireless network technologies for enabling the IoT: A

comprehensive review. 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, ss. 807-814, doi: 10.1109/IC3I59117.2023.10397952.

Davahli, A., Shamsi, M. & Abaei, G. (2020). A lightweight anomaly detection model using SVM for WSNs in IoT through a hybrid feature selection algorithm based on GA and GWO. Journal of Computing and Security, 7(1), 63–79. doi:10.22108/jcs.2020.119468.1033

De Boer, P. T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. Annals of Operations Research, 134(1), 19–67. https://doi.org/10.1007/s10479-005-5724-z

Dina, A.S., Siddique, A.B. & Manivannan, D. (2023). A deep learning approach for intrusion detection in Internet of Things using focal loss function. Internet of Things, 22, 100699. https://doi.org/10.1016/j.iot.2023.100699

Dizdarević, J., Carpio, F., Jukan, A. & Masip-Bruin, X. (2019). A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration. ACM Computing Surveys (CSUR), 51, 116.

Dong, S. & Sarem, M. (2020). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. IEEE Access, 8, 5039–5050.

Duangphasuk, S., Duangphasuk, P. & Thammarat, C. (2020). Review of Internet of Things (IoT): Security Issue and Solution. In Proceedings of the 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (ss. 559–562). doi:10.1109/ECTI-CON49241.2020.9157904

El Sayed, A.I., Abdelaziz, M., Hussein, M. & Elbayoumy, A.D. (2024). DDoS mitigation in IoT using machine learning and blockchain integration. IEEE Networking Letters, 6(2), 152–155.

El Zein, Y., Lemay, M. & Huguenin, K. (2024). PrivaTree: Collaborative Privacy-Preserving Training of Decision Trees on Biomedical Data. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 21(1), 1–14.

Elhefnawy, R., Abounaser, H. & Badr, A. (2020). A hybrid nested genetic-fuzzy algorithm framework for intrusion detection and attacks. IEEE Access, 8. doi:10.1109/ACCESS.2020.2996226

En, V.T. & Selvarajah, V. (2022). Cross-Site Scripting (XSS). In Proceedings of the 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC) (ss. 1–6).

Facchini, S., Giorgi, G., Saracino, A. & Dini, G. (2020). Multi-level Distributed Intrusion Detection System for an IoT based Smart Home Environment. In Proceedings of ICISSP (ss. 705–712).

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log-dönüşümü ve bunun veri analizi üzerindeki etkileri. Shanghai Archives of Psychiatry, 26(2), 105–109. https://doi.org/10.3969/j.issn.1002-0829.2014.02.009

Filus, K., Domańska, J. & Gelenbe, E. (2021). Random Neural Network for Lightweight Attack Detection in the IoT. In Modelling, Analysis, and Simulation of Computer and Telecommunication Systems. Springer. https://doi.org/10.1007/978-3-030-68110-4_5

Fix, E. & Hodges, J. L. (1951). Discriminatory analysis, nonparametric discrimination: consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field.

Gao, J., Li, L., Kong, P., Bissyandé, T. F. & Klein, J. (2019). Should you consider adware as malware in your study? 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER), Hangzhou, China, ss. 604-608. doi:10.1109/SANER.2019.8668010.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Ghosh, D. & Cabrera, J. (2022). Enriched random forest for high dimensional genomic data. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(5), 2817–2828. doi:10.1109/TCBB.2021.3089417

Gomes, T., Salgado, F., Pinto, S., Cabral, J. & Tavares, A. (2018). A 6LoWPAN Accelerator for Internet of Things Endpoint Devices. IEEE Internet of Things Journal, 5(1), 371–382.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (ss. 2672-2680). https://papers.nips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

Guan, Y. & Ge, X. (2017). Distributed Secure Estimation Over Wireless Sensor Networks Against Random Multichannel Jamming Attacks. IEEE Access, 5, ss. 10858-10870. https://doi.org/10.1109/ACCESS.2017.2713807.

Gupta, S., Singhal, A. & Kapoor, A. (2016). A literature survey on social engineering attacks: Phishing attack. 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, ss. 537-540. doi:10.1109/CCAA.2016.7813778.

Halimaa, A. A. & Sundarakantham, K. (2019). Machine learning based intrusion detection system. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, ss. 916-920, doi: 10.1109/ICOEI.2019.8862784.

Haykin, S. (2008). Neural Networks and Learning Machines. Pearson Prentice Hall New Jersey USA 936 pLinks, 3. https://doi.org/978-0131471399.

He, D., Gu, H., Zhu, S., Chan, S. & Guizani, M. (2024). A Comprehensive Detection Method for the Lateral Movement Stage of APT Attacks. IEEE Internet of Things Journal, 11(5), 8440–8447. doi:10.1109/JIOT.2023.3322412

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J. & Scholkopf, B. (1998). Support vector machines. IEEE Intelligent Systems and Their Applications, 13(4), ss. 18-28. https://doi.org/10.1109/5254.708428

Heaton, J. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(3–4), 305–307. https://doi.org/10.1007/s10710-017-9314-z

Hnamte, V. & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. Telematics and Informatics Reports, 10, 100053. https://doi.org/10.1016/j.teler.2023.100053

Hou, Y., He, H., Jiang, X., Chen, S. & Yang, J. (2023). Deep-Reinforcement-Learning-Aided Loss-Tolerant Congestion Control for 6LoWPAN Networks. IEEE Internet of Things Journal, 10(21), 19125–19134.

Hu, W., Gao, J., Wang, Y., Wu, O. & Maybank, S. (2014). Online Adaboost-Based Parameterized Methods for Dynamic Distributed Network Intrusion Detection. IEEE Transactions on Cybernetics, 44(1), 66-82, doi:10.1109/TCYB.2013.2247592.

Hua, J., Zhou, Z. & Zhong, S. (2021). Flow misleading: Worm-hole attack in software-defined networking via building in-band covert channel. IEEE Transactions on Information Forensics and Security, 16, 1029–1030. doi:10.1109/TIFS.2020.3013093

Huo, F., Mitran, P. & Gong, G. (2016). Analysis and Validation of Active Eavesdropping Attacks in Passive FHSS RFID Systems. IEEE Transactions on Information Forensics and Security, 11(7).

Hyman, B., Alisha, Z. & Gordon, S. (2019). Secure controls for smart cities. Applied Intelligence in Transportation Systems and Smart Buildings.

Ideses, I. & Neuberger, A. (2014). Adware Detection and Privacy Control in Mobile Devices. In Proceedings of the 2014 IEEE 28th Convention of Electrical and Electronics Engineers in Israel, 1–5.

Iqbal, A., Khan, S. U. & Niazi, M. (2024). Advancing database security: A comprehensive systematic mapping study of potential challenges. Wireless Networks, 30, 6399–6426. doi:10.1007/s11276-023-03436-z.

Islam, S. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K. S. (2015). The internet of things for health care: A comprehensive survey. IEEE Access, 3, 678-708. https://doi.org/10.1109/ACCESS.2015.2437951

Jourdan, P. & Stavrou, E. (2019). Towards designing advanced password cracking toolkits: optimizing the password cracking process. In Proceedings of the Adjunct

Publication of the 27th Conference on User Modeling, Adaptation and Personalization (ss. 203–208). ACM.

Kantardzic, M. (2011). Data mining: Concepts, models, methods, and algorithms (2nded). Hoboken, New Jersey, US: Wiley.

Kareem, F. Q., Ameen, S. Y., Salih, A. A., Ahmed, D. M., Kak, S. F., Yasin, H. M., Ibrahim, I. M., Ahmed, A. M., Rashid, Z. N. & Omar, N. (2021). SQL injection attacks prevention system technology: Review. Asian Journal of Research in Computer Science, 10(3), 13-32. doi:10.9734/ajrcos/2021/v10i330242.

Kasongo, S.M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199, 113–125. https://doi.org/10.1016/j.comcom.2022.12.010

Khanna, A. & Kaur, S. (2019). Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. Computers and Electronics in Agriculture, 157, 218–231.

Khirirat, S., Feyzmahdavian, H. R., & Johansson, M. (2017). Mini-batch gradient descent: Faster convergence under data sparsity. 2017 IEEE 56th Annual Conference on Decision and Control (CDC), 2880–2887. https://doi.org/10.1109/CDC.2017.8264077

Khosravi, M. & Ladani, B. T. (2020). Alerts correlation and causal analysis for APT-based cyber attack detection. IEEE Access, 8, 162642-162656. doi:10.1109/ACCESS.2020.3021499.

Khosravi, M. & Ladani, B.T. (2020). Alerts Correlation and Causal Analysis for APT Based Cyber Attack Detection. IEEE Access, 8.

Kilic, I. & Yaman, O. (2024). A New Approach to Spyware Classification Using a 1D-Gaussian Filter on Network Packets. In Proceedings of the 8th International Artificial Intelligence and Data Processing Symposium (IDAP).

Kilic, I. & Yaman, O. (2024). Classification of Spyware from Network Packets with Decision Trees Using Recursive Feature Elimination (RFE).

Kim, Y. & Kyung, Y. (2024). AoI-Aware Retained Message Policy in MQTT-Based IoT Networks. IEEE Sensors Journal, 24(21).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv. https://doi.org/10.48550/arXiv.1412.6980

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. Mechanical Systems and Signal Processing, 151, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kivinen, J., Smola, A. J. & Williamson, R. C. (2004). Online learning with kernels. IEEE Transactions on Signal Processing, 52(8), ss. 2165-2176. https://doi.org/10.1109/TSP.2004.830991

Kosugi, A., Teranishi, K. & Kogiso, K. (2024). Experimental validation of the attack-detection capability of encrypted control systems using Man-in-the-Middle attacks. IEEE Access, 12, 10535–10544. doi:10.1109/ACCESS.2024.3353289

Krishnan, A. J. & Binu, G. S. (2017). Energy efficient tree construction for ZigBee router network. 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, ss. 746-751, doi: 10.1109/CESYS.2017.8321182.

Kumar, A., Shridhar, M., Swaminathan, S. & Lim, T.J. (2022). Machine learning-based early detection of IoT botnets using network-edge traffic. Computers & Security, 117, 102693. https://doi.org/10.1016/j.cose.2022.102693

Kumar, A., Somani, G. & Agarwal, M. (2024). Comparing HAProxy scheduling algorithms during the DDoS attacks. IEEE Networking Letters, 6(2), 139–142.

Kumari, A., Gupta, D. & Uppal, M. (2024). Enhanced Brute Force Attack Detection in Remote Access Security: Integrating ANN and SVM. 2024 Asia Pacific Conference on Innovation in Technology (APCIT), Mysore, India, 1-5, doi:10.1109/APCIT62007.2024.10673550.

Lang, L. & Zilong, X. (2022). Auxiliary Technology of Firewall Based on Computer Virus Detection. In Proceedings of the 2022 International Conference on Artificial Intelligence in Everything (AIE) (ss. 12–14).

Lashkari, A. H. (2018), Cicflowmeter-v4.0 Anomali tespiti için ağ trafik analiz aracı, URL: https://github.com/iscx/cicflowmeter, Son Erişim Tarihi: 17.04.2023.

Lee, J., Lee, Y., Lee, D., Kwon, H., & Shin, D., "Classification of Attack Types and Analysis of Attack Methods for Profiling Phishing Mail Attack Groups," IEEE Access, vol. 9, ss. 80866–80879, May 2021. DOI: 10.1109/ACCESS.2021.3084897.

Lee, J., Sim, M.K. & Hong, J.-S. (2024). Assessing decision tree stability: a comprehensive method for generating a stable decision tree. IEEE Access, 12, 90061–90072.

Lee, S.J., Yoo, P.D., Asyhari, A.T., Jhi, Y., Chermak, L., Yeun, C.Y. & Taha, K. (2020). IMPACT: Impersonation Attack Detection via Edge Computing Using Deep Autoencoder and Feature Abstraction. IEEE Access, 8, 65520–65532. https://doi.org/10.1109/ACCESS.2020.2985089

Letafati, M., Behroozi, H., Khalaj, B.H. & Jorswieck, E.A. (2022). Hardware-impaired PHY secret key generation with Man-in-the-Middle adversaries. IEEE Wireless Communications Letters, 11(4), 856–859. doi:10.1109/LWC.2022.3147952

Li, J., Sisodia, D. & Stafford, S. (2023). On the detection of smart, self-propagating internet worms. IEEE Transactions on Dependable and Secure Computing, 20(4), 3051–3053. doi:10.1109/TDSC.2022.3194127

Li, P., Pan, L., Chen, F., Hoang, T. & Wang, R. (2023). TOTPAuth: A Time-based One Time Password Authentication Proof-of-Concept against Metaverse User Identity Theft. In Proceedings of the 2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom).

Liu, M., Zhang, B., Chen, W. & Zhang, X. (2019). A Survey of Exploitation and Detection Methods of XSS Vulnerabilities. IEEE Access, 7, 182003–182013.

Liu, Q., Zhao, P. & Chen, F. (2019). A Hardware Trojan Detection Method Based on Structural Features of Trojan and Host Circuits. IEEE Access, 7, 44632–44644. doi:10.1109/ACCESS.2019.2908088

Liu, Y., Yang, C., Jiang, L., Xie, S. & Zhang, Y. (2019a). Intelligent edge computing for IoT-based energy management in smart cities. IEEE Network, 33, 111–117.

Lu, D. & Liu, L. (2023). Research on Cross-site Scripting Attack Detection Technology Based on Few-shot Learning. IEEE ITNEC, 6(3), 1425–1432.

Lu, K., Zhang, W. & Sun, B. (2017). Multidimensional Data-Driven Life Prediction Method for White LEDs Based on BP-NN and Improved-Adaboost Algorithm. IEEE Access, 5, 21660-21668, doi:10.1109/ACCESS.2017.2761802.

Ma, Y. & Ding, X. (2005). Real-Time Multi-View Face Detection and Pose Estimation Based on Cost-Sensitive AdaBoost. Tsinghua Science & Technology, 10(2), 152-157, doi:10.1016/S1007-0214(05)70047-X.

Manjula, R., & Vijaya, M. S. (2020). Deep neural network for evaluating web content credibility using Keras sequential model. In T. Sengodan, M. Murugappan, & S. Misra (Eds.), Advances in electrical and computer technologies (Lecture Notes in Electrical Engineering, Vol. 672, ss. 13–24). Springer, Singapore. https://doi.org/10.1007/978-981-15-5558-9_2

Mayuranathan, M., Saravanan, S.K., Muthusenthil, B. & Samydurai, A. (2022). An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique. Advances in Engineering Software, 173, 103236. https://doi.org/10.1016/j.advengsoft.2022.103236

Mei, Y., Han, W., Li, S., Lin, K., Tian, Z. & Li, S. (2024). A Novel Network Forensic Framework for Advanced Persistent Threat Attack Attribution Through Deep Learning. IEEE Transactions on Intelligent Transportation Systems, 25(9), 12131–12140. doi:10.1109/TITS.2024.3360260

Merhav, N. & Cohen, A. (2020). Universal Randomized Guessing With Application to Asynchronous Decentralized Brute–Force Attacks. IEEE Transactions on Information Theory, 66(1).

Mienye, I.D. & Jere, N. (2024). A survey of decision trees: concepts, algorithms, and applications. IEEE Access, 12, 86716–86735.

Modiri, E., Azmoodeh, A., Dehghantanha, A., Newton, D.E., Parizi, R.M. & Karimipour, H. (2022). Fuzzy pattern tree for edge malware detection and categorization in IoT.

Mosenia, A., & Jha, N. K. (2016). A comprehensive study of security of Internet-of-Things. IEEE Transactions on Emerging Topics in Computing, 5(4), 586-602. https://doi.org/10.1109/TETC.2016.2606384

Msadek, N., Soua, R. & Engel, T. (2019). IoT device fingerprinting: machine learning based encrypted traffic analysis. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC) (ss. 1–8). IEEE.

Mukhopadhyay, S. C., Tyagi, S. K. S., Suryadevara, N. K., Piuri, V., Scotti, F. & Zeadally, S. (2021). Artificial intelligence-based sensors for next generation IoT applications: A review. IEEE Sensors Journal, 21(22), 24920-24932, doi: 10.1109/JSEN.2021.3055618.

Muzammil, M.B., Bilal, M., Ajmal, S., Shongwe, S.C. & Ghadi, Y.Y. (2024). Unveiling vulnerabilities of web attacks considering Man in the Middle attack and session hijacking. IEEE Access, 12, 6365–6375. doi:10.1109/ACCESS.2024.3350444

Naik, K. P. & Joshi, U. R. (2017). Performance analysis of constrained application protocol using Cooja simulator in Contiki OS. 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kerala, Hindistan, ss. 547-550. https://doi.org/10.1109/ICICICT1.2017.8342622

Nandgaonkar, S., Attar, V.Z. & Sinha, P.K. (2014). Efficient decision tree construction for classifying numerical data. IEEE Transactions on Knowledge and Data Engineering, 26(5), 1225–1234.

Naqvi, B., Perova, K., Farooq, A., Makhdoom, I., Oyedeji, S. & Porras, J. (2023). Mitigation strategies against phishing attacks: A systematic literature review. Computers & Security, 132, 103387. doi:10.1016/j.cose.2023.103387.

Nugroho, K., Hendrawan & Iskandar (2024). Overfitting Defeat with Dropout for Image Classification in Convolutional Neural Networks. 2024 10th International Conference on Wireless and Telematics (ICWT), Batam, Endonezya, 1-5. https://doi.org/10.1109/ICWT62080.2024.10674725.

Nyang, D., Mohaisen, A. & Kang, J. (2014). Keylogging-Resistant Visual Authentication Protocols. IEEE Transactions on Mobile Computing.

Obaidat, M. S., Nicopolitidis, P., & Zarai, F. (2020). Security and Privacy in Internet of Things (IoT): Models, Algorithms, and Implementations. Springer.

Okesola, J. O., Ogunbanwo, A. S., Owoade, A., Olorunnisola, E. O. & Okokpuji, K. (2023). Securing web applications against SQL injection attacks: A parameterised query perspective. 2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG), Omu-Aran, Nigeria, ss. 1-6, doi: 10.1109/SEB-SDG57117.2023.10124613.

Ortolani, S., Giuffrida, C. & Crispo, B. (2013). Unprivileged Black-Box Detection of User-Space Keyloggers. IEEE Transactions on Dependable and Secure Computing.

Özer, E., İskefiyeli, M. & Azimjonov, J. (2021). Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset. International Journal of Distributed Sensor Networks, 17(10). https://doi.org/10.1177/15501477211052202

Patil, A., Laturkar, A., Athawale, S. V., Takale, R. & Tathawade, P. (2017). A multilevel system to mitigate DDOS, brute force and SQL injection attack for cloud security. 2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC), Indore, India, 1-7, doi:10.1109/ICOMICON.2017.8279028.

Pereira, D.S., De Morais, M.R., Nascimento, L.B.P., Alsina, P.J. & Santos, V.G. (2020). Zigbee Protocol-Based Communication Network for Multi-Unmanned Aerial Vehicle Networks. IEEE Access.

Pérez-Díaz, J.A., Valdovinos, I.A., Choo, K.K.R. & Zhu, D. (2020). A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access, 8, 155859–155874.

Petäjäjärvi, J., Mikhaylov, K., Hämäläinen, M. & Iinatti, T. (2017). Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. International Journal of Distributed Sensor Networks, 13(3).

Pham Anh, S. & Nakamura, Y. (2024). A baseline investigation into the evolution and prevalence of Mirai and Hajime utilizing a network telescope. IEEE Access, 12. doi:10.1109/ACCESS.2024.3434555

Pooja, T.S. & Shrinivasacharya, P. (2021). Evaluating neural networks using Bi-Directional LSTM for network IDS (intrusion detection systems) in cybersecurity. https://doi.org/10.1016/j.gltp.2021.08.017

Prechelt, L. (1998). Early stopping - but when?. In G. B. Orr & K. R. Müller (Eds.), Neural networks: Tricks of the trade (Lecture Notes in Computer Science, Vol. 1524, ss. 55–69). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-49430-8_3

Purnama, I. P. C., Ginantra, N. L. W. S. R., Darma, I. W. A. S. & Udayana, I. P. A. E. D. (2023). Comparison of Support Vector Machine (SVM) and Linear Regression (LR) for stock price prediction. 2023 Eighth International Conference on Informatics and Computing (ICIC), Manado, Endonezya, ss. 1-6. https://doi.org/10.1109/ICIC60109.2023.10381982

Qingyu, O. & Dawei, Z. (2010). Research on Active Learning Based Computer Viruses Detection Approaches. In Proceedings of the 2010 Second International Conference on Computational Intelligence and Natural Computing (CINC) (ss. 101–102).

Qiu, Y. & Ma, M. (2017). A secure PMIPv6-based group mobility scheme for 6LoWPAN networks. 2017 IEEE International Conference on Communications (ICC), Paris, France, ss. 1-6, doi: 10.1109/ICC.2017.7997471.

Quan, W.W., Kumar, V.V., Sundaresan, S. & Sutaria, I. (2019). Data Streaming Service for an Internet-of-Things Platform. Google Patents..

Ray, S. (2019). A quick review of machine learning algorithms. 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, Hindistan, ss. 35-39. https://doi.org/10.1109/COMITCon.2019.8862451

Ren, K., Yuan, S., Zhang, C., Shi, Y. & Huang, Z. (2023). CANET: A hierarchical CNN-Attention model for Network Intrusion Detection. Computer Communications, 205, 170–181. https://doi.org/10.1016/j.comcom.2023.04.018

Researcher, L. (2020). Energy Efficiency in Bluetooth Low Energy. Journal of Network and Computer Applications, 116, 42–55.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. The Annals of Mathematical Statistics, 22(3), 400–407. https://doi.org/10.1214/aoms/1177729586

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv. https://arxiv.org/abs/1609.04747

Saba, T., Rehman, A., Sadad, T., Kolivand, H. & Bahaj, S.A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering, 99, 107810. https://doi.org/10.1016/j.compeleceng.2022.107810

Safavian, S.R. & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics, 21(3), 660–674.

Salmani, H., Tehranipoor, M. & Plusquellic, J. (2012). A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20(1), 112–125. doi:10.1109/TVLSI.2010.2093547.

Saranya, T., Sridevi, S., Deisy, C., Chung, T.D. & Khan, M.A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: a review. Procedia Computer Science, 171, 1251–1260.

Selvan, E.G.S.R., Azees, M., Vinodkumar, C.R. & Parthasarathy, G. (2022). Hybrid optimization enabled deep learning technique for multi-level intrusion detection. Advances in Engineering Software, 173, 103197. https://doi.org/10.1016/j.advengsoft.2022.103197

Seo, S. B. & Singh, D. (2018). Smart town traffic management system using LoRa and machine learning mechanism. IEEE Technology Policy and Ethics, 3(6), 1-4, doi:10.1109/NTPE.2018.9778109.

Shaashua, T.M. & Shaashua, O. (2019). Physical Environment Profiling Through Internet of Things Integration Platform. Google Patents.

Shabani, A. & Alizadeh, B. (2021). Enhancing Hardware Trojan Detection Sensitivity Using Partition-Based Shuffling Scheme. IEEE Transactions on Circuits and Systems II: Express Briefs, 68(1), 266–270.

Shah, D., Xue, Z. Y., & Aamodt, T. M. (2022). Label encoding for regression networks. arXiv. https://doi.org/10.48550/arXiv.2212.01927

Shahzad, R.K., Lavesson, N. & Johnson, H. (2011). Accurate Adware Detection Using Opcode Sequence Extraction. In Proceedings of the Sixth International Conference on Availability, Reliability and Security, 189–197.

Sharma, B., Sharma, L., Lal, C. & Roy, S. (2023). Anomaly-based network intrusion detection for IoT attacks using deep learning technique. Computers and Electrical Engineering, 107, 108626. https://doi.org/10.1016/j.compeleceng.2023.108626

Sheta, M.A., El Hadad, K.A.E.S., M, H.A. & Zaki, M. (2016). Anti-spyware Security Design Patterns. In Proceedings of the Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control.

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146-164. https://doi.org/10.1016/j.comnet.2014.11.008

Smith, L. N. (2015). Cyclical learning rates for training neural networks. arXiv. https://doi.org/10.48550/arXiv.1506.01186

Solairaj, A., Prabanand, S.C., Mathalairaj, J., Prathap, C. & Vignesh, L.S. (2019). Keyloggers Software Detection Techniques. Nadar Saraswathi College Journal.

Spachos, P. & Plataniotis, K. (2020). BLE beacons in the smart city: Applications, challenges, and research opportunities. IEEE Internet of Things Magazine, 3(1), 14-18, doi:10.1109/IOTM.0001.1900073.

Statista. (2024). Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025 (in zettabytes). Statista. Erişim adresi: https://www.statista.com/statistics/871513/worldwide-data-

created/#:~:text=The%20total%20amount%20of%20data,to%20more%20than%20394%20zettabytes.

Subramaniyaswamy, V., Manogaran, G., Logesh, R., Vijayakumar, V., Chilamkurti, N., Malathi, D. & Senthilselvan, N. (2019). An ontology-driven personalized food recommendation in IoT-based healthcare system. Journal of Supercomputing, 75, 3184–3216.

Subramanya, S.R. & Lakshminarasimhan, N. (2001). Computer Viruses. IEEE Potentials, 27(6), 16–27.

Sun, H.M., Chen, Y.H. & Lin, Y.H. (2012). oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks. IEEE Transactions on Information Forensics and Security.

Suykens, J. A. K. (2001). Nonlinear modelling and support vector machines. IMTC 2001. Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference. Rediscovering Measurement in the Age of Informatics (Cat. No.01CH 37188), Budapeşte, Macaristan, ss. 287-294, Cilt 1. https://doi.org/10.1109/IMTC.2001.928828

Süzen, A. (2023). Examining social engineering attack vector in line of data breach. Teknik Bilimler Dergisi, 13, 50-56. doi:10.35354/tbed.1310185.

Taher, M. A., Tariq, M., Behnamfar, M. & Sarwat, A. I. (2023). DC mikroşebeke mutabakat kontrolünde tekrarlama saldırısı etkisinin analizi: Kalman filtre tabanlı gözlemci ile tespit ve önleme. IEEE Access, 11, ss. 121368-121378. https://doi.org/10.1109/ACCESS.2023.3317799.

Tahir, R. & McDonald-Maier, K. (2012). Improving resilience against node capture attacks in wireless sensor networks using ICmetrics. 2012 Third International Conference on Emerging Security Technologies, Lisbon, Portugal, ss. 127-130. https://doi.org/10.1109/EST.2012.44.

Tang, J., Xu, M., Fu, S. & Huang, K. (2018). A scheduling optimization technique based on reuse in spark to defend against APT attack. Tsinghua Science and Technology, 23(5), 550–560. doi:10.26599/TST.2018.9010022

Thankappan, M., Rifà-Pous, H. & Garrigues, C. (2024). A signature-based wireless intrusion detection system framework for multi-channel Man-in-the-Middle attacks against protected Wi-Fi networks. IEEE Access, 12, 23096–23104. doi:10.1109/ACCESS.2024.3362803

Tian, J., Tan, R., Guan, X., Xu, Z. & Liu, T. (2020). Moving target defense approach to detecting Stuxnet-like attacks. IEEE Transactions on Smart Grid, 11(1), 291–292. doi:10.1109/TSG.2019.2921245

Tiwari, N. & Hubballi, N. (2023). Secure Socket Shell Bruteforce Attack Detection With Petri Net Modeling. IEEE Transactions on Network and Service Management, 20(1).

Tsimenidis, S., Lagkas, T. & Rantos, K. (2022). Deep learning in IoT intrusion detection. Journal of Network and Systems Management, 30(1), 1–40.

Vakil, A., Mirzaeian, A., Homayoun, H., Karimi, N. & Sasan, A. (2021). AVATAR: NN-Assisted Variation Aware Timing Analysis and Reporting for Hardware Trojan Detection. IEEE Access, 9, 92881–92900. doi:10.1109/ACCESS.2021.3093160

Vapnik, V. N. & Chervonenkis, A. Y. (1964). Об одном классе алгоритмов обучения распознаванию образов (On a class of algorithms of learning pattern recognition). Avtomatika i Telemekhanika, 25(6).

Vashist, A., Keats, A., Pudukotai Dinakarrao, S. M. & Ganguly, A. (2019). Securing a Wireless Network-on-Chip Against Jamming-Based Denial-of-Service and Eavesdropping Attacks. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(12), 2781-2791, doi:10.1109/TVLSI.2019.2928960.

Venkatraman, S & Muthusamy, P. & Balusa, Bhanuchander & Thangaiyan, Jayasankar & Kavithaa, G. & Sekar, Dr K R & Bharatiraja, C. (2021). Time dependent anomaly detection system for smart environment using probabilistic timed automaton. Journal of Ambient Intelligence and Humanized Computing, 1–9.

Victoria, E.C., Adebiyi, A.A. & Igbekele, E.O. (2021). Keylogger Detection: A Systematic Review. IEEE Transactions on Information Forensics and Security.

Wager, S., Wang, S., & Liang, P. (2013). Dropout training as adaptive regularization. arXiv. https://doi.org/10.48550/arXiv.1307.1493

Wagner, M. & Fröhlich, A.A. (2024). Securing Cyber-Physical Systems Against GPS Spoofing Attacks Using Confidence Attribution. In Proceedings of the IEEE.

Wan, J., Lai, C.-F., Song, H., Imran, M. & Jia, D. (2019). Software-defined industrial Internet of Things. Wireless Communications and Mobile Computing, 2019.

Wang, F., Li, Z., He, F., Wang, R., Yu, W. & Nie, F. (2019). Feature Learning Viewpoint of Adaboost and a New Algorithm. IEEE Access, 7, 149890-149899, doi:10.1109/ACCESS.2019.2947359.

Wang, H. & Hu, D. (2005). Comparison of SVM and LS-SVM for regression. 2005 International Conference on Neural Networks and Brain, Pekin, Çin, ss. 279-283. https://doi.org/10.1109/ICNNB.2005.1614615

Wang, J., Yang, Q. & Ren, D. (2011). An intrusion detection algorithm based on decision tree technology. Journal of Computer Applications, 31(4), 981–984.

Wang, X. & Yamagishi, J. (2023). Spoofed Training Data for Speech Spoofing Countermeasure Can Be Efficiently Created Using Neural Vocoders. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

Wazid, M., Katal, A., Goudar, R.H., Singh, D.P. & Tyagi, A. (2012). A Framework for Detection and Prevention of Novel Keylogger Spyware Attacks. In IEEE Conference Paper.

Wazid, M., Katal, A., Goudar, R.H., Singh, D.P., Tyagi, A., Sharma, R. & Bhakuni, P. (2013). A framework for detection and prevention of novel keylogger spyware attacks. In Proceedings of the 7th International Conference on Intelligent Systems and Control (ISCO).

Wei, D., Wu, H., Wu, M., Chen, P.-Y., Barrett, C., & Farchi, E. (2023). Convex bounds on the softmax function with applications to robustness verification. arXiv. https://doi.org/10.48550/arXiv.2303.01713

Wenhua, X. & Zheng, Q. (2012). Constructing decision trees for mining high-speed data streams. Chinese Journal of Electronics, 21(2), 215–222.

Why, B.A. & Pottas, D. (2021). Best Practices to Address Medical Identity Theft Awareness. In IST-Africa 2021 Conference Proceedings.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. arXiv. https://doi.org/10.48550/arXiv.1705.08292

Wu, S. & Nagahashi, H. (2014). Parameterized AdaBoost: Introducing a Parameter to Speed Up the Training of Real AdaBoost. IEEE Signal Processing Letters, 21(6), 687-691, doi:10.1109/LSP.2014.2313570.

Wu, S. (2013). A review on coarse warranty data and analysis. Reliability Engineering & System Safety, 114, 1–11. https://doi.org/10.1016/j.ress.2012.12.021

Xu, H. & Yuan, H. (2020). An SVM-Based AdaBoost Cascade Classifier for Sonar Image. IEEE Access, 8, 115857–115867.

Xu, M., Baraldi, P., Lu, X. & Zio, E. (2022). Generative Adversarial Networks With AdaBoost Ensemble Learning for Anomaly Detection in High-Speed Train Automatic Doors. IEEE Transactions on Intelligent Transportation Systems, 23(12), 23408-23421, doi:10.1109/TITS.2022.3203871.

Yang, Z., Li, Q., Li, Y. & Wang, Z. (2019). A LBP texture analysis based liveness detection for face authentication. Machine Learning for Cyber Security (ML4CS 2019), Lecture Notes in Computer Science (Vol. 11806). Springer, Cham. doi:10.1007/978-3-030-30619-9_3.

Yangchun, Z., Zhao, Y. & Yang, J. (2020). New Virus Infection Technology and Its Detection. In Proceedings of the 2020 IEEE 2nd International Conference on Artificial Intelligence and Engineering (AIE) (ss. 388–390).

Yassein, M. B., Shatnawi, M. Q. & Al-zoubi, D. (2016). Application layer protocols for the Internet of Things: A survey. 2016 International Conference on Engineering & MIS (ICEMIS), Agadir, Morocco, ss. 1-4, doi: 10.1109/ICEMIS.2016.7745303.

Yungaicela-Naula, N.M., Vargas-Rosales, C., Pérez-Díaz, J.A., Jacob, E. & Martinez-Cagnazzo, C. (2023). Physical assessment of an SDN-based security framework

for DDoS attack mitigation: Introducing the SDN-SlowRate-DDoS dataset. IEEE Access, 11, 46820–46832.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arXiv. https://doi.org/10.48550/arXiv.1611.03530

Zhang, F., Zhou, H. & Zhou, X. (2009). A routing algorithm for ZigBee network based on dynamic energy consumption decisive path. 2009 International Conference on Computational Intelligence and Natural Computing, Wuhan, China, ss. 429-432, doi: 10.1109/CINC.2009.193.

Zhang, S., Xie, X. & Xu, Y. (2020). A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity. IEEE Access, 8, 128250-128263, doi:10.1109/ACCESS.2020.3008433.

Zhou, H., Hu, Y., Yang, X., Pan, H., Guo, W. & Zou, C. C. (2020). A worm detection system based on deep learning. IEEE Access, 8, 205444-205454. doi:10.1109/ACCESS.2020.3023434.

Zhou, P. & Chou, W.-K. (2019). IBBO-LSSVM-based network anomaly intrusion detection. International Journal of Embedded Systems, 11, 352–362.

Zhou, X.-H., Zhou, C., Lui, D., & Ding, X. (2014). Applied missing data analysis in the health sciences (1sted.). Wiley

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (ss. 2223-2232). https://doi.org/10.1109/ICCV.2017.244

Zilberman, A., Dvir, A. & Stulman, A. (2024). SPRINKLER: A multi-RPL Man-in-the-Middle identification scheme in IoT networks. IEEE Transactions on Mobile Computing, 23(10), 9971–9983. doi:10.1109/TMC.2024.3370894.

ÖZGEÇMİŞ

Kemal AKKANAT, Lise öğrenimini Bayrampaşa Sabit Büyükbayrak Lisesi'nde tamamladı. 1997 yılında girdiği İstanbul Üniversitesi İşletme Fakültesi İşletme (İng) Bölümü'nden, 2002 yılında mezun oldu. Daha sonra Sakarya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne 2010 yılında giriş yapıp 2014 yılında mezun oldu. 2018 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD'nda Yüksek Lisansı bitirdi. 2018 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü, Hesaplamalı Bilimler ABD'nda Doktora çalışmalarına başladı. 2015 yılından beri Trakya Üniversitesi Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi olarak çalışmaktadır.