T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BİLİM DALI

NESNELERİN İNTERNETİ İÇİN AKILLI SALDIRI TESPİT SİSTEMLERİ GELİŞTİRİLMESİ

DOKTORA TEZİ

Oğuzhan TAŞ

İstanbul
Ağustos-2022

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ BİLİM DALI

NESNELERİN İNTERNETİ İÇİN AKILLI SALDIRI TESPİT SİSTEMLERİ GELİŞTİRİLMESİ

DOKTORA TEZİ

Oğuzhan TAŞ

Tez Danışmanı
Dr. Öğr. Üyesi Amani YUSUF

İstanbul
Ağustos-2022

TEZ ONAYI

Lisansüstü Eğitim Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Bilgisayar Bilimleri ve Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman	Dr. Öğr. Üyesi Amani YUSUF	……
Üye	Dr. Öğr. Üyesi Muhammed DAVUD	……
Üye	Dr. Öğr. Üyesi Hakan GENÇOĞLU	……
Üye	Dr. Öğr. Üyesi Jawad RASHEED	……
Üye	Dr. Öğr. Üyesi Özge Yücel KASAP	……

Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Prof. Dr. Metin TOPRAK

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Doktora tezi olarak hazırladığım **"Nesnelerin İnterneti için Akıllı Saldırı Tespit Sistemleri Geliştirilmesi"** adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Oğuzhan TAŞ

ÖN SÖZ

Doktora tezi sürecinde pozitif ve yapıcı kişiliği yanında bilgi ve deneyiminden yararlandığım tez danışmanım Dr. Amani YUSUF'a, yurt dışında olmasına rağmen bilgi ve desteğini esirgemeyen Doç. Dr. Mustafa MISIR'a, bu aşamaya gelene kadar jürilerde yer almış ve değerli görüşleriyle katkıda bulunmuş hocalarıma, Fen Bilimleri Enstitüsü çalışanlarına, güzel bir çalışma ortamı sağladıkları ve güzel bir kütüphane oluşturdukları için Sabahattin Zaim Üniversitesi kütüphanesi çalışanlarına şükranlarımı sunmak isterim.

Tezin hazırlanmasında bana moral ve motivasyon olarak sürekli destek olan, yorucu ve yıpratıcı bu süreçte hep yanımda olan eşime, yeterince zaman ayıramadığım biricik kızlarım Neslişah ve Aslışah'a, akademik hayatımın her aşamasında maddi manevi olarak beni destekleyen rahmetli annem ve babam emekli öğretmen İbrahim TAŞ'a teşekkür ederim.

Oğuzhan TAŞ

İstanbul-2022

ÖZET

NESNELERİN İNTERNETİ İÇİN AKILLI SALDIRI TESPİT SİSTEMLERİ GELİŞTİRİLMESİ

Oğuzhan TAŞ

Doktora, Bilgisayar Bilimleri ve Mühendisliği Ana Bilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Amani YUSUF

Ağustos, 2022- 139 Sayfa

Nesnelerin İnterneti (IoT-Internet of Things) kavramı, günlük hayatta kullandığımız cihazlarla beraber üretilen yeni nesil teknolojilerin internete bağlandığı hatta bir ağ üzerinden birbirleriyle iletişim halinde olabildiği teknolojiyi tanımlamaktadır. Birçok üretici tarafından farklı altyapılarla üretilen IoT cihaz sayısının artması ve birçok alanda kullanılmasıyla birlikte IoT güvenliği problemleri ortaya çıkmıştır. Bu tez çalışmasında ilk bölümde IoT güvenliğine ağ üzerinden yapılan saldırılar, IoT katmanlarına göre ayrı ayrı incelenmiş ve savunma teknikleri tartışılmış ve yeni yöntemler önerilmiştir. İkinci bölümde ağ tabanlı saldırıların tespit edilmesi için Meta sezgisel Optimizasyon algoritmaların, Topluluk Öğrenme Algoritmalarıyla birlikte kullanıldığı bir model önerilmiştir. Önerilen model, klasik makine öğrenmesi algoritmalarıyla karşılaştırılmış, ağ tabanlı saldırıların yer aldığı NSL-KDD ile yeni nesil saldırıların da yer aldığı özellikle nesnelerin internetine yönelik saldırıları içeren BoT-IoT veriseti üzerinde %99,63 değerine ulaşan, doğruluk oranı yüksek sonuçlar alınmıştır.

Anahtar Kelimeler: Nesnelerin İnternetinde Güvenlik, Saldırı Tespit Sistemleri, Makine Öğrenmesi, Topluluk Öğrenme Algoritmaları

ABSTRACT

DEVELOPMENT OF AN INTELLIGENT INTRUSION DETECTION SYSTEM FOR INTERNET OF THINGS

Oğuzhan TAŞ

Ph. D. Department of Software Engineering

Supervisor: Asst. Prof. Dr. Amani YUSUF

August, 2022 - 139 Pages

The concept of the Internet of Things (IoT) defines the technology in which new generation technologies produced together with the devices we use in daily life are connected to the Internet and can even communicate with each other over a network. With the increase in the number of IoT devices produced by many manufacturers with different infrastructures and their use in many areas, IoT security problems have emerged. In this thesis, in the first chapter, attacks on IoT security over the network are examined separately according to IoT layers, defense techniques are discussed and new methods are proposed. In the second part, a model in which Metaheuristic Optimization algorithms are used together with Community Learning Algorithms is proposed to detect network-based attacks. The proposed model was compared with classical machine learning algorithms, and results with high accuracy of 99.63% were obtained on the NSL-KDD, which includes network-based attacks, and the BoT-IoT dataset, which also includes next-generation attacks, especially attacks against the Internet of Things.

Keywords: Internet of Things Security, IoT Security, Intrusion Detection Systems, Machine Learning, Ensemble Learning

İÇİNDEKİLER

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

SEMBOLLER

$P(A|B)$: B olayı doğru tahmin edildiğinde A olayının gerçekleşme olasılığı

$P(B|A)$: A olayı doğru tahmin edildiğinde B olayının gerçekleşme olasılığı

$P(A)$: A olayının gerçekleşme olasılığı

$P(B)$: B olayının gerçekleşme olasılığı

r : Korelasyon sabiti

x_i : bir örnekteki x değişkeninin değeri

$\bar{x}$: x değişken değerlerinin ortalaması

y_i : bir örnekteki y değişkeninin değeri

$\bar{y}$: y değişkeni değerlerinin ortalaması

μ : Ortalama değer

σ : Standart Sapma

X : Anlık değer

X' : Standartlaştırılmış değer

X_{maks} : Maksimum değer

X_{min} : Minimum değer.

KISALTMALAR

ABC : Artificial Bee Colony

AE : Auto-encoder

AES : Advanced Encryption Standard

ANN : Artificial Neural Networks

CNN : Cellular Neural Networks

CoAP : Constrained Application Protocol

CRC : Cyclic Redundancy Check

DBF : Deep Belief Network

DDoS : Distributed Denial of Service

DHSS : Direct Sequence Spread Spectrum

DoS : Denial of Service (Servis Yalanlaması)

DT : Decision Tree

EDA : Energy Deplation Attacks

FHSS : Frequency-hopping Spread Spectrum

GTS : Guaranteed Time Slot

IDS : Intrusion Detection System

IoT : Internet of Things

IoE : Internet of Everything

KAA : Kablosuz Algılayıcı Ağlar

kNN : k Nearest Neighbor

LoRaWAN : Long Range Wide Area Network

LoWPAN : Low Power Wireless Personel Area Networks

LPW : Low-Power Wireless

LR : Logistic Regression

M2M : Machine to Machine

MAC : Media Access Protocol

ML : Machine Learning

MLP : Multi Layer Perceptron

MQTT : Message Queuing Telemetry Protocol

NB : Naive Bayes

PAN : Personal Area Network

RBM : Restricted Boltzmann Machine

RF : Random Forest

RFID : Radio Frequency IDentification

RNN : Recurrent Neural Networks

RPL : Routing Protocol for low power and lossy network

RREQ : Route Request

SHA : Secure Hash Algorithm

SMOTE : Synthetic Minority Oversampling TEchnique

STS : Saldırı Tespit Sistemi

SVM : Support Vector Machine

T2M : Things to Machine

T2T : Things to Things

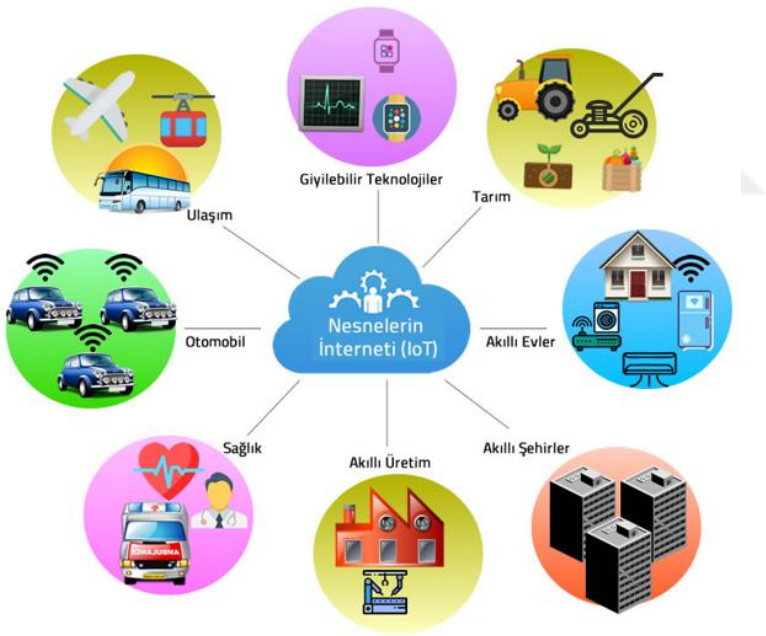
Wi-fi : Wireless Fidelity

WSN : Wireless Sensor Networks

BİRİNCİ BÖLÜM

NESNELERİN İNTERNETİNE YAPILAN SALDIRILARIN ÖNLENMESİ

Nesnelerin İnterneti Wifi, BlueTooth, GSM, ZigBee ve RFID gibi haberleşme teknolojisini bünyesinde barındıran İnternete bağlı cihazların oluşturduğu bir bilgisayar paradigmasıdır(Yaqoob vd., 2017). IoT aslında kablolu ve kablosuz telekomünikasyon ağlarından oluşan bir ağ alt yapısıdır. IoT uygulamalarını başta sağlık, otomotiv, askeri, Telekom olmak üzere birçok sektörde görmekteyiz. IoT kavramındaki "şey (things)" kelimesi, akıllı bileklik, akıllı bir elektrik süpürgesi, akıllı telefon, akıllı saat, otonom araçlar, akıllı bir şebeke ve akıllı bir ev olabilmektedir.



Şekil 1.1: Nesnelerin İnternetinin Bazı Kullanım Alanları.

Endüstride de geniş kullanım alanları mevcuttur, makine performanslarının takip edilmesinde, makine arıza verdiğinde ilgili personelin uyarılmasında, akıllı şehirleşmenin bir örneği olarak şehir içi trafik yoğunluğunun izlenmesinde, park yeri tespitinde, sarsıntıların algılayıcılarla tespit edilip olabilecek bir depremin haber verilmesinde, askeri teknolojide hudut güvenliğinin sağlanmasında IoT cihazlar

karşımıza çıkmaktadır. IoT teknolojisinin sağlık sektöründeki uygulamalarına da her geçen gün şahit olmaktayız. Örneğin uzaktan hasta takibinde, görme özürlü insanların algılayıcılar ile çevrelerini hissetmesinde, hasta kalp atışlarının takibinde, hastanın ilaç zamanı geldiğinde uyarı verilmesinde, ilacını alıp almadığının tespit edilmesinde vb. sağlık alanlarında IoT uygulamaları geniş olarak görülmektedir. Bu cihazlar kendi aralarında da İnternet olmadan sadece yerel kablosuz ağı kullanarak da haberleşebilmektedir. IoT teknolojisi her geçen gün kullanım sahası genişleyen ve giderek geniş kitlelere yayılan bir teknolojidir.

IoT teknolojisinin kapsamına bakıldığında Kablosuz Algılayıcı Ağ Teknolojileri (KAA-WSN), Makineden Makineye Haberleşme (M2M), Düşük Güçle çalışan Kişisel Ağlar (LoWPAN) gibi ağlardan oluştuğunu görmekteyiz (Granjal vd., 2015). Cisco 2018-2023 raporuna göre; Makineden Makineye(M2M) bağlantılar 2018'de %33'ten %50'ye yükselecektir. 2023 yılına kadar 14,7 milyar cihaz arasında M2M bağlantı olacaktır(*Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco*, y.y.).

IoT cihazların farklı teknolojilerden oluşması, bu cihazların entegrasyonunu ve güvenlik sorunlarını da beraberinde getirmektedir. IoT çatısı içinde birçok kablosuz teknoloji yer almaktadır. Wi-Fi, ZigBee, RuBee, Bluetooth, 6LowPan, LoRA, RFID, Thread, Z-Wave, Wavenis, 4G/LTE, NBIoT, RIIoT, Wi-Max, Tinymesh, 5G kablosuz teknolojilere örnek verilebilir. Kablolu teknolojilere örnek olarak PPP, X25, xDSL, FDDI, ARCNet, Ethernet, Token Ring, Frame Relay, ISDN teknolojileri örnek verilebilir(Lounis & Zulkernine, 2020).

IoT sisteminin güvenliği; Hücresel Ağlar, İnternet ve Kablosuz Algılayıcı Ağar'ın güvenliğine de bağlıdır, bu ağlarda olabilecek problemler IoT ekosistemini etkilemektedir. Ayrıca IoT cihazların büyük miktarda veriyi toplayıp gönderdiği Bulut Sistemlerin güvenliği de araştırmacılar tarafından üzerinde çalışılan diğer bir konudur.

IoT cihazlara karşı şimdiye kadar yapılmış en büyük saldırı olan Mirai saldırısı, 2016 yılının ağustos ayında gerçekleşti. Bu DDoS saldırısı ile başta IP kameralar ve ev yönlendiricileri olmak üzere internete bağlı 2,5 milyon cihaz etkilendi(*Understanding the Mirai Botnet | USENIX*, y.y.).

Klasik kriptografi biliminde yıllardır kullanılan ve geliştirilen algoritmalarla güvenlik, gizlilik, bütünlük, erişim kontrolü, inkâr edememe ve kimliğini doğrulama gibi

güvenlik gerekleri yerine getirilmektedir (Stallings, y.y.). Fakat IoT teknolojisinde gerek depolama alanı gerek bant genişliği gerekse hesaplama bakımından güçsüz IoT cihazlarda geleneksel kriptografi tekniklerinin uygulanması pratik olmamakla birlikte, zordur. Örneğin genel anahtar (public key) yapısına sahip RSA gibi asimetrik şifreleme algoritmasının, AES gibi tek anahtarla şifreleyip çözen (secret key) simetrik şifreleme algoritmasının, SHA-512, RIPEM-D gibi özellikle bütünlük kontrolü için kullanılan algoritmaların çalıştırılması zordur. Sonuç olarak, geleneksel tekniklerin hesaplama karmaşıklığının yüksek olmasından dolayı, IoT cihazlarda verimli olması mümkün değildir, çünkü hesaplamalar güçlü işlemci ve büyük miktarda bellek gerektirmektedir. Sistem kaynaklarının limitli olmasından ötürü, bahsi geçen güçlü algoritmalarla ve güvenlik standartlarıyla kimlik doğrulama, gizlilik, erişilebilirlik, inkâr edememe gerekleri sağlanamamaktadır, bu sebeple IoT cihazlar saldırıları karşı savunmasız duruma gelmektedir. Özellikle askeri alanda ve tıp alanında güvenlik daha öne çıkmaktadır. Bir ameliyat gerçekleştirilirken IoT cihaza yapılan saldırı hasta hayatını etkileyebilir, hudut geçişini kontrol eden bir algılayıcı kümesinin devre dışı kalması düşmanlara karşı savunmasız bırakabilir. Geleneksel ağlardan farklı olarak IoT cihazlar erişilemez veya ulaşılamaz durumda değildir. Geniş bir hudut alanına bırakılan kablosuz algılayıcı cihazlara düşmanın erişmesi ve kurcalayarak yeniden programlanması gibi ekstra risklerle karşılaşılabilir.

IoT ve Kablosuz Algılayıcı Ağların(WSN) mimarisini katmanlara göre karşılaştıracak olursak; farklı kaynaklarda ara katmanlar eklense de temel olarak Şekil 1.2'de görüldüğü gibi KAA beş katmanda(Tomić & McCann, 2017), Şekil 1.3'de görüldüğü üzere Nesnelerin İnterneti ise dört temel katmanla (Hassija vd., 2019) gösterilmektedir.

UYGULAMA KATMANI	Veri birleştirme, son kullanıcıyla etkileşim yapılır.
TAŞIMA KATMANI	Güvenilir veri taşıma işlemi bu katmanda yapılır.
AĞ KATMANI	Yönlendirme, ağ topoloji yönetimi yapılır.
VERİ BAĞI KATMANI	Hata Kontrolü, Veri çerçevesi tespit etme, çoğullama
FİZİKSEL KATMAN	Modülasyon, frekans ve kanal seçimi, sinyal işleme

Şekil 1.2: Kablosuz Algılayıcıların Katmanları

UYGULAMA KATMANI	Son kullanıcıyla etkileşim kuran soyutlanmış çözümler.
ORTA KATMAN	API, Web Servis ve Bulut vb.
AĞ KATMANI	Kablolu ve kablosuz sistemlerden oluşur. Algılama katmanından gelen veriyi işleme ve iletme işlemleri.
ALGILAMA KATMANI	Ortam verisini algılama işlemleri. mekaniksel, elektriksel, elektronik ve kimyasal algılayıcılar

Şekil 1.3: Nesnelerin İnternetinde Katmanlar

Uygulama ve Ağ arasında Veri İşleme (Data Processing) isimli bir katman da bazı kaynaklarda eklenmektedir. Tıpkı OSI katmanlarında olduğu gibi veriyi bir alt katmandan alıp işleme, daha sonra analiz etme ve değerlendirip sonuca göre davranma (karar alma) ve diğer IoT cihazlarla bilgiyi paylaşma gibi görevleri bulunmaktadır. Örneğin Akılı ev hub'ı ve akıllı saat gibi IoT cihazlarda daha evvel analiz edilen ve kayıt edilenler kullanıcı verileri sayesinde, kullanıcı deneyimi artırılmaktadır(Sikder vd., 2018).

Bu bölümün kalan kısmında, Tablo 1.1'de görülen Nesnelerin İnternetine gerçekleştirilebilen saldırıları katmanlara göre detaylı şekilde incelenecek ve saldırılar için savunma teknikleri önerilecektir.

Tablo 1.1: Nesnelerin İnternetinde Katmanlara Göre Saldırılar

Fiziksel Katman	Yayını Bozma Saldırısı, Kurcalama Saldırısı, Zararlı Kod Aşılama, Yan Kanal Saldırısı, Uykudan Yoksun Bırakma Saldırısı
Veri Bağı Katmanı	Çakışma, Tükenme, Yayın Bozma, Geri Çekilme, Değişmeyen Tek Kullanımlık Sayıya Yapılan Saldırı GTS (Garantilenmiş Zaman Slotu) Tam Hâkimiyet, Zeki Olmayan Tekrarlama, Asıllanmış Yayın,
Ağ Katmanı	MERHABA Akını Saldırısı, DATA Akını Saldırısı, Solucan Deliği Saldırısı, Sybil Saldırısı, Gider Deliği (Sinkhole), Gri Delik, Kara Delik, Düğüm Tekrarlanması
Taşıma Katmanı	SYN Akını, Eşleme (Senkronizasyon) Bozulması
Uygulama Katmanı	Servis Reddi Saldırısı, Ortadaki Adam Saldırısı

1.1. Fiziksel Katman Üzerine Yapılan Saldırılar

1.1.1. Yayın Bozma Saldırısı

Yayın bozma Saldırısı (Jamming Attack) düşmanın güçlü bir anten kullanıp sinyal üreterek parazitler oluşturduğu ve iletişimi engellediği saldırıdır. Farklı türlerde yayın bozma gerçekleşebilir. Sabit yayın (Constant Jamming) bozma saldırısında sürekli olarak radyo sinyali gönderilerek gerçekleştirilir, ya da rastgele birlerin MAC katman etiketi olmadan kanala gönderimi gerçekleştirilir (Xu vd., 2006). Diğer bir tür olan Aldatıcı Yayın (Deceptive Jammer) Bozma tekniğinde ise, aldatıcı paketlerin düzenli paketler arasına karıştırılarak, saldırının kamufle edildiği bir saldırı türüdür. Rastgele

yayın (Random Jammer) bozma saldırısında ise bir süre sinyal gönderilir, daha sonra uyku modunda beklenilir, belli bir zaman sonra uyku modundan uyanıp tekrar sinyal gönderilerek tespit edilme ihtimali azaltılır. Uyku modu esnasında aldatıcı veya sabit yayın bozma gibi hareket edilmektedir. Tepkili Yayın (Reactive Jammer) bozma saldırısında diğerlerinden farklı olarak ağdaki trafik kontrol edilir, trafik eğer yoksa beklenilir varsa aktif moda geçiş gerçekleşir, zor tespit edilen saldırı tiplerinden biridir (X. Chen vd., 2009).

Bu saldırı tipleri bazı çalışmalarda DoS (Servis reddi ya da yalanlaması) kategorisi içinde gruplandırılmıştır. Bu tip saldırı farklı katmanlara da uygulanabileceğinden her bir katman için ayrı bir önlem veya savunma tekniği düşünmek gerekmektedir. Fiziksel katmanda, haberleşme bitleri akıp giderken sinyalin ya da ilgili paket bitlerinin değiştirilmesiyle de bu saldırı gerçekleşebilmektedir. ACK paketlerine saldırılar ayrı bir bölümde ele alınacaktır.

Savunma tekniği olarak, sürekli ağı izleyen ve performansını değerlendiren teknikler kullanılabilir. Sinyal gücü yanında, paketin ne kadar oranda teslim edildiği vb. bilgiler ile hesaplar yapılarak sinyalin ne kadar bozulduğu anlaşılabilir.

Ek bir savunma yöntemi olarak, algılayıcı düğümlere yayın bozma saldırısını önlemek için frekans karıştırma yöntemi kullanılabilir. FHSS ve DSSS dağınık yayılma yöntemleri kullanılabilir. FHSS hem vericinin hem de alıcının bildiği rastgele bir diziyi taşıyıcıyı frekans kanalı arasında hızlı bir şekilde değiştirerek radyo sinyalleri oluşturur. DSSS yönteminde ise çok miktarda bitten oluşan örüntü ile orijinal sinyal temsil edilir. Bitlerden birisi veya ikisi bozulursa, istatistiksel yöntemler kullanarak haberleşmesi yinelemeden orijinal bilgiye geri dönülür.

Kızılötesi haberleşmede veya kablosuz haberleşmede, mod değiştirilerek (haberleşmede) saldırıdan korunulabilir, düşman bertaraf edilebilir. Diğer bir yaklaşım da Bölgesel Planlama yöntemidir, bu yöntem ile olası yayın bozma yapılan bölgeler belirlenerek, etkilenebilecek ağ düğümlerinden bir grup meydana getirilir. Eğer algılayıcı düğümlerden bir tanesi saldırı fark ederse, hemen kanal değiştirilip, düşman bertaraf edilir.

1.1.2. Kurcalama Saldırısı

Düşmanın algılayıcı bölgesine girerek, düğümlere fiziksel olarak etki ettiği saldırı türüdür. Düşman cihazın içindeki yazılıma erişerek değiştirerek bir saldırıda bulunabilir. Hiyerarşik sistemdeki bir ağda en kilit düğüme ulaşabilir ve yalnızca bu düğüme saldırılarak tüm ağ sistemini çökertip devre dışı bırakabilir.

Düşman devreyi ele geçirdikten sonra baskı devresini tekrar çıkarabilir, hafıza(memory) okuması yapılabilir, ROM ve RAM bellek içeriği tekrar istenilen şekilde doldurulabilir, yongada araştırma yapılabilir (on-chip probing), elektromanyetik tarama yapılabilir, derin mikron saldırıları gibi başka donanıma zarar verecek saldırılar yapılabilir(Pacalet, 2022).

Savunma tekniği olarak, bilgi sızıntısının engellemek için, eğer bir müdahale şeklinde saldırı varsa otomatik olarak hafızanın silinmesi gerçekleştirilebilir. Şifreleme algoritmalarıyla hafıza içeriği kriptolanabilir, performans kaybı olmaması için verimli algoritmalar seçilmelidir. Algılayıcıların kamufle edilmesi de diğer bir çözümdür, böylece fiziksel olarak bulmak güçleşecektir. Farklı yazılım koruma metotları uygulanabilir örneğin kodun karıştırılması (obfuscation) yöntemi ile düşmandan kaynak kodlar gizlenebilir ve değiştirilmesi önlenebilir.

Anahtar Dağıtım ve Yönetim Teknikleri (Key Distribution and Management) ile algılayıcıların arasında pasif kalan düğümlerin tespit edilmesi, bir tanesinin alınıp yerine düşmanın kendi istediği düğümü yerleştirmesi gibi durumlar hemen fark edilebilir.

Blok zinciri teknolojisi(TAŞ & KİANİ, 2018) ile ağda bir düğümler arası bir mesaj özütü (hash) zinciri oluşturularak herhangi bir düğümde olabilecek yer değiştirme, devre dışı kalma vb. bir problem tespit edilebilir.

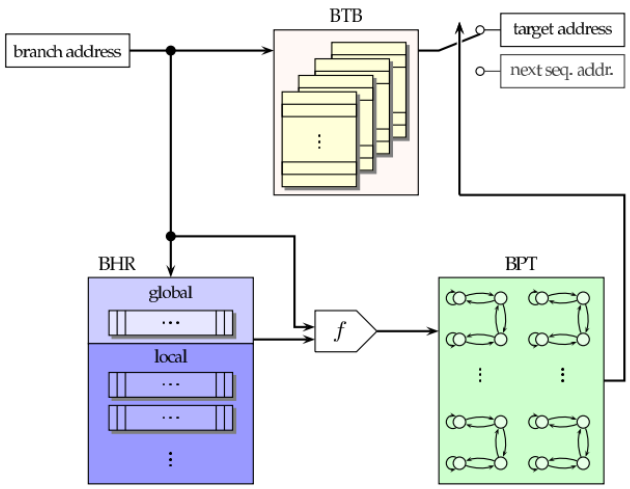
1.1.3. Zararlı Kod Aşılama Saldırısı

Zararlı Kod Aşılama (Malicious Code Injection) saldırısı, IoT veya KAA(WSN) ağındaki bir algılayıcının belleğine dışarıdan zararlı kodları enjekte edildiği saldırıdır. Güvenlik amacıyla belli periyotlarla bu yazılımların güncellenmesi riski azaltmak birlikte tamamen yok etmemektedir. IoT cihaza yazılımla müdahale edip, istediği şekilde kontrol altına alarak ağdaki diğer düğümlerle olan iletişimi takip edebilir.

Savunma tekniği olarak, cihaz Komut Setinin Rastgele Hale Getirilmesi(ISR-Instruction Set Randomization) yöntemi (Wai Kyi Kyi Oo, 2019) ile zararlı kodların enjekte edilmesi önlenebilir. Algoritmalar tarafından rastgele olarak yapılacak işe özel komut seti oluşturularak IoT cihaza zararlı kod enjekte edilmesine karşı önlem alınabilir.

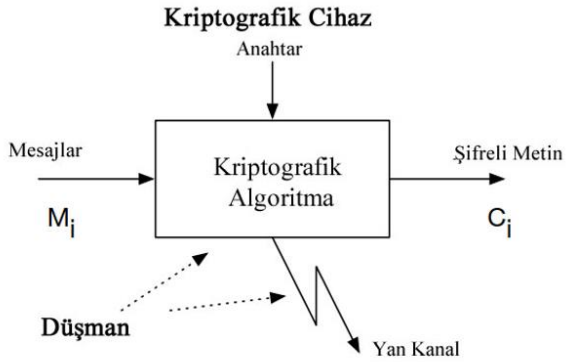
1.1.4. Yan Kanal Saldırısı

Yan Kanal Saldırısı (Side Channel Attack) şifrelenecek metin (plaintext) veya şifrelenmiş metin (ciphertext) olmadan sadece şifreleme yapan fiziksel cihazdan elde edilen bilgilere dayanılarak yapılan saldırıdır. Cihazın mikroişlemci mimarisi, güç tüketimi incelenerek elde edilen bilgiler, yayılan elektromanyetik sinyaller gibi şifreleme işlemi sırasında önemli bilgiler elde edilebilir(Babar vd., 2011). Lazer ile yapılan saldırılar, Zamanlama saldırıları gibi kritik bilgilerin elde edilmesi için birçok saldırı tekniği bulunmaktadır. Şekil 1.4'de görülen Acıiçmez ve takım arkadaşları (Acıiçmez vd., y.y.) tarafından mikroişlemci Dallanma İşlem Birimi (BPU-Branch Prediction Unit)'ne saldırı gerçekleştirilmiştir. Çok kullanılan ve güçlü RSA algoritmasının kullanıldığı internette iki taraf arasında güvenliği sağlayan OpenSSL (*Open SSL*, y.y.) uygulamasına dört farklı saldırı senaryosu gerçekleştirilmiştir. Şekil 1.4'te ilgili makaleden alınan BPU şeması görülmektedir.



Şekil 1.4: Dallanma Tahmin Birimi (BPU) Mimarisi.

Bu saldırıya savunma yaklaşımı olarak, işlemci içerisine entegre edilecek kriptografik modüller aracılığı ile bu tip saldırılar önlenebilir. Şifreleme cihazının çalıştığı ortamın elektromanyetik izolasyonu sağlanabilir.



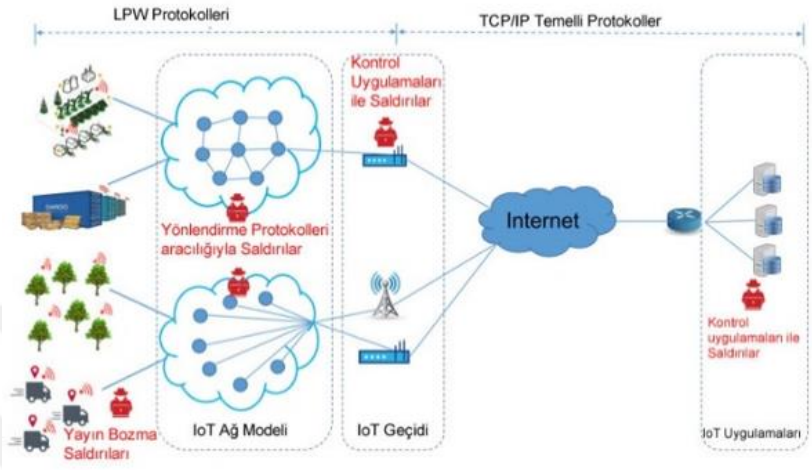
Şekil 1.5: Yan Kanal Saldırısı

1.1.5. Uykudan Yoksun Bırakma Saldırısı

IoT cihazların batarya ya da pil ömrü zayıf olduğundan dolayı güç tasarrufu için kendilerini kullanılmadıkları müddetçe uyku moduna geçirirler, böylece ayakta kalma süreleri artar. Bu saldırıda, cihazın kendisini uyku moduna geçirmesini engellemek için sürekli meşgul edilir. Bu saldırıda, bataryası biten düğümler algılayıcı ağına dahil olamayacağından ve hizmet veremeyeceğinden bu saldırı çeşidi DoS (Hizmet Reddi) saldırısının bir varyasyonu olarak da açıklanabilir. Örneğin, zararlı kodların sisteme yüklenmesi ve sonsuz döngü ile algılayıcının sürekli çalıştırılması durumunda bataryası bitecek ve algılayıcı devre dışı kalacaktır.

Savunma tekniği olarak, algılayıcı cihazın uykuya girmeyi engelleme problemi için dağıtık işbirlikçi bir yapı sunulmuştur (Bhattasali vd., 2012). Sunulan model ile algılayıcı ağda meydana gelebilecek anormallik tespiti için iki adımda ihlaller azaltılmıştır, sonuçta düğümlerin enerji sarfiyatı korunmuş ve ağın ayakta kalma (yaşam) süresi artırılmıştır. Bu saldırıya önlem olarak yapılan diğer bir çalışmada (Nguyen vd., 2019), Düşük Güçlü (Low Power) Kablosuz IoT Ağlar için Enerji Tüketimine Yönelik Saldırılar (EDA) Fiziksel Katmanı, MAC, Ağ Katmanı ve Uygulama katmanına göre ayrı ayrı incelenmiştir. Fiziksel katman için yapılan

saldırılar Yayın Bozma Saldırıları başlığında incelendiği için tekrar bahsetmeyeceğiz, MAC katmanı için yapılanlar ise Tekrar Gönderme (Replay Protection) saldırısına karşı korunma başlığında açıklanmıştır. Uygulama katmanı için yapılan Yeniden Programlama ve Kod Enjekte Etme (Aşılama) saldırıları da önceki kısımda incelenmiştir.



Şekil 1.6: EDA Saldırıları ve LPW Ağa Saldırılar

1.2. Veri Bağı katmanına Saldırılar

1.2.1.Çakışma Saldırısı

İki düğüme aynı frekansta eşzamanlı olarak paket gönderildiğinde çakışma (collision) meydana gelir. Paketlerde çakışma olduğunda, veri kısmında değişiklik meydana gelir ve sinyalin doğruluğunu kontrol eden CRC (Cyclic Redundancy Check) uyuşmazlık olduğu anlaşılır. CRC sonucuna bağlı olarak paket atılacaktır. Diğer bir çalışma saldırısında da, saldırgan sürekli ACK mesajı göndererek paketlerde çarpışmaya sebep olabilir, gönderilen paketlerin tekrar tekrar gönderilmesi algılayıcıların enerji tüketerek devre dışı kalmasına ve ağda iletişimin engellenip, zaman kaybedilmesine sebep olur (Payandeh & Mozayyani, y.y.).

Savunma tekniği olarak, baz istasyonuna gönderilip alınan paketlerin akış miktarını baz alan farklı tespit yöntemleri ve algoritmaları kullanılarak bu saldırı tespit edilebilir ve önlenebilir.

1.2.2. Tükenme Saldırısı

Algılayıcı ağ içerisindeki düşman ya da kötü niyetli bir düğüm tarafından, kanalda sürekli yapılan istek-iletimi ile MAC protokolünün bozulması gerçekleşir. Tekrarlanan bu saldırılar sürekli olduğundan sistem kaynaklarının aşırı tüketilmesine sebep olur. Sonuçta daha önceki bölümlerde de anlatıldığı gibi bataryanın erken tükenmesi ve ağın iş göremez hale gelmesine neden olur.

Savunma tekniği olarak, MAC kabul ve kontrol birimine yerleştirilecek İstek Oran Limiti ile gelebilecek aşırı isteklerin yok edilmesi sağlanabilir(Law vd., 2009). İkinci bir yaklaşım, zaman bölmeli çoğullama tekniği kullanılabilir (Abdelzaher vd., 2004). Modekurthy (Modekurthy vd., 2018) ve arkadaşları tarafından yapılan çalışmada çok sekmeli(multi-hop), çok kanallı(multi-channel) endüstriyel kablosuz algılayıcı-aktüatör ağları için kullanım limitini baz alan bir çizelgelenebilirlik analizi önerilmiştir, yapılan çalışma TOSSIM benzetim aracı ile değerlendirilmiştir.

1.2.3. Yayın Bozma Saldırısı

Fiziksel katmanda daha önce değindiğimiz Yayın Bozma (Jamming Attack) Saldırısı, Veri bağı katmanında daha karmaşık ve daha fazla enerji bakımından etkisi olan saldırılardır. Fiziksel katmanda veri paketleri sadece hedef olurken bu katmanda hemen her paket hedef olabilir, tespit edilmesi daha güç bir saldırı türüdür.

Kötü niyetli kişi, algılayıcıların iletim zamanında paketler göndererek tıkanıklığa sebep olur, normal kullanıcılar böylece ağı kullanamaz. IoT cihazlarda MAC protokolleri birbirinden farklı olsa da, her biri için MAC protokol tipine göre tıkama saldırısı yapılır. Saldırgan için zor olan veri paketlerinin erişim zamanını kontrol etmektir. Bu bilgiden yola çıkarak Law ve ekibi (Law vd., 2009), veri paketlerinde zaman aralıklarının olasılık dağılımı çıkarıp kümelere ayıran bir saldırı tekniği sunmuşlardır.

Savunma tekniği olarak, veri iletiminin en başında bir mesaj iletim tablosu düğümlerle paylaşılabilir. Buradaki mesaj iletim tablosun baz alınarak paket anahtar yöntemleri ile paketler birer birer şifrelenebilir.

1.2.4. Geri Çekilme Saldırısı

Geri Çekilme Saldırısı (Back-off Manipulation Attack) birçok veri bağı katmanında algılayıcıların aynı zamanda erişmesini engellemek için bir geri çekilme (back-off time) zamanı belirlenmiştir. Saldırgan bu zamanda bir hileye başvurarak, daha küçük bir gecikme zamanı belirler ve diğer düğümlere göre daha avantajlı duruma gelir. Ek olarak, kendisinin belirlediği bir geri çekilme zamanında ne zaman isterse başka düğümlerle iletişime geçebilir (Radosavac vd., 2007).

Savunma tekniği olarak Dasari ve arkadaşları (Dasari, 2017), IEEE 802.11 kablosuz ağlarda MAC katmanı DoS ataklarını minimum tespit gecikmesi ile gerçek zamanlı olarak tespit eden bir yöntem önermişlerdir. Gecikme ve çıktı verilerini toplayıp ve dağıtım değişimini gözlemlemek için bir değişim noktası algılama algoritması uygulamışlardır. Bu saldırıların etkisini gözlemlemek için, Ağ Simülatörü (NS-3) üzerinde benzetimler yapmış, gecikme ve çıktı sonuçları açısından algılama algoritmasının etkinliğini göstermişlerdir.

Diğer bir savunma tekniği olarak CSMA-CA yapısına dayanan protokoller kullandığında bu saldırı önlenebilir. Ayrıca Gecikme Zamanı değeri rastgele olarak ayarlanıp, düğümün kendisinin ayarlamasına müsaade edilmeyebilir. Ek olarak, bir başka yöntem de protokol paketlerini performans düşmeyecek şekilde şifrelemektir, böylece saldırgan gecikme zamanını istediği gibi değiştiremez ve saldırı önlenmiş olur.

1.2.5. Değişmeyen Tek Kullanımlık Sayıya Yapılan Saldırı

Değişmeyen Tek Kullanımlık Sayı Saldırısı (Sastry & Wagner, 2004) bütün protokollerde karşımıza çıkan bir saldırı tipidir. Bir defa kullanılan sayı (bks) kavramı, kriptografi biliminde karşılıklı doğrulama protokollerinde oldukça sık kullanılmaktadır. Fakat bu sayının her defasında farklı seçilmesi ve tekrar etmemesi, gerçekten rastgele üretilmesi önemlidir. Algılayıcılar hedef adres, bks bilgisi ve anahtar bilgisini kendi erişim kontrol tablolarında şifreleyerek tutabilirler. Eğer aynı şifreleme anahtarı ve bks değeri kullanılırsa, ortamı dinleyen ve gözlemleyen saldırgan, şifreli metinlerden bazı bilgileri ele geçirebilecektir.

Bir ZigBee ağında da ağı dinleyen saldırgan ele geçirdiği iki paket arasında XOR işlemi yaparak bazı bilgiler ele geçirebilir. Bu durum çok az ihtimalle gerçekleşse de Güven Merkezi'nin (Trust Center) bataryaları devre dışı bırakılıp elektrik kesintisi ile bu durumun gerçekleşmesi mümkündür. Bu durumda güven merkezinin bks değerini sıfırlayıp aynı bks değerinin tekrar göndermesi mümkündür(Vidgren vd., 2013).

Savunma tekniği olarak, erişim kontrol listesine yeni alanlar ilave edilebilir, çerçeve sayısından (frame counter) bks değeri ayrıştırılabilir (Sokullu vd., 2008). Ayrıca çerçeve sayacı yerine zaman damgası kullanılabilir, böylece iletim zamanı hesaplanıp, saldırganın paketleri ele geçirip indirmesi ve değiştirmesi gibi durumlar anlaşılabilir.

1.2.6. GTS-Garantiye Alınmış Zaman Slotuna Yapılan Saldırı

IEEE 802.15.4. MAC standardı PAN (Kişisel Bölge Ağı) kontrolcüsü tarafından yönetilir. Süper çerçeve alt yapısı ile çekişmeli ya da çekişmeli olmadan hizmetler yönetilmektedir. PAN, ağ cihazlarının her biri için ayrı oluk(slot) tahsis ederek, mesajlarda çarpışma olmasını engeller. Bu saldırı tipinde saldırgan, GTS(Garantiye Alınmış Zaman Slotu) oluklardaki bir zayıflıktan yola çıkarak PAN koordinatörü ile cihaz arasındaki iletişimi bozar. Kötü niyetli kişi kendisini diğer düğümlerle senkronize ederek ağ hakkındaki detay bilgileri içeren mesajları (beacon-fener mesajları) alır. İlk önce düğümlerden biri GTS oluğunda yer tahsis edilmesi için bir talep gönderir. PAN koordinatörü ya olumlu cevap verip GTS oluğu tahsis eder, ya da bu talebi reddeder. GTS talebi olumlu ise fener mesajı ile tüm düğümlere bildirim yapılır. Ağ dinleyen saldırgan GTS açıklama bilgisini inceleyip araya girer ve resmi GTS düğümü ile PAN Koordinatörü arasında gidip gelen veri paketlerinde çarpılmalar ve bozulmaları oluşmasına sebep olur (Sokullu vd., 2008).

Savunulması zor olan bu saldırı tipinde PAN koordinatörüne gelen GTS talepleri şifrelenebilir ve farklı kimlik asıllama (doğrulama) yöntemleriyle düşmanın sahte mesaj göndermesi engellenebilir. Sajjad ve arkadaşları (Sajjad & Yousaf, 2014) IEEE 802.15.4 için güvenli anahtar oluşturma, değiştirme ve macKeyTable'in hangi şekilde oluşturulacağı gibi noktalara değinmişlerdir.

Daidone ve arkadaşları (Daidone vd., 2013) da 802.15.4 ağlarında seçimsel yayın bozma saldırılarına karşı yayın bozma saldırısına karşı dirençli bir GTS önermişlerdir.

1.2.7. Tekrar Gönderme Saldırısından Korunulmasını Önleyen Saldırı

Kablosuz ağda tekrar gönderme gibi saldırı tiplerinden korunmak amacıyla son gönderilen mesajdaki çerçeve numarası önceki mesajdakinden büyük olmalıdır. Tekrar gönderme korunması (Replay Protection) yapısını bilen saldırgan daha büyük çerçeve numarası içeren mesaj göndererek saldırı gerçekleştirebilir (Pawar vd., 2012).

Savunma tekniği olarak, artık çerçeve numarasından ziyade zaman damgasını(timestamp) kullanmak bu saldırı tipini önleyecektir. Na ve arkadaşları (Na vd., 2017) tarafından LPWAN (Düşük güçlü Geniş Bölge Ağları) teknolojisinin pazardaki adının LoRaWAN olarak değiştiğini fakat ağa katılma prosedüründe saldırganın güvenlik açıklarını kullanarak sömürebileceğini belirtmişlerdir. Bu saldırı senaryosuna karşı farklı bir ağa katılma prosedürü önermişlerdir.

Diğer bir çalışmada (Kim & Song, 2017) LoRaWAN'ın yanlışlıkla normal mesajları yeniden tekrarlama saldırısı olarak gördüğünden dolayı yeni bir LoRaWAN şema önerilmiştir. Mevcut paket yapısı ile uyumlu ve cihaz sıfırlama gibi özel durumlar için tasarlanmıştır. Yapılan çalışmada normal mesajın, tekrarlama saldırısı gibi algılanma oranı mevcut LoRaWAN alt yapısına göre %60-89 daha düşüktür.

1.2.8. Tam Hâkim Olma Saldırısı

Tam Hakimiyet (Full Domination) Saldırı tipinde, saldırgan MAC protokolü hakkında geniş bilgiye sahiptir ve ağa gizlice katılabilir. Ağ üzerinde güvenilir bir trafik üretip, uyku yalanlaması saldırısını kullanarak maksimum fayda sağlayabilir. Ağdaki düğümlerden bir veya birkaçına yerleşebilir. MAC katmanı bu tip saldırıları karşı zafiyet göstermektedir.

Savunma tekniği olarak, ağ protokol paketlerinin performansı düşürmeyecek şekilde şifrelenmesi ile bu saldırı engellenebilir.

1.2.9. Zeki Olmayan Tipte Tekrarlama Saldırısı

Zeki Olmayan Tekrarlama (Unintelligent Replay Attack) saldırı tipinde saldırgan MAC protokolü hakkında geniş bilgiye sahip değildir. Daha önce kaydedilen olaylar

ağda yinelenerek düğümlerin uyku moduna geçmesi önlenir ve düğümler çok fazla paket iletimiyle meşgul olduklarından çok fazla enerji tüketirler (Pawar vd., 2012).

Savunma tekniği olarak, zaman damgası ile paketlerin ne zaman ulaştığı takip edilebilir, böylece bu saldırıp tipi bertaraf edilebilir.

1.2.10. Asılanmamış (Kanıtlanmamış) Yayın Saldırısı

MAC protokolü ile ilgili detaylı bilgiye sahip olduğu ve ağa sızma yaptığı saldırı türüdür. Ağda kimliği tespit edilemeyen mesajlar yayınlayarak düğümlerin uyku ve dinleme döngüsünü saldırgan bozar. Ağada aşırı enerji tüketilmesi ve ağın hayatta kalma süresinin azalmasına neden olur.

Savuma tekniği olarak paketlerin şifrelenmesi ve her ağ paketine zaman damgası (timestamp) eklenebilir. Ek olarak her düğüm, paketlerin gönderilme geçmişini düğümün ilgili iletim tablosunda saklayarak paketin atılacağına ya da saklanacağına her düğüm kendisi karar verebilen AT2A yöntemi(*Publication - AT2A: Defending Unauthenticated Broadcast Attacks in Mobile Wireless Sensor Networks*, y.y.) önerilebilir.

1.3. Ağ Katmanına Saldırılar

Literatürde yönlendirme saldırıları olarak da geçen bu saldırılar özellikle mesajların düğümler arasında yönlendirilmesi sırasında meydana gelmektedir. İlk değineceğimiz ve en çok bilinen saldırı türü Akın saldırıları (Flood Attack) adıyla geçen şüpheli düğümlerin ağa yanlış paketler göndererek ağın çalışmasını engellediği saldırı türüdür. Akın saldırıları (Flood Attack) aslında DATA, HELLO ve RREQ (Route Request-Yönlendirme İsteği) saldırıları olmak üzere üç tipte sınıflandırılır.

1.3.1. DATA Akını Saldırısı

Bir DoS Saldırı tipi olan DATA akını saldırısında şüpheli düğüm ağdaki veri paketlerini ağdaki herhangi bir düğüme iletebilir. Bu saldırı tipi hedeflere veri yönlendirilirken daha fazla etkilidir. Bu saldırı tipine yönelik savunma yaklaşımına MERHABA Akını Saldırısı bölümünde değinilecektir.

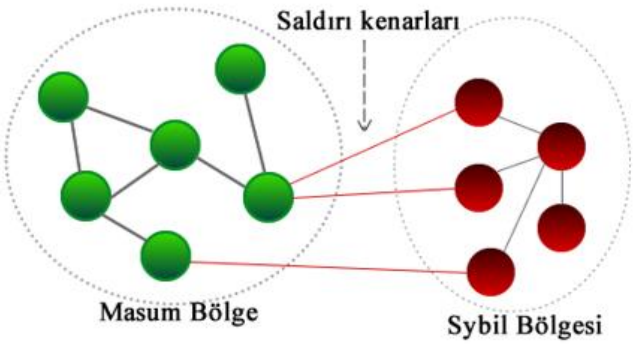
1.3.2. MERHABA Akını Saldırısı

Merhaba Akını (HELLO Flood Attack) olarak da geçen bu saldırı tipinde; ağdaki düğüm, komşu düğümlerin keşfi için HELLO paketlerini belli aralıklarla gönderir. Güçlü paket iletim gücüne sahip düşman ağdaki düğümlerine hepsine HELLO cevabı ürettirir ve sonuçta düğümler çok fazla enerji tüketirler (Deng vd., 2006). Bu saldırı türü de bir DoS saldırı tipidir.

Savunma tekniği olarak, çift taraflı düğüm kimlik doğrulama sağlanarak bu saldırı önlenebilir. Alternatif olarak, kimliklendirilmiş yayın tarzında çalışan protokoller kullanılabilir. Adiya ve arkadaşları Derin Öğrenme tabanlı BAU-ROA-DBN yöntemiyle ağdaki MERHABA Akını saldırısına neden olacak düğümleri belirlemişlerdir (Aditya Sai Srinivas & Manivannan, 2020).

1.3.3. Sybil Saldırısı

Bu saldırı tipinde saldırgan sahte kimlikler kullanarak veya var olan bir düğümün kimliklerinin kopyaları ile ağa sızar. Bu saldırı oldukça etkilidir, son yıllarda da önlemek için yoğun çalışmalar yapılmaktadır. Dağıtık Algoritmalar tarafından sağlanan veri bütünlüğü, veri güvenliği ve verimli kaynak kullanımını düşürür. Dağıtık saklama ünitelerine, farklı yönlendirme mekanizmalarına karşı Sybil saldırısı zarar verebilir. Eşler arası ağlar (Peer to Peer) genel olarak Sybil saldırısına karşı güçsüz olan ağlardır. Aşağıdaki Şekil 1.7'de görüldüğü gibi kırmızı düğümler yapay şekilde üretilmiş ya da ağdaki herhangi bir düğümden klonlanmış(kopyalanmış) Sybil düğümlerdir (Lakhanpal & Sharma, 2016).



Şekil 1.7: Ağ Katmanı Saldırılarından Sybil Saldırısı

Savunma tekniği olarak Arshad ve arkadaşları (Arshad vd., 2021) Sybil saldırısına karşı koymak için güncel yöntemler önermişlerdir. Bu yöntemler, Anahtar Yönetimi, Kaynak Testi, Şifreleme, Güven, RSSI (Received Signal Indicator-Alınan sinyal göstergesi), Watchdog, TDOA, Yapay zekâ, Teşvik Temelli yaklaşımlardan oluşur.

1.3.4. Gider Deliği Saldırısı

Gider Deliği (Sinkhole Attack) saldırı tipinde, saldırgan ağdaki baz istasyonunun bilgiye ulaşması engellenmeye çalışılır. Saldırgan trafiği belli bölgeye yönlendirir. Yanlış yönlendirme verisi paylaşarak, kendi istediği yolu cazip olarak sunar. İstediği yolun bant genişliği ve düşük gecikme bakımından avantajlı bir yol olduğuna diğer düğümleri ikna eder. Bu saldırı tipi aslında seçici yönlendirme ve bilgi sızdırma gibi saldırıların ilk adımı olarak da kullanılmaktadır (Ioannou & Vassiliou, 2020).

Savunma tekniği olarak, veri tutarlılığı önerilen algoritmalarla kontrol edilerek şüphelenilen düğümler tespit edilir, ayrıca ağın akış trafiği de detaylı şekilde analiz edilerek zararlı düğümler tespit edilir (Kibirige & Sanga, y.y.). Farklı bir yaklaşım olarak bir düğüm komşu düğümlerdeki trafiği gözlemleyerek davetsiz misafirleri tespit edebilir, bu teknik atlama sayısı (hop count) tekniği olarak da geçer. (Abdullah vd., 2015). Her düğüm için anomali tespiti yapılarak bu saldırı önlenebilir. Mathew ve arkadaşları (Mathew & Terence, 2018) ajan tabanlı tespit, kriptografi tabanlı tespit, sıra numarası tabanlı tespit olmak üzere gider deliği saldırısı için savunma tekniklerine detaylı olarak değinmiştir. Kriptografi tabanlı teknikler için imza tabanlı saldırı tespit sistemi, AODV yönlendirme protokolü üzerinden sayısal imza ve mesaj özütü (hash) fonksiyonu tabanlı tespit sistemi örnek olarak verilebilir.

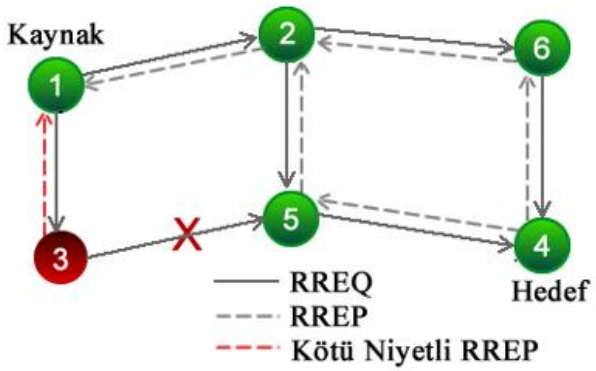
1.3.5. Gri Delik Saldırısı

Bu saldırı türü özellikle Örgü (Mesh) tipteki ağlarda meydana gelir. Örgüsel ağlar statik olmayan ve ağların aktif kablosuz bağlantı noktaları yardımıyla döngü oluşturduğu ağlardır. Saldırganın bazı mesajları göndermeyip, paketleri yayılmasını ağdan düşürerek engellediği saldırı türüdür. Saldırının tespit edilme olasılığını azaltmak için, istediği herhangi bir paketi seçer ve bu paket içeriğini değiştirirken diğerlerini değiştirmeden ağa bırakır. Bu saldırının tespit edilmesi oldukça zordur.

Savunma tekniği olarak AODV (Ad-hoc On-Demand Distance Vector) yönlendirme protokolüne dayanan bir algoritma ile ilk aşamada paket sayıcısı ve eşik değeri (tespit için) kullanılır. Algoritmanın ikinci aşaması sorgu tabanlı olup, ara düğümler kullanılarak saldırganın pozisyonu tespit edilir. Kara Delik, Gri Delik ve Gider Deliği saldırılarını tespit etmek için, dinamik ve merkezi olmayan MANET(Mobil tasarsız ağ) yapılarında Yapay Arı Optimizasyon algoritması ile Yapay Sinir Ağlarını kullanan bir model önerilmiştir(Rani vd., 2020). Testlerde PDR (Paket Delivery Ratio-Paket Dağıtım Oranı) %97,55 seviyesine ulaşmıştır.

1.3.6. Kara Delik Saldırısı

Kara Delik (Black Hole Attack) Saldırısı da tıpkı Gri Delik saldırısı gibi özellikle Örgü tipteki ağlarda en çok gerçekleşen saldırılardır. Bu saldırı bir yönlendirme saldırısı türüdür. Ağa eklenen düşman düğüm, yönlendirme tablosunda güncellemeler yaparak komşu düğümleri veri göndermeye zorlar. Daha sonra tıpkı bir kara delik gibi ele geçirdiği paketleri asla göndermez, yönlendirmez, hepsi ağdan atılır(Ali vd., 2018). Aşağıdaki şekilde Düğüm 1 kaynak düğüm, Düğüm 4 ise hedef düğümdür. Düğüm 1 RREQ (Route Request-Yönlendirme İsteği) paketini gönderdiğinde Düğüm 2 ve Düğüm 3 paketleri ağdan alır. Kırmızı ile gösterilen Düğüm 3, düşman düğüm olup kaynaktan gelen RREQ mesajına hızla karşı cevap verir. Düşman düğüm, en kısa yolun (rotanın) kendi üzerinden geçtiğini belirterek yanıltır. Bundan sonra Düğüm 1, veri paketleri kırmızı düşman düğüm 3'e gönderir, bu düğümden gelen paketleri ağdan düşürür ve imha eder. Ağda birkaç tane düşman düğüm olabilir, buna Çoklu ya da İşbirlikçi Kara Delik Saldırısı adı verilir.

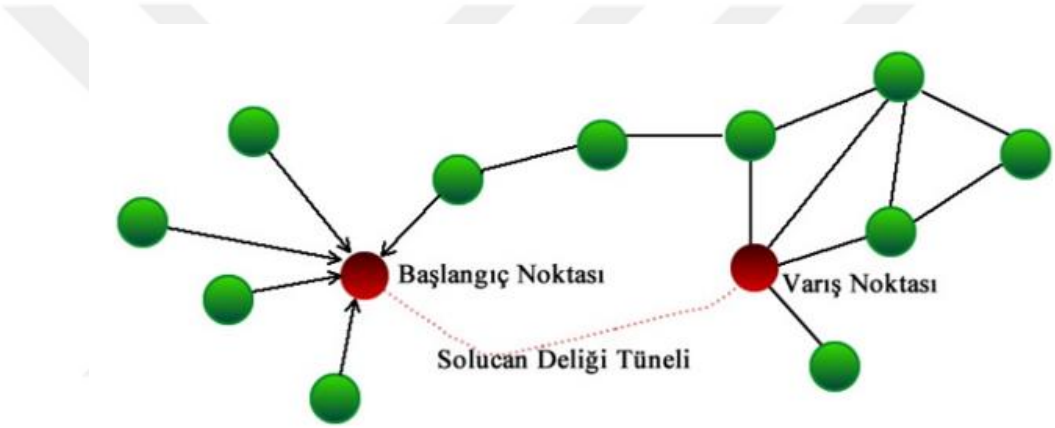


Şekil 1.8: Ağ Katmanı Saldırılarından Kara Delik Saldırısı

Savunma tekniği olarak, ağ trafiği sürekli taranarak ağda kayıp paketlerden şüpheli olan düğümlerin tespiti yapılarak ağdan çıkarılabilir. Gri delik saldırısında değindiğimiz yapay zekâ tabanlı yöntemler de uygulanabilir.

1.3.7. Solucan Deliği Saldırısı

Solucan Deliği (Wormhole Attack) saldırı tipinde aşağıdaki Şekil 1.9'da kırmızı iki düşman düğüm arasında bir bağlantı(tünel) kurulur, bu bağlantıda gecikme süresi düşüktür.



Şekil 1.9: Solucan Deliği Tüneli Saldırısı

Düşman düğüm, aldığı paketleri tünelin karşı tarafındaki düğüme iletir, böylece tünelin karşı tarafındaki düğümler iletişimde oldukları düşman düğümün bir atlama(hop) mesafesinde ya da hemen yanındaki komşusu olduğunu sanır. Bu aşamadan sonra kablosuz ağ, düşman düğümlerin ele geçirdiği paketlerin defalarca gönderilmesi sonucu çökertilir. Düşman düğümler isterse paketleri değiştirir, isterse ağdan düşürerek yok eder, isterse DoS gibi başka bir saldırının başlamasına sebep olur.

Bu saldırı tipi, daha verimli haberleşme için düğümler arasındaki en kısa yolu hesaplayan protokoller için ve bazı MAC teknikleri için (komşu değişimi yapan) tehlikelidir (Ali vd., 2018).

IoT ağları için özel olarak tasarlanan RPL (Düşük güçlü ağlar için yönlendirme) protokolünde solucan Deliği saldırısı ağın 6LoWPAN adaptasyon katmanında

gerçekleşen saldırılarından biridir. Deskmukh ve arkadaşları tarafından önerilen Solucan Saldırısı ve saldırgan için geliştirilen Saldırı Tespit Sistemi Contiki OS işletim sistemi üzerinde Cooja Simülatörü kullanılarak geliştirilmiştir (Deshmukh-Bhosale & Sonavane, 2019).

Seo ve arkadaşları (Seo & Lee, 2012) iki aşamalı bir savunma mekanizması önermişlerdir. İlk aşamada saldırı tespiti için bir yöntem, gelen saldırıya cevap için ikinci bir yöntem önerilmiştir. Bu saldırının tespitinde, komşu düğümlerden yararlanılmaktadır. Her paketin içinde gönderen paketin kimlik bilgileri kanıt olarak yer alır. Arada kalan düğümler haberleşirken kanıtları kontrol ederek komşu düğümlere ait olup olmadığına bakar ve yönlendirmede bu saldırıyı tespit edebilir.

1.3.8. Düğüm Tekrarlanması-Kimlik Sahtekarlığı Saldırısı

Düğüm tekrarlanması (Node Replication) saldırısı ya da diğer adıyla Kimlik Sahtekarlığı (Identity) saldırısında saldırganın yalnızca bir düğüme erişmesi ve elde etmesi yeterlidir. Bu saldırı tipi Fiziksel katmana gerçekleştirilen bir saldırı olarak da ifade edilebilir. Elde edilen düğümdeki şifrelenmiş bilgiler deşifre edilir ve saldırgan tarafından yeniden programlanır. Daha sonra sahte düğümler klonlanarak çoğaltılır ve daha önce planlanan kritik bölgelere konumlandırılır. Tekrar tekrar yeni oturumlar ile ağ meşgul edilip, çökertilir.

Savunma tekniği olarak, statik yapıdaki ağlar için "Komşu merkezli Saldırı Tespit Şeması" önerilmiştir (Ko vd., 2009). Bu şema önceki yaklaşımlardan farklı olarak periyodik olarak kopya düğümleri denetlemektedir. Bu şema sadece gerçek zamanlı tespit yapmakla kalmayıp aynı zamanda daha düşük haberleşme ve bellek maliyeti bakımından avantajlıdır.

1.4. Taşıma Katmanına Yapılan Saldırılar

1.4.1. SYN Akını Saldırısı

Bu saldırı tipinde düşman büyük miktarda SYN istediğini kurban düğüme gönderir. Kurban düğüm, SYN/ACK ile karşı cevap verir. Kurban daha sonra herhangi bir eylem gerçekleştirmeden bekler. Belli bir süre sonra diğer bir SYN bağlantı isteği gönderir. Bu sürecin sonunda birçok tamamlanmamış yarım kalmış TCP protokol bağlantısı

meydana gelir. Ağ bu şekilde sürekli meşgul edilir ve belli bir süre sonra iş göremez. Bu saldırı aynı zamanda bir DoS saldırı tipidir. TCP protokolünün üçlü el sıkışma işlemini kötüye kullanır.

Savunma tekniği olarak SYN çerezleri ile SYN akını engellenebilir. Bağlantı istekleri bu savunma tipinde hedefte tutulmaz, hedef düğüm bağlantının son durumunu tekrar düşmana gönderir ve düğümde bir çereze depolar. Bu yaklaşım ile SYN saldırısı engellenir, düğümde bellek taşması da bu yaklaşım ile önlenir, ağdaki trafik duraksamadan devam eder.

1.4.2. Eşleme Bozulma Saldırısı

Bu saldırı tipinde, gönderici düğüm ile alıcı düğüm arasında senkronizasyonun (eşlemenin) sağlanması amacıyla bir sıra numarası (sequence number) kullanılır. Saldırgan, gönderici düğüm ile alıcı düğüm arasına girerek bağlantıyı kesip sıra numarası üzerinde oynama yapar, sıra numarasını değiştirebilir veya iletişimdeki düğümlere sahte sıra numarası içeren veri paketleri yollar. Bu şekilde, düğümler arasında senkronizasyon bozulur ve düğümler paketleri bilinçsizce tekrar tekrar iletirler.

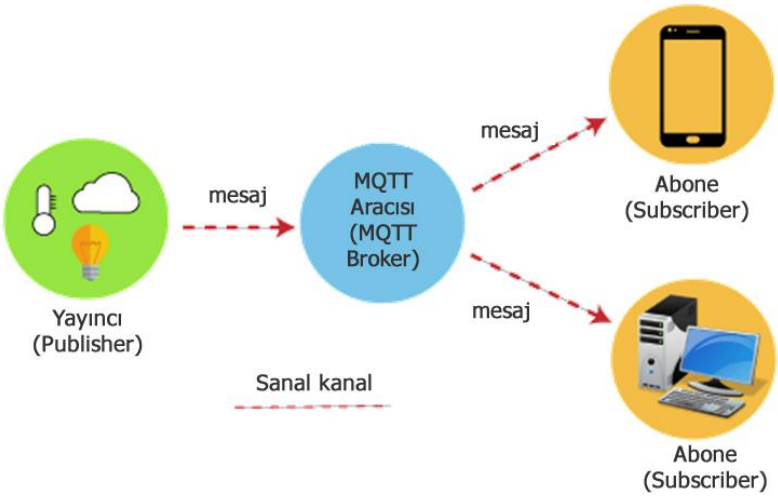
Savunma tekniği olarak, veri paket başlıklarına doğrulama imzası (ya da doğrulama özütü) eklendiğinde bu saldırı önlenecektir.

1.5. Uygulama Katmanına Saldırılar

Uygulama katmanı, çok farklı tipte saldırıların gerçekleştirildiği en çok saldırılardan etkilenen katmandır. Bir algılayıcı ağında, DoS saldırısı örneği olarak bir düğüme çok sayıda mesaj göndererek "Algılayıcı düğümün mesaja boğulması" saldırısı, hiyerarşik bir ağ yapısında el alttaki yaprak düğümün defalarca mesaj göndererek ağı çökerttiği "Yol Tabanlı DoS Saldırısı" gerçekleştirilebilir. Bu saldırıları düşük güçteki IoT ağlarında fazla enerji tüketimi, bant genişliğinin sömürülmesi, ağın meşgul edilerek hizmet verememesi gibi olumsuz sonuçlara neden olur.

1.5.1. Ortadaki Adam Saldırısı

Haberleşen iki taraf arasındaki trafiğin davetsiz bir misafir(saldırgan) tarafından dinlendiği saldırı türüdür. Normal bir istemci – sunucu mimarili bir ağda da gerçekleşebilen eski bir saldırı tekniğidir. IoT mimarisinde Orta-Katmanda bulunan MQTT protokolüne karşı bu saldırı yapılabilir. MQTT protokolü, Yayıncı (Publisher) ve Abone (Subscription) mantığına göre çalışır. Bu protokolde istemci düğümler ile üye düğümler arasında MQTT Aracısı, bir vekil gibi aracı olarak iş görür. Böylece hiçbir detaylı bilgiye sahibi olmadan Yayıncı ve Abone birbirinden ayrılarak mesaj gönderilebilir. Fakat bu kolaylık, bir güvenlik zafiyetine neden olmaktadır. Araya giren bir saldırgan "Ortadaki Adam" saldırısını kolaylıkla gerçekleştirir, yayıncı ve aboneler hakkında detaylı bilgiye sahip olur, haberleşmeyi kontrolü altına alır.



Şekil 1.10: MQTT Mimarisi

Bu saldırı tipine önlem olarak, güncel anahtar yönetim (key management) protokolleri uygulanabilir, ayrıca mesajlar internet gibi açık bir ortam üzerinden gönderilmeden önce gizlilik prensibinin sağlanması amacıyla kriptolanmalı, paket bütünlüğünün korunması için de mesaj özütü/imza teknikleri kullanmak gereklidir.

1.5.2. Hizmet Reddi Saldırısı (DDoS/DoS)

Bu saldırı tipinde, ağı tamamen yok etmek ya da iş göremez duruma getirmek amaçlanmaktadır. Hizmet Reddi ya da Servis Yalanlaması olarak farklı kaynaklarda geçmektedir. Saldırı tek bir kaynaktan geliyorsa DoS, birden fazla kaynaktan eş

zamanlı olarak geliyorsa DDoS olarak adlandırılır. Günümüzde DDoS tabanlı saldırılar daha ağırlıklı görülmektedir.

Düşük güçlü bir IoT ağında düğümün fonksiyonları tamamen engellenebilir ya da azaltılabilir. Şimdiye kadar incelediğimiz saldırıların birçoğu aslında DoS/DoS saldırısıdır, düğümü iş göremez hale getirmek, aşırı mesaj göndererek mesajlaşmasını engellemek ve ağdan düşürmek amaçlandığı için bu kategori altında toplayabiliriz.

Saldırgan bir IoT cihazı veya bir bilgisayarı ele geçirip, içine zararlı kodları yerleştirebilir ve bu cihazlar saldırganın planladığı şekilde zamanı gelince bir kaynağa saldırı düzenlemek üzere devreye alınabilir. Zombi IoT cihaz ya da Zombi bilgisayar dediğimiz bu yöntemle saldırgan herhangi bir yatırım yapmadan ele geçirdiği cihazlar üzerinden istediği hedefe saldırıda bulunur. Örneğin önceden belirlenen bir IP adresine 22.06.2022 saat 20:00'de sürekli mesaj gönderecek şekilde programlarsa, kullanıcının haberi olmadan bu saldırı gerçekleşebilir.

DoS saldırılarını Hacim tabanlı (Volume-Based) saldırılar, Protokol veya Ağ Tabanlı Saldırılar, Uygulama Katmanlı Saldırılar olarak ayırabiliriz.

Hacim tabanlı saldırılar, web kaynaklarına yüksek miktarda trafik gönderirler. Saldırının hacmini ölçmek için saniyede gönderilen bit miktarı yani BPS (Bit Per Second) birimi kullanılır. ICMP (Internet Control Message Protocol) Akını Saldırıları, UDP (User Datagram Protocol) Akını saldırıları ve sahte ağ paketi saldırıları örnek verilebilir. ICMP mesajları normalde ağdaki cihazların durumlarını kontrol etmek için kullanılır, ağ cihazlarına bir ICMP mesajı gönderilerek aktif olup olmadığı kontrol edilir, eğer cihaz aktifse cevap niteliğinde bir echo mesajı gönderir. Bu durumu saldırganlar kullanarak çok miktarda echo mesajı göndererek sunucuyu ya da IoT cihazı hizmet veremez hale getirirler. UDP protokolü tabanlı DoS saldırısında, sunucunun UDP paketlere verdiği cevap kullanılarak, defalarca hedef sunucu UDP paketi bombardımanına tutularak gerçekleştirilir.

Savunma tekniği olarak ICMP paketlerinin cevaplama oranı sınırlandırılabilir. Fakat, bu teknikte normal paket isteklerinin de reddedilme ihtimali vardır, ayrıca UDP akını yeterince büyükse, tutulan sunucu ateş duvarı ve durum tablosu etkilenir ve bu durum ağda tıkanmaya sebep olabilir.

Protokol ve Ağ tabanlı DoS saldırılarını önce bölümde incelemiştik. SYN Akını bu saldırılara örnektir. Saniyedeki paket sayısına bağlı olarak bu saldırının ölçümü yapılır, yani PPS (Paket Per Second) birimi kullanılır.

DoS/DDoS saldırılarını önlemek için sonraki bölümde inceleyeceğimiz makine öğrenmesi tekniklerini kullanan yöntemlere değineceğiz.

İKİNCİ BÖLÜM

SALDIRI TESPİT SİSTEMLERİ

2.1. Giriş

Teknolojinin gelişmesiyle birlikte, ağ ve sistem güvenliğinin önemi artmıştır. Finansal işlemlerin, eğitimin, e-ticaretin tamamen internet üzerinden yürümesi ve internetten başka bir alternatifin olmaması siber güvenliğe yapılan saldırıların artmasına neden olmuştur. Siber güvenlik kavramı teknoloji ilerledikçe hayatımıza daha çok girmeye başlamış ve popüler bir çalışma alanı haline gelmiştir.

Siber güvenlik tanım olarak; bilgisayarları, ağı, programları ve veriyi saldırılardan, yetkisiz girişlerden, değiştirme ve yok etme işlemlerinden korumak için geliştirilen bir dizi teknolojidir (Buczak & Guven, 2016). Symantec 2020 raporuna göre, COVID-19 salgını siber suçlulara ara verilen bir yıl olmadı, aksine hastanelere, sağlık kurumlarına, medikal araştırma tesislerine, bunun yanında medikal personele ve uluslararası sağlık organizasyonlarına yapılan saldırılar bir kademe daha yükseldi (*Symantec Security Summary - June 2020 | Symantec Blogs*, y.y.).

Bilinen saldırıların yanında, her geçen gün bilinmeyen (sıfır gün-zero day) saldırılarla da karşılaşılmaktadır. Sürekli değişen, geliştirilen ve daha önce görülmemiş ciddi güvenlik tehditlerinden dolayı, saldırı tespit sistemlerinin sürekli geliştirilmesi ve yenilenmesi gerekmektedir. Cisco raporuna göre IoT cihazlar operasyonel teknolojide (OT) önemli bir rol oynamaya başladı, IP kameralar kritik sistemleri ve akıllı sistemleri üretim ortamlarında gözlemleyip, önemli verileri bulut ortama aktarmaktadır. Bu gelişme kaliteyi artırarak daha hızlı hareket etmeyi sağlamaktadır, fakat bu gelişme endüstriyel ortamlar için riskler taşımaktadır (stockvisual, 2020).

Siber saldırıların frekansı ve karmaşıklığı gün geçtikçe artırdığından dolayı, saldırıları en verimli şekilde tespit etmek ve önlemek, yeni tip saldırılara karşı koymak için sürekli farklı saldırı tespit sistemleri (IDS) önerilmektedir. Saldırı tespit sistemleri ağ veya sistemi gözetleyerek zararlı aktiviteleri tanımlayıp anında uyarı veren bir yazılım ve donanım kombinasyonu olarak ifade edilebilir. Yanlış alarm vermeyen, yüksek doğruluk oranına sahip bir IDS tasarlamak halen üzerinde çalışılan sıcak konulardan biridir, çünkü saldırılar gün geçtikçe artmakta ve farklılaşmaktadır. Birçok saldırıya

karşı koyma yöntemi düşünülse de yapay zekâ-makine öğrenmesi yaklaşımlarını kullanmaktan başka bir alternatif sıfır gün saldırılarını önlemek için yeterli değildir. Fakat makine öğrenmesi algoritmaları da hemen saldırıları tanıyıp engelleyecek bir konuma tam olarak gelememiştir, veri bilimi uzmanları tarafından farklı parametrelerle, çeşitli hiperparametre denemeleri yapılması ve verinin bir dizi ön işlemden geçirilmesi gerekmektedir.

Burada amaç, insan müdahalesini minimuma indirerek saldırıyı otomatik olarak algılayıp önleyecek güvenli bir sistem tasarlamaktır. Bu bölümde geniş bir literatür çalışması yapılarak saldırı tespit sistemleri incelenmiştir. İmza tabanlı, anomali tabanlı ve melez(hibrit) tabanlı olmak üzere üç tip saldırı tespit sisteminden bahsetmek mümkündür.

IDS'ler saldırının yapılacağı noktaya göre Ağ-Tabanlı (NIDS), Bilgisayar-Tabanlı (HIDS) ve Uygulama-Tabanlı (AIDS) olarak sınıflandırılabilir. Ağ tabanlı IDS'ler (NIDS) ağı, dış dünyadaki saldırılardan korumak amacıyla geliştirilmiştir. Dışarıdan yetkisiz bir kullanıcı sisteme saldırmaya çalıştığından alarm vererek siber güvenlik uzmanını ya da bilişim personelini uyarmaktadır. Yazılım veya donanım tabanlı olan IDS'ler, genellikle ateş duvarının önüne konur. Tek bir algılayıcı IDS birçok bilgisayarı araştırabilir. Yazılım tabanlı saldırı tespit sistemleri kurum veya kuruluştaki her bir sunucuya ağ trafiğini gözetlemek üzere yüklenir. Donanım tabanlı saldırı tespit sistemleri ise özel bir yazılım içererek algılayıcı ile her bir sunucuya trafiği yakalamak ve analiz etmek üzere yerleştirilir.

Bilgisayar tabanlı saldırı tespit sistemleri (HIDS), bilgisayara gelen ve giden paketleri gözetlemek üzere her bir bilgisayara yüklenen tespit yazılımlarıdır. Paketleri gözetleyen sistem, kullanıcı aktivitelerini inceler ve kütük dosyalarını analiz ederek, mesaj özütü algoritmaları (MD5, SHA, RIPEMD) gibi teknikler kullanarak dosya bütünlüğünü kontrol eder. Eğer dosyanın özütü kendi sakladığı özüt ile uyuşmuyorsa alarm verir. Bu tespit sistemleri, sistemdeki dosyaların anlık görüntüsünü alıp, bir önceki görüntü ile karşılaştırır, aralarında fark var ise uyarı verir. Bu sistemlere örnek olarak Cisco Security Agent ve Tripwire yazılımları verilebilir.

Uygulama tabanlı saldırı tespit sistemleri (AIDS) ise bir sistem veya yazılın ajanı olabilir, genelde sunucu bilgisayarlarda yer alarak, uygulama bazlı özel

protokollerdeki haberleşmeyi gözetler. Örneğin, web sunucu ile veritabanı yönetim sistemi arasına yerleştirilerek SQL protokolünü gözetleyebilir.

2.2. Saldırı Tespit Sistemleri

2.2.1. İmza Tabanlı Saldırı Tespit Sistemleri

İmza Tabanlı (Signature-based) veya Kötüye Kullanım (Misuse-Based) Tabanlı saldırı tespit sistemleri (SIDS), bilinen saldırı tiplerine karşı etkilidir, sistem her bir saldırının imzasını veya örüntüsünü kendi veritabanında saklar, eğer veritabanında olan bir saldırı imzasına rastlanırsa alarm verilir. Genelde düzenli ifadeler (regular expression) veya metinsel karşılaştırma algoritmaları kullanır. Örneğin sistemdeki bayt sayısını, 1 olan bitlerin sayısını ya da 0 olan bitlerin sayısını karşılaştırarak saldırı tespit etmeye çalışır. İmza tabanlı IDS'lerde katarlar(stringler), olayların sırası (event sequences), aktivite çizgeleri, saldırı senaryoları ile karakterize edilebilir. Sonlu durum makinaları(Fu vd., 2017), Renkli Petri Ağlar, İlişkilendirme Kuralları, Uzman sistemlerin üretim kuralları kullanılır.

İmza tabanlı sistemler var olan birçok IDS sisteminde kullanılmaktadır, Sektörde en çok kullanılan IDS'lere örnek olarak SNORT (Snort, 2016)(Gaddam & Nandhini, 2017), Suricata (Nam & Kim, 2018)(Thongkanchorn vd., 2013), Bro-IDS (yeni ismi Zeek)(*The Zeek Network Security Monitor*, y.y.), Kismet (*Alerts and WIDS - Kismet*, y.y.), Security Onion (*Security Onion*, y.y.), Sagan (*Sagan User Guide — Sagan User Guide 1.2.2 documentation*, y.y.) verilebilir. Geleneksel ID/IPS sistemlerin dışında WireShark, Nagious, Cacti yazılımları bilgisayar ve ağ tabanlı seviyesinde aktivite izleme sistemleri aktivitelerin veri kütüklerini ayrıntılı takip etme imkanı sağlar.

İmza tabanlı sistemlerin avantajı, hızlı olması ve yanlış alarm oranının düşük olmasıdır, dezavantajları ise sadece bilinen saldırıları engelleyebilmesi, sıfır gün(zero-day) saldırılarını engelleyememesi ve manuel olarak imza veritabanının düzenlemek gerekmesidir.

İmza tabanlı sistemler nokta bazlı çözümlerdir ve heterojen kaynaklardan gelen verinin birleştirilmesinde yetersizlerdir (More vd., 2012).

2.2.2. Anomali Tabanlı Saldırı Tespit Sistemleri

Anomali tabanlı sistemleri (Anomaly Based Intrusion Detection Systems-AIDS) ise ağ ve sistemi modelleyip, normal davranışın dışında bir sapma olduğunda uyarı verir. Bu sistemin en büyük avantajı, daha önce benzeri görülmemiş yeni tip saldırılara karşı tespit yapabilmesidir, yani sıfır gün saldırılarına karşı etkindir. Fakat sistem yanlış alarm verebilmektedir, bu durum bu IDS tipinin en büyük dezavantajı olarak görülmektedir. İmza tabanlı sistemlerde bilinen saldırılar, küçük bir değişime uğratılıp tekrar kullanıldığında tespit etmek zor iken, anomali tabanlı sistemlerde kolaylıkla tespit edilebilir.

Anomali tabanlı sistemlerde saldırıları tespit etmek sınıflandırma problemine dönüşmüştür. İkili ve çoklu sınıf olarak saldırılar sınıflandırılabilir. İkili sınıflandırma, Normal veya Normal olmayan, ya da "saldırı yok" veya "saldırı var" diye iki gruba ayrılır. Çoklu sınıflandırma ise kullanılan veri setindeki kategoriye bağlı olarak sınıflandırılabilir. Örneğin KDDCup99 veri setinde yer alan Denial of Service (DoS), User to Root (U2R), Probe, Root to Local (R2L) saldırıları yer aldığından beş farklı kategoriye göre sınıflandırma yapılmaktadır.

2.2.3. Melez (Hibrit) Tabanlı Saldırı Tespit Sistemleri

Melez tabanlı IDS sistemleri (Hybrid based Intrusion Detection Systems(HIDS), anomali tabanlı ve imza tabanlı sistemlerin avantajlarını birleştirir. Bilinen saldırılar için imza tabanlı IDS, bilinmeyen saldırılar için anomali tabanlı IDS sistemler kullanılır. Bilinen saldırılar tekrar makine öğrenmesi algoritmalarıyla öğrenilmeye çalışılmadığından hem eğitim için zaman kaybedilmemiş olur, hem de yanlış alarm oranı (FAR) düşürülmüş olur. Sıfır gün saldırılarını sistem tespit edebilir. Günümüzde birçok sistem, melez tabanlı IDS yapısını kullanmaktadır.

Buczak ve arkadaşları(Buczak & Guven, 2016) siber analitik için Makine Öğrenmesi ve Veri Madenciliği metotlarını detaylı şekilde sunmuşlardır. ML/DM algoritmaları ile İmza tespiti, anomali tespiti ve hibrit tespiti bu çalışmada incelenmiştir. Yapay Sinir Ağları, Bulanık Mantık, Bayes Ağları, Karar Ağaçları, Kümeleme, Evrimsel Hesaplama, Destek Vektör Makineleri, Saklı Markov Modeli, Tümevarımsal öğrenme(Inductive Learning), Sıralı Örüntü Madenciliği, Topluluk Öğrenmesi metotları açıklanmış ve her metot ile ilgili iki veya üç makale özetlenmiştir. Fakat

Yarı Eğiticili öğrenme ve Derin Öğrenme algoritmaları çalışmada yer almamaktadır ayrıca KDDCup 99 ve NSL-KDD gibi genel veri setlerine sadece yer verilmiştir.

Tablo 2.1: Saldırı Tespit Sistemlerinin Karşılaştırılması

	İmza Tabanlı	**Anomali Tabanlı**	**Hibrit Tabanlı**
Avantajlar	Sadece bilinen saldırıları tespit eder, Düşük Yanlış alarm oranı(FAR), Örüntü ve imza kullanarak hızlı tespit zamanı	Sıfır gün saldırılarını tespit etme	İmza tabanlı ve Anomali tabanlı IDS'lerin avantajlarını birleştirir. Bilinen saldırıları imza tabanlı IDS ile tespit ederken, sıfır gün saldırılarını ise Anomali tabanlı IDS ile tespit eder.
Dezavantajlar	Sıfır gün saldırılarını tespit edemez. Veritabanının manuel olarak sık sık güncellenmesine ihtiyaç duyar. Heterojen kaynaklardan gelen verinin bütünleştirilmesinden etkili değildir.	Yüksek FAR oranı. Makine öğrenmesi algoritmaları, eğiticili öğrenmelerde normal dışı davranışları tespit edebilmek için eğitim ve test zamanına gereksinim duymaktadır. Şifrelenmiş paketler tespit edilemez.	Yanlış Alarm oranını (FAR) düşürerek, sıfır gün saldırılarını tespit eder.

Chaabouni ve arkadaşları (Chaabouni vd., 2019) ise Ağ Tabanlı Saldırı Tespit Sistemleri(NIDS) üzerine odaklanmış ve NIDS araçları ve veri setleri

karşılaştırılmıştır. Ayrıca, IoT tabanlı NIDS için öğrenme teknikleri incelenmiştir fakat referans verilen verisetleri güncel değildir. Bu çalışmada, sınırlı sayıda veri setleri karşılaştırılmış ve anomali tespitinde derin öğrenme yaklaşımlarından kısmi olarak bahsedilmiştir.

Tablo 2.2: En çok referans alan son inceleme makalelerinin karşılaştırılması

Makale	**Atıf sayısı**	**Yıl**	**İncelenen kısımlar**	**İncelenmeyen/Eksik kısımlar**
Buczak (Buczak & Guven, 2016)	1241	2016	Eğiticili, eğiticisiz, topluluk metotları	Yarı eğiticili, derin öğrenme metotları
Chaabouni (Chaabouni vd., 2019)	106	2019	Derin öğrenme ve Makine Öğrenmesi	Sınırlı Veriseti
Bhuyan (Bhuyan vd., 2014)	471	2014	Eğiticili, eğiticisiz, yarı eğiticili ve topluluk metotları	Derin Öğrenme metotları, Sınırlı ve güncel olmayan veri setleri
Kraisat (Khraisat vd., 2019a)	87	2020	Eğiticili, eğiticisiz, yarı eğiticili ve topluluk metotları	Derin Öğrenme metotları, Sınırlı veri seti
Zarpelao (Zarpelão vd., 2017)	404	2017	IoT'deki anomali tespit için sadece istatiksel metotlar.	Makine Öğrenmesi, Derin Öğrenme, veri seti kullanılmamış
Ferrag (Ferrag vd., 2020)	55	2020	Sadece Derin Öğrenme metotları (RNN, CNN, Derin AE, DBM, DBN, RBM, DNN), Bot-IoT ve CSE-CIC-IDS20108 Verisetleri	Makine Öğrenmesi metotları

Bhuyan ve arkadaşları (Bhuyan vd., 2014) yaptıkları çalışmada yapısal ve geniş bir şekilde saldırı tespit sistemleri metotlarını tartışmış ve karşılaştırmıştır. Ayrıca; istatiksel tabanlı, sınıflandırma tabanlı, kümeleme ve aykırı değer tabanlı, esnek hesaplama(soft-computing) tabanlı, bilgi tabanlı, füzyon tabanlı ve topluluk öğrenme tabanlı anomali tespit metotları karşılaştırılmıştır. Fakat 2014 yılında yayınlanmasına rağmen, sınırlı ve güncel olmayan veri setleri üzerine yapılan çalışmalara referanslarda değinilmiştir. Derin öğrenme tabanlı yaklaşımlar 2014 yılında bilinmesine rağmen, bu çalışmada bahsedilmemiştir.

Khraisat ve arkadaşları (Khraisat vd., 2019b) tarafından yapılan çalışmada ise IDS sistemleri taksonomi olarak incelenmiş ve güncel veri setleri açıklanmıştır. Özellikle bu çalışmanın diğer çalışmalardan farkı, saldırganlar tarafından kullanılan kaçınma tekniklerine değinmesidir. İstatiksel tabanlı, bilgi ve makine öğrenmesi tabanlı yaklaşımlar açıklanmıştır fakat sınırlı çalışma referans verilmiştir.

ÜÇÜNCÜ BÖLÜM

ANOMALİ TABANLI SALDIRI TESPİTİ

3.1. İstatistik Tabanlı Teknikler

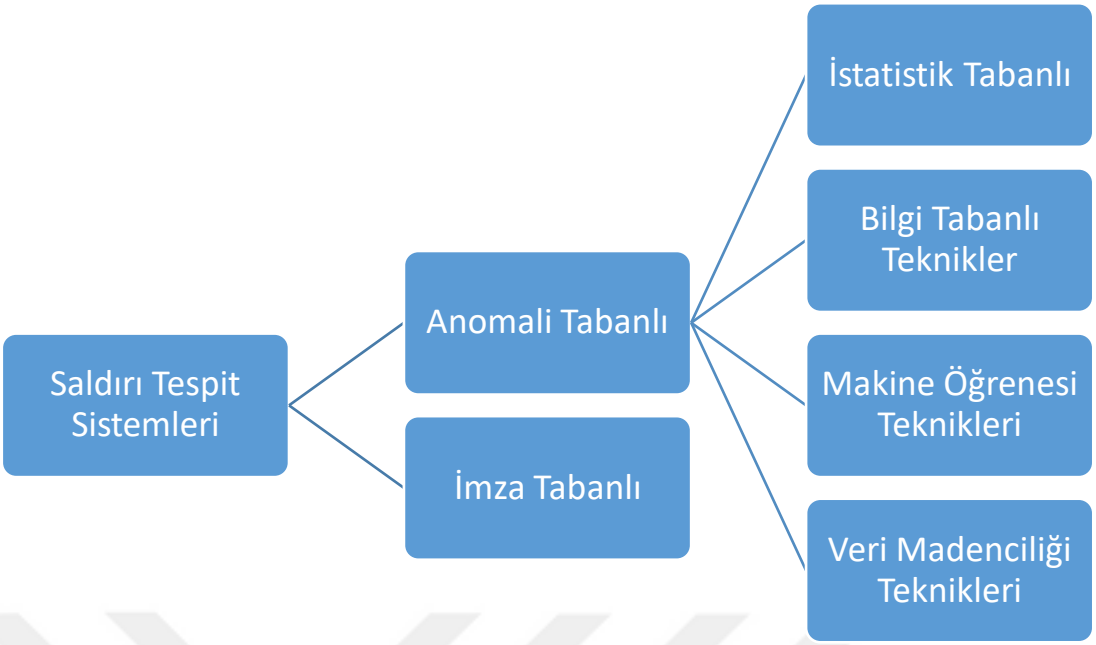
İstatistiksel temelli saldırı tespit sistemleri, normal davranış profili için bir dağıtım modeli oluşturur. İstatistiksel teknikler genelde paketlerin medyan, ortalama, standart sapma gibi istatistiksel ölçülerini dikkate alır. Diğer bir deyişle veri trafiğini araştırmak yerine, veri akışının parmak izi olan her bir paket gözlemlenir. İstatistiksel AIDS'ler normal davranışla şu anki davranış arasındaki fark tiplerini tanımlar. İstatistiksel tabanlı teknikler tek değişkenli veri (univariate), çok değişkenli veri (multivariate), zaman Serileri Modeli olabilir.

Bilgi sistemlerinde Tek değişkenli istatistik tekniği, istatistiksel norm profili tek bir davranış ölçüsü tarafından oluşturulmuş ise kullanılır. Tek değişkenli IDS her bir metrikte anormalliklere bakar. Diğer taraftan izinsiz girişler genellikle birden fazla faaliyet ölçüsünü etkiler.

Çok değişkenli istatistik tekniğinde ise değişkenler arasındaki ilişkiyi anlamak için iki veya daha fazla metriğin arasındaki ilişkiye bakılır. Bu tekniğin en büyük dezavantajı, çok boyutlu veri için dağılımları tahmin etmenin zorluğudur (Ye vd., 2002).

3.2. Bilgi Tabanlı Teknikler

Bilgi tabanlı Teknikler (Knowledge Based Techniques), **u**zman sistemler metodu olarak da adlandırılabilir, kurallar kümesinin belirlenmesi bakımından insan bilgisi taban alınarak standart profil modeli oluşturulur. Standard normal profilden farklı davranışlar, izinsiz giriş olarak algılanır. Diğer AIDS sınıflarından farklı olarak kurallar kümesinin oluşturulması bakımından bu model insan bilgisine dayalı olarak belirlenir (More vd., 2012). Uzman Sistemler, Petri Ağları, İmza Analizi, Sonlu durum makineleri bu gruba dahil edilebilir.



Şekil 3.1: Saldırı Tespit Sistemleri Sınıflandırması.

3.3. Makine Öğrenmesi Teknikleri

Eğiticili öğrenmede (Supervised Learning) her bir özellik veya bağımsız değer için veri setleri etiketlenmiştir. Giriş ve çıkış çiftlerinden oluşan veri seti kullanılır. Veri seti eğitim ve test olmak üzere iki kısma ayrılır. Eğitim aşamasında normal kullanıcı veya saldırganı örüntüsünü sistem öğrenir. Test aşamasında, STS sistemim doğruluğu, performansı test edilir. Bu algoritmalara örnek olarak Destek Vektör Makineleri, Naive Bayes, Karar Ağaçları (Decision Tree) verilebilir.

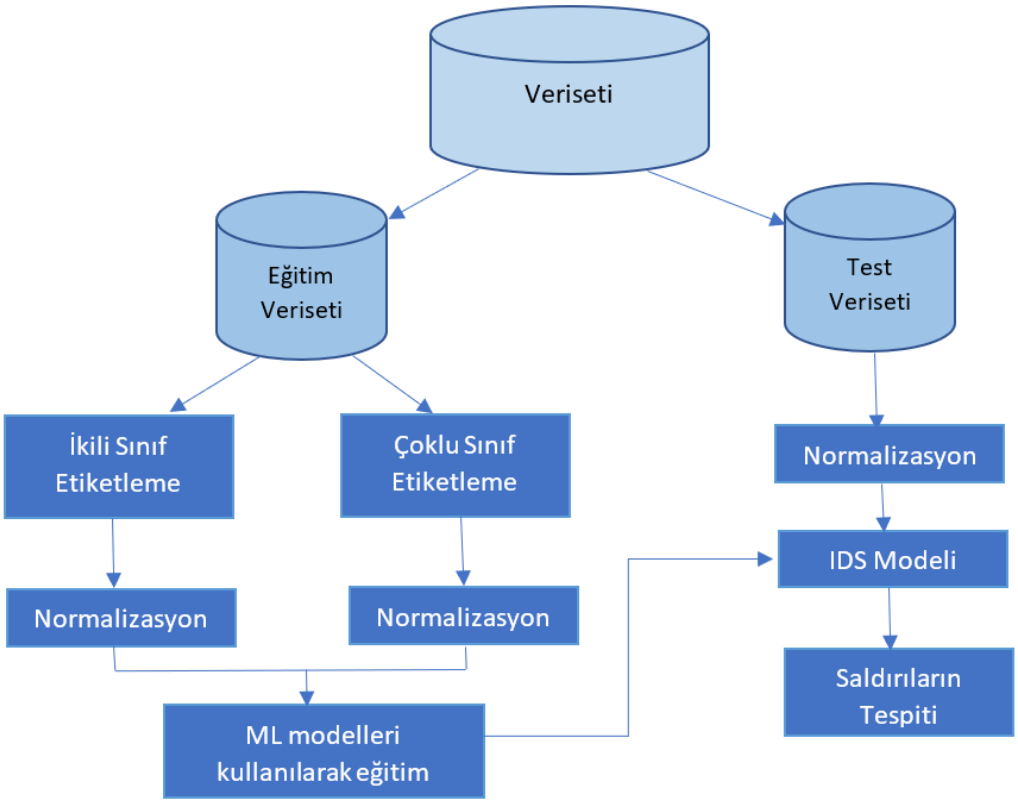
Eğiticisiz öğrenme (Unsupervised Learning) ise, Eğiticili öğrenmenin aksine veri seti etiketlendirilmemiştir. Sadece giriş verisi vardır, verilen bir giriş verisinini doğru çıkışı bilinmemektedir. Aynı gruptaki veri noktaları arasındaki benzerlik göz önüne alınarak kümeleme(clustering) teknikleri kullanılır. Saldırı tespiti için kümelendirme en popüler tekniklerden biridir. Eğiticili öğrenme ile karşılaştırıldığında doğruluk oranı daha düşüktür. Yapay Sinir Ağlarından SOM (Self organization map) ve Adaptive Resonance örnek verilebilir.

Yarı Eğiticili öğrenmede (SSL-Semi Supervised Learning) ise etiketli verilerin yanında etiketsiz verilerin de olduğu bir öğrenme tekniğidir. Genelde verinin çok az olduğu durumlarda kullanılır, bu algoritmalar etiketlenmemiş örnekleri kullanarak performansı genellikle artırırlar. Beklenti maksimizasyonu (Expectation

Maximization), Güçlendirilmiş yarı eğiticili öğrenme(Boosted semi supervised learning), yarı eğiticili SVM(Laplace düzenlemesi ile SVM), ortak Eğitim (co-training), kendi kendine eğitim(self-training), çizge tabanlı metodlar (Label propagation), Min-cuts, MRFs, GRFs, LDS, SGY, SSM, ASSEMBLE, Mixture of Experts, EM-Naïve Bayes, TSVM, S3VM, Gaussian Process (Kumar Mallapragada vd., y.y.).

Takviyeli öğrenme (Reinforcement Learning) ise her bir eylemde modelin öğrendiği özel bir makine öğrenmesi yöntemidir. Model her doğru davranışında ödüllendirilir, yanlış davranışında ise cezalandırılır ve böylece örüntülerden öğrenilir. Her seferinde bilinmeyen veri üzerinde daha doğru karar alınır.

Sınıflandırma tanım olarak, yeni gözlemin hangi kategoriye ait olduğunu tanımlama işlemidir. K-means, Destek Vektör makineleri (SVM), Yapay Sinir ağları (ANN) gibi algoritmalar örnek verilebilir.



Şekil 3.2: Anomali Tespiti için IDS Tasarımı

Sınıflandırma tabanlı anomali tespit metotları, etiketlenen eğitim örneklerinin kullanımından dolayı eğiticisiz metotlara (kümeleme gibi) göre daha iyi sonuçlar vermektedir. Geleneksel sınıflandırmada yeni gelen bilgi tüm veri seti ile birleştirilip, tüm veri seti tekrar tekrar eğitilir, bu işlem uzun zaman almaktadır. Bu sorunun çözümü için artırımlı sınıflandırma kullanılıp verim artırılmaktadır. Sınıflandırma algoritmaları popüler olmasına rağmen, veri setinde sızıntı ile ilgili bilgiler yoksa tahmin etme ve tespit etmede başarısız olabilir (Bhuyan vd., 2014).

3.3.1. Naive Bayes

Naive Bayes algoritması denetimli öğrenme kategorisinde yer alan bir sınıflandırma algoritmasıdır. Thomas Bayes tarafından 18. Yüzyılda geliştirilen bu algoritma özellikler arasında bağımsızlık varsayımına dayalıdır. Önceki bilgiden yola çıkarak hipotezin olasılığını bulmamıza sağlar. İstatistik terimlerinde Bayes Kuralı ya da Bayes Kanunu olarak geçen bu teoremde bir olayın olasılığı tanımlanır. Olaya bağlı olarak önceki bilgilerimizden hareketle olasılık hesaplarız. Aşağıdaki denklem ile tek bir özellik için ifade edilir.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{3.1}$$

P(A|B) : B olayı doğru tahmin edildiğinde A olayının gerçekleşme olasılığı.

P(B|A) : A olayı doğru tahmin edildiğinde B olayının gerçekleşme olasılığı

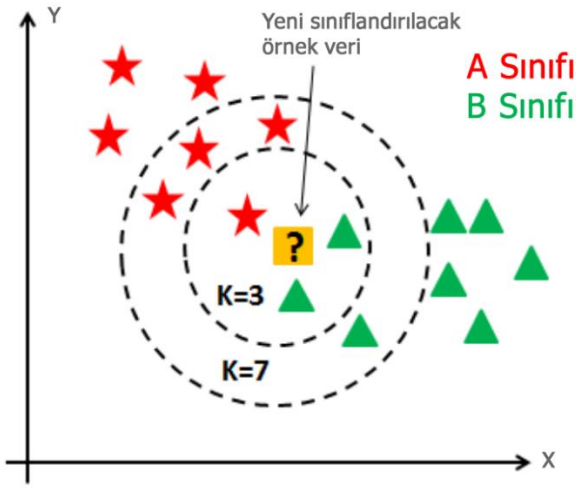
P(A) : A olayının gerçekleşme olasılığı

P(B) : B olayının gerçekleşme olasılığı

Bu algoritmanın en büyük avantajı küçük bir verisetinde iyi sonuçlar verebilmesidir ve formülü basit olduğundan kolayca anlaşılabilmesidir. Bu algoritma metin sınıflandırmada, duygu analizinde, tavsiye(öneri) sistemlerinde, e-posta önemsiz e-posta filtrelenmesi gibi birçok alanda kullanılmaktadır.

3.3.2. k En Yakın Komşu

K En Yakın Komşu (KNN), Makine Öğrenmesinde denetimli öğrenme kategorisine ait popüler sınıflandırma algoritmalarından biridir. Ayrıca KNN, sistemin eğitim verilerini genelleştiren tembel(lazy) bir öğrenme algoritmasıdır.



Şekil 3.3: kNN Algoritmasında Sınıflandırma

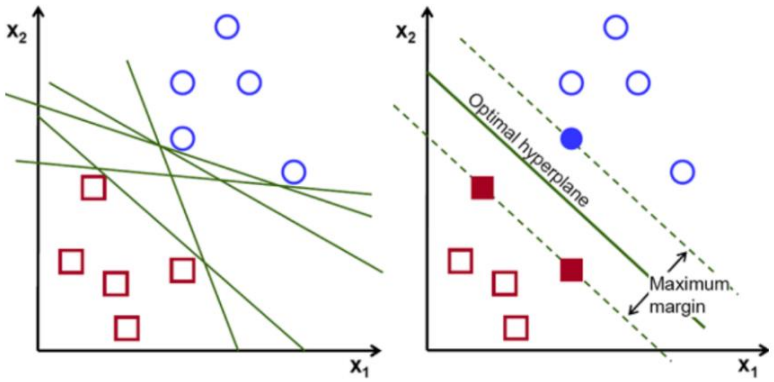
Çevrimiçi öneri sistemleri, örüntü tanıma, veri madenciliği ve saldırı tespiti, bu algoritmanın sınıflandırma amacıyla uygulandığı alanlardan bazılarıdır. Algoritma, verileri sınıflandırmak için etiketlenmemiş test vektörü ile en yakın k komşu arasındaki mesafeyi hesaplar. KNN algoritmasında k değeri kullanıcı tarafından seçilir ve mesafe hesabı Öklid, Manhattan veya Minkowski yöntemlerinden biri ile yapılır (Abu Alfeilat vd., 2019).

3.3.3. Destek Vektör Makineleri

Destek Vektör Makineleri (Support Vector Machines-SVM), değişkenler arasındaki örüntülerin bilinmediği veri setlerindeki sınıflandırma problemleri için önerilmiş bir makine öğrenmesi yöntemidir. İstatiksel öğrenme teorisine ve yapısal risk minimizasyonuna dayanmaktadır (Taş, 2016).

SVM, özellik uzayını bulunan hiperdüzlemler aracılığı ile bölen bir sınıflandırıcıdır. Amaç her bir sınıfın veri noktaları arasındaki uzaklığı maksimize etmektir. Eğer iki

sınıf ayrılamazsa gevşek(slack) değişlenler eklenerek ve maliyet parametresi çakışan veri noktaları için atanır.



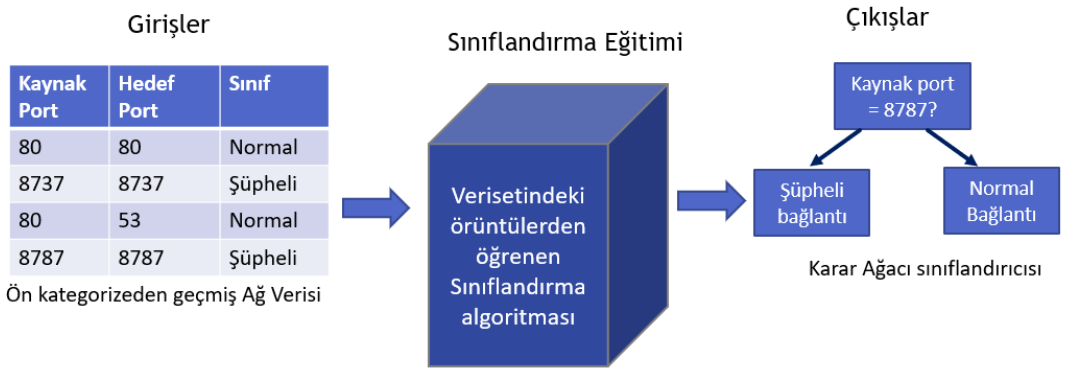
Şekil 3.4: SVM ile İki Boyutlu Uzayda Sınıflandırma İşlemi.

Kuadratik En İyileştirme (Quadratic Optimization) ile maksimum marjın ve hiper düzlemin yeri $O(n^2)$ karmaşıklığında çalışma zamanında belirlenir. Çekirdek fonksiyonu ile sınıflandırma yüzeyi bölünür. Desktek vektör makinelerinde çekirdek fonksiyonları doğrusal(lineer), polinomsal, Gaussian Radial Tabanlı (RBF) olabilir.

3.3.4. Karar Ağaçları

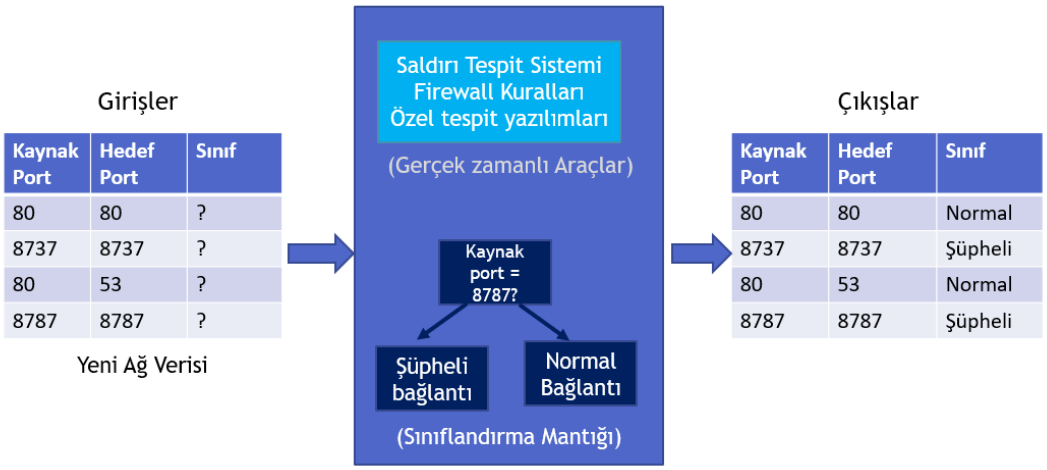
Karar Ağaçları (Decision Trees-DT), dallardaki düğümlerden ve yapraklardan oluşan bir veri madenciliği metodudur. Karar Ağacı algoritmalarına örnek olarak ID3, C4.5, CART, Best First Tree ve J48 verilebilir. Bu algoritmalar içinde en popular olanı J48 algoritmasıdır, veriyi kategorik olarak ve sürekli bir biçimde sınıflandırır.

Şekil 3.5'de Karar ağacına örnek olarak bilinen şüpheli 8787 numaralı port için tek bir karar alma kuralı gösterilmiştir. Sınıflandırma algoritmaları bir veri setinden örüntüleri tanıyarak ve bu bilgiyi kullanarak şekildeki gibi bir karar ağacı oluşturur (Markey, 2020).



Şekil 3.5: Karar Ağacı Sınıflandırma Giriş ve Çıkışlar

Şekil 3.5'te karar ağacı eğitim aşaması görülmektedir. Şekil 3.6'da Karar Ağacı sınıflandırma mantığı gösterilmiştir, burada öğrenme algoritmasından elde edilen kurallar kullanılarak gerçek zamanlı ağ araçları ile yeni veri kategorize edilmektedir.



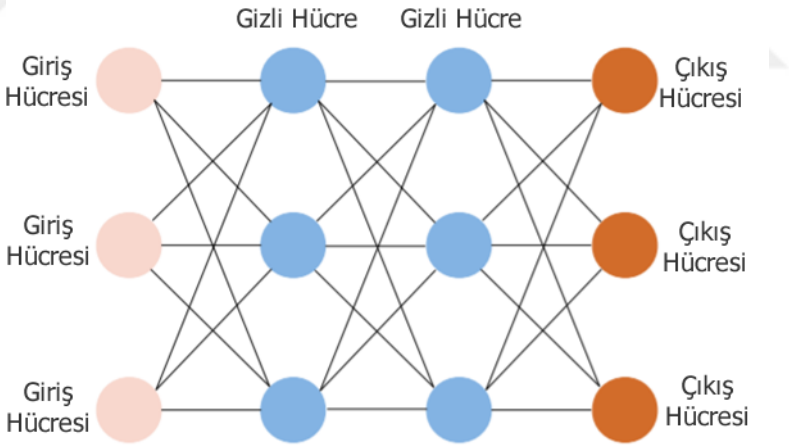
Şekil 3.6: Karar Ağacı Eğitimi ve Giriş-Çıkış Verileri

Ağaç üretme algoritmaları kullanılarak kısa bir zaman dilimi içerisinde iteratif olarak Karar ağaçları oluşturulur. En optimum ağacı hesapsal olarak oluşturmak imkansızdır, çünkü özellik kümesine bağlı olarak olası ağaç sayıları üstel olarak büyümektedir. Hunt'ın algoritması gibi çoğu Aç Gözlü algoritma (Greedy Algoritmaları) farklı kural setine bağlı olarak rekürsif (özdevinimli) bir şekilde hızlı, efektif fakat optimal ağaçları entropiyi minimize ederek üretirler (Markey, 2020).

3.3.5. Yapay Sinir Ağları ve Derin Sinir Ağları

Derin Sinir Ağları, daha önceki yıllarda önerilen Yapay Sinir Ağlarının daha fazla katmanlı ve yeni aktivasyon fonksiyonlarına sahip hali gibi düşünülebilir. Karmaşıklığı oldukça fazla olan problemlerde özellikle de büyük veri kavramının ortaya çıkmasıyla derin sinir ağları modellerine ilgi de artmıştır. Görüntüyü sınıflandırma, Nesne Bulma, Doğal Dil İşlem gibi birçok alanda problemlerin çözümü için derin sinir ağları gerekmektedir.

Derin sinir Ağlarında katman sayısı ve her katmandaki nöron sayısı çok fazla olduğundan özellik çıkarımı gibi klasik makine öğrenmesi algoritmalarındaki aşama gerçekleştirilebilmektedir. Derin Sinir ağları ile çok büyük veriler üzerinde çalışmalar yapılabilmektedir. Yeni nesil Grafik İşlem birimleri (GPU) aracılığı ile daha fazla çekirdek üzerinde paralel olarak problemlerin daha kısa sürede çözümü mümkün hale gelebilmektedir. Özellikle NVidia CUDA ekran kartları üzerinde yer alan grafik işlemciler ile çok çekirdekli işlemciden daha iyi sonuçlar alınabilmektedir.

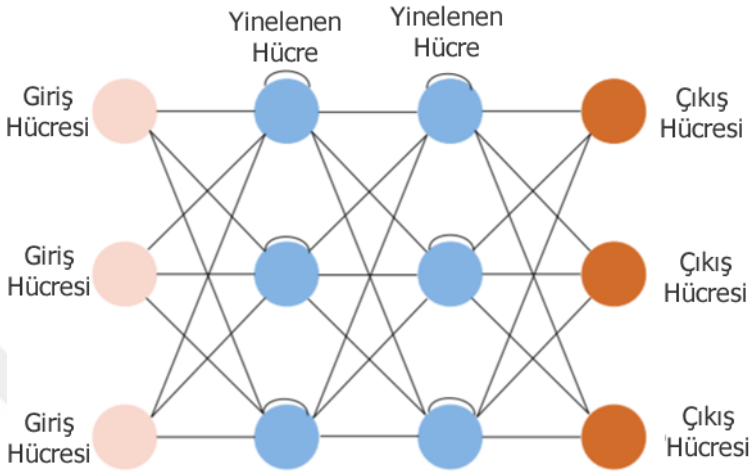


Şekil 3.7: Derin Sinir Ağları

3.3.6. Yinelenen Sinir Ağları

Yinelenen Sinir Ağları (Recurrent Neural Networks-RNN) bir önceki girişi hafızasında saklayabilen ilk derin öğrenme algoritmalarından biridir. RNN'in arkasındaki fikir sıralı bilginin kullanımıdır, klasik yapay sinir ağlarında tüm girişler ve çıkışlar birbirinden bağımsız olarak kabul edilir. Örneğin bir cümledeki sonra gelen

kelimeyi tahmin etmek isterseniz, ondan önce gelen kelimeyi bilmeniz gerekmektedir. RNN modelinde yinelenen(tekrarlanan) denmesinin sebebi sıranın her bir elemanı için aynı işlemi gerçekleştirmesinden gelir, çıkış önceki hesaplamalara bağlıdır ve sistemdeki "hafıza" sayesinde o ana kadar yapılan hesaplamaları zaten bilirsiniz.



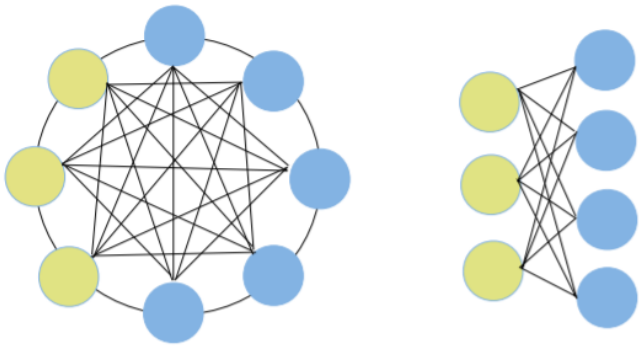
Şekil 3.8: Yinelenen Sinir Ağı

3.3.7. Boltzmann Makinesi ve Kısıtlı Boltzmann Makinesi

Boltzmann Makinesi, orijinal veri setinden olasılık dağılımlarını içeren ve daha önce görülmeyen veri hakkında çıkarımlar yapan eğiticisiz modellerden biridir. Boltzmann Makinesi, görülen katman olarak adlandırılan bir giriş katmanına ve bir veya daha fazla gizli katman(ya da katmanlara) sahiptir. Boltzmann Makinesi, yalnızca diğer katmanlardaki diğer nöronlarla değil, aynı katmandaki nöronlara da bağlı nöronlarla sinir ağları kullanır. Her şey birbirine bağlıdır, bağlantı iki yönlüdür, giriş(görülen) katman nöronları birbirine bağlıdır ve gizli katman nöronları da birbirine bağlıdır.

Boltzmann makinesi, giriş verisi beklememektedir, kendisi veriyi üretmektedir. Nöronlar gizli ve görülebilir olup olmadıklarına bakmaksızın bilgi üretirler. Boltzmann makinesinde tüm nöronlar aynıdır, gizli ve görünen nöronlar arasında bir ayrım yoktur. Düğümler stokastik karar verebilirler, açık veya kapalı olabilirler. Sera örneği ile bu model açıklanabilir, nem, sıcaklık, hava akışı ve ışık gibi farklı parametrelere bir serada ihtiyaç duyulmaktadır. Boltzmann makinesi de tıpkı sera gibi bir sistemdir, nem, sıcaklık, hava akışı, toprak durumu ve ışık arasındaki ilişkileri öğrenir.

Kısıtlı Boltzman Makinesi (Restricted Boltzman Machine-RBM) ise yönsüzdür ve bu model giriş ve gizli katman olmak üzere yalnızca iki katmana sahiptir. Simetrik İki Parçalı Grafik (Symmetrical bipartite graph) olarak da adlandırılır.



Şekil 3.9: Boltzmann Makinesi ve Kısıtlı Boltzmann Makinesi

3.3.8. Derin İnanç Ağları

Derin İnanç Ağları (Deep Belief Networks-DBF), makine öğrenimi ve sinir ağları ile istatistiğin bir karışımıdır. DBF'ler birden çok katmandan oluşur, burada katmanlar arasında bir ilişki bulunmaktadır ama değerler arasında yoktur. Temel amaç sistemin verileri farklı kategorilere ayırmasına yardımcı olmaktır.

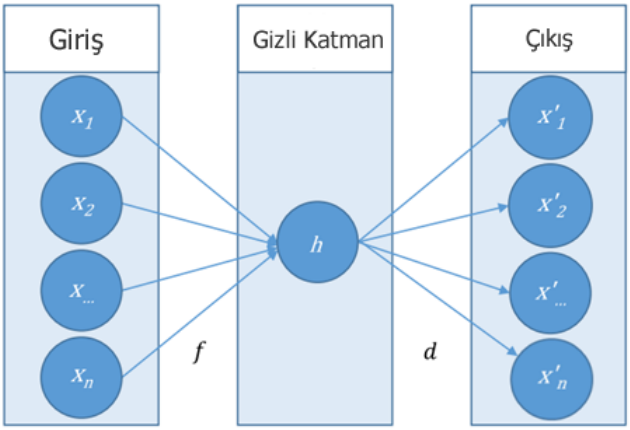
Derin İnanç Ağları, RBM gibi eğiticisiz ağlardan oluşur. Burada her bir alt ağın görünmez katmanı, bir sonrakini görünür katmanıdır. Gizli ve görünmez katmanlar birbirine bağlı değildir ve koşullu olarak bağımsızdır.

Özetle, Kısıtlı Boltzmann Makinelerinin (RBM) birbiri ardına eklenmesiyle Temel İnanç Ağları (DBF) yapısı oluşturulur. Eğitim yapılırken de Kısıtlı Boltzmann Makineleri sırasıyla eğitimden geçirilip sonuç alındığı bir dizi işlem gibi düşünülebilir. Daha çok resim ve video işleme, ses tanıma gibi alanlarda kullanılır.

3.3.9. Derin Oto Kodlayıcılar

Derin Otokodlayıcılar (Deep autoencoder-DAE) modeli özellik çıkarma tabanlı bir eğiticisiz yapay sinir ağı algoritmasıdır. DAE algoritmaları, öğrenerek boyut azaltmayı

amaçlarlar. Ayrıca, giriş verisinin sıkıştırılmış gösteriminden en iyi özellikleri öğrenirler. Tipik bir Tekli oto-kodlayıcılar giriş katmanı, çıkış katmanı ve gizli katmandan oluşur. Bu modelde, çıkış katmanı ve giriş katmanı aynı boyuttadır, gizli katman ise giriş katmanından daha küçük boyuttadır.



Şekil 3.10: Derin Oto-Kodlayıcı

Literatürde anomali tespitine yönelik makine öğrenmesi metodunu kullanan diğer saldırı tespiti çalışmalarına da değinelim. Verma ve arkadaşları(Verma & Ranga, 2020) sınıflandırıcının performansını IoT saldırı tespit sisteminde istatiksel olarak analiz etmişlerdir, sınıflandırıcının çalışmasını Raspberry Pi IoT donanımı üzerinde test etmilerdir. Friedman Testi (parametrik olmayan istatistiksel bir test) ve Nemenyi testi performans sonuçları için kullanılmıştır. Topluluk öğrenme algoritmaları olan Rastgele Orman (Random Forest), Adaboost (AB), Aşırı Gradyan Artırma (Extreme Gradient Boosting-XGB), Gradyan Artırma Makinesi (Gradient Boosting Machine-GBM), Aşırı Rastgele Ağaçlar (Extremely Randomized Trees-ETC) topluluk öğrenme algoritmaları ve tekli sınıflandırıcılar olarak CART ve MLP (Multi-layer Perception) algoritmaları incelenmiştir. CIDDS-001, UNSW-NB15 and NSL-KDD verisetleri değerlendirmede kullanılmıştır.

Karataş ve arkadaşları (Karatas vd., 2020), CSE-CIC-IDS 2018 veri seti üzerinde Rastgele Orman(RF), Karar Ağaçları(DT), Adaboost, K-En Yakın Komşu(KNN) ve Doğrusal Ayrımcı Analizi(LDA) makine öğrenmesi algoritmaları kullanmışlardır. Dengesiz veri setinden dolayı, dengesizlik oranı SMOTE tekniği ile azaltılmıştır. Bu metot KNN algoritması kullanmaktadır. ADASYN ve RandomoverSampler metodları

da kullanılmıştır. Deneysel sonuçlarda, doğruluk oranı ADASYN %5 ile SMOTE tarafından üretilenden daha düşük sonuçlar alınmıştır.

Kitsune (Mirsky vd., 2018) sistemi, kendi tabirleriyle tak ve çalıştır özelikli ağ tabanlı IDS(NIDS) sistemini saldırıları tespit etmek için geliştirmişlerdir. Mirsky ve arkadaşları diğer çalışmalardan farklı olarak, veri seti ve gerçek zamana yakın makine öğrenme açısından yeni bir yaklaşım sunmuşlardır. Bu çalışmada, genel verisetleri yerine IP kamera video ağına yapılan saldırılar toplanmış ve incelenmiştir. Kitsune algoritması (Kitnet) kullanılarak yapay sinir ağları topluluğu ile anormal trafik örüntüleri tespit edilmeye çalışılmıştır.

Vinayakumar ve arkadaşları(Vinayakumar vd., 2019), Derin Sinir Ağları (DNN) tabanlı IDS ile bilinen ve bilinmeyen saldırıları karşı verimli bir şekilde tespit etmeye çalışmışlardır. Önerilen DNN modelini test etmek için KDDCup99, UNSW-NB15, Kyoto, WSN-DS ve CICIDS 2017 verisetleri kullanılmıştır. Ayrıca, Scale-Hybrid-IDS-AlertNET isimli yeni bir hibrit DNN çatısı önerilmiş, gerçek zamanlı trafikteki anormallikleri de tespit etmek amaçlanmıştır.

Shone ve arkadaşları (Shone vd., 2018) ise eğiticisiz öğrenme için Simetrik olmayan derin oto-kodlayıcı isimli yeni bir Derin Öğrenme tekniği sunmuşlardır. Ayrıca, yeni bir derin öğrenme sınıflandırıcısı önermişlerdir. Deneysel çalışmalarda KDDCup 99 ve NSL-KDD verisetleri kullanmışlardır.

Diro ve arkadaşları(Diro & Chilamkurti, 2018) tarafından önerilen çalışmada, sıfır gün saldıırlarını ve varolan saldırıların mutasyona uğratılmış olanları IoT ortamında tespit etmek Derin Öğrenme teknikleri IDS için önerilmiş ve geleneksel makine öğrenmesi algoritmaları ile karşılaştırılmıştır. Bu çalışmanın diğer çalışmalardan farkı hem Merkezi hem de dağıtık yapıdaki saldırı tespit sistemini değerlendirmesidir. Dağıtık Sis(Fog) ağları modellerin eğitiminden sorumludur, koordinatör yönetici düğüm ise paylaşım ve optimizasyon için işbirliği parametrelerini tutmaktadır. Elde edilen sonuçlara göre; Derin öğrenme tabanlı yaklaşım kullanılarak dağıtık yapı tercih edildiğinde merkezi yapıya göre daha iyi sonuçlar alınmıştır. KDDCup 99, ISCX, NSL-KDD veri setleri deneysel çalışmalarda kullanılmıştır.

Pajouh ve arkadaşları (Pajouh vd., 2019) IoT Backbone ağları için bir saldırı tespit sistemi önermiştir, bu sistem LDA ve Bileşen Analizi (CA) kullanarak iki katmanlı boyut azaltma uygulamaktadır. Ayrıca Naïve Bayes ve Kesinlik faktör versiyonlu

KNN ile iki katmanlı sınıflandırma sistemi sunulmuştur. Deneysel çalışmalarda 41 özellikten 35'i NSL-KDD veri setinde kullanılarak SVM, Naïve Bayes, Decision Trees(J48), Random Forest algoritmaları ile karşılaştırılmıştır.

Chawla ve arkadaşları(Chawla vd., 2019) tarafından önerilen yaklaşımda ise CNN/RNN modelini kullanan bilgisayar tabanlı saldırı tespit sistemi önerilmiştir. Uzun Kısa Hafızalı Model(LSTM) sinir ağlarından ziyade Kapılı tekrarlayan birimler(GRU) kullanılarak daha iyi sonuçlar alınmıştır. ADFA veri seti üzerinde deneysel çalışmalar gerçekleştirilmiştir. Yığılmış CNN ile GRU'ların birleşimi, anomali tabanlı IDS'in performansını artırmıştır.

DÖRDÜNCÜ BÖLÜM

TOPLULUK ÖĞRENME VE OPTİMİZASYON ALGORİTMALARI

4.1. Topluluk Öğrenme Yöntemleri

Topluluk yöntemleri, daha iyi bir tahmin elde etmek amacıyla bir problemi çözmek için birden fazla öğrenme algoritmasının kullanıldığı metotlardır. Klasik yaklaşımda bir öğrenme algoritması ile eğitim yapılır, topluluk öğrenme algoritmalarında öğrenme algoritmalarından oluşan bir küme ile eğitim yapılarak birleştirilir. Topluluk öğrenme yöntemleri, komite tabanlı öğrenme veya çoklu sınıflandırıcı öğrenme olarak da adlandırılır (Zhi-Hua Zhou, 2012). Genel olarak ifade edecek olursak, topluluk öğrenme yöntemleri dört ana kategori altında toplanabilir. Denetimli topluluk sınıflandırma, yarı denetimli topluluk sınıflandırma, kümeleyerek toplu sınıflandırma ve yarı denetimli kümeleyerek toplu sınıflandırma olarak ayırabiliriz (Dong vd., 2019).

Topluluk öğrenme algoritmalarını aşağıdaki üç temel kategoride ele almak mümkündür.

- Oylamalı (Voting) Topluluk Öğrenme: Karar Ağaçları, Rastgele Orman
- Artırmalı (Boosting) Topluluk Öğrenme: AdaBoost, XGBoost, LightGBM, BrownBoost, CoBoosting algoritmaları örnek verilebilir. Artırma (Boosting) algoritmaları sıralı topluluk öğrenme prensibine dayanır. Model performansını artırmak için zayıf öğrenme algoritmalarının birleşiminden oluşan bir algoritma oluşturma prensibine dayanır. Bir bir ardınca hesaplama yapılır.
- Torbalama (Bagging) Topluluk Öğrenme: Ektsra Ağaçlar (Extremely Randomized Trees). Paralel şekilde hesaplama yapılır.

4.1.1. Rastgele Orman

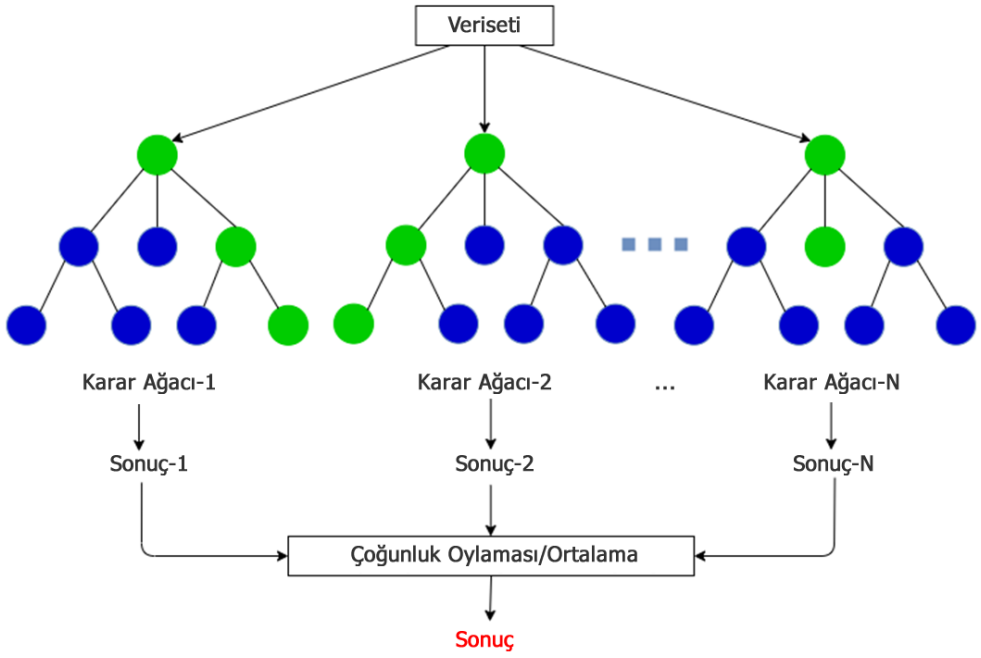
Rastgele Orman (Random Forest) algoritması, rastgele oluşturulan birçok karar ağacından elde edilen sonucun birleştirilerek nihai sonucun elde edildiği bir topluluk öğrenme algoritmasıdır. Gerek sınıflandırma gerekse regresyon problemlerinde oldukça sık kullanılan popüler bir algoritmadır. Sınıflandırma problemlerinde rastgele

ağacın çıkışı ağaçları çoğu tarafından seçilen bir sınıftır. Regresyon problemlerinde ise tekil ağaçların ortalama tahmin döndürülmektedir. Verisetin göre değişmekle birlikte genelde doğruluk oranı yüksek sonuçlar elde edilir. Gini indeks veri setinde yanlış sınıflandırılan verilerin olasılığını verir. Aşağıdaki Gini indeks (G) formülünde p(i) belirli bir sınıfta sınıflandırılma olasılığıdır. CART algoritması Gini Indeksi kullanmaktadır, kusursuz bir sınıflandırma yapılmışsa Gini Indeks sıfırdır

$$Gini\ Indeks = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{4.1}$$

Entropi formülü de aşağıda verilmiştir, fakat hesapsal karmaşıklığı artırdığı için genellikle tercih edilmemektedir.

$$Entropi = \sum_{i=1}^{c} -p_i\ log2p_i \tag{4.2}$$



Şekil 4.1: Rastgele Orman Topluluk Öğrenme Algoritması.

4.1.2. AdaBoost

Freund ve Schapire tarafından geliştirilen AdaBoost Algoritması, birçok farklı alanda uygulanan ve Gödel ödülünü kazanan ilk pratik boost algoritmasıdır. Güçlendirme

Algoritmaları, birçok zayıf ve yanlış kuralı birleştirir ve yüksek doğrulukla tahminler yapar (Schapire, 2013; Sagi & Rokach, 2018; Tyralis & Papacharalampous, 2021).

Her model veya öğrenen algoritma, bir önceki modelin hatasını düzeltmeye çalışır. Hesaplanan ağırlıklar her seferinde modele veya öğrenen algoritmaya atanır. Ardı ardına sıralı bir şekilde modeller hatayı düzeltmektedir. Ağırlıklar bir sonraki sınıflandırıcı ile güncellenir ve tekrar eğitilir ve tekrar eğitim seti üzerinde bir tahmin yapılır ve bu şekilde devam edilir. Sınıflandırıcı algoritmalar SVM, ANN, Karar Ağacı veya kNN algoritmaları olabilir.

Eğer Algoritma 1'e bakacak olursak, başlangıçta ağırlık değerleri w_i *(1,2, 3,…,N)* eşit şekilde *1/m (m=1,2,3,…M, örnek sayısı*) formülüne göre atanır. İlk sınıflandırıcı ağırlık w_i değerlerini kullanarak $G_m(x)$ tarafından eğitilir ve ağırlıklandırılmış hata err_m hesaplanır. Burada err_m terimi ağırlıkları bulma ve güncelleme için en önemli terimdir.

$$err_m = \frac{\sum_{i=1}^{N} w_i I\left(y_i \neq G_m(x_i)\right)}{\sum_{i=1}^{N} w_i} \tag{4.3}$$

Bu formülde I fonksiyonu yanlış olarak sınıflandırılmış örnekler için 1, doğru sınıflandırılanlar için 0 döndürür. err_m hesaplandıktan sonra tahmin edicinin ağırlığı olan $\alpha_m$hesaplanır. Burada α_m değeri tüm sınıflandırıcıların birleşiminden oluşan yeni bir ağaçtır. Formüldeki η değeri öğrenme oranını ifade eder ve bir hiperparametre'dir, varsayılan değeri 1'dir, daha doğru tahmin için daha yüksek değerler alacaktır.

$$\alpha_m = \eta log\ (\frac{(1 - err_m)}{err_m}) \tag{4.4}$$

Sonuç olarak AdaBoost algoritması sınıflandırıcı oylarının ağırlıklı bir sınıflandırıcı oyları toplamı verecektir. İkili sınıflandırmada G(x) değeri -1 veya +1 olarak atanacaktır.

Algoritma 1. AdaBoost Algoritmasının sözde-kodu

1. Başlangıç ağırlıklarını ayarla $w_i=1/N$, $i=1, 2, 3, ..., N$
2. **for** *m=1 to M* **do**

    (a) veriyi eğitime için wi ağırlıklarını kullanarak sınıflandırıcı $G_m(x)$

(b) Hesaplama işlemi, $err_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}$

(c) Hesaplama işlemi, $\alpha_m = \eta log\ (\frac{(1-err_m)}{err_m})$

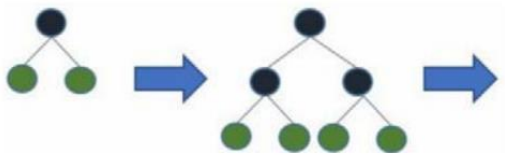
(d) i $= 1,2,3, \ldots,$ N değerleri için ağırlıkları güncelle.

$w_i \leftarrow w_i \cdot exp[\alpha_m \cdot I(y_i \neq G_m(x_i)]$

3. Çıkış $G(x) = [\sum_{m=1}^{M} \alpha_m G_m(x)]$

4.1.3. XGBoost

The eXtreme Gradient Boosting (XGBoost) algoritması (T. Chen & Guestrin, y.y.) tarafından 2016 yılında geliştirilen, son tasarım bir denetimli makine öğrenimi algoritmasıdır. XGBoost, birden çok sınıflandırma ve Regresyon Ağacını (CART) birleştiren bir topluluk öğrenme modeli olan açık kaynaklı Gradient Boosting Tree uygulamasıdır. XGBoost algoritması son derece esnek bir yapıya sahiptir. Sınıflandırma, regresyon, derecelendirme ve diğer kullanıcı-tanımlı hedeflerde kullanılabilir. XGBoost algoritması katman mantığında ağaç büyümesi gerçekleştirir.



Şekil 4.2: **XGBoost Katman Mantığında Ağaç Büyümesi.**

Klasik makine öğrenmesi algoritmaları verisetine bağlı olarak zayıf doğruluk oranı, yüksek kayıp oranı ve büyük miktarda varyans değeri gibi negatif özellikler gösterebilir. FAR dediğimiz Yanlış Tespit Oranı, klasik algoritmalarda yine verisetine bağlı olarak yüksek olabilir. XGBoost Algoritması bir topluluk öğrenme algoritması olduğundan farklı verisetleri üzerinde daha başarılı sonuçlar elde edilmiştir. Bu nedenle Kaggle gibi veri bilimi makine öğrenmesi yarışmalarında tercih edilir olmuştur. XGBoost algoritması optimize edilmiş yapısıyla düşük donanım konfigürasyonunda dahi en iyi performansı gösterebilmektedir.

XgBoost algoritması, amaç fonksiyonunu ($\mathcal{L}$) optimize etmek için ek bir eğitim yöntemi kullanır. Bu metotta, sonraki adımın optimizasyon sonucu önceki adım olan t değerine bağlıdır. Aşağıdaki formül her bir adımdaki amaç fonksiyonunu ifade eder. t parametresi döngüde kayıp değeri (loss term) ifade eder ve Ω değeri regülarizasyon yani düzenleme değeridir.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) \tag{4.5}$$

Yukarıdaki eşitlikteki Ω teriminin açılımı aşağıdaki gibidir.

$$\Omega(f_t) = \gamma . T_t + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2 \tag{4.6}$$

Burada γ ve λ değerleri optimizasyon parametreleridir, bu değerler büyüdükçe ağaç yapısı basitleşir ve aşırı uydurma (overfitting) problemi çözülür. Eğer ikinci dereceden Taylor açılımı uygulanırsa, aşağıdaki denklem elde edilir. Burada *g* birinci dereceden kısmi türev, *h* ise ikinci dereceden kısmi türevdir.

$$\mathcal{L}^{(t)} \cong \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_{i)} \right] + \Omega(f_t) \tag{4.7}$$

Burada g ifadesi kayıp fonksiyonunda birinci dereceden gradyanı(eğimi) ifade eder.

$$g_i = \partial_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \tag{4.8}$$

h ikinci dereceden gradyan istatistiklerini temsil eder.

$$h_i = \partial^2_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1}) \tag{4.9}$$

Optimal ağırlık değeri olan w_j^* aşağıdaki gibi hesaplanır.

$$w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{4.10}$$

Burada $\hat{\mathcal{L}}^{(t)}$ kayıp fonksiyonu skorunu ifade eder.

$$\hat{\mathcal{L}}^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\left(\sum_{i\in I_j} h_i + \lambda\right)} + \gamma.T \tag{4.11}$$

Bu formülde kayıp fonksiyonu değeri ($\hat{\mathcal{L}}^{(t)}$) daha iyi bir ağaç yapısı için düşük olmalıdır ve w_j^* değeri ağırlıkların çözümüdür.

4.1.4. CatBoost

Yandex tarafından 2017 yılında geliştirilen bir topluluk öğrenme algoritmasıdır. Kategorik Artırma (Categorical Boosting) algoritması permutasyon teknikleri kullanarak kategorik sütunlara odaklanır. CatBoost eksponansiyel olarak büyüyen özellikleri, mevcut ağacı Aç gözlü algoritma (Greedy Metotu) kullanarak bölerek çözer(Dorogush vd., y.y.). CatBoost Algoritması finans gibi farklı alanlarda kullanılmaktadır. Ağaç yapısını seçerken rastgele permutasyonlarla ağaç değerlerini tahmin etmesi ile aşırı uydurma probleminin üstesinden gelir, bu yönüyle geleneksel artırma algoritmalarından ayrılır. CatBoost tahmin ediciye göre, alt tarafta ikili karar ağaçları kullanır.

$$h(\mathbf{x}) = \sum_{j=1}^{J} b_j \mathbb{1}_{\{\mathbf{x}\in R_j\}}, \tag{4.12}$$

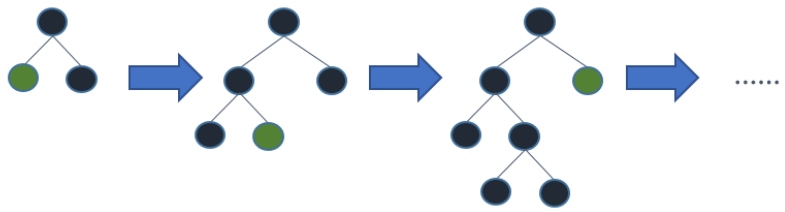
Bir karar ağacı mantığında özellik uzayı rekürsif olarak parçalara ayrılır. Ayırma özelliği a değerine bağlı olarak R^m ayrık bölgelere (ağaç düğümleri) ayrılır. Burada Rj, ağacın yapraklarına karşı gelen ayrık bölgelerdir.

$$a = \mathbb{1}_{\{x^k > t\}} \tag{4.13}$$

Özellikler genellikle *t* eşik değerini geçen x^k ile tanımlanan bazı özelliklerdir, genelde ikili değerler olur, yukarıda formülde x^k ifadesi sayısal veya ikili bir özellik olabilir. Her bir final bölgesine (ağacın yaprakları) bir değer atanır, regresyon işleminde bir değer veya sınıflandırma probleminde tahmin edilen sınıf için y'nin cevabı tahmin edilmeye çalışılır.

4.1.5. LightGBM

Microsoft tarafından 2017 yılında geliştirilen bir topluluk öğrenme algoritmasıdır. LightGBM algoritması yaprak mantığında ağaç büyümesi gerçekleştirir. LightGBM, GOSS(Gradient Based One Side Sampling- Gradyan Tabanlı Tek Taraflı Örnekleme) ve EFB(Exclusive Feature Bundling- Özel Özellik Paketleme) kullanarak verileri ve özellikleri örnekleyerek histogram oluşturmanın karmaşıklığını azaltmayı amaçlar. GOSS, gradyan(eğim) temeline göre çalışan yeni bir tek taraflı örnekleme metodur. Küçük gradyanlı örnekler üzerinde rastgele olarak örnekleme gerçekleştiren büyük gradyanlı örnekleri koruyup, küçük gradyana sahip örnekleri atar. EFB'nin ilk aşamasında ise birlikte paketlenebilecek özellikleri belirlemek için bir algoritma çalışır, ikinci bir algoritma özellikleri birleştirir.



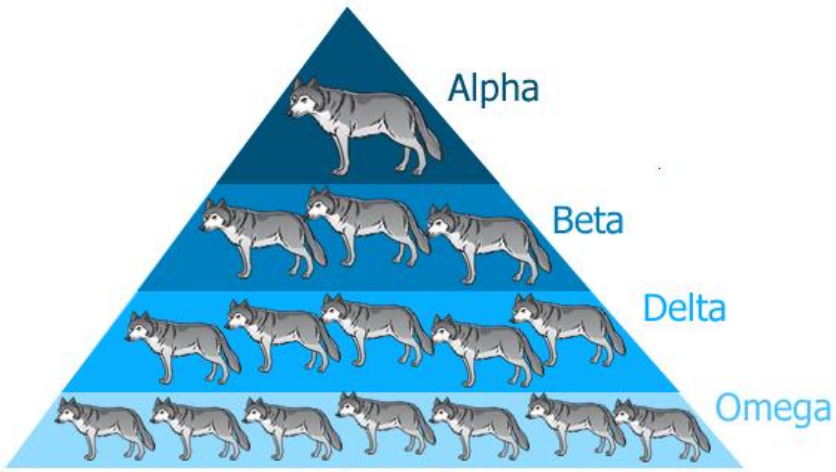
Şekil 4.3: LightGBM Yaprak Mantığında Ağaç Büyümesi

4.2. Optimizasyon Algoritmaları

4.2.1. Gri Kurt Optimizasyon (GWO) Algoritması

Gri Kurt Optimizasyon algoritması, Mirjalili tarafından 2014 yılında geliştirilen bir meta-sezgisel algoritmadır Gri kurtların avlanma sistemi örnek alınarak oluşturulan algoritma, mühendislik problemlerine kolayca uyarlanabilmesi nedeniyle popülasyon tabanlı algoritmalar arasında popüler hale gelmiştir. Gri kurtlar arasında dört seviyeli bir hiyerarşi vardır: Alfa(α), Beta(β), Delta(δ), ve Omega(ω). Avın yakalanmasında takip, kuşatma ve saldırı aşamalarında bu hiyerarşiye göre görev dağılımı yapılır. Alfa Kurt, sürünün lideridir ve kritik kararlar verir. Örneğin, nerede kamp kurulacağına, hangi ava saldırılacağına ve nerede uyuyacağına Alfa Kurt karar verir. Hiyerarşide alfa kurtların hemen altında yer alan beta kurtlar, alfa kurtlara tabidir ve alfa kurtlara en yakın arkadaş kurtlardır ve sürüye disiplin sağlarlar.

Alfa kurt yaşlanır veya ölürse beta kurtlar arasından yeni bir alfa kurt seçilir. Öte yandan, Delta kurtları hiyerarşide beta kurtlarının hemen altındadır; delta kurtları güçlüdür ancak liderlik yeteneğinden yoksundur veya kendilerini liderlik sorumluluğunu üstlenecek bir konumda görmezler. Hiyerarşinin en altında yer alan Omega kurtlarının herhangi bir gücü yoktur, avı en son yerler ve genç kurtların gözetiminden sorumludurlar. Mirjalili tarafından oluşturulan matematiksel model, avın çevrelenmesi simülasyonu için aşağıdaki matematiksel denklemleri kullanır.



Şekil 4.4: Gri Kurt Optimizasyonunda Kurt Popülasyonu hiyerarşisi

$$\vec{D} = \left| \vec{C} - \overrightarrow{X_p}(t) - \vec{X}(t) \right| \tag{4.14}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \,.\, \vec{D} \tag{4.15}$$

Burada t mevcut iterasyonu gösterir, $\vec{A}$ ve $\vec{C}$ katsayı vektörleridir, $\overrightarrow{X_p}$ avın pozisyonu ve $\vec{X}$ vektörü gri kurtların pozisyonunu ifade eder.

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{4.16}$$

$$\vec{C} = 2.\vec{r}_2 \tag{4.17}$$

$$\vec{a} = 2(\frac{1-t}{T}) \tag{4.18}$$

burada $\vec{a}$ değişkeni her döngü adımında 2'den 0'a lineer olarak azalır. *T* maksimum iterasyon sayısını ifade eder. $\vec{r}_1$ ve $\vec{r}_2$ değerleri [0,1] aralığında rastgele sayılardır Aşağıdaki matematiksel denklemler kurtların avlanmasını ifade eder. Bu formülde alfa kurt avın pozisyonu hakkında en iyi bilgiye sahiptir, ikinci en iyi çözüme beta kurt, üçüncüye de delta kurt sahiptir.

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1} . \overrightarrow{X_\alpha} - \vec{X}\right| \tag{4.19}$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2} . \overrightarrow{X_\beta} - \vec{X}\right| \tag{4.20}$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3} . \overrightarrow{X_\delta} - \vec{X}\right| \tag{4.21}$$

Burada $\overrightarrow{C_1}, \overrightarrow{C_2}$ ve $\overrightarrow{C_3}$ rastgele vektörlerdir, $\overrightarrow{X_\alpha}, \overrightarrow{X_\beta}$ ve $\overrightarrow{X_\delta}$ vektörleri sırasıyla alfa, beta ve delta kurtları ifade eder. $\vec{X}$ Vektör değeri ise mevcut çözümün pozisyonudur.

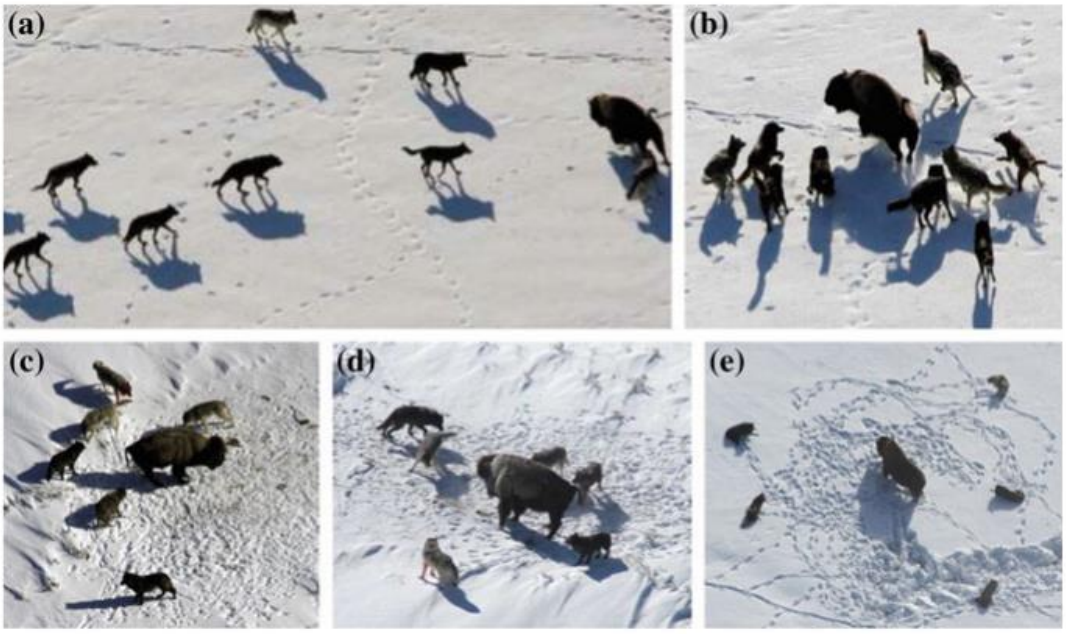
$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} . \overrightarrow{D_\alpha} \tag{4.22}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} . \overrightarrow{D_\beta} \tag{4.23}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} . \overrightarrow{D_\delta} \tag{4.24}$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3}}{3} \tag{4.25}$$

Yukarıdaki matematiksel modellerden görülebileceği gibi, alfa, beta ve delta kurtlar, 2D arama uzayındaki her döngü adımında avlarına göre konumlarını ayarlar. GWO algoritmasında avlanma, av hareketsiz kalana kadar devam eder.



Şekil 4.5: Kurtların sürü halinde avlanması.

Matematiksel modelde $\vec{A}$ değeri $\vec{a}$ değerine bağlıdır ve her döngü adımında azalır. Burada $\vec{a}$ değerinin 2'den 0'a azalması sonucu $\vec{A}$ değeri [-2a, 2a] aralığında değerler alır. Eğer$|A| < 1$, kurtlar ava saldırır, eğer$|A| > 1$ kurtlar avdan uzaklaşır. Tüm avlanma adımlarının sözde-kodu aşağıdaki gibidir.

Algoritma. Gri Kurt Optimizasyon Algoritması sözde-kodu.

Kurt popülasyonu için başlangıç değerlerini ayarla X_i *(i=1,2, …, n)*
Başlangıç değerlerini a, A, ve C için ayarla
Amaç fonksiyonuna göre her arama ajanı değerini hesapla

X_α*=alfa kurt arama ajanı (en iyi değer)*
X_β*=beta kurt arama ajabı (ikinci en iyi değer)*
X_δ*=delta kurt arama ajanı (üçüncü en iyi değer)*
while *(t<Max_iterasyon_sayısı)*
    for each *kurt arama ajanı için* ***do***
        Arama ajanı pozisyonlarını güncelle
    end for
    Güncelleme işlemi: A, C ve a değerleri için.
    Tüm kurtlar için amaç fonksiyonunu hesapla (tüm ajanlar)

Güncelleme işlemi: X_α, X_β ve X_δ

t=t+1

end while

X_α değerini döndür

BEŞİNCİ BÖLÜM

MAKİNE ÖĞRENMESİNDE ÖN İŞLEMLER

5.1. Veri Önişleme

Makine Öğrenme algoritmasında modelleme yapılmadan önce verilerin ön işleme (pre-processing) aşamasından geçirilmesi gerekmektedir. Verisetindeki eksik ve null değerler içeren, ölçeklenmemiş, normalize edilmemiş, çift kayıtlar bulunduran, etiketler dışında sayısal olmayan, gerekli tip dönüşümü yapılmamış veriler istenen sonucun alınmasını güçleştirir. Ön işleme işlemi ham verinin modellemeye uygun hale getirilmesi için yapılan dönüşümlerdir. Tahmini modelleme projelerinde en zaman alıcı ve en önemli kısım verinin hazırlanmasıdır. Fakat verisetinin ön işlemden geçirilmemesi zaman ve yer karmaşıklığını artırabilir, bu durumda işlem süresi ve depolama alanı artacaktır. Sınıflandırma ve regresyon gibi tahmin projelerinde verinin daima ön hazırlık işleminden geçirilmesi gerekir. Bu sayede veriseti modellemeye uygun hale getirilir ve algoritmalar daha hızlı, daha doğru sonuç üretebilirler. Veri temizleme, özellik seçimi, boyut azaltma, eksik verileri atma veya değerle doldurma gibi aşamalar verinin hazırlanmasında sonuca direkt etki eden önemli faktörlerdir.

Veri hazırlama adımında ham veri istenen biçime dönüştürülerek modellemeye uygun hale getirilir. Proje genellikle problemin tanımlanması (amaç nedir, neyi tahmin etmek istiyoruz vb.) verinin hazırlanması, modelin değerlendirilmesi gibi alt adımlardan oluşur. Verinin Önişleme aşamasında aşağıdaki işlemler yapılmaktadır.

- Gereksiz boş kayıt veya null değer içeren veriler atılmakta, ilgisiz sütunların veri setinden kaldırılmaktadır. Veri setinde olabilecek istatistiksel gürültü, ölçüm hatası ve giriş hataları düzeltilmektedir.
- Eksik değerler yerine, uygun değerlerin konması gerekmektedir. Genellikle ortalama değer, medyan değer veya bir algoritma ile bu değerler doldurulmaktadır. Scikit-learn içindeki SimpleImputer sınıfı NaN(Not a number-sayı değil) olarak işaretlenen verinin dönüşümünü yapmaktadır.
- Sınıflandırma veya Regresyon modelleri için veri setindeki aykırı değerlerin (Outlier) tespit edilmesi önemlidir, bu değerler tahmin performansını

düşürmektedir. Veri setlerinde standart sapma veya çeyrekler arası oran (interquartile range) kullanılabilir.

- Veri seti çok büyük ise istatistiksel tekniklerin alternatifi olarak otomatik aykırı değer ayıklayan algoritmalar kullanmak gerekebilir. Bunlardan biri İzolasyon Ağaçları (Isolation Forest) kısaca iForest olarak adlandırılan algoritmadır. Scikit-learn kütüphanesinde yer alan IsolationForest sınıfı, kirlilik (contamination) parametresi (0.0-0.5 aralığında, varsayılan 0.1) alarak kullanılabilir. Eğer veri Gauss eğrisi şeklinde ise Minimum Kovaryans Belirleyici (MCD) algoritması aykırı değer tespiti içim kullanılabilir, scikit-learn kütüphanesinde EllipticEnvelope sınıfı contamination parametresi ile kullanılabilir. Local Outlier Factor(LOF) algoritması birkaç özelliğin olduğu veri setinde kullanılabilir. One-Class SVM ise regresyon ve sınıflandırma problemlerinde aykırı değer tespitinde kullanılmaktadır.
- Standardizasyon işlemi yapılır. Standard Scaler() fonksiyonu kullanılarak Veri Çerçevesi (DataFrame) sütunu ölçeklendirilebilir.
- Normalizasyon işlemi ile her bir giriş verisi 0-1 aralığına ölçeklendirilmektedir. Bu ölçekleme için verinin minimum ve maksimum değerleri bilinmelidir. Scikit-learn kütüphanesindeki MinMaxScaler nesnesi ile veriseti normalize edilmektedir.
- Tüm sütunlardaki veri sayısal olmakta, kategorik veriler de sayısal bir değere çevrilmektedir. Tek Sıcak Kodlama (One Hot Encoding) ile kategori sütunları sayısal değerlere kodlanmaktadır. Örneğin otomobil veri setinde her bir otomobil rengi sırasıyla kodlanmakta, örneğin beyaz=1, siyah=2, kırmızı=3 şeklinde devam etmektedir. İkili (Binary) şekilde de kodlama yapılabilmektedir, örneğin öğrenci veri setinde durum sütunu “geçti” ise 1, “kaldı” ise 0 şeklinde kodlanabilmektedir. Scikit-learn kütüphanesindeki OneHotEncoder fonksiyonu ile kodlama yapılabilmektedir.
- Özellik mühendisliği (Feature Engineering) ile mevcut veriden yeni değişkenler türetme ve özellik seçimi yapılmaktadır. En popüler algoritmalardan bir olan RFE (Recursive Feature Elimination) ile veri setinden en önemli özelliklerin seçimi saplanmaktadır, scikit-learn kütüphanesinde RFE

isimli n_features_to_select parametresi ile seçilecek özellik sayısı ayarlanmaktadır.

- Boyut azaltma ile büyük verinin daha kolay işlenebilmesi için sonucu etkilemeyecek veya az etkileyecek sütunların sayısında azaltmaya gidilmektedir. PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), SVD (Singular Value Decomposition) veya AutoEncoder gibi teknikler kullanılmaktadır.

- Kategorilerdeki veri sayısında dengesizlik olduğu zaman Makine Öğrenme Algoritmalarının veri sayısı fazla olan kategorileri daha iyi tespit ettiği, veri sayısı az olan kategorileri ise düşük olasılıkla tespit etme durumu olabilir. Bu nedenle ya sentetik veri üretilerek kategorilerdeki veri sayıları eşitlenir, ya da bir kategoride veri sayısı çok az ise o kategori silinebilir. SMOTE, ADASYN, Oversampling ve Undersampling gibi yöntemlerle sentetik veri üretilerek, her bir kategorideki veri sayısı dengelenir.

5.1.1. Standardizasyon

Standardizasyon işleminde aşağıdaki formül ile veriler standart hale getirilmiştir. Burada mevcut veriden, ortalama değer çıkarılarak standart sapma değerine bölünmektedir. Veri Standardizasyon Formülü aşağıdaki şekilde hesaplanır.

$$X' = \frac{X - \mu}{\sigma} \tag{5.26}$$

μ : Ortalama değer

σ : Standart Sapma

X : Anlık değer

X' : Standartlaştırılmış değer

5.1.2. Normalizasyon

Normalizasyon işleminde aşağıdaki formül ile veriler normalize edilmektedir. Burada mevcut veriden minimum değer çıkarılarak, verinin maksimum ve minimum değer farkına bölme işlemi yapılmıştır. Veri Normalizasyon Formülü aşağıdaki şekilde ifade edilir.

$$X' = \frac{X - X_{min}}{X_{maks} - X_{min}} \tag{5.27}$$

X : Anlık değer

X' : Normalize edilmiş değer

X_{maks} : Maksimum değer

X_{min} : Minimum değer.

Tek Sıcak Kodlama (One Hot Encoding) ile kategori sütunlarına bir sayısal değer atanmaktadır. Örneğin daha önceki bölümlerde gördüğümüz NSL-KDD veri setinde «protocol_type», «service», «flag» sütunlarına panda kütüphanesi ile değerler atanarak ön işleme yapılmaktadır.

5.1.3. Regresyon Algoritmalarında Metrikler

Regresyon algoritmalarında karşılaştırma için aşağıdaki metrikler kullanılmaktadır. Regresyon algoritmaları, ev fiyatı gibi genelde sayısal sürekli bir verinin tahmin edilmesinde kullanılır.

- Ortalama Karesel Hata (MSE-Mean Squared Error)
- Ortalama Mutlak Hata (MAE-Mean Absolute Error)
- Ortalama Karesel Hatanın Kökü (RMSE-Root Mean Square Error)

5.1.4. Sınıflandırma Algoritmalarında Metrikler

Sınıflandırma algoritmaları verileri belli kategorilere göre gruplandırmak için kullanılmaktadır. Sınıflandırma için aşağıdaki metrikler kullanılmaktadır. E-postaların önemli ve önemsiz olarak sınıflandırılması, görüntü işlemede cisimlerin sınıflandırılması gibi örnekler verilebilir. Bu çalışmada özellikle saldırı tespitini sınıflandıracağız. Makine öğrenme algoritmalarının başarımının karşılaştırılması açısından literatürde aşağıdaki metrikler tercih edilmektedir.

- Doğruluk(accuracy),
- Kesinlik(precision),

- Duyarlılık(recall),
- F1-Score,
- Kesinlik (harmonik ortalaması),
- ROC ve AUC.

Karmaşıklık Matrisi (Confusion Matrix) sınıflandırma problemleri içindir. Karmaşıklık Matrisinde 4 durum vardır. Örneğin bir hastalık üzerinden örneklendirirsek;

Doğru Pozitif (True Positive-TP): Hasta olduğu tahmin edilen, gerçekte de hasta olunan durum.

Doğru Negatif (True Negatives-TN): Hasta olmadığı tahmin edilen, gerçekte de hasta olunmayan durum.

Yanlış Pozitif (False Positive-FP): Hasta olduğu tahmin edilen ama gerçekte hasta olunmayan durum. Tip 1 hata, yanlış teşhis.

Yanlış Negatif (False Negative-FN): Hasta değil diye tahmin edilen ama gerçekte hasta olan durum. Tip 2 hata, yanlış teşhis**.**

Tablo 5.1: Karmaşıklık Matrisi Örneği

	Tahmin EVET	**Tahmin HAYIR**	
Gerçekte EVET	TP=100	FP=10	110
Gerçekte HAYIR	FN=5	TN=50	55
	105	60	165

Doğruluk (Accuracy): Bir modelin başarısını ölçmek amacıyla kullanılan en önemli metriklerden biri olmasına rağmen, tek başına yeterli olduğunu söyleyemeyiz. Özellikle veri setinin eşit dağılmaması durumunda doğru sonuçlar vermemektedir.

$$\frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5.28}$$

Yukarıdaki formüle göre Doğruluk değeri = (100+50)/165=0.91 çıkmaktadır.

Kesinlik (Precision): Kesinlik değeri Doğru Pozitif değerinin, pozitif değerlerin toplamına bölümü ile hesaplanmaktadır.

$$\frac{TP}{TP + FP} \tag{5.29}$$

Yukarıdaki formüle göre kesinlik değeri = 100/110=0.91 çıkmaktadır.

Duyarlılık (Sensivity, Recall, TPR): Duyarlılık değeri Doğru Pozitif değerinin, doğru pozitif ve yanlış negatif toplamına bölümü ile hesaplanmaktadır.

$$\frac{TP}{TP + FN} \tag{5.30}$$

Duyarlılık değeri ise; 100/105=0.95 çıkmaktadır.

Özgüllük (Specifity, Selectivity, TNR-True Negative Rate): Özgüllük değeri ise Doğru Negatif değerinin, Doğru Negatif ve Yanlış Pozitif değeri toplamına bölümü ile hesaplanmaktadır.

$$\frac{TN}{TN + FP} \tag{5.31}$$

Özgüllük değeri ise 50/60 = 0.83 çıkmaktadır.

F1 skoru: Bu metrik bize Kesinlik ve Duyarlılık değerlerinin harmonik ortalamasını göstermektedir. Veri kümelerinde eşit dağılmama durumu var ise doğruluk değeri yanlış sonuçlar verecektir, bu durumda F1 skorunun kullanılması daha uygundur. Tüm hata değerlerini içeren bir metrik olması açısından da F1 skor önemlidir.

$$\frac{2 * Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \tag{5.32}$$

Doğru Pozitif Oranı (True Positive Rate- TPR)

$$\frac{TP}{TP + FN} = 1 - FNR \tag{5.33}$$

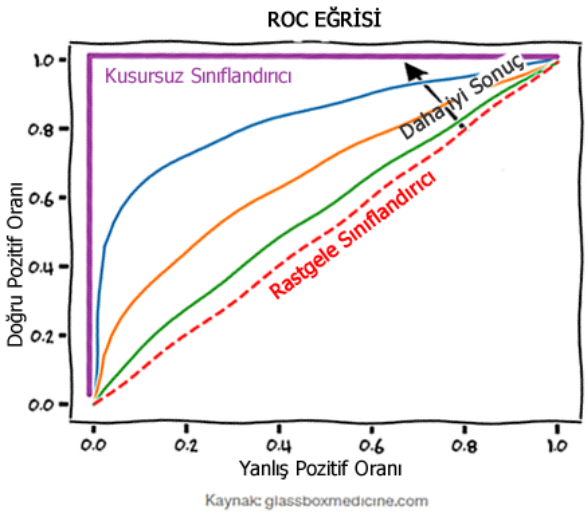
Yanlış Pozitif Oranı (False Positive Rate – FPR – Tip 1 Hata)

$$\frac{FP}{FP + TN} = 1 - TNR \tag{5.34}$$

Yanlış Negatif Oranı (False Negative Rate - Miss Rate – FNR Tip 2 Hata)

$$\frac{FN}{TP + FN} \tag{5.35}$$

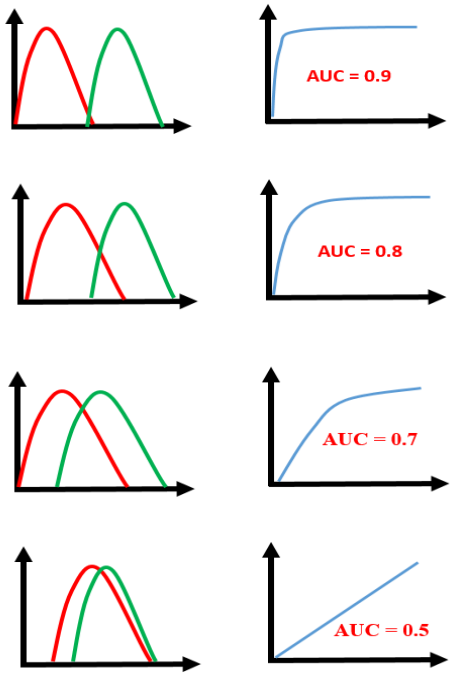
Diğer bir başarım ölçüm metriği olan ROC Eğrisi de sınıflandırma modelinin performansını tüm sınıflandırma eşiklerinde grafiksel olarak gösterilir. TP ve FP değerlerine göre çizmektedir. Eğri sol üst köşeye ne kadar yakınsa sınıflandırma o kadar iyidir.



Şekil 5.1: ROC Eğrisi.

Eğri Altındaki Alan (Area Under Curve-AUC): Tüm ROC eğrisi altında kalan alanın tam ölçüsüdür. Aşağıdaki örnekte de görüldüğü gibi kırmızı ile belirtilen sınıf ile yeşil ile belirtilen sınıf ilk şekilde tam ayrı iken AUC değeri 0.9 olur. Son şekilde

iki sınıf birbirine oldukça yaklaştığından ayırt edilemez duruma gelmiştir ve AUC değeri 0.5'e düşmüştür.



Şekil 5.2: AUC Eğrisi.

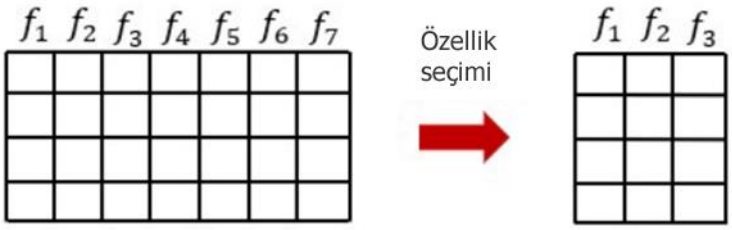
Sonuç olarak AUC değeri ne kadar yüksek ise sınıflandırma o kadar iyi yapılmıştır.

İkili Sınıflandırma ve Çoklu Sınıflandırma: İkili Sınıflandırmada "Saldırı var" ya da "Yok" şeklinde iki durum göz önüne alınır. Örneğin durum; Normal veya Anormal olabilir. Çoklu sınıflandırmada ise saldırı türlerine ilaveten Normal durum da vardır. Çoklu sınıflandırmaya örnek olarak NSL-KDD verisetinden örnek verecek olursak DOS saldırısı, U2R saldırısı, R2L saldırısı ve Normal başlıkları olabilir. İkili sınıflandırmaya örnek olarak Saldırı Var=1 durumu ya da Saldırı Yok=0 durumu kullanılır.

5.2. Özellik Seçimi

Özellik seçimi tanım olarak; sadece ilgili, tutarlı ve gereksiz olmayan veri kullanarak verideki gürültüden kurtulmak için giriş verisini azaltma işlemidir. Veri işlemenin önemli adımlarından biridir, makine öğrenmesi problemleri için yüksek boyutlu veri indirgenerek daha etkin ve verimli sonuçlar elde edilir.

Özellik seçiminin amacı, alaka düzeyini en üst düzeye çıkarmak ve fazlalığı en aza indirmektir. Veri setinde çok fazla veri olduğunda, makine modelin geliştirme ve eğitim süresi uzamakta, sonuca ulaşmak uzun zaman almaktadır. Yüksek boyutta verilerin işlenmesi için çok fazla donanım kaynağı (bellek, GPU, TPU vb.) tüketilmesi gerekmektedir. Çok fazla veri olmasının diğer bir dezavantajı, modelin ilişkisiz verilerden öğrenmesi ve doğru sonuç alınamamasıdır. Özellik seçimi ile sadece eğitim zamanı kısalmaz, aynı zamanda model karmaşıklığı da azalır ve sonuçta aşırı uydurma(overfitting) önlenir. Özellik seçimi, modeli basitleştirerek daha rahat anlaşılmasını da sağlar.



Şekil 5.3: Özellik seçimi ve verinin boyutu azaltılması.

Örneğin; bir kütüphane eski kitapları bağışlamakta ve raflara yeni kitapları koymak istemektedir. Bu işlemi otomatikleştirmek için bir model eğitmek isteyelim. Aşağıdaki örneği inceleyecek olursak, burada renk alanı bizim problemimiz için gereksiz bir alandır, çıkarılması sonucu etkilemez.

Kitap adı	Okunma Sayısı	Kitabın Durumu	**Renk**

Özellik seçimi aşağıdaki avantajlara sahiptir.

- Gürültüden öğrenmeyi engelleme,
- Doğruluk ölçeğini artırma,
- Eğitim zamanını azaltma,
- Modelin karmaşıklığını azaltma,
- Yüksek boyut probleminden kaçınma,

- Modeli basitleştirip daha rahat anlaşılmasını sağlama.

Filtreleme metotları eğiticili ve eğiticisiz olmak üzere iki ana gruba ayrılır. Özellik seçimi, çok geniş bir çalışma alanı olduğundan ve sürekli üzerinde çalışıldığından dolayı bu çalışmada özellikle eğiticili metodlar üzerinde durulacaktır.

5.2.1. Eğiticili Metotlar

Özellik seçiminde Eğiticili metodlar (Supervised Methods) ana grubu, Filtreleme metotları, Sarmalayıcı metotlar, Hibrit Metotlar, Gömülü metodlar gibi alt gruplara ayrılır.

a) Filtreleme Metotları

Çıkıştaki ilişkiye bağlı olarak ya da çıkıştaki korelasyonun nasıl olduğuna göre özellikler çıkarılır. Avantajları; sarmalayıcı ve gömülü metotlar ile karşılaştırıldığında hesaplama karmaşıklığı bakımından daha iyidir. En hızlı çalışma zamanına sahiptir. Aşırı uydurma (overfitting) riski daha azdır, genelleştirme yeteneğine sahiptir. Sınıflandırıcıdan bağımsızdır ayrıca yüksek boyuttaki veri setlerinde kolaylıkla ölçeklenebilir.

Dezavantajları ise; özellik seçimi için sınıflandırıcı model ile etkileşimi yok sayar, özellik bağımlılıklarını yok sayar ve tek değişkenli(univariate) teknikler olması durumunda her bir özelliğe ayrı ayrı karar verir, bu durum düşük hesaplama performansına neden olur.

Şekil 5.4. Filtreleme Metodu

Filtreleme metotlarından Pearson Yöntemi, X ve Y gibi iki sürekli değişken arasındaki doğrusal bağımlılığı ölçmek için kullanılır. X ve Y değerleri -1 ile +1 arasında değerler değişir. +1 pozitif ilişki, -1 negatif ilişki, 0 ilişki yok anlamına gelir. Eğer iki özellik kendi aralarında yüksek korelasyona sahipse ikisinden birini eleyebiliriz.

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}} \quad (5.36)$$

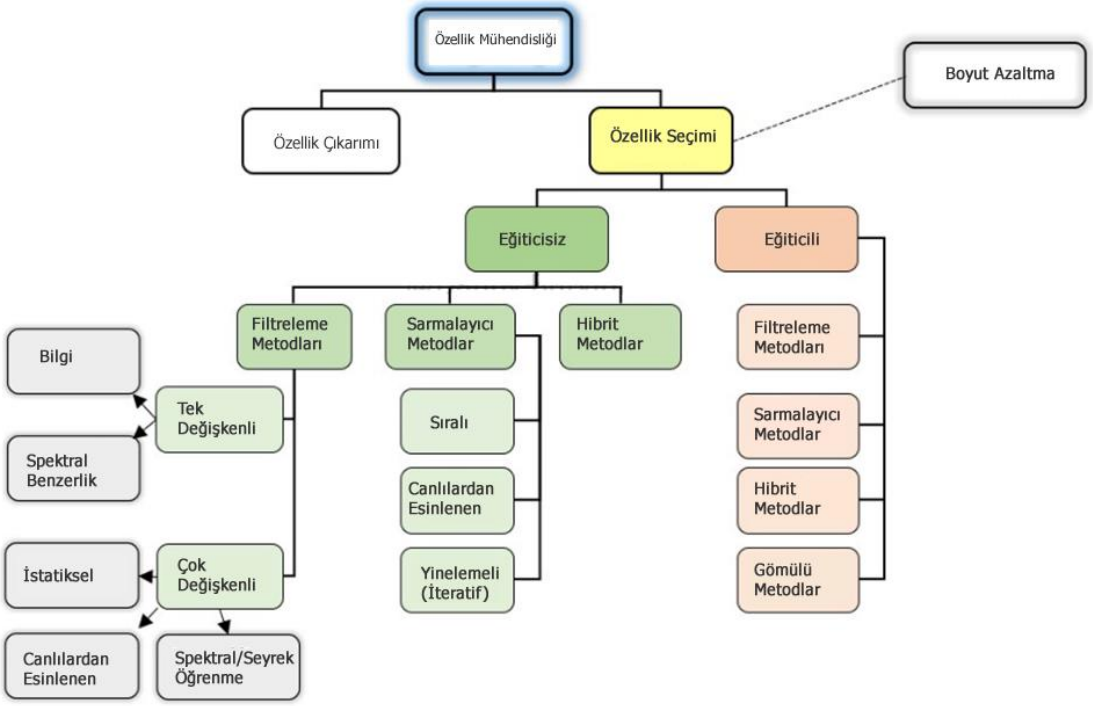
r : Korelasyon sabiti

x_i : bir örnekteki x değişkeninin değeri

$\bar{x}$: x değişken değerlerinin ortalaması

y_i : bir örnekteki y değişkeninin değeri

$\bar{y}$: y değişkeni değerlerinin ortalaması



Şekil 5.5: Özellik Seçimi Sınıflandırması.

LDA Yöntemi, Doğrusal Ayırt Edici Analiz (Lineer Discriminant Analysis) ile kategorik bir değişkenin iki veya daha fazla sınıfını karakterize eden veya ayıran özelliklerin doğrusal kombinasyonunu bulmak için kullanılır.

ANOVA Yöntemi, varyans analiz etme üzerine kuruludur. LDA'ya benzerdir fakat ANOVA'da bir veya daha fazla kategorik bağımsız özellik ile bir sürekli bağımlı özellik kullanılarak işlem yapılır. ANOVA, birden fazla grubun ortalamalarının eşit olup olmadığını belirleyen bir istatistiksel test sunar.

Chi-squared test Yöntemi, Kategorik değişkenler arasındaki ilişki olasılığını veya korelasyonu değerlendirmek için frekans dağılımlarını kullanan istatiksel bir testtir.

Bilgi Kazanımı (Information Gain) Yöntemi, entropideki azalmayı belirler. Fisher's Skoru Yöntemi, azalan sırada özellik sıralamasını döndürür. Kayıp Değer Oranı (Missing Value Ratio) Yöntemi, özellik kümesini eşik değerine)karşı değerlendirir. Diğer Yöntemler ise F-Test, FCBF Hızlı Korelasyon Filtresi, Düşük Varyans Filtresi yöntemleridir.

b) Sarmalayıcı Metodlar

En iyi performans veren özellik alt kümlerini araştırma üzerine kuruludur. Avantajları; özellik seçimi için sınıflandırıcı ile etkileşim kurar, daha kapsamlı bir şekilde özellik uzayı seçimi yapar, model özellik bağımsızdır, filtreleme metodundan daha iyi genelleştirme yapar. Özellik bağımlılıklarını hesaba katar. Dezavantajları ise; filtreleme ve gömülü metotlar ile karşılaştırıldığında aşırı uydurma riski vardır, hesapsal maliyeti yüksektir, daha uzun çalışma zamanına sahiptir.

Şekil 5.6: Sarmalayıcı metot sıralaması.

Sarmalayıcı metodlar, dört alt gruba ayrılmaktadır; İleri Yönde Özellik Seçimi, Geri Yönde Özellik Seçimi, Özdevinimli Özellik Seçimi, Kapsamlı Özellik Seçimi.

İleri Yönde Özellik Seçiminde, ilk başta modelde hiçbir özellik yoktur, her bir iterasyonda modeli geliştiren bir özellik eklenir, bu işlem artık model gelişmediği tespit edilene kadar devam eder.

Geri Yönde Özellik Seçiminde, ilk başta tüm özelliklerle başlanır, daha sonra modelin performansına etki etmeyen özellikler her bir iterasyonda elenir, bu işlem modelin performansında ilerleme görülmediği zamana kadar tekrar eder.

Özdevinimli Özellik Seçiminde, en iyi performansı veren özellik alt kümesini bulan Aç gözlü (Greedy) optimizasyon algoritması kullanılır. Her bir iterasyonda tekrar tekrar modeller oluşturulur, en iyi ve en kötü performansı gösteren özellik bir kenara bırakılır. Tüm özellikler tükenene kadar soldaki özelliklerle bir sonraki modeli oluşturur, daha sonra eleme sırasına bağlı olarak özelliklere bir puanlama yapılır.

Kapsamlı özellik seçiminde ise mümkün olan her bir özellik kombinasyonu denenmekte, en iyi performans gösteren alt küme döndürülmektedir.

Sarmalayıcı metodlarla özellik seçiminde kullanılabilecek en iyi yollardan biri Boruta paketidir, bu paket kullanılarak gölge özelliklerle özelliğin derecesi belirlenir.

c) Gömülü Metodlar

Hem sınıflandırma hem de özellik seçimini aynı anda yaptıklarından filtreleme yöntemlerine göre daha maliyetlidir(Budak, 2108). Filtreleme ve sarmalama metotların birleşimini kullanarak daha kaliteli bir özellik seçimi yapar. Gömülü metotlar, eğitim esnasında algoritmanın otomatik olarak özellik seçimini gerçekleştirir. Bu metota örnek olarak, aşırı uydurmayı azaltmak için dahili ceza fonksiyonlarına sahip olan LASSO ve RIDGE regresyonları verilebilir. LASSO regresyonu, katsayıların büyüklüğünün mutlak değerin eşdeğer ceza ekleyen L1 düzenlemesini gerçekleştirir. RIDGE regresyonu ise katsayıların büyüklüğünün karesine eşdeğer ceza ekleyen L2 düzenlemesini gerçekleştirir. Karar ağaçları, destek vektör makineleri de gömülü yöntemlere örnek olarak verilebilir.

d) Hibrit Metodlar

Hibrit özellik seçme metodu neleri kombine edeceğine göre değişir. Önce kullanılacak yöntemler seçilir daha sonra bu yöntemlerin süreçleri takip edilir. İlk adımda, sıralama metodları kullanılarak özellik sıra listesi üretilir, sonra listeden en iyi k tane özellik için sarmalayıcı metodlar gerçekleştirilir. Böylece, sarmalayıcı metodların zaman karmaşıklığını azaltmak amacıyla bu filtre tabanlı sistemleri kullanarak verisetinin özellik uzayını azaltabiliriz.

5.2.2. Eğiticisiz Metodlar

Eğiticisiz metodlar (Unsupervised Methods) grubu Filtreleme, Sarmalayıcı, Hibrit Metotlar alt gruplarına ayrılmaktadır (Solorio-Fernández vd., 2020). Yüksek boyuttaki verilerin analizinde eğiticisiz metodlar yaygın olarak kullanılır. Bu yöntemlerin çoğunda özelliklerin önemine odaklanılırken, özellikler arasındaki fazlalıklar göz ardı edilir.

a) Filtreleme Metodları

Filtreleme metotları, tek değişkenli(univariate) ve çok değişkenli(multivariate) olarak iki kısma ayrılır. Tek değişkenli yöntemlerde sıralı bir liste elde etmek için her bir değişkeni kesin bir kritere göre ele alarak sıralar ve final sonuç alt kümesi bu sıraya göre seçilir. Alakasız özellikleri kaldırmak için etkili olsa da özellikler arasındaki bağımlılıkları göz önüne almadıkları için gereksiz özellikleri kaldırmada verimli değillerdir. Çok değişkenli yöntemler ise gereksiz ve alakasız olan özellikleri ele alabilir, bu nedenle çoğu durumda çok değişkenli metotlarla seçilen özellik alt kümesi üzerinde öğrenme algoritmaları tek değişkenli yöntemlere nazaran daha yüksek doğruluk oranına ulaşırlar.

b) Sarmalayıcı Metodlar

Sıralı (Sequential), Calılardan Esinlenen (bio-inspared) ve Yinelemeli (Iteratif) olmak üzere üç gruba ayrılırlar. Sıralı metodlarda özellik ekleme ve çıkarma sıra ile yapılır, uygulaması kolay ve hızlıdır. Canlılardan esinlenen metotlar ise arama sürecinde rastgeleliği kullanarak lokal minimumdan kaçmayı amaçlayan metotlardır. Yinelemeli metotlar ise eğiticisiz özellik seçim problemini bir tahmin problemine dönüştürerek kombinasyonel aramadan kaçmayı amaçlayan metotlardır. Sarmalayıcı metotların dezavantajı hesapsal maliyetlerinin yüksek olmasıdır.

c) Hibrit Metodlar

Filtreleme ve Sarmalama metotların kaliteli yönlerini birleştiren, etkinlik ve verimlik arasındaki dengeyi sağlayarak daha iyi sonuca ulaşmayı amaçlayan metotlardır. Filtreleme aşamasında verinin içsel özelliklerine dayalı bir yöntem kullanılarak sıralama yapılır, sarmalama aşamasında kümeleme algoritmaları ile özellik alt kümesi en iyiyi bulmak için değerlendirilir.

ALTINCI BÖLÜM

SALDIRI TESPİTİ İÇİN VERİSETLERİ

Makine öğrenmesi tabanlı IDS sistemin geliştirilmesinde kullanılan veri seti, saldırıların yüksek doğrulukta tespit edilebilmesi için oldukça önemlidir. Ülkelerin kendi ağ kaynaklarından topladıkları bazı özel verisetleri güvenlik nedeniyle paylaşılmamaktadır. Bilimsel çalışmalarda genelde anonim veri setleri kullanılmaktadır. Bazı anonim veri setleri mevcut ihtiyaçları karşılamaktan uzaktır, yeni tip saldırıları, yeni trendleri karşılamamaktadır, bazıları istatistiksel bakımdan yetersizdir, bu nedenle kusursuz veri seti tam anlamıyla yoktur. Yeni tip saldırıları tespit edebilmek için veri setleri hem saldırı çeşitliliği hem de veri sayısı bakımından yeterli olmalıdır. Literatürde yapılan çalışmaların birçoğunda yeni nesil makine öğrenmesi teknikleri kullanılmasına rağmen seçilen veri setleri eski nesil olduğundan önerilen fikir ve metotlar yeni tip saldırıları tam olarak tespit edemeyeceği açıktır. Çok fazla sayıda veriseti olduğundan dolayı sadece makalelerde karşılaştırma amaçlı en çok kullanılan verisetlerine değineceğiz.

6.1. DARPA/KDDCup99 Veriseti

Saldırı tespit sistemleri için ilk veri seti DARPA (Defense Advanced Research Projects Agency) tarafından 1998 (R. P. Lippmann vd., 2000) ve DARPA 1999 (R. Lippmann vd., 2000) veri setleri olarak MIT Lincoln Laboratuvarında geliştirilmiştir, bilimsel çalışmalarda bu veri setlerine yoğun şekilde referans verilmiştir. Ağ için IDS tasarımı yapmak amacıyla geliştirilen veri setleri, 9 hafta boyunca kaydedilen verilerden oluşmaktadır. Veri setinin 7 haftası eğitim için, 2 haftası da test için kullanılmaktadır. TCP/IP ağ verisi, Solaris Temel Güvenlik Modülü kütük verisi, root kullanıcı göre derlenmiş ve inşa edilmiştir. 38 farklı saldırı 300 farklı işlemi içeren veri setinde, 5 farklı sınıf bulunmaktadır. Bu sınıflar;

Normal: Saldırı verisi yok.

Probe: Bu saldırı türünde sunucu veya herhangi bir kişisel bilgisayarın işletim sistemini öğrenmek ve ona göre saldırı geliştirmek, işletim sistemini portlarını taramak, belli IP aralığını taramak amacıyla yapılan saldırılardır.

Remote to Local (R2L): Bu saldırı tipi, saldırgana tüm yerel ağ erişimi haklarını verir.

User to Root (U2R): Kullanıcının tam yetkili kullanıcı olan root (Mac, Linux) veya administrator (Windows) yetkisine sahip olmasıdır. Yönetici yetkisine sahip olmayan bir kullanıcının, yöneticiymiş gibi bazı komutları işletim sisteminde çalıştırması için yaptığı saldırılardır.

Denial of Service(DoS): Hizmet reddi saldırısı, birçok tipleri bulunan bu saldırı ile ilgili açıklamalar önceki bölümde yapıldı. Kısaca TCP/IP protokolünün açıklarından yararlanarak çok fazla istek gönderilmesi sonucu, sunucunun devre dışı kalması veya hizmet akışının yavaşlamasıdır.

Veri setinin yaklaşık olarak %80'i saldırı verisi iken kalan %20'si tehlikesizdir. Makine Öğrenme tabanlı IDS tasarımı için en popüler veri seti olan KDDCup99 (Knowledge Discovery and Data Mining) veri seti(*KDD Cup 1999 Data*, y.y.), DARPA 1998 veri setinden türetilmiştir, 42 özellik yer almaktadır. 41 özellik bağlantı ile ilgili bilgileri gösterirken, son özellik saldırı tipini gösterir. Normal ve saldırı trafiği hakkında dört gigabayt ağ paketi yaklaşık 4,9 milyon kayıt barındırmaktadır. Dezavantajı, eski bir veri seti olduğundan yeni tip saldırıları içermemektedir. Örneğin malware saldırıları bu veri setinde bulunmamaktadır.

6.2. NSL-KDD Veriseti

NSL KDD (*NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, y.y.) veri seti, KDDCUP 99'un iyileştirilmiş halidir (Tavallaee vd., 2009). KDD veri setinde yer alan tekrarlı veriler silinip, homojen dağılım sağlanarak NSL-KDD veri seti oluşturulmuştur. Fazla bağlantı kayıtları (136.489 kayıt) ve bağlantı kayıtları (136.497 kayıt) test verisinden kaldırılmıştır. Böylece makine öğrenmesi algoritmaları için daha uygun bir veri seti elde edilmiştir, fakat gerçek zamanlı ağ trafiği karakteristiğini bu şekilde göstermemektedir. Veri seti etiketlendirilmiştir ve 42 özellik içermektedir, bu özellikler dört ana grupta toplanmıştır. Genel özellikler, İçerik özellikleri, Sunucu tabanlı trafik özellikleri, Zaman bağımlı trafik özellikleridir. Tıpkı KDD 99'da olduğu gibi saldırı tipleri dört farklı kategoride yer almaktadır. DoS, Probe, U2L ve R2L.Aşağıdaki çizime bakılarak veri setlerininin dönüşümü görülebilir.

Tablo 6.1: NSL KDD Veri Seti Saldırı Kategorileri

Sınıf	Alt Sınıf	Tanım
Normal	Normal	Benign
DoS	apache2, back, land, neptune, mailbomb, pod, processtable, smurf, teardrop, udpstorm, worm	DoS Saldırısında saldırgan makine veya ağ kaynakları çökertmek amacıyla saldırı düzenler.
Probe	Nmap, Ipsweep, Portsweep, Satan, Mscan, Saint	Sistem ve ağ hakkında detaylı istatiksel bilgi elde etmek için kullanılan saldırı türüdür.
R2L	httptunnel, ftp_write, guess_passwd, snmpguess, snmpgetattack, imap, spy, warezclient, warezmaster, multihop, phf, imap, named, sendmail, xlock, xsnoo, snmpgettattack	Uzaktaki makineye yetkisiz giriş. Örneğin şifrenin tahmin edilmesi
U2R	Ps, buffer_overflow, perl, rootkit, loadmodule, xterm, sqlattack, httptunnel	İşletim sisteminde yönetici (Linux işletim sisteminde root, Windows'ta administrator ayrıcalıklarına yetkisiz etişim. Örneğin çeşit bellek taşması saldırıları

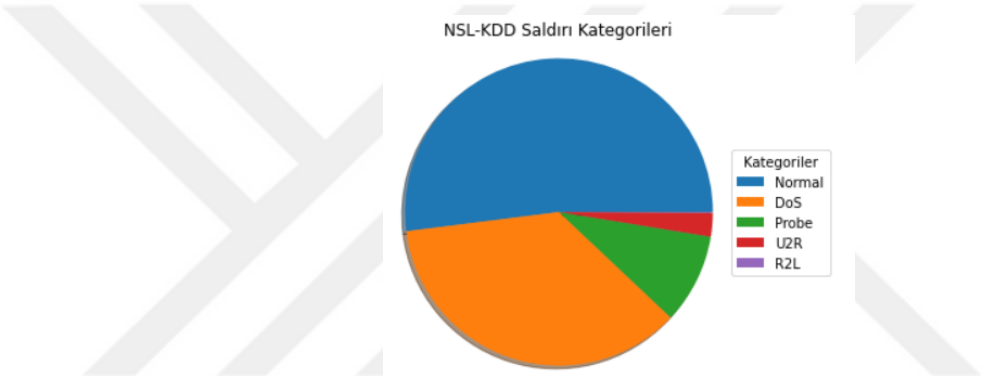
NSL-KDD verisetinde eğitim ve test verileri 4 tip saldırı, 1 normal durum olmak üzere beş farklı kategoride toplanmıştır. Verisetinde saldırılar arasında dengesiz bir dağılım vardır.

Şekil 6.1: NSL-KDD Veriseti Oluşumu.

Aşağıdaki tabloda saldırı kategorilerine göre veri boyutu görülmektedir. Bu veri setinde de kategorilere göre veri dengesizdir.

Tablo 6.2: NSL-KDD Verisetinde Eğitim ve Test verisi boyutu.

Kategori	**Eğitim**	**Test**
Normal	67343	9711
DoS	45927	7460
Probe	11656	2421
R2L	995	2885
U2R	52	67
Toplam	**125973**	**22544**



Şekil 6.2: NSL -KDD Veriseti Saldırı Kategorileri

6.3. UNSW-NB15 Veriseti

UNSW-NB15 veri setinin ham paketleri, Australian Centre Cyber Range Laboratuvarında IXIA trafik üretme aracı, üç sanal sunucu ayarlanarak oluşturulmuştur. Sunucu 1 ve 3 normal aktiviteli trafik, Sunucu 2 ise anormal aktiviteli trafik yayılımı yapmaktadır. 100 GB ham trafik, TcpDump aracı kullanılarak elde edilmiştir. Linux Ubuntu 14.04'te Argus ve Bro-IDS yardımıyla PCAP dosyalarından özellikler çıkarılmıştır. Bu veri seti, 2.540.044 kayıt ve 9 tip saldırı içermektedir (Moustafa & Slay, 2015).

Veri setinde Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode ve Worms saldırıları bulunmaktadır. Eğitim setinde 175.341 kayıt ve test için 82.332 kayıt bulunmaktadır(*The UNSW-NB15 data set description*, y.y.). Bu veri seti, ağ trafiğinin gerçek modern normal ve çağdaş sentezlenmiş saldırı aktivitelerinin

birleşiminden oluşur. Var olan ve yeni metotlardan yararlanılarak UNSW-NB15 veri seti özellikleri üretilmiştir.

Tablo 6.3: UNSW-NB15 Veriseti

Kategori	**Eğitim**	**Test**
Normal	56000	37000
Analysis	2000	677
Backdoor	1746	583
DoS	12264	4.089
Exploits	33393	11132
Fuzzers	18184	6062
Generic	40000	18871
Reconnaissance	10491	3496
Shell Code	1133	378
Worms	130	44
Toplam	**175.341**	**82.332**

6.4. BoT-IoT Veriseti

BoT-IoT veri seti (*The Bot-IoT Dataset | UNSW Research*, y.y.) Cyber Range laboratuvarlarında doğal bir ağ ortamı tasarlanarak ve IoT altyapısına odaklanılarak oluşturulmuştur. Ağ izleme tarafından toplanan ham veri seti, 70 GB paket yakalama dosyası (PCAP) içerir. Makine öğrenimi modelleri ile toplanan veriler Wireshark (*Wireshark · Go Deep.*, y.y.), Argus (*Argus - System and Network Monitoring Software*, y.y.) veya Zeek (*The Zeek Network Security Monitor*, y.y.) gibi Ağ Analizi Araçları kullanılarak işlenir; yazarlar BoT-IoT veri seti için Argus Network Monitory aracını kullanmışlardır. Bot-IoT veri kümesi 72 milyondan fazla kayıt, üç bağımlı özellik ve 43 bağımsız özellik içerir. BoT-IoT veri setinde, saldırı kategorisi sütununda DoS(Denial of Service), DDoS (Distributed DoS), OS Service Scan ve Keylogging saldırıları bulunmaktadır. Veri setinde pratik çalışabilmek için MySQL veritabanı sorguları ile orijinal veri setinin %5'i çıkartılmıştır. Son 10 En iyi alt küme, orijinal veri kümesinin %5'idir ve 43 yerine on bağımsız özellik içerir. Bu alt küme 3 bağımlı (saldırı, kategori, alt kategori) ve 16 bağımsız özellik içerir, 16 özelliğin 6'sı akış tanımlayıcıdır ve geri kalan 10'u (Koroniotis vd., 2019) tarafından en iyisi olarak kabul edilir. Daha önceki çalışmalara baktığımızda bu veri setinin genellikle Nesnelerin İnterneti'nde saldırı tespiti için tercih edildiğini görüyoruz.

Tablo 6.4: BoT-IoT %5 Alt küme/ 10 En iyi sınıf dağılımı

Kategori	Veri Sayısı	Alt Kategori	Veri Sayısı
Normal	477	Normal	477
DDoS	1926624	TCP	977380
		UDP	948255
		HTTP	989
DoS	1650260	TCP	615800
		UDP	1032975
		HTTP	1485
Reconnaissance	91082	OS Fingerprinting	17914
		Service Scanning	73168
Theft	70	Keylogging	73
		Data Exfiltration	6

YEDİNCİ BÖLÜM

SALDIRI TESPİT SİSTEMLERİNDE AKILLI YAKLAŞIMLAR

Bu bölümde inceleyeceğimiz makine öğrenmesinde en önemli noktalardan biri özellik seçimidir, önceki bölümlerde özellik seçimi için birçok istatistiksel yöntemlere değinildi. Makine öğrenmesi algoritmaları her zaman performanslı sonuçlar vermeyebilir, eğitim çok uzun sürebilir, özellik büyük veri olduğunda performansı düşebilir, sistem veya donanım kaynakları modeli eğitmek için yetersiz kalabilir. Diğer taraftan veriye aşırı uydurma gerçekleşebilir. Ayrıca, veri kümesi ne kadar büyük olursa, eğitim ve test süreleri artmakta, büyük miktarda bellek gibi daha güçlü donanımlara ihtiyaç duyulmaktadır. Çünkü önemsiz özelliklerin elenmesi ve sadece önemli özelliklerin seçilmesi bu gibi dezavantajların önüne geçecek ve sonucu hem hesapsal hem de zaman bakımından yönde etkileyecektir.

Optimizasyon algoritmaları ile topluluk algoritmalarının beraber kullanılarak performansı daha yüksek sonuçlar alınmaktadır. Bu çalışmada topluluk öğrenme algoritmalarıyla birlikte özellik seçimi için Gri Kurt Optimizasyon yöntemi önerilmiştir. Özellikle Nesnelerin İnternetine yapılan saldırıların yer aldığı BoT-IoT veriseti ve ağ tabanlı saldırıların yer aldığı NSL-KDD veri seti üzerinde önerilen model farklı algoritmalarla test edilmiştir.

Topluluk öğrenme algoritmaları daha önceki bölümde anlatıldığı ve alt tarafta kullandıkları matematiksel yöntemler farklı olmakla birlikte çoğu karar ağacı tabanlıdır. Örneğin yine önceki bölümde değinilen Rastgele Orman algoritması bir torbalama(bagging) algoritmasıdır ve karar ağaçları birleşiminden oluşur. AdaBoost, XGBoost, LightGBM, CatBoost gibi artırma algoritmaları ise sıralı şekilde çalışır. Biz bu algoritmaların performanslarını kullandığımız veri setleri üzerinde ölçüyoruz.

Önerilen Topluluk Öğrenme Algoritması + GWO modeli ile veri setinden uygun özniteliklerin seçimi GWO algoritması ile gerçekleştirilmiştir. Bu çalışmadaki amacımız;

- Aşırı uydurmanın(overfitting) önlenmesi,
- Doğruluk oranı vb. performans kriterlerinin artırılması,

- Büyük veri üzerinde özellik seçimi ile eğitim süresini kısaltılarak, güçlü donanım kaynağı ihtiyacının azaltılması,
- Saldırılara karşı daha erken cevap verilerek, güvenliğin sağlanması

Gri Kurt algoritmasının (GWO) birçok mühendislik problemine uygulandığı gibi saldırı tespit sisteminde kullanacağımız makine öğrenmesi algoritmalarıyla kolay uygulanabilmesi en önemli avantajlarından biridir. GWO Algoritması gücü düşük donanımlarda bile çalışan hesapsal karmaşıklığı düşük bir algoritmadır.

7.1. Saldırı Tespitinde Makine Öğrenmesi Algoritmaları

Saldırı Tespit Sistemlerinde şimdiye kadar yapılan çalışmalar incelendiğinde anomali tabanlı saldırılar için k En Yakın Komşu(kNN), Destek Vektör Makineleri (SVM), Yapay Sinir Ağları (ANN), Naive Bayes(NB) ve Karar Ağaçları(DT) gibi klasik makine öğrenmesi algoritmaları tek olarak veya hibrit bir şekilde kullanılmıştır.

Makine öğrenmesinde boyut azaltmak için en eski teknik özellik çıkarımıdır. Özellik çıkarımı için PCA(Principal Component Analysis-Temel Bileşenler Analizi), LDA (Local Discriminate Analysis-Yerel Ayrımcılık Analizi), CCA(Canonical Correlation Analysis-Kanonik Korelasyon Analizi) gibi teknikler kullanılmıştır(Ray vd., 2021).

Serpen ve arkadaşları(Serpen & Aghaei, 2018) ADFA-LD veri seti üzerinde bilgisayar tabanlı bir kötüye kullanım tespit çalışması sunmuşlardır. Öznitelik çıkarımı için PCA ve sınıflandırma için KNN algoritması kullanmışlardır. Sınıflandırıcı performans sonuçlarına göre, ikili ve çok sınıflı sınıflandırıcılar için doğruluk 99.98 veya daha yüksektir.

Saleh ve arkadaşları (Saleh vd., 2019) çoklu sınıflandırma problemini çözmek için bir Hibrit IDS(HIDS) tasarlamışlar, verilerin boyutunu küçültmek için Naive Bayes öznitelik seçme tekniğini uygulamışlardır. Ek olarak, yüksek oranda yanlış sınıflandırmayı önlemek için Optimize Edilmiş Destek Vektör Makinesi (OSVM) kullanan aykırı değer reddetme tekniği sunmuşlardır. Daha sonra saldırıların tespiti için Öncelikli K-En Yakın Komşu (PKNN) algoritması kullanılmıştır. Önerilen yöntem KDD Cup'99, NSL-KDD ve Kyoto 2006+ veri setleri ile karşılaştırılmıştır. Test sonuçlarına göre, Kyoto veri seti üzerinde OSVM uygulandığında tespit oranı

%92'dir. Ayrıca OSVM algoritması KDD Cup'99 veri setinde 91,70 tespit oranı, algoritma NSL-KDD veri setinde %95,77 tespit oranı vermektedir.

Abuoromman et al (Aburomman & Ibne Reaz, 2016), izinsiz giriş tespiti için KNN-PSO-SVM topluluk yöntemi önermiler ve KDD Cup'99 verisetinden rastgele beş alt küme seçmişlerdir.

Li (Li vd., 2012) tarafından yapılan çalışmada, KDDCup99 veri seti RBF(Radyal tabanlı) çekirdek fonksiyonu kullanılarak önceden belirlenmiş kategorilere göre (Dos, Probe or Scan, U2R, R2L, Normal) sınıflandırılmıştır. 41 özellikten, alt özellik kümesi özellik atma politikası kullanılarak 19 özelliğe düşürülmüştür. Aynı çalışmada Karınca Koloni Optimizasyonu kullanılarak eğitim kümesinin alt kümesi belirlenmiştir. Bu yaklaşımda sınıflandırıcı genelleştirmek ve KDDCup99 setindeki bias'ı minimize etmeyi amaçlamışlardır. 10 katmanlı çapraz doğrulama performansında doğruluk oranı %98'dir. En kötü performans %52 ile U2R kategorisindedir.

Amiri ve arkadaşları (Amiri vd., 2011) tarafından yapılan çalışmada en küçük kareli SVM kullanılarak büyük boyuttaki veri setlerini hızlıca eğitmek amaçlanmıştır. KDDCup99 veri setindeki özellik 41'den 19'a düşürülmüş, üç farklı özellik çıkarım yöntemi kullanılmıştır. Bunlar; Sınıflandırma performansını maksimize eden özelliği çıkarma, Çoklu bilgi tabanlı (Multi information based) ve Korelasyon tabanlı (Correlation based) tekniklerdir. Sonuçlar çoklu bilgi tabanlı yaklaşımın diğer ikisine göre daha iyi olduğunu göstermektedir.

SVM ile anormallik tespiti ile ilgili diğer bir çalışmada Hu ve arkadaşları(Hu & Hu, 2003) Robust SVM (RSVM) kullanmışlardır. RSVM, ayırt edici hiper düzlemin daha düzgün olduğu ve regülasyon parametresinin çalışmalarındaki anormallik sınıflandırıcısı olarak otomatik olarak belirlendiği bir SVM varyasyonudur. Bu çalışmada veri seti olarak DARPA 98 veri seti eğitim ve testte kullanılmıştır. Sonuç olarak; yanlış alarm oranı olmadan %75 doğrulukta, iyi bir sınıflandırma performansı %3 FAR ve %100 doğruluk oranı elde edilmiştir.

Al-Qatf (Al-Qatf vd., 2018) ve arkadaşları bir derin öğrenme yaklaşımı olan Self Taught Learning (STL)'i önermişlerdir. Bu yaklaşım Özellik öğrenme ve boyut azaltma amacıyla önerilmiştir. Eğiticisiz ayrık otokodlayıcı (sparse-autoencoder) ile ön işlemeden sonra SVM ile NSL-KDD veri seti üzerinde sınıflandırma yapılmıştır.

Sahu ve arkadaşları (Sahu & Mehtre, 2015), Kyoto 2006+ veri seti kullanılarak J48 algoritması ile sınıflandırma yapılmıştır. Eğitim ve test esnasında 134665 ağ örneği kullanılmış, %97,2 doğruluk oranıyla bağlantının saldırı olup olmadığı tespit edilmiştir. Shadi ve arkadaşları (Aljawarneh vd., 2019), NSL KDD veri seti ile WEKA uygulaması ile %99,88 doğruluk oranında saldırı tespiti yapmışlardır. 10 katlı çapraz doğrulama test uygulandığından %90.01 doğruluk oranı elde edilmiştir.

Otoum ve arkadaşları (Otoum vd., 2019), Restricted Boltzmann Machine-based Clustered IDS(RBC-IDS) önermişlerdir. Daha önce yine kendi geliştirdikleri ASCH-IDS (Otoum vd., 2018) ile karşılaştırmışlardır. Doğruluk oranı ve tespit oranı ASCH-IDS ile karşılaştırıldığında sonuçlar aynı çıkmakla birlikte zaman olarak iki kat daha iyi sonuç elde etmişlerdir. Deneysel çalışmalarda KDDCup 99 veri seti kullanılmıştır.

Gao ve arkadaşları (Gao vd., 2015) tarafında yapılan çalışmada saldırı tespit sisteminde büyük veri sınıflandırması yapılmıştır. KDDCup 99 veri seti kullanılarak yapılan çalışmada Derin İnanç Ağlarının, SVM ve ANN'den performans açısından daha iyi sonuç verdiği belirtilmiştir.

Saleh ve diğerleri (Saleh vd., 2017) kNN, SVM ve NB algoritmalarını kullanan bir hibrit sistem önermiştir. Bu çalışmada, öznitelik seçimi için bir NB sınıflandırıcısı kullanılmıştır; aykırı değer reddi için SVM algoritması, karar verme için kNN sınıflandırıcısı kullanılmıştır.

Yin ve arkadaşları (Yin vd., 2017), önerdikleri RNN derin öğrenme yaklaşımı ile NSL-KDD veri seti üzerinde ikili ve çoklu sınıflandırma deneyler yaparak performansı değerlendirmişlerdir. Önerilen RNN-IDS modelini J48, Yapay Sinir Ağları, Rastgele Orman ve Dektek Vektör Makineleri gibi algoritmalarla karşılaştırmışlardır. Chawla ve arkadaşları (Chawla vd., 2019), bilgisayar tabanlı bir IDS modeli önermişlerdir, bu modelde RNN/CNN kombinasyonu kullanmışlardır. Kapılı RNN (Gated RNN) ile CNN'lerin birleşimi ile IDS de başarımı artırmışlardır. Deneysel çalışmalarda ADFA veri seti kullanmışlardır. LSTM modeli ile karşılaştırıldığında tespit zamanı düşüktür.

Wu ve arkadaşları, Evrişimsel Sinir Ağı (CNN) ağını kullanarak NSL-KDD veri seti üzerinde yaptıkları çalışmada tek katmanlı ağda %78,33, iki katmanlı ağda %79,48 doğruluk oranı elde etmişlerdir. (Wu vd., 2018)

7.2. Topluluk Öğrenme Algoritmaları ile Yapılan Çalışmalar

Topluluk öğrenme algoritmaları farklı veri setleri üzerinde yapılan modelleme çalışmaları incelendiğinde, genelde topluluk öğrenme algoritmalarının klasik algoritmalara göre daha başarılı olduğu görülmektedir. Wang ve arkadaşları rüzgar gülünün hızını komşu konumlardan alınan ölçümleri kullanarak AdaBoost ile ELM algoritmasını kullanarak tahmin edilen bir çalışma yapmışlardır.(Wang vd., 2022). Qu ve arkadaşları bina elektrik tüketim sisteminde bir saldırı olduğunda tespit eden(Qu vd., 2021) sistemi Genetik Algoritma kullanan AdaBoost temeli bir anomali tespit sistemi geliştirmişlerdir. M-AdaBoost-A ismini verdikleri topluluk öğrenme algoritması ile ağdaki sınıf dengesizliklerini eğrinin altında kalan alanı arturma(boosting) sürecine dahil eden bir yaklaşım sunmuşlardır (Y. Zhou vd., 2020).

7.3. Gri Kurt Algoritması ile Yapılan Çalışmalar

Makine öğrenme algoritmalarıyla birlikte Gri Kurt gibi birçok optimizasyon algoritması beraber kullanılmıştır. Bunlardan en çok kullanılan Destek Vektör Makineleri (SVM) algoritmasıdır.

Safaldin ve arkadaşları (Safaldin vd., 2021) ikili gri kurt optimizasyon algoritması ile SVM algoritmasını beraber kullanarak 3,5 ve 7 kurttan oluşan çalışmalarını NSL-KDD99 veriseti üzerinde yapmışlar. Bu veriseti eski olduğundan IoT için uygun değildir ve sadece Parçacık Sürü Optimizasyonu ile karşılaştırılmıştır.

ZHou ve arkadaşları (J. Zhou vd., 2021), SVM makine öğrenme algoritması ile birlikte genetik algoritma ve Gri Kurt Optimizasyon algoritmalarını kullanarak zeminin deprem kaynaklı sıvılaşma modelini tahmin eden bir çalışma sunmuş ve üzerinde %90'ın üzerinde bir doğruluk oranına ulaşmışlardır.

7.4. Önerilen Yaklaşım ve Deneysel Sonuçlar

NSL-KDD veriseti Makine öğrenme algoritmalarıyla modellemeden önce bir dizi ön işlemeden geçirilmiştir.

- null ya da boş değer olup olmadığı kontrol edilmiştir, verisetinde herhangi bir boş değer bulunmamaktadır.

- Gereksiz özellik elemesi: NSL-KDD verisetinde 43 özellik bulunmaktadır, difficulty_level özelliği çıkarılmıştır.
- Sayısal değer içermeyen sütunlar belirlemiştir. Protocol_type, service ve flag sütunları metinsel değerler içermektedir. Protokol tipini barındıran protocol_type sütunu TCP, UDP, ICMP gibi protokol değerleri içermektedir.
- Sayısal değer içeren sütunlar normalizasyon işleminden geçirilmiştir.
- Sayısal değer içermeyen protocol_type, service , flag gibi kategori sütunları Tek Sıcak Kodlama(One Hot Encoding) işleminden geçirilerek sayısal değerler alması sağlanmıştır.
- Saldırılar 4 ana kategori altında toplanmıştır.

Tablo 7.1: NSL-KDD Önişleme- Saldırı Sınıflandırma

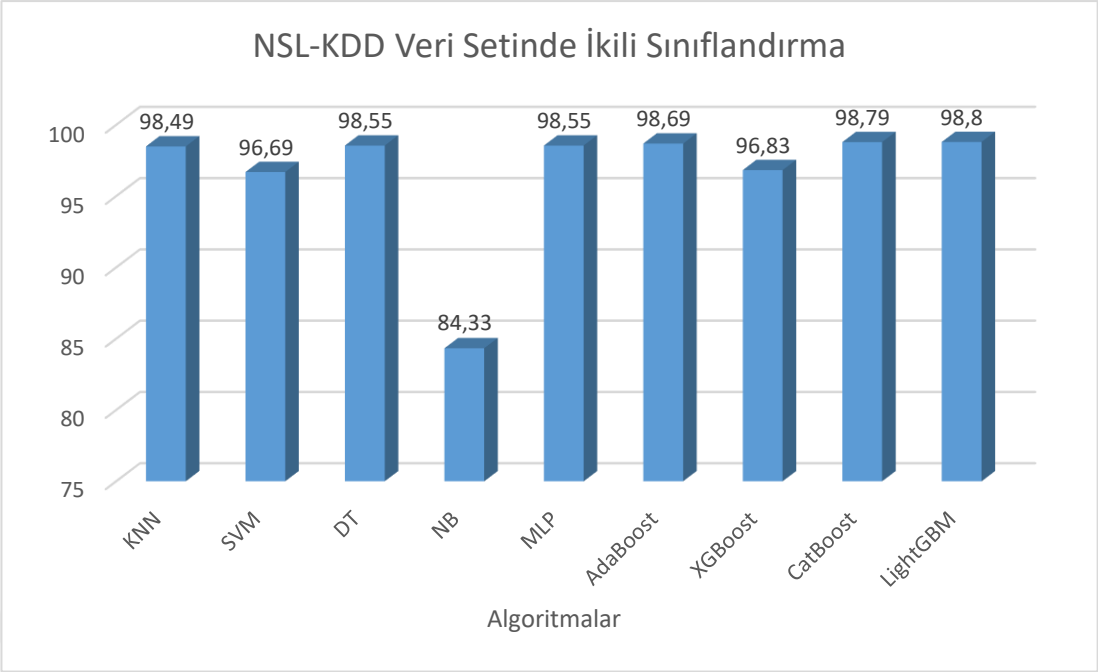
Alt Sınıf	Toplandıkları Sınıf
Normal	Normal
apache2, back, land, neptune, mailbomb, pod, processtable, smurf, teardrop, udpstorm, worm	DoS
Nmap, Ipsweep, Portsweep, Satan, Mscan, Saint	Probe
httptunnel, ftp_write, guess_passwd, snmpguess, snmpgetattack, imap, spy, warezclient, warezmaster, multihop, phf, imap, named, sendmail, xlock, xsnoo, snmpgettattack	R2L
Ps, buffer_overflow, perl, rootkit, loadmodule, xterm, sqlattack, httptunnel	U2R

- 5 katlı çapraz doğrulama yapılarak makine öğrenmesi algoritmalarıyla modellenmiştir. Çapraz doğrulamada eğitim verisi belirlenen sayı kadar parçaya bölünerek her bir parça üzerinden ayrı işlem yapılır ve daha sağlıklı sonuçlar elde edilir.

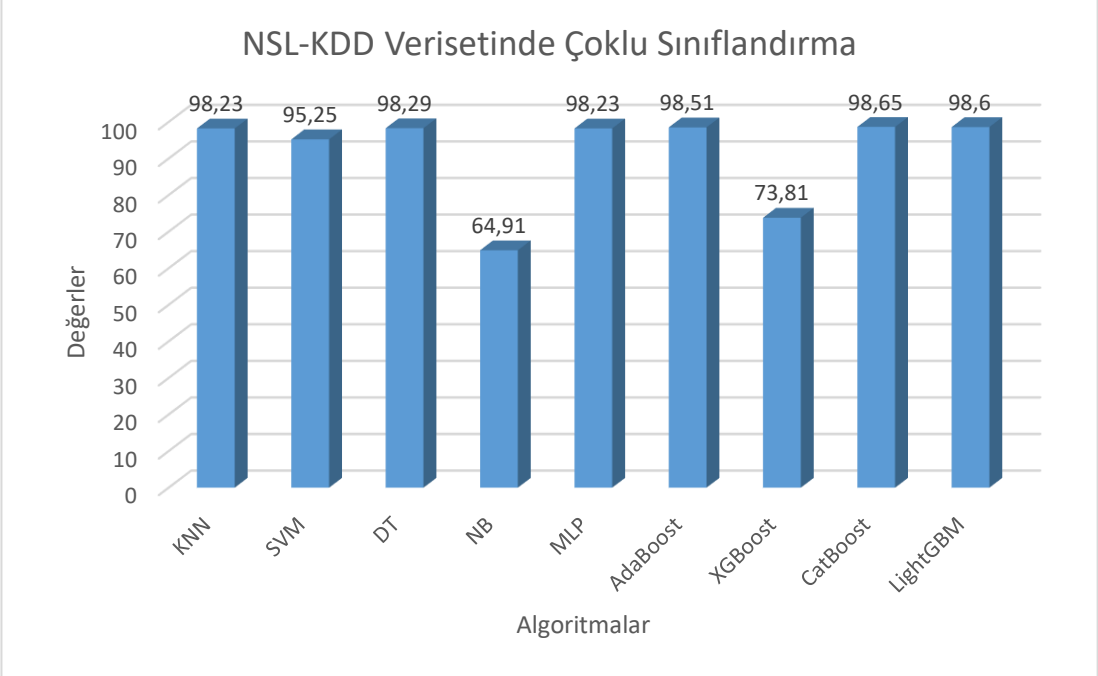
Tablo 7.2'de görüldüğü üzere ikili sınıflandırmada genelde algoritmalar başarılıdır. Fakat ikili sınıflandırma sadece Saldırı Var/Yok bilgisi vermektedir. Saldırının hangi kategoriden olduğunun bilinmesi önemlidir. Saldırı türüne göre savunma taktiği geliştirilmesi için kategorinin mutlaka algoritmalar tarafından doğru tahmin edilmesi gerekir. Beklenilen şekilde Karar Ağaçları, AdaBoost, XGBoost ve LightGBM gibi topluluk öğrenme algoritmalarında doğruluk oranı çoklu sınıflandırmada %98 ve üstü olacak şekilde yüksektir. XGBoost algoritması sadece %73,81 oranında kalmıştır.

Tablo 7.2: NSL-KDD Veriseti üzerinde Doğruluk Oranına göre Sonuçlar

Algoritma	**İkili Sınıflandırma (%)**	**Çoklu Sınıflandırma(%)**
LR	98.49	95.41
KNN	98.49	**98.23**
SVM	96.69	95.25
DT	98.55	98.29
NB	84.33	64.91
MLP	98.55	**98.23**
AdaBoost	98.69	**98.51**
XGBoost	96.83	73.81
CatBoost	98.79	**98.65**
LightGBM	98.80	98.60



Şekil 7.1: NSL-KDD verisetinin İkili Sınıflandırılması



Şekil 7.2: NSL-KDD Veri Seti Çoklu Sınıflandırma

GWO algoritmasında diğer Genetik Algoritma gibi birçok optimizasyon algoritmasında olduğu gibi amaç fonksiyonu (fitness function) seçimi problemin çözümünde çok önemlidir. GWO algoritmasında tercih ettiğimiz Amaç fonksiyonu aşağıdaki formülle ifade edilmiştir. Aşağıdaki ifadede c değeri 0,9 olarak seçilmiştir.

*Amaç Fonksiyonu= -(c*doğruluk + (1-c) * 1/(özellik sayısı)))* (7.1)

Burada amaç fonksiyonu seçilirken iki önemli kriter gözetilmektedir;

- Doğruluk (accuracy) değerini yüksek tutmak,
- Özellik sayısını mümkün olduğunca düşürmek.

Önerilen amaç fonksiyonu dışında başka amaç fonksiyonları da denenmiştir. Al-tashi ve arkadaşları aşağıdaki amaç fonksiyonunu Koroner Arter hastalığı sınıflandırmasında kullanmışlardır. Burada FN tüm verisetindeki toplam özellik sayısı, FL seçilen özelliğin uzunluğu, α sınıflandırma doğruluk ağırlığı, β özellik seçimi kalitesi ve β = 1- α olmak üzere SVM sınflandırıcı ile doğruluk değerini hesaplamışlardır (Al-Tashi vd., 2019).

$$Amaç\ Fonksiyonu\ =\ \alpha * doğruluk\ +\ \beta\frac{FN-FL}{FL} \quad (7.2)$$

Fakat bu amaç fonksiyonu bu çalışmada kullanılan verisetlerinde eğitim süresinin çok uzun zaman almasına neden olduğundan dolayı kullanılmamıştır.

Bu çalışmada Intel i7-7700 HQ 2.8 Mhz(8 Core) işlemci, 16 GB RAM, NVidia Geforce 1050TX ekran kartına sahip bir bilgisayar ortamında kullanılmıştır. Geliştirmeler Python Scikit-learn kütüphanesi ile Anaconda ortamında çalıştırılmıştır. CatBoost hariç, diğer algoritmalar Google Colab ortamında TPU üzerinde de denenmiştir. CatBoost algoritmaları kendi dosya ve klasörlerini oluşturup üzerinde işlem yaptığı için şu an için Colab üzerinde yazma yetkisi kısıtlamasından dolayı mümkün değildir. Diğer algoritmalar sorunsuz olarak çalışmaktadır.

Daha önceki bölümde anlatıldığı gibi NSL-KDD veriseti üzerinde önişleme yapıldıktan sonra topluluk öğrenme algoritmalarında denenmiştir. Topluluk öğrenme

algoritmalarının parametreleri değiştirilmeden varsayılan parametreler kullanılarak eğitim ve test aşamaları gerçekleştirilmiştir. Gri Kurt algoritması amaç fonksiyonuna göre veri setinden özellikleri seçip numaralarını döndürmektedir. Aşağıdaki tabloda onlu şekilde özellikler sıralanmıştır. Tek Sıcak Kodlama (One Hot Encoding) yapılmasından dolayı özellik sayısı son etiket dahil 123 sütundan (0-122 arası) oluşmaktadır. Ekler kısmında bu liste sunulmuştur.

7.4.1. Önerilen AdaBoost + GWO ile NSL-KDD Veriseti Test Sonuçları

Klasik AdaBoost Algoritması en eski algoritmalardan biridir, diğer Artırma(Boosting) algoritmalarına göre eğitim ve test süresi verisetine göre de değişmekle birlikte sonraki bölümlerde de göreceğimiz gibi daha yavaş olabilmektedir.

Klasik AdaBoost algoritmasında %72,98 doğruluk oranı yakalanırken, önerilen AdaBoost+GWO ile özellik seçimi yapılarak doğruluk oranı %80,98'e çıkmıştır. Kesinlik değeri %66,62'den %72,82'ye, Duyarlılık değeri %45,79'dan %56,15'e, F1-Skor değeri de %47,56'dan %57,88'e çıkmıştır.

Tablo 7.3: NSL-KDD verisetinde AdaBoost ve AdaBoostGWO Karşılaştırması

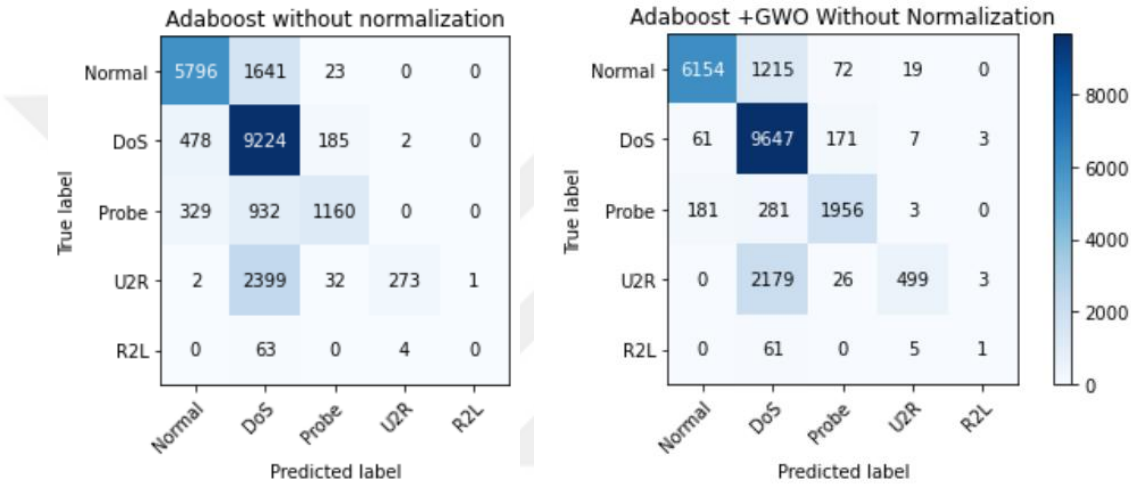
	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
AdaBoost	%72.9817	% 66.6295	% 45.7938	%47.5631
AdaBoost+GWO	%80.9839	%72.8234	%56.1531	%57.8882

AdaBoost algoritmasının temel tahmin edici (base_estimator) olarak maksimum derinliği başlangıçta 1 olan Karar Ağacı sınıflandırıcı (DecisionTreeClassifier) kullanmaktadır. Diğer bir parametre olan n_estimator parametresi tahmin edicilerin maksimum sayısını belirtmektedir, 1 ile sonsuz arasında değerler alabilmektedir, varsayılan değeri 50'dir. Öğrenme oranı (Learning_rate) parametresi her bir iterasyonda ağırlık her bir sınıflandırıcıya uygulanmaktadır, varsayılan değeri 1.0 olmaktadır.

```
[  0   1   9  10  12  14  15  18  25  26  30  34  36  44  52  53  54  55
  56  58  62  63  64  65  66  68  72  74  77  81  83  85  86  87  95  97
 101 103 104 105 106 107 112 114 116 117 118 120]
```

Şekil 7.3: AdaBoost+GWO yaklaşımı tarafından seçilen özellikler

Önerilen AdaBoost+GWO algoritması 122 özellikten 48 tanesini seçmiştir. Bu özellikler diğer çalışmalarda kullanılabilir fakat diğer algoritmaların seçtiği özellikleri de inceledikten sonra karar vermek gerekmektedir.



Şekil 7.4: Karmaşıklık Matrisi AdaBoost ve AdaBoost+GWO

Her iki algoritmanın Karışıklık Matrisleri karşılaştırıldığında; DoS, Probe, U2R ve R2L saldırılarını önerilen modelin daha iyi tespit ettiği Şekil 7.4'de görülebilir.

7.4.2. Önerilen CatBoost+GWO ile NSL-KDD Veriseti Test Sonuçları

CatBoost algoritması diğer algoritmalardan daha sonra geliştirilen bir algoritma olmasına rağmen oldukça yavaştır, test sonuçları bölüm sonunda grafik olarak sunulacaktır. Doğruluk oranı AdaBoost algoritmasından yüksektir. Önerilen GWO yaklaşımıyla doğruluk oranı %76,1'den %81,44'e çıkmıştır. Kesinlik, Duyarlılık ve F1 Skor değerlerinde de benzer artımlar söz konusudur.

Tablo 7.4: NSL-KDD verisetinde CatBoost ve CatBoost+GWO Karşılaştırması

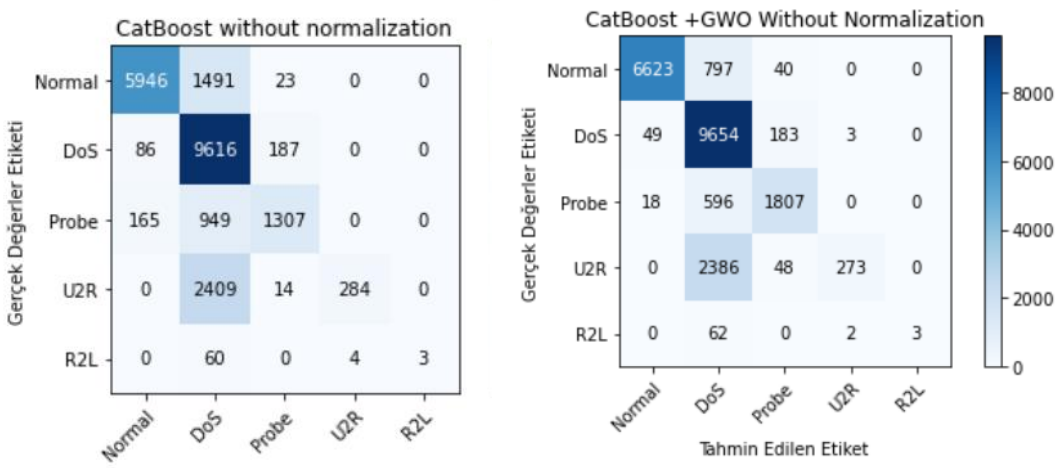
	Doğruluk	Kesinlik	Duyarlılık	F1-Skor
CatBoost	%76.1000	%89.2266	%49.1799	%51.9062
CatBoost+GWO	%81.4407	%91.1392	%55.1210	%56.6745

Testler sonrasında NSL-KDD verisetindeki 122 özellikten 53'ü CatBoost+GWO tarafından seçilmiştir.

```
[  1   3   7   9  10  12  13  19  23  24  25  26  27  32  33  36  39  40
  41  42  43  44  47  51  52  53  55  56  58  62  65  67  70  71  72  74
  75  78  81  84  86  91  92  93  94 104 105 109 112 113 115 118 120]
```

Şekil 7.5: CatBoost+GWO yaklaşımı tarafından seçilen özellikler

Karmaşıklık matrisinden görüleceği gibi önerilen CatBoost+GWO yaklaşımı başta DoS olmak üzeri tüm saldırı tiplerinde daha yüksek tahmin etmiştir.



Şekil 7.6: Karmaşıklık Matrisi CatBoost (Solda) ve CatBoost+GWO(Sağda)

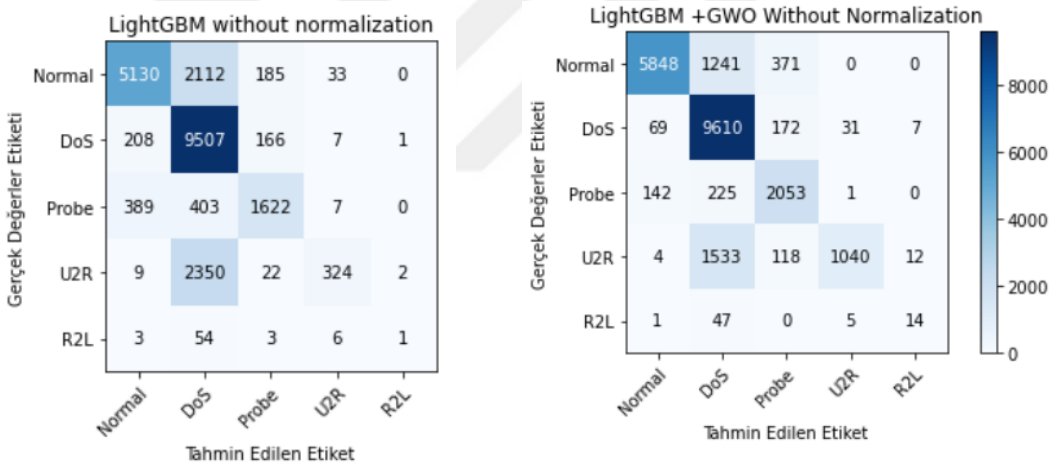
7.4.3. Önerilen LightGBM+GWO ile NSL-KDD Verseti Test Sonuçları

LightGBM Algoritmasının diğer algoritmalara göre bu veri seti üzerinde daha hızlı çalıştığı görülmüştür. Özellikle işlemci çekirdekleri arasında işi dağıtan n_jobs parametresi ayarlanarak performans artırılmıştır, tüm algoritmalarda bu parametre

maalesef yoktur, deneyde kullanılan işlemci çekirdek sayısı sekiz olduğu için performans artmıştır, daha yüksek çekirdeğe sahip sistemlerde daha iyi sonuçlar alınabilir. NSL-KDD veri seti üzerinde LightGBM algoritması CatBoost algoritmasına göre yaklaşık 18 kat daha hızlı çalışmaktadır. Önerilen Gri Kurt Algoritmasıyla yapılan özellik seçiminde Doğruluk Oranı %73,56'dan %82,35'e çıkmıştır. Kesinlik, Duyarlılık ve F1 Skor metriklerinde de benzer artışlar görülmüştür. Fakat F1 skorunda görülen %16 artış diğer algoritmalara göre fazladır. F1 skor Kesinlik metriği ile Duyarlılık metriği arasındaki harmonik ortalamayı temsil eder.

Tablo 7.5: NSL-KDD ile LightGBM ve LightGBM+GWO Karşılaştırması

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
LightGBM	%73.5628	%69.4826	%49.0725	%50.6341
LightGBM+GWO	%82.3501	%77.4008	%63.9368	%66.9329



Şekil 7. 7: Karmaşıklık Matrisi LightGBM ve LightGBM+GWO.

Karmaşıklık Matrisi incelendiğinde DoS saldırılarının önerilen LightGBM+GWO ile 9610 adetinin doğru tahmin edildiği görüşmektedir. Probe saldırısına baktığımızda klasik LightGBM'de doğru tahmin 1622 iken önerilen yaklaşımda 2053 doğru tahmin yapılmıştır, diğer U2R ve R2L saldırıları için de benzer artışlar görülmektedir.

```
[  0   6   7  12  15  16  18  21  23  24  25  26  33  38  39  41  43  46
  54  55  58  60  62  65  66  69  71  72  75  78  79  82  83  86  88  93
  94  95 100 102 104 105 111 113 117 119]
```

Şekil 7.8: LightGBM + GWO yaklaşımı tarafından seçilen özellikler.

LightGBM+GWO yaklaşımı ile 122 özellikten 46 tanesi seçilmiştir. Amacımız daha az özellikle daha doğru sonuca ulaşmak olduğu için LightGBM+GWO modelini özellik seçmede de başarılı olduğunu söyleyebiliriz.

7.4.4. Önerilen XGBoost +GWO ile NSL-KDD Veriseti Test Sonuçları

XGBoost algoritması Kaggle Makine Öğrenmesi yarışmalarında çok tercih edilen bir algoritmadır. Önerilen XGBoost yaklaşımı ile doğruluk oranı %76,40' dan %82,54'e çıkarken diğer algoritmalardan farklı olarak Kesinlik değeri %85,22'den, %70,86'ya düşmüştür. Duyarlık ve F1-Skor değerlerindeki artış da tablodan görülmektedir.

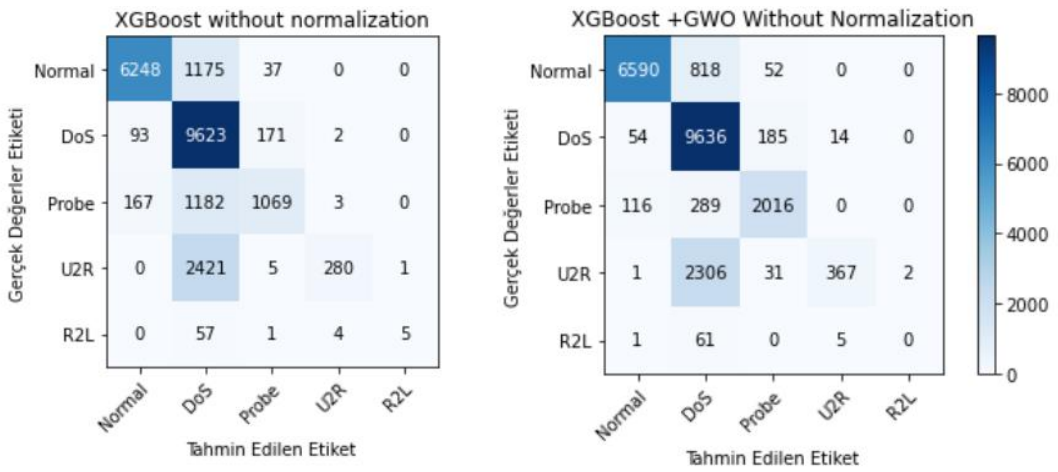
Tablo 7.6: NSL-KDD ile XGBoost ve XGBoost+GWO Karşılaştırması

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
XGBoost	%76.4061	%85.2205	%48.605	%51.7244
XGBoost+GWO	%82.5452	%70.8603	%56.5216	%57.1791

```
[  7   9  11  12  13  14  15  16  22  23  24  25  26  30  33  36  38  41
  42  43  44  45  48  49  50  52  56  58  60  61  63  64  65  66  67  74
  85  87  88  91  93  98  99 102 104 107 108 111 112 113 114 115 117 118
 120 121]
```

Şekil 7.9: XGBoost+GWO yaklaşımı tarafından seçilen özellikler.

XGBoost+GWO algoritması tarafından 122 özellikten 56 tanesi seçilmiştir. Bu verisetinde XGBoost'un özellik seçimi AdaBoost, LightGBM ve CatBoost'a göre yüksektir.



Şekil 7.10: Karmaşıklık Matrisi XGBoost ve XGBoost+GWO

7.4.5. Rastgele Orman+GWO ile NSL-KDD Veriseti Test Sonuçları

Ratsgele Orman algoritması diğer algoritmalara göre daha fazla modeli optimize etmek için daha fazla hiperparametre seçeneği sunmaktadır. Hiperparametre değerlerindeki küçük değişiklikler performansı önemli ölçüde eklemektedir. Bu algoritmada da n_jobs parametresini kullanarak işlemci çekirdeklerinden yararlandık, diğer parametreleri varsayılan şekilde bıraktık.

Tablo 7.7: NSL-KDD ile RF ve RF+GWO Karşılaştırması.

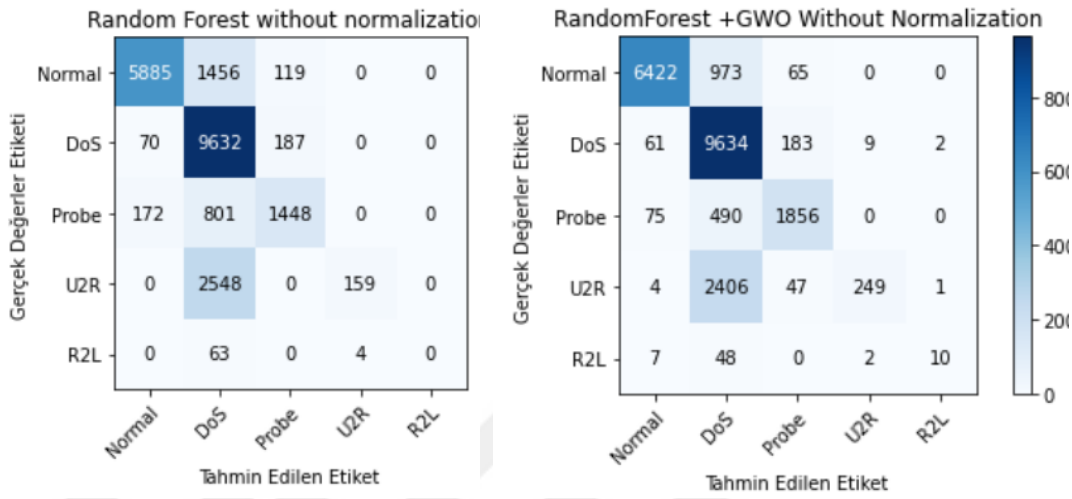
	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
Rastgele Orman	%75.9581	%68.5156	%48.3944	%49.2118
Rastgele Orman+GWO	%80.6024	%85.5669	%56.8587	%59.3458

Tablodan görüldüğü üzere önerilen Rastgele Orman +GWO modeli ile doğruluk oranı %75,95'ten %80,60'a çıkarılmıştır.

```
[  1   6  10  11  13  14  17  18  19  20  21  23  25  26  33  36  38  41
  43  46  48  49  57  59  60  61  63  64  66  71  72  75  76  80  83  84
  86  92  96 100 104 107 109 110 112 113 114 117 119 120 121]
```

Şekil 7.11: Rastgele Orman+GWO yaklaşımı tarafından seçilen özellikler

Önerilen RF+GWO modeli ile 122 özellikten 51 tanesi seçilmiştir, özellik seçiminde iyi olmamasına rağmen sonuç bölümünde göreceğimiz gibi algoritma oldukça hızlı ve verimlidir.



Şekil 7.12: Karmaşıklık Matrisi RF (Solda) ve RF+GWO(Sağda)

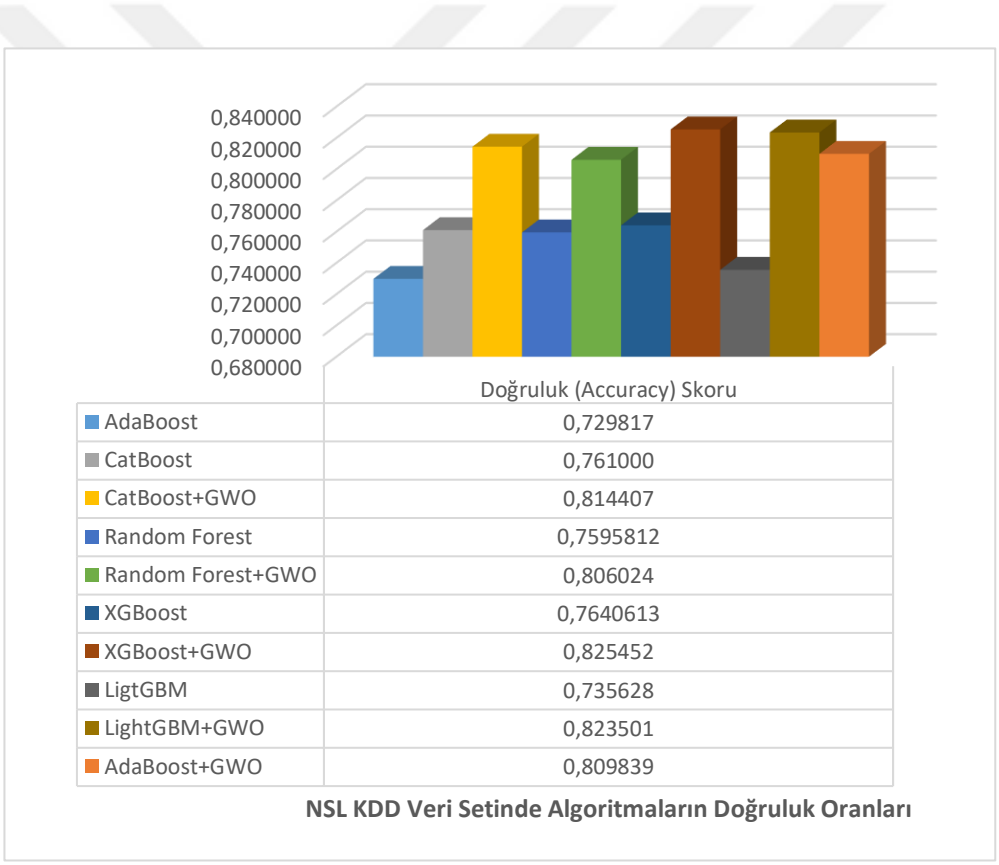
Karmaşıklık matrisi incelendiğinde DoS saldırısında hemen hemen aynı değerleri aldığını ama Probe, U2R ve R2L saldırılarında klasik Rastgele Orman algoritmasına göre daha iyi sonuç verdiğini söyleyebiliriz.

7.4.6. NSL-KDD Veriseti için Sonuç ve Tartışma

NSL-KDD veriseti üzerinde yapılan testlerde LightGBM+GWO algoritması %82,35 ve XGBoost+GWO Algoritması %82,54 ile en yüksek sonuçları vermişlerdir. Daha sonra %81,44 doğruluk oranı ile CatBoost algoritması gelmektedir. Rastgele Orman ve AdaBoost algoritmalarının doğruluk oranı %80 civarındadır. Yukarıdaki tabloyu grafiğe döktüğümüzde sadece Doğruluk Oranını değerlendirecek olursak en kötü performansı %72,98 ile klasik AdaBoost algoritması göstermektedir.

Tablo 7.8: Tüm Algoritmaların NSL-KDD Veriseti Üzerinde Karşılaştırılması

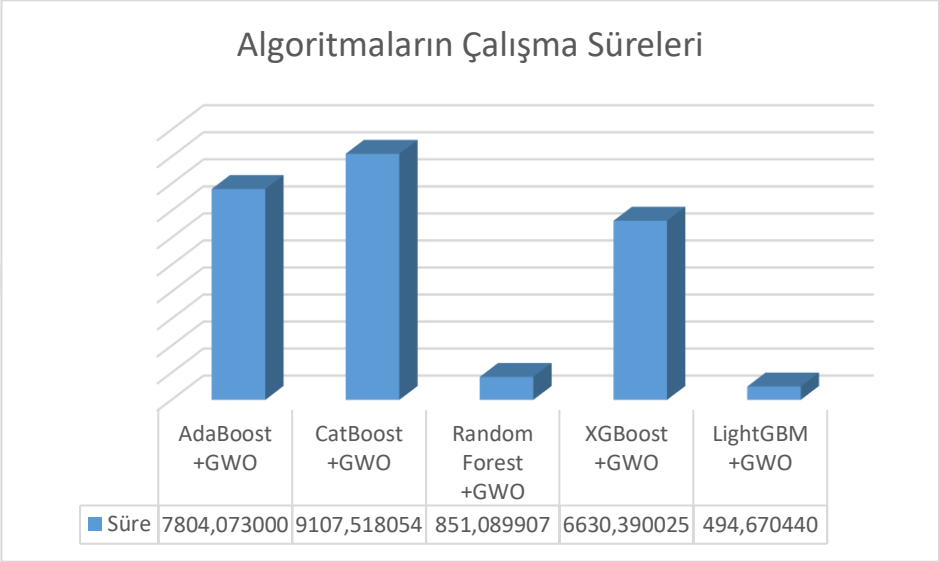
	Doğruluk	Kesinlik	Duyarlılık	F1-Skor
AdaBoost	%72.9817	% 66.6295	% 45.7938	%47.5631
AdaBoost+GWO	%80.9839	%72.8234	%56.1531	%57.8882
CatBoost	%76.1000	%89.2266	%49.1799	%51.9062
CatBoost+GWO	%81.4407	%91.1392	%55.1210	%56.6745
LightGBM	%73.5628	%69.4826	%49.0725	%50.6341
LightGBM+GWO	**%82.3501**	%77.4008	%63.9368	%66.9329
XGBoost	%76.4061	%85.2205	%48.605	%51.7244
XGBoost+GWO	**%82.5452**	%70.8603	%56.5216	%57.1791
Rastgele Orman	%75.9581	%68.5156	%48.3944	%49.2118
Rastgele Orman+GWO	%80.6024	%85.5669	%56.8587	%59.3458



Şekil 7.13: NSLKDD Veriseti üzerinde Doğruluk Oranları Karşılaştırması

Özellik seçimi için kullanılan Gri Kurt Algoritması amaç fonksiyonu üzerinde denemeler yaparak çözüme ulaşmaktadır, bu nedenle de belli bir süre kaybedilmesi beklenilen bir durumdur, büyük veri üzerinde çok fazla özellik incelenerek kaybedilen zamana göre kıyaslanırsa, kaybedilen zaman telafi edilebilir. Algoritmaların çalışma

süreleri incelendiğinde en hızlısı Microsoft tarafından geliştirilen LightGBM, daha sonra ise paralel yapısından dolayı Rastgele Orman (Random Forest), en yavaş çalışanı CatBoost ve AdaBoost algoritmalarıdır. Çözüme ulaşmaktaki yavaşlıkları nedeniyle bir ağ üzerindeki trafiği izleyip anında cevap verebilecek bir STS için yeterli değillerdir.



Şekil 7.14: Algoritmaların Çalışma Süreleri

7.4.7. Önerilen AdaBoost ile BoT-IoT Veriseti Test Sonuçları

Tezin ana konusu özellikle Nesnelerin İnternetine yönelik saldırıların tespit edilmesi olduğundan Nesnelerin İnternetine yönelik saldırıları içeren BoT-IoT veri setinin test sonuçları daha çok önem taşımaktadır. BoT – IoT test verisetinde aşağıdaki özellikler bulunmaktadır. Gri Kurt Optimizasyon Algoritması bu özellikler içinden seçim yapılacaktır. Sonunda enc uzantısı olan özellikler daha önceki bölümlerde anlatıldığı gibi kodlanmış alanlardır. Makine öğrenmesi algoritması algoritmaları sadece sayısal verilerle çalıştığından DoS, DDoS, Reconnaisance, Theft gibi saldırı kategorileri etiket Kodlama (Label Encoding) tekniği ile kodlanmıştır. SAddr alanı yani kaynak IP adresi ve Daddr yani hedef IP adresi de kodlanmıştır.

DoS/DDoS Saldırısı, normal kullanıcıların ağ kaynaklarına erişemediği bir hizmeti kesintiye uğratmak veya tamamen durdurmak için yapılan bir saldırı türüdür. DoS saldırısında tek bir kaynaktan saldırı başlatılırken, DDoS saldırı türünde birden çok kaynak tarafından gerçekleştirilir. Genellikle sunucu bilgisayarda, daha önce saldırgan

tarafından yakalanan ve kontrol edilen, bot adı verilen bir grup makine tarafından toplu olarak düzenlenir. Saldırı metodolojisine göre DDoS/DoS saldırıları hacimsel ve protokol tabanlı olmak üzere iki ana gruba ayrılır(Doshi vd., 2018). Hacimsel saldırı türünde, saldırgan büyük miktarda ağ trafiği oluşturarak kurban makinenin yasal/normal isteklere yanıt vermesini engeller ve sonuç olarak hizmet dışı kalır. Protokol tabanlı saldırılarda ise internet protokollerini hedef alarak CPU ve bellek kaynaklarının tükenmesine neden olarak hedef makinenin isteklere yanıt verememesine neden olur. BoT-IoT veri seti, TCP, UDP ve HTTP protokollerine karşı DDoS ve DoS saldırılarını içerir.

Tablo 7.9: BoT-IoT Verisetinde 15 Özellik

No	Özellik Adı	Açıklama
0	sport	Kaynak Port
1	dport	Hedef Port
2	Stddev	Toplanan kayıtların Standart Sapması
3	N_In_Conn_P_SrcIP	Her bir kaynak IP'den gelen paket sayısı
4	Min	Toplanan kayıtların minimum süresi
5	State_number	Özellik durumunun sayısal gösterimi
6	Mean	Toplanan kayıtların ortalama süresi
7	N_IN_Conn_P_DstIP	Her bir hedef IP den gelen toplam paket sayısı
8	Drate	Hedeften kaynağa paketler
9	Srate	Kaynaktan hedefe saniyedeki paketler
10	Max	Toplanan kayıtların maksimum süresi
11	Saddr_enc	Kaynak IP Adresi kodlanmış hali
12	Daddr_enc	Hedef IP Adresi kodlanmış hali
13	Proto_enc	Protokollerin kodlanmış hali
14	Category_enc	Kategorilerin kodlanmış hali

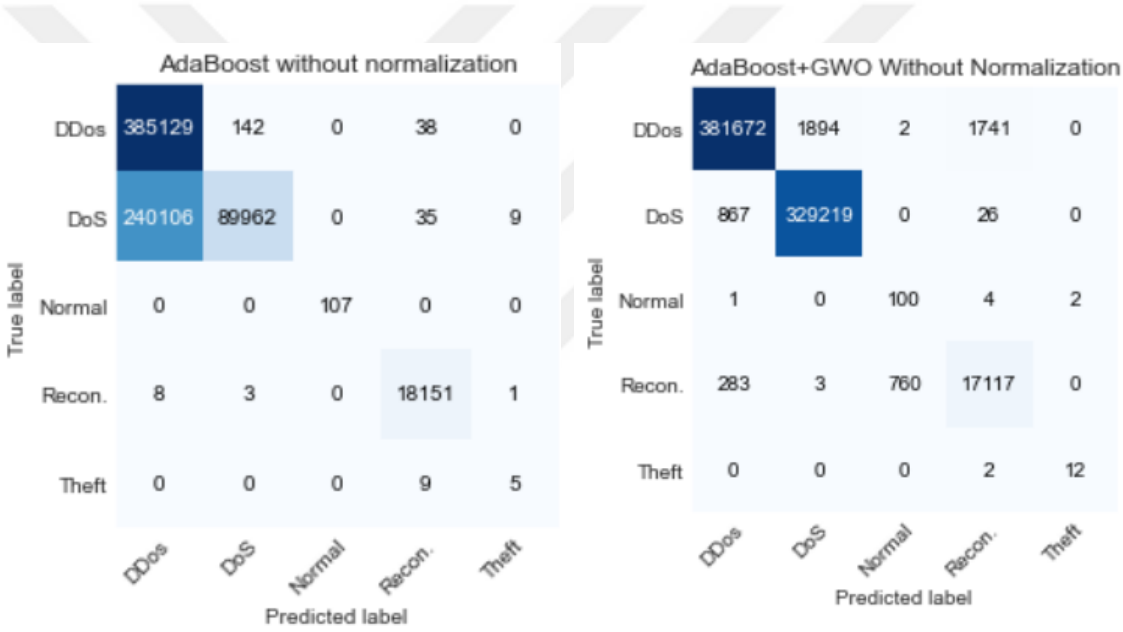
Hırsızlık(Theft) saldırıları(*Reconnaissance attacks, Tools, Types, and Prevention*, y.y.), saldırgan kuruluşların bilgisayar güvenliğini atlayarak hassas bilgileri elde etmelerine yönelik saldırı türüdür. Bu saldırı türü de birçok alt kategoriye ayrılmıştır. Saldırgan, yetkilendirme ve hırsızlık verileri olmadan uzak makineye bağlanabilir.

Başka bir saldırı türü olan keylogging tekniğinde saldırgan, kullanıcının klavye hareketlerini kaydederek tüm bilgilere erişebilir.

AdaBoost Algoritması ile özellik seçimi için AdaBoost+GWO kullanımı ile doğruluk değeri %67,24'ten %99,23'e çıkmıştır.

Tablo 7.10: BoT-IoT Verisetinde AdaBoost ve AdaBoost+GWO Karşılaştırması

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
Adaboost	%67.2415	%78.8639	%72.5707	%70.6525
AdaBoost+GWO	%99.2388	%77.4111	%94.4398	%79.5402



Şekil 7. 15: Adaboost ve AdaBoost+GWO Algoritması Karmaşıklık Matrisi

Yukarıdaki karmaşıklık matrisine baktığımızda DoS saldırılarının tahmin değerinin 89962'den 329219'a çıktığını görmekteyiz. Theft saldırısı ise 5'den 12'ye çıkmıştır. Veri sayısı az olsa bile GWO'nun etkisi ile tahmin artmıştır. Reconnaisance saldırılarında 18151'den 17117 değerine tahmin düşmüştür.

```
Total time(Tamamlanma süresi):  15624.10108089447
Son Seçilen Özellikler
[ 6  9 12 14]
0.9923879488350222
```

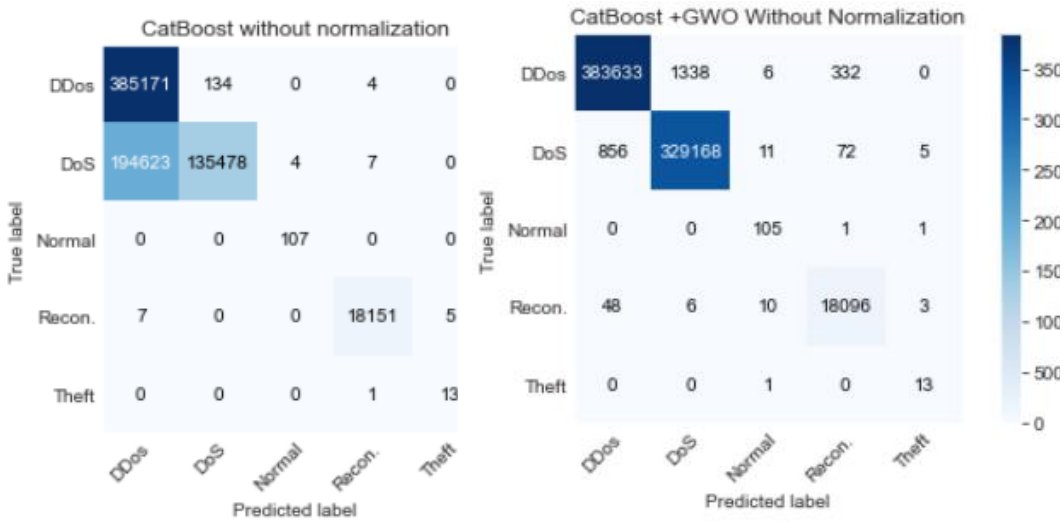
Şekil 7. 16: AdaBoost+GWO Seçimi Sonuçları

7.4.8. Önerilen CatBoost ile BoT-IoT Veriseti Test Sonuçları

CatBoost Algoritması önerilen teknikler beklenilen üstünde başarı göstererek %73.45'den %99.63'e ulaşmıştır. Kesinlik, Duyarlılık ve F1-Skor metriklerinin hepsinde de başarılı bir performans göstermiştir. Çalışma zamanı bakımından algoritmaların performansını son bölümde karşılaştıracağız.

Tablo 7.11: BoT-IoT Verisetinde CatBoost ve CatBoost+GWO Karşılaştırması.

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
CatBoost	%73.4519	%86.9771	%86.7591	%83.4695
CatBoost+GWO	**%99.6334**	%87.0415	%97.9796	%91.5507



Şekil 7.17: Karmaşıklık Matrisi - CatBoost(solda) ve CatBoost+GWO(Sağda)

Karmaşıklık matrisinden görüldüğü DoS ve DDoS saldırılarını tahmin etme yeteneği önerilen yöntemle artmıştır. Recon.ve Theft saldırılarında çok az düşüş vardır.

Algoritmanın %99'un üzerinde doğruluğu yakaladığını karmaşıklık matrisi de göstermektedir.

```
Total time(Tamamlanma süresi):  24950.42485308647
Son Seçilen Özellikler
[ 0  6  7  9 12]
0.9963336763413089
```

Şekil 7.18: CatBoost+GWO Özellik Seçimi Sonuçları

Önerilen CatBoost+GWO yaklaşımının özellik seçimi sonuçları incelendiğinde 0,6,7,9 ve 12 numaralı özelliklerin 15 özellik içinden seçildiği görülmektedir.

7.4.9. Önerilen LightGBM+GWO ile Bot-IoT Veriseti Test Sonuçları

LightGBM algoritması tıpkı NSL-KDD verisetinde olduğu gibi BoT-IoT verisetinde de başarılı bir performans skoru yakalamıştır. Fakat bu veri setinde %99,29 doğruluk oranı ile daha iyi tahmin değerine ulaşmıştır.

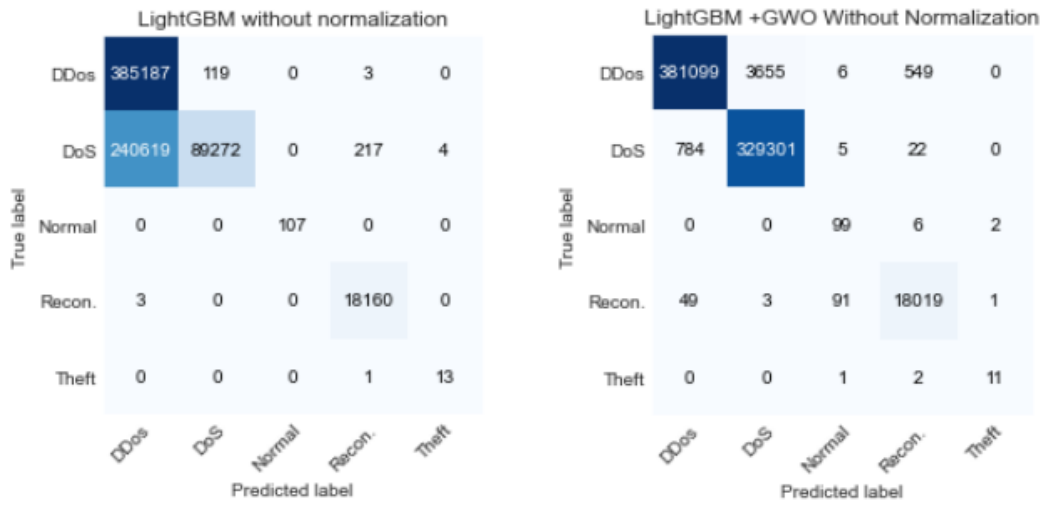
Tablo 7.12: BoT-IoT Veriseti LightGBM ve LightGBM+GWO Karşılaştırması

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
LightGBM	%67.1576	%87.3371	%83.9704	%80.4018
LightGBM+GWO	**%99.2945**	%84.6303	%93.7927	%87.8702

Seçilen özellikler incelendiğinde [0 5 9 10] ifadesinden de anlaşılacağı gibi 0, 5, 9 ve 10. özellikler seçilmiştir, yani 15 özellikten 4 tanesini seçmiştir.

```
['Iteration number 19 the best fitness values -0.9186508542261537']
alpha: [ 0  5  9 10]
Total time(Tamamlanma süresi):  4021.785990715027
```

Şekil 7.19: LigthGBM Algoritması tarafından seçilen Özellikler



Şekil 7.20: Karmaşıklık Matrisi LightGBM ve LightGBM+GWO

Karmaşıklık matrisi incelendiğinde önerilen LightGBM+GWO yaklaşımın özellikle DDoS ve DoS saldırılarını mükemmele yakın şekilde tahmin ettiği görüşmektedir.

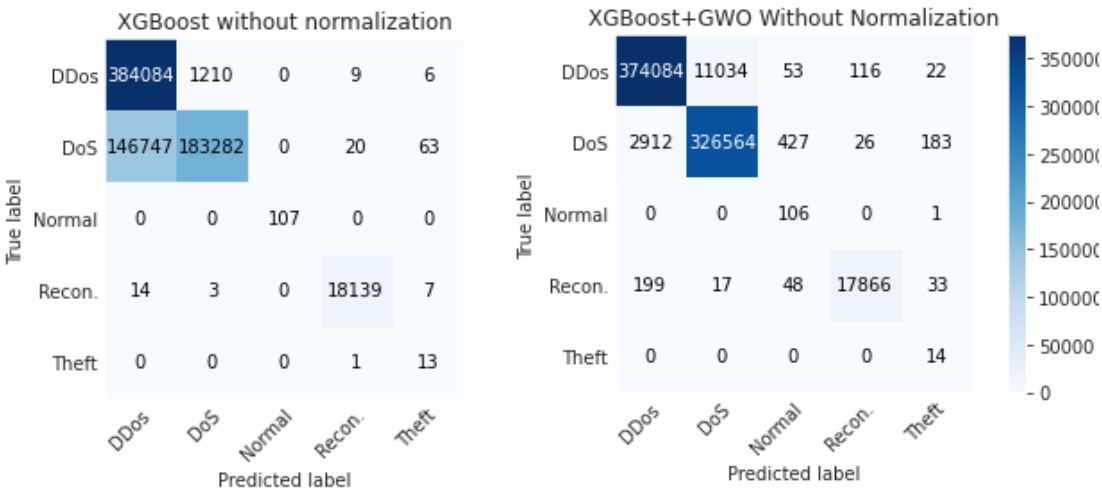
7.4.10. Önerilen XGBoost+GWO ile BoT-IoT Veriseti Test Sonuçları

Doğruluk Oranı XGBoost algoritmasında %79,81'den %97,94'e çıkmıştır. Kesinlik, duyarlılık ve F1-Skor metrik değerlerinden Kesinlik ve Duyarlılık artarken F1-Skor değerinde düşüş olmuştur.

Tablo 7.13: BoT-IoT Veriseti XGBoost ve XGBoost+GWO Karşılaştırması

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
XGBoost	%79.81	%77.22	%89.58	%76.03
XGBoost+GWO	**%97.94**	%63.47	%98.68	%66.76

Karmaşıklık Matrisine baktığımızda DoS Saldırılarını daha iyi tanıdığını diğer saldırılarda biraz düşüş olduğunu görmekteyiz.



Şekil 7.21: XgBoost ve XGBoost+GWO Karmaşıklık Matrisi

Seçilen özelliklere baktığımızda 3,5,7,9,10 ve 12 numaralı özelliklerin algoritma tarafında seçildiğini görmekteyiz. 15 özellikten 5 tanesi seçilmiştir.

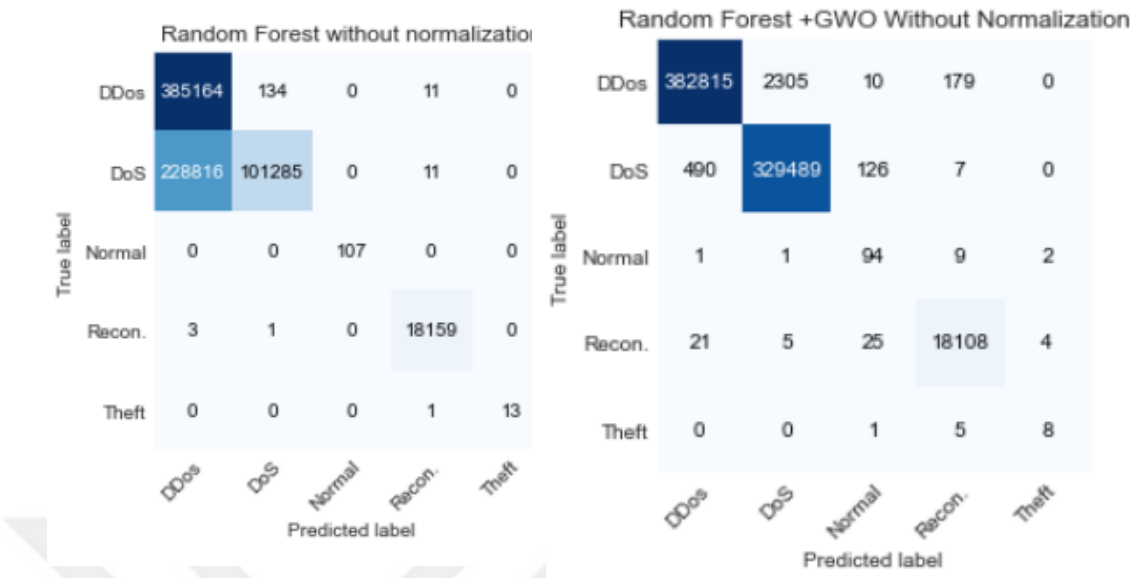
7.4.11. Önerilen Rastgele Orman+GWO ile Bot-IoT Test Sonuçları

Torbalama tekniği kullanan Rastgele orman algoritması önerilen Gri Kurt Optimizasyon algoritması ile başarısını %69,08'den %97,81'e yükseltmiştir. Diğer metrik değerlerde de önerilen yöntemde iyileşme görülmektedir. Sadece F1 değerinde bir farklılık söz konusudur.

Tablo 7.14: BoT-IoT Veriseti Rastgele Orman+GWO Karşılaştırması

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
Rastgele Orman	% 68.79	% 92.49	% 84.69	% 84.05
Rastgele Orman+GWO	% 97.81	% 78.3878	% 88.7709	% 81.48

Karmaşıklık matrisi incelendiğinde DoS saldırından algoritmanın başarısı oldukça yüksektir, DDoS, Reconnaisance saldırısında da yakın değerler almıştır. Theft saldırısında da az veri olmasına rağmen tahmin oranı yüksektir.



Şekil 7.22: Karmaşıklık Matrisi Rastgele Orman ve Rastgele Orman+GWO

Seçilen özellikler incelendiğinde [0 5 9] ifadesinden de anlaşılacağı gibi 0, 5 ve 9. özellikler seçilmiştir, yani 15 özellikten 3 tanesini seçmiştir.

```
Total time(Tamamlanma süresi):  2637.6076335906982
Son Seçilen Özellikler
[0 5 9]
0.9956508405966976
```

Şekil 7.23: Rastgele Orman Algoritması tarafından seçilen Özellikler

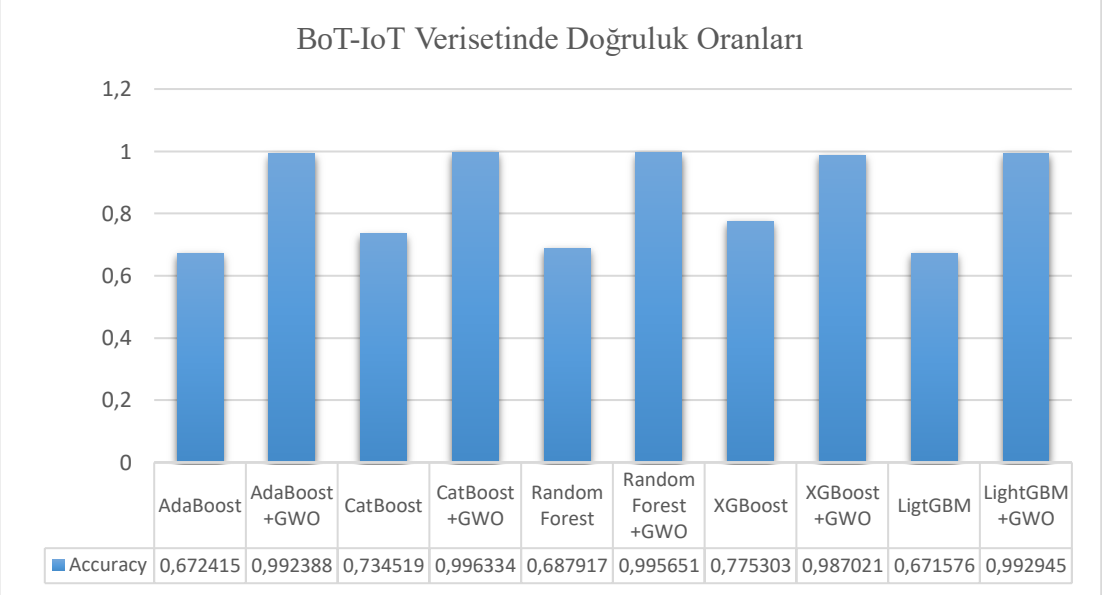
7.4.12. BoT-IoT Veriseti için Sonuç ve Tartışma

Önerilen Gri Kurt Optimizasyon Algoritması ile özellik seçiminin BoT-IoT veriseti üzerinde NSL-KDD verisetine göre daha başarılı sonuçlar verdiğini görüyoruz. NSL-KDD verisetinde en iyi tahmin %82,54 ile XGBoost algoritmasına ait iken, BoT-IoT verisetinde LightGBM algoritması önerilen GWO ile kullanıldığında %99,29, CatBoost ise önerilen yöntemle %99,63'e ulaşmıştır. doğruluk oranına ulaşmıştır. BoT-IoT verisetinde 15 özellik olması, NSL-KDD verisetinde 122 özellik olması NSL-KDD verisetinde arama uzayı bakımından algoritmanın daha fazla çaba sarf

ettiğini göstermektedir, bu nedenle de özellik seçimi işlemi de daha fazla süre almaktadır.

Tablo 7.15: Tüm Algoritmaların Karşılaştırmalı Verileri

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F1-Skor**
Adaboost	%67.2415	%78.8639	%72.5707	%70.6525
AdaBoost+GWO	**%99.2388**	%77.4111	%94.4398	%79.5402
CatBoost	%73.4519	%86.9771	%86.7591	%83.4695
CatBoost+GWO	**%99.6334**	%87.0415	%97.9796	%91.5507
LightGBM	%67.1576	%87.3371	%83.9704	%80.4018
LightGBM+GWO	**%99.2945**	%84.6303	%93.7927	%87.8702
XGBoost	%79.81	%77.22	%89.58	%76.03
XGBoost+GWO	**%97.94**	%63.47	%98.68	%66.76
Rastgele Orman	%69.0815	%92.5428	%84.8268	%84.2357
Rastgele Orman+GWO	**%97.81**	%56.4998	%77.4280	%61.1444



Şekil 7.24: BoT IoT Veri Seti Doğruluk Oranları

Leevy ve diğerleri (Leevy vd., 2021) BoT-IoT veriseti üzerinde sadece Theft saldırılarını tahmin ederken bizim çalışmamızda beş kategoriden saldırı tahmin edilmiştir. Leevy ve diğerleri CatBoost algoritmasıyla %98,34 doğrulukta, önerdiğimiz CatBoost+GWO yaklaşımımızda %99,63 oranında, yine LightGBM ile %98,78 tahmin ederken, önerdiğimiz metot %99,29 tahmin etmiştir. Aynı çalışmada

Karar Ağaçları %96,16, Lojistik Regresyon %97,37, Naive Bayes %95,55, MLP %97,38, Random Forest %97,18, XGBoost %97,75 doğrulukta tahmin edilmiştir.

Tablo 7.16: Literatürde Yapılan Çalışmaların Karşılaştırılması

Çalışma	**Algoritma**	**Doğruluk**
(Leevy vd., 2021)	CatBoost	%98.34
(Leevy vd., 2021)	Karar Ağaçları	%96.16
(Leevy vd., 2021)	LightGBM	%98.78
(Leevy vd., 2021)	Lojistik Regresyon	%97.37
(Leevy vd., 2021)	Rastgele Orman	%95.59
(Leevy vd., 2021)	Naive Bayes	%97.18
(Leevy vd., 2021)	MLP	%97.18
(Leevy vd., 2021)	XGBoost	%97.75
(Saba vd., 2022)	CNN	%95.55
(Swarna Sugi & Ratna, 2020)	LSTM	%97.28
(Swarna Sugi & Ratna, 2020)	KNN	%92.29
Önerilen Çalışma-1	**CatBoost+GWO**	**%99.63**
Önerilen Çalışma-2	**LightGBM+GWO**	**%99.29**
Önerilen Çalışma-3	**Adaboost+GWO**	**%99.23**

Saba ve arkadaşları Bot-IoT veri seti üzerinde CNN(Evrişimsel Sinir Ağları) ile yaptıkları çalışmada düşük saldırı sayısına sahip Theft saldırısını verisetinden kaldırdıkları halde %95.55 başarı elde etmişlerdir (Saba vd., 2022). Bu çalışmada küçük sayıda olsa da tüm saldırıları önemsediğimiz için Theft saldırısını kaldırmadık, bu saldırı kalkarsa daha iyi sonuçlar alınır, Karmaşıklık Matrislerinden görüleceği üzere algoritmaların az sayıda veri bulunan kategorilerde diğerlerine göre daha düşük performans gösterdiği söylenebilir.

Sonuç olarak önerilen Gri Kurt Optimizasyon algoritması ile özellik seçimi %99'un üzerine taşıdığı başarı oranıyla akıllı saldırı tespiti için kullanılabileceğini göstermiştir.

SONUÇ VE TARTIŞMA

İnternetin güvenliğine yapılan tehditler, Nesnelerin İnterneti cihazlarını yoğun şekilde tehdit ettiğinden dolayı bu alanda sibergüvenlik çalışmaları artmıştır. İlk bölüme Nesnelerin İnternetine yapılan saldırıları katmanlara göre incelendi ve her tehdit için farklı savunma teknikleri sunuldu. Sonraki bölümlerde Akıllı Saldırı Tespit Sistemi Gri Kurt Optimizasyon Algoritmasının özellik seçiminde kullanıldığı Topluluk Öğrenme Algoritmaları önerildi. İlk önce ağ üzerinden tehditlerin yer aldığı NSL-KDD veriseti üzerinde daha sonra tezin ana konusu olan Nesneleri İnterneti için açık kaynak olarak paylaşılan BoT-IoT veriseti üzerinde testler yapıldı. Her iki veriseti üzerinde de önerilen yöntemin başarısı Doğruluk Oranı, Kesinlik, Duyarlılık ve F1-Skor gibi metriklerle karşılaştırılarak sunuldu. Özellikle BoT-IoT veriseti üzerinde AdaBoost (%99,23), LightGBM (%99,29), CatBoost (%99,63) algoritmalarının %99'un üzerinde başarı göstermesi ve Rastgele Orman algoritmasının %97,81 başarı göstermesi önerilen tekniğin akıllı savunma sisteminde kullanılabileceğini gösterdi.

Literatürde yer alan birçok çalışmada özellikle algoritmaların ne kadar sürede çalıştığı hakkında bir istatistik bulunmamaktadır. Bu çalışmada algoritmaların süre bakımından çalışmaları ölçülerek, performansları yanında çalışma süreleri de incelendi.

Gelecek çalışmalarda yeni çıkacak veri setleri üzerinde de bu algoritmaların başarımı değerlendirilecek farklı amaç fonksiyonlarında algoritmaların hem süre hem de metrik değerler bakımından başarısı incelenecektir.

KAYNAKLAR

Abdelzaher, T. F., Prabh, S., & Kiran, R. (2004). On real-time capacity limits of multihop wireless sensor networks. *Proceedings - Real-Time Systems Symposium*: 359–370. https://doi.org/10.1109/REAL.2004.37

Abdullah, I., Muntasir Rahman, M., & Chandra Roy, M. (2015). Detecting Sinkhole Attacks in Wireless Sensor Network using Hop Count. *International Journal of Computer Network and Information Security*, *7*(3): 50–56. https://doi.org/10.5815/IJCNIS.2015.03.07

Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S. (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. Içinde *Big Data* (C. 7, Sayı 4, ss. 221–248). Mary Ann Liebert Inc. https://doi.org/10.1089/big.2018.0175

Aburomman, A. A., & Ibne Reaz, M. Bin. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing Journal*, *38*, 360–372. https://doi.org/10.1016/j.asoc.2015.10.011

Acıiçmez, O., Seifert, J.-P., Etin, C. ¸, & Koç, K. (y.y.). *Predicting Secret Keys via Branch Prediction*.

Aditya Sai Srinivas, T., & Manivannan, S. S. (2020). Prevention of Hello Flood Attack in IoT using combination of Deep Learning with Improved Rider Optimization Algorithm. *Computer Communications*, *163*: 162–175. https://doi.org/10.1016/J.COMCOM.2020.03.031

Al-Qatf, M., Lasheng, Y., Al-Habib, M., & Al-Sabahi, K. (2018). Deep Learning Approach Combining Sparse Autoencoder with SVM for Network Intrusion Detection. *IEEE Access*, *6*: 52843–52856. https://doi.org/10.1109/ACCESS.2018.2869577

Al-Tashi, Q., Rais, H., & Jadid, S. (2019). Feature selection method based on grey wolf optimization for coronary artery disease classification. *Advances in Intelligent Systems and Computing*, *843*: 257–266. https://doi.org/10.1007/978-3-319-99007-1_25/COVER/

Alerts and WIDS - Kismet. (y.y.). Tarihinde 29 Haziran 2020, adresinden erişildi https://www.kismetwireless.net/docs/readme/alerts_and_wids/

Ali, S., Khan, M. A., Ahmad, J., Malik, A. W., & Ur Rehman, A. (2018). Detection and prevention of Black Hole Attacks in IOT & WSN. *2018 3rd International Conference on Fog and Mobile Edge Computing, FMEC 2018*: 217–226. https://doi.org/10.1109/FMEC.2018.8364068

Aljawarneh, S., Yassein, M. B., & Aljundi, M. (2019). An enhanced J48 classification algorithm for the anomaly intrusion detection systems. *Cluster Computing*, *22*(5): 10549–10565. https://doi.org/10.1007/s10586-017-1109-8

Amiri, F., Rezaei Yousefi, M., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual information-based feature selection for intrusion detection systems. *Journal of Network and Computer Applications*, *34*(4): 1184–1199. https://doi.org/10.1016/j.jnca.2011.01.002

Argus - System and Network Monitoring Software. (y.y.). Tarihinde 12 Haziran 2022, adresinden erişildi http://argus.tcp4me.com/

Arshad, A., Hanapi, Z. M., Subramaniam, S., & Latip, R. (2021). A survey of Sybil attack countermeasures in IoT-based wireless sensor networks. *PeerJ Computer Science*, *7*, 1–33. https://doi.org/10.7717/PEERJ-CS.673

Babar, S., Stango, A., Prasad, N., Sen, J., & Prasad, R. (2011). Proposed embedded security framework for Internet of Things (IoT). *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE 2011*. https://doi.org/10.1109/WIRELESSVITAE.2011.5940923

Bhattasali, T., Chaki, R., & Sanyal, S. (2012). Sleep Deprivation Attack Detection in Wireless Sensor Network. *International Journal of Computer Applications*, *40*(15): 975–8887.

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems and tools. *IEEE Communications Surveys and Tutorials*, *16*(1), 303–336. https://doi.org/10.1109/SURV.2013.052213.00046

Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys and Tutorials*, *18*(2): 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Budak, H. (2108). *Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım*. Süleyman Demirel Üniversitesi Dergisi. https://doi.org/10.19113/sdufbed.01653

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Communications Surveys and Tutorials*, *21*(3): 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Chawla, A., Lee, B., Fallon, S., & Jacob, P. (2019). Host Based Intrusion Detection System with Combined CNN/RNN Model. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11329 LNAI*: 149–158. https://doi.org/10.1007/978-3-030-13453-2_12

Chen, T., & Guestrin, C. (y.y.). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672

Chen, X., Makki, K., Yen, K., & Pissinou, N. (2009). Sensor network security: A survey. *IEEE Communications Surveys and Tutorials*, *11*(2): 52–73. https://doi.org/10.1109/SURV.2009.090205

Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper - Cisco. (y.y.). Tarihinde 20 Şubat 2021, adresinden erişildi https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

Daidone, R., Dini, G., & Tiloca, M. (2013). A solution to the GTS-based selective jamming attack on IEEE 802.15.4 networks. *Wireless Networks*, *20*(5): 1223–1235. https://doi.org/10.1007/S11276-013-0673-Y

Dasari, M. (2017). Real time detection of MAC layer DoS attacks in IEEE 802.11 wireless networks. *2017 14th IEEE Annual Consumer Communications and Networking Conference, CCNC 2017*: 939–944. https://doi.org/10.1109/CCNC.2017.7983259

Deng, J., Han, R., & Mishra, S. (2006). INSENS: Intrusion-tolerant routing for wireless sensor networks. *Computer Communications*, *29*(2): 216–230. https://doi.org/10.1016/J.COMCOM.2005.05.018

Deshmukh-Bhosale, S., & Sonavane, S. S. (2019). A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things. *Procedia Manufacturing*, *32*: 840–847. https://doi.org/10.1016/J.PROMFG.2019.02.292

Diro, A. A., & Chilamkurti, N. (2018). Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Generation Computer Systems*, *82*: 761–768. https://doi.org/10.1016/j.future.2017.08.043

Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2019). A survey on ensemble learning. *Frontiers of Computer Science*, *14*(2): 241–258. https://doi.org/10.1007/S11704-019-8208-Z

Dorogush, A. V., Ershov, V., & Yandex, A. G. (y.y.). *CatBoost: gradient boosting with categorical features support*. Tarihinde 23 Haziran 2022, adresinden erişildi https://github.com/Microsoft/LightGBM

Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning DDoS detection for consumer internet of things devices. *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*: 29–35. https://doi.org/10.1109/SPW.2018.00013

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *50*: 102419. https://doi.org/10.1016/j.jisa.2019.102419

Fu, Y., Yan, Z., Cao, J., Koné, O., & Cao, X. (2017). *An Automata Based Intrusion Detection Method for Internet of Things*. https://doi.org/10.1155/2017/1750637

Gaddam, R. T., & Nandhini, M. (2017). An analysis of various snort based techniques to detect and prevent intrusions in networks: Proposal with code refactoring snort tool in Kali Linux environment. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2017*, 10–15. https://doi.org/10.1109/ICICCT.2017.7975177

Gao, N., Gao, L., Gao, Q., & Wang, H. (2015). An Intrusion Detection Model Based on Deep Belief Networks. *Proceedings - 2014 2nd International Conference on Advanced Cloud and Big Data, CBD 2014*: 247–252. https://doi.org/10.1109/CBD.2014.41

Granjal, J., Monteiro, E., & Sa Silva, J. (2015). Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys and Tutorials*, *17*(3): 1294–1312. https://doi.org/10.1109/COMST.2015.2388550

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. Içinde *IEEE Access* (C. 7, ss. 82721–82743). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2019.2924045

Hu, W., & Hu, W. (2003). Robust Support Vector Machines for Anomaly Detection. *IN PROC. 2003 INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND APPLICATIONS (ICMLA'03*, 23--24. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.4085

Ioannou, C., & Vassiliou, V. (2020). Accurate Detection of Sinkhole Attacks in IoT Networks Using Local Agents. *2020 Mediterranean Communication and Computer Networking Conference, MedComNet 2020*. https://doi.org/10.1109/MEDCOMNET49392.2020.9191503

Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. *IEEE Access*, *8*: 32150–32162. https://doi.org/10.1109/ACCESS.2020.2973219

KDD Cup 1999 Data. (y.y.). Tarihinde 17 Aralık 2020, adresinden erişildi http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019a). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1): 1–22. https://doi.org/10.1186/s42400-019-0038-7

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019b). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1): 1–22. https://doi.org/10.1186/S42400-019-0038-7/FIGURES/8

Kibirige, G. W., & Sanga, C. (y.y.). *A Survey on Detection of Sinkhole Attack in Wireless Sensor Network*.

Kim, J., & Song, J. (2017). A simple and efficient replay attack prevention scheme for LoRaWAN. *ACM International Conference Proceeding Series*: 32–36. https://doi.org/10.1145/3163058.3163064

Ko, L. C., Chen, H. Y., & Lin, G. R. (2009). A neighbor-based detection scheme for wireless sensor networks against node replication attacks. *2009 International Conference on Ultra Modern Telecommunications and Workshops*. https://doi.org/10.1109/ICUMT.2009.5345637

Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems*, *100*: 779–796. https://doi.org/10.1016/J.FUTURE.2019.05.041

Kumar Mallapragada, P., Jin, R., Jain, A. K., & Liu, Y. (y.y.). *SemiBoost: Boosting for Semi-supervised Learning*.

Lakhanpal, R., & Sharma, S. (2016). Detection & Prevention of Sybil attack in Ad hoc network using hybrid MAP & MAC technique. *2016 International Conference on Computation of Power, Energy, Information and Communication, ICCPEIC 2016*: 283–287. https://doi.org/10.1109/ICCPEIC.2016.7557211

Law, Y. W., Palaniswami, M., Hoesel, L. Van, Doumen, J., Hartel, P., & Havinga, P. (2009). Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. *ACM Transactions on Sensor Networks (TOSN)*, *5*(1). https://doi.org/10.1145/1464420.1464426

Leevy, J. L., Hancock, J., Khoshgoftaar, T. M., & Peterson, J. (2021). Detecting Information Theft Attacks in the Bot-IoT Dataset. *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021*: 807–812. https://doi.org/10.1109/ICMLA52953.2021.00133

Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., & Dai, K. (2012). An efficient intrusion detection system based on support vector machines and gradually feature removal method. *Expert Systems with Applications*, *39*(1): 424–430. https://doi.org/10.1016/j.eswa.2011.07.032

Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, *34*(4): 579–595. https://doi.org/10.1016/S1389-1286(00)00139-0

Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K., & Zissman, M. A. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *Proceedings - DARPA Information Survivability Conference and Exposition, DISCEX 2000*, *2*: 12–26. https://doi.org/10.1109/DISCEX.2000.821506

Lounis, K., & Zulkernine, M. (2020). Attacks and Defenses in Short-Range Wireless Technologies for IoT. *IEEE Access*, *8*: 88892–88932. https://doi.org/10.1109/ACCESS.2020.2993553

Markey, J. (2020). *SANS Institute Information Security Reading Room Using Decision Tree Analysis for Intrusion Detection: A How-To Guide Using Decision Tree Analysis for Intrusion Detection: A How-To Guide Using Decision Tree Analysis for Intrusion Detection: A How-To Guide! 1 !*

Mathew, A., & Terence, J. S. (2018). A survey on various detection techniques of sinkhole attacks in WSN. *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, *2018-January*, 1115–1119. https://doi.org/10.1109/ICCSP.2017.8286550

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). *Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection*. https://doi.org/10.14722/ndss.2018.23204

Modekurthy, V. P., Ismail, D., Rahman, M., & Saifullah, A. (2018). A Utilization-Based Approach for Schedulability Analysis in Wireless Control Systems. *Proceedings - 2018 IEEE International Conference on Industrial Internet, ICII 2018*, 49–58. https://doi.org/10.1109/ICII.2018.00014

More, S., Matthews, M., Joshi, A., & Finin, T. (2012). A knowledge-based approach to intrusion detection modeling. *Proceedings - IEEE CS Security and Privacy Workshops, SPW 2012*: 75–81. https://doi.org/10.1109/SPW.2012.26

Moustafa, N., & Slay, J. (2015, Aralık 7). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*. https://doi.org/10.1109/MilCIS.2015.7348942

Na, S., Hwang, D., Shin, W., & Kim, K. H. (2017). Scenario and countermeasure for replay attack using join request messages in LoRaWAN. *International Conference on Information Networking*: 718–720. https://doi.org/10.1109/ICOIN.2017.7899580

Nam, K., & Kim, K. (2018). A Study on SDN security enhancement using open source IDS/IPS Suricata. *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*: 1124–1126. https://doi.org/10.1109/ICTC.2018.8539455

Nguyen, V. L., Lin, P. C., & Hwang, R. H. (2019). Energy depletion attacks in low power wireless networks. *IEEE Access*, *7*: 51915–51932. https://doi.org/10.1109/ACCESS.2019.2911424

NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (y.y.). Tarihinde 17 Aralık 2020, adresinden erişildi https://www.unb.ca/cic/datasets/nsl.html

Open SSL. (y.y.). Tarihinde 20 Haziran 2022, adresinden erişildi https://www.openssl.org/docs/

Otoum, S., Kantarci, B., & Mouftah, H. (2018). Adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures. *IEEE International Conference on Communications*, *2018-May*. https://doi.org/10.1109/ICC.2018.8422401

Otoum, S., Kantarci, B., & Mouftah, H. T. (2019). On the Feasibility of Deep Learning in Sensor Network Intrusion Detection. *IEEE Networking Letters*, *1*(2): 68–71. https://doi.org/10.1109/lnet.2019.2901792

Pacalet, R. (2022). *Hardware Security Probing attacks*.

Pajouh, H. H., Javidan, R., Khayami, R., Dehghantanha, A., & Choo, K. K. R. (2019). A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Transactions on Emerging Topics in Computing*, *7*(2): 314–323. https://doi.org/10.1109/TETC.2016.2633228

Pawar, P. M., Nielsen, R. H., Prasad, N. R., Ohmori, S., & Prasad, R. (2012). Behavioural modelling of WSN MAC layer security attacks: A sequential UML approach. *Journal of Cyber Security and Mobility*, *1*(1): 65–82. https://doi.org/10.13052/JCSM2245-1439.115

Payandeh, A., & Mozayyani, N. (y.y.). Detection Collision Attacks In Wireless Sensor Network Usingrule-Based Packet Flow Rate. *SaeedSedighianKashi/ International Journal of Engineering Research and Applications (IJERA)*, *3*, 261–268. Tarihinde 20 Haziran 2022, adresinden erişildi www.ijera.com

Publication - AT2A: Defending Unauthenticated Broadcast Attacks in Mobile Wireless Sensor Networks. (y.y.). Tarihinde 20 Haziran 2022, adresinden erişildi https://www.ijecce.com/index.php/issues?view=publication&task=show&id=926

Qu, Z., Liu, H., Wang, Z., Xu, J., Zhang, P., & Zeng, H. (2021). A combined genetic optimization with AdaBoost ensemble model for anomaly detection in buildings electricity consumption. *Energy and Buildings*, *248*: 111193. https://doi.org/10.1016/J.ENBUILD.2021.111193

Radosavac, S., Cárdenas, A. A., Baras, J. S., & Moustakides, G. V. (2007). Detecting IEEE 802.11 MAC layer misbehavior in ad hoc networks: Robust strategies against individual and colluding attackers. *Journal of Computer Security*, *15*(1): 103–128. https://doi.org/10.3233/JCS-2007-15105

Rani, P., Kavita, Verma, S., & Nguyen, G. N. (2020). Mitigation of Black Hole and Gray Hole Attack Using Swarm Inspired Algorithm with Artificial Neural Network. *IEEE Access*, *8*: 121755–121764. https://doi.org/10.1109/ACCESS.2020.3004692

Ray, P., Reddy, S. S., & Banerjee, T. (2021). Various dimension reduction techniques for high dimensional data analysis: a review. *Artificial Intelligence Review*, *54*(5): 3473–3515. https://doi.org/10.1007/S10462-020-09928-0/FIGURES/27

Reconnaissance attacks, Tools, Types, and Prevention. (y.y.). Tarihinde 08 Temmuz 2022, adresinden erişildi https://www.computernetworkingnotes.com/ccna-study-guide/reconnaissance-attacks-tools-types-and-prevention.html

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

Safaldin, M., Otair, M., & Abualigah, L. (2021). Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*, *12*(2): 1559–1576. https://doi.org/10.1007/S12652-020-02228-Z/FIGURES/18

Sagan User Guide — Sagan User Guide 1.2.2 documentation. (y.y.). Tarihinde 29 Haziran 2020, adresinden erişildi https://sagan.readthedocs.io/en/latest/

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4). https://doi.org/10.1002/widm.1249

Sahu, S., & Mehtre, B. M. (2015). Network intrusion detection system using J48 Decision Tree. *2015 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2015*: 2023–2026. https://doi.org/10.1109/ICACCI.2015.7275914

Sajjad, S. M., & Yousaf, M. (2014). Security analysis of IEEE 802.15.4 MAC in the context of Internet of Things (IoT). *Conference Proceedings - 2014 Conference on Information Assurance and Cyber Security, CIACS 2014*: 9–14. https://doi.org/10.1109/CIACS.2014.6861324

Saleh, A. I., Talaat, F. M., & Labib, L. M. (2017). A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers. *Artificial Intelligence Review*, *51*(3): 403–443. https://doi.org/10.1007/S10462-017-9567-1

Sastry, N., & Wagner, D. (2004). *Security Considerations for IEEE 802.15.4 Networks*.

Schapire, R. E. (2013). Explaining AdaBoost. *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik*: 37–52. https://doi.org/10.1007/978-3-642-41136-6_5

Security Onion. (y.y.). Tarihinde 29 Haziran 2020, adresinden erişildi https://securityonion.net/

Seo, J., & Lee, G. (2012). An Effective Wormhole Attack Defence Method for a Smart Meter Mesh Network in an Intelligent Power Grid: *https://doi.org/10.5772/45995*, *9*. https://doi.org/10.5772/45995

Serpen, G., & Aghaei, E. (2018). Host-based misuse intrusion detection using PCA feature extraction and kNN classification algorithms. *Intelligent Data Analysis*, *22*(5): 1101–1114. https://doi.org/10.3233/IDA-173493

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(1): 41–50. https://doi.org/10.1109/TETCI.2017.2772792

Sikder, A. K., Petracca, G., Aksu, H., Jaeger, T., & Uluagac, A. S. (2018). *A Survey on Sensor-based Threats to Internet-of-Things (IoT) Devices and Applications*. https://doi.org/10.48550/arxiv.1802.02041

Snort. (2016). *Snort - Network Intrusion Detection & Prevention System*. Snort.Org. https://www.snort.org/

Sokullu, R., Dagdeviren, O., & Korkmaz, I. (2008). On the IEEE 802.15.4 MAC layer attacks: GTS attack. *Proceedings - 2nd Int. Conf. Sensor Technol. Appl., SENSORCOMM 2008, Includes: MESH 2008 Conf. Mesh Networks; ENOPT 2008 Energy Optim. Wireless Sensors Networks, UNWAT 2008 Under Water Sensors Systems*: 673–678. https://doi.org/10.1109/SENSORCOMM.2008.75

Solorio-Fernández, S., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. F. (2020). A review of unsupervised feature selection methods. *Artificial Intelligence Review*, *53*(2): 907–948. https://doi.org/10.1007/S10462-019-09682-Y/TABLES/8

Stallings, W. (y.y.). *Network security essentials : applications and standards* (Sixth).

stockvisual. (2020). *White paper Cisco public IT and OT Cybersecurity: United We Stand, Divided We Fall*.

Swarna Sugi, S. S., & Ratna, S. R. (2020). Investigation of machine learning techniques in intrusion detection system for IoT network. *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 1164–1167. https://doi.org/10.1109/ICISS49785.2020.9315900

Symantec Security Summary - June 2020 | Symantec Blogs. (y.y.). Tarihinde 24 Aralık 2020, adresinden erişildi https://symantec-enterprise-blogs.security.com/blogs/feature-stories/symantec-security-summary-june-2020

Taş, O. (2016). *Destek Vektör Makineleri - Support Vector Machine*. Presentation. https://www.slideshare.net/oguzhantas/destek-vektr-makineleri-support-vector-machine

TAŞ, O., & KİANİ, F. (2018). Blok Zinciri Teknolojisine Yapılan Saldırılar Üzerine bir İnceleme. *Bilişim Teknolojileri Dergisi*, *11*(4): 369–382. https://doi.org/10.17671/gazibtd.451695

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, Aralık 16). A detailed analysis of the KDD CUP 99 data set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009*. https://doi.org/10.1109/CISDA.2009.5356528

The Bot-IoT Dataset | UNSW Research. (y.y.). Tarihinde 12 Haziran 2022, adresinden erişildi https://research.unsw.edu.au/projects/bot-iot-dataset

The UNSW-NB15 data set description. (y.y.). Tarihinde 17 Aralık 2020, adresinden erişildi https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/

The Zeek Network Security Monitor. (y.y.). Tarihinde 12 Haziran 2022, adresinden erişildi https://zeek.org/

Thongkanchorn, K., Ngamsuriyaroj, S., & Visoottiviseth, V. (2013). Evaluation studies of three intrusion detection systems under various attacks and rule sets. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*. https://doi.org/10.1109/TENCON.2013.6718975

Tomić, I., & McCann, J. A. (2017). A Survey of Potential Security Issues in Existing Wireless Sensor Network Protocols. *IEEE Internet of Things Journal*, *4*(6), 1910–1923. https://doi.org/10.1109/JIOT.2017.2749883

Tyralis, H., & Papacharalampous, G. (2021). Boosting algorithms in energy research: a systematic review. *Neural Computing and Applications 2021 33:21*, *33*(21), 14101–14117. https://doi.org/10.1007/S00521-021-05995-8

Understanding the Mirai Botnet | USENIX. (y.y.). Tarihinde 19 Haziran 2022, adresinden erişildi https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

Verma, A., & Ranga, V. (2020). Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wireless Personal Communications*, *111*(4): 2287–2310. https://doi.org/10.1007/s11277-019-06986-8

Vidgren, N., Haataja, K., Patiño-Andres, J. L., Ramírez-Sanchis, J. J., & Toivanen, P. (2013). Security threats in ZigBee-enabled systems: Vulnerability evaluation, practical experiments, countermeasures, and lessons learned. *Proceedings of the Annual Hawaii International Conference on System Sciences*: 5132–5138. https://doi.org/10.1109/HICSS.2013.475

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, *7*: 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

Wai Kyi Kyi Oo, H. K. (2019). *A Framework of Moving Target Defenses for the Internet of Things*. Bulletin of Networking, Computing, Systems, and Software. http://w.bncss.org/index.php/bncss/article/view/115

Wang, L., Guo, Y., Fan, M., & Li, X. (2022). Wind speed prediction using measurements from neighboring locations and combining the extreme learning machine and the AdaBoost algorithm. *Energy Reports*, *8*: 1508–1518. https://doi.org/10.1016/J.EGYR.2021.12.062

Wireshark · Go Deep. (y.y.). Tarihinde 12 Haziran 2022, adresinden erişildi https://www.wireshark.org/

Wu, K., Chen, Z., & Li, W. (2018). A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. *IEEE Access*, *6*: 50850–50859. https://doi.org/10.1109/ACCESS.2018.2868993

Xu, W., Ma, K., Trappe, W., & Zhang, Y. (2006). Jamming sensor networks: Attack and defense strategies. *IEEE Network*, *20*(3): 41–47. https://doi.org/10.1109/MNET.2006.1637931

Yaqoob, I., Ahmed, E., Hashem, I. A. T., Ahmed, A. I. A., Gani, A., Imran, M., & Guizani, M. (2017). Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges. *IEEE Wireless Communications*, *24*(3): 10–16. https://doi.org/10.1109/MWC.2017.1600421

Ye, N., Emran, S. M., Chen, Q., & Vilbert, S. (2002). Multivariate statistical analysis of audit trails for host-based intrusion detection. *IEEE Transactions on Computers*, *51*(7): 810–820. https://doi.org/10.1109/TC.2002.1017701

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, *5*: 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things. Içinde *Journal of Network and Computer Applications* (C. 84, ss. 25–37). Academic Press. https://doi.org/10.1016/j.jnca.2017.02.009

Zhi-Hua Zhou. (2012). *Ensemble learning: foundations and algorithms*.

Zhou, J., Huang, S., Wang, M., & Qiu, Y. (2021). Performance evaluation of hybrid GA–SVM and GWO–SVM models to predict earthquake-induced liquefaction potential of soil: a multi-dataset investigation. *Engineering with Computers*: 1–19. https://doi.org/10.1007/S00366-021-01418-3

Zhou, Y., Mazzuchi, T. A., & Sarkani, S. (2020). M-AdaBoost-A based ensemble system for network intrusion detection. *Expert Systems With Applications*, *162*: 113864. https://doi.org/10.1016/j.eswa.2020.113864

EKLER

Tablo Ek. 1: NSL-KDD Veriseti 0-29 Arası Özellik Tablosu

No	Özellik	No	Özellik	No	Özellik
0	duration	10	root_shell	20	srv_count
1	src_bytes	11	su_attempted	21	serror_rate
2	dst_bytes	12	num_root	22	srv_serror_rate
3	land	13	num_file_creations	23	rerror_rate
4	wrong_fragment	14	num_shells	24	srv_rerror_rate
5	urgent	15	num_access_files	25	same_srv_rate
6	hot	16	num_outbound_cmds	26	diff_srv_rate
7	num_failed_logins	17	is_host_login	27	srv_diff_host_rate
8	logged_in	18	is_guest_login	28	dst_host_count
9	num_compromised	19	count	29	dst_host_srv_count

Tablo Ek. 2: NSL-KDD Veriseti 30-59 Arası Özellik Tablosu

No	Özellik	No	Özellik	No	Özellik
30	dst_host_same_srv_rate	40	protocol_type_udp	50	service_daytime
31	dst_host_diff_srv_rate	41	service_IRC	51	service_discard
32	dst_host_same_src_port_rate	42	service_X11	52	service_domain
33	dst_host_srv_diff_host_rate	43	service_Z39_50	53	service_domain_u
34	dst_host_serror_rate	44	service_aol	54	service_echo
35	dst_host_srv_serror_rate	45	service_auth	55	service_eco_i
36	dst_host_rerror_rate	46	service_bgp	56	service_ecr_i
37	dst_host_srv_rerror_rate	47	service_courier	57	service_efs
38	protocol_type_icmp	48	service_csnet_ns	58	service_exec
39	protocol_type_tcp	49	service_ctf	59	service_finger

Tablo Ek. 3: NSL-KDD Veriseti 60-89 Arası Özellik Tablosu

No	Özellik	No	Özellik	No	Özellik
60	service_ftp	70	service_iso_tsap	80	service_netbios_ssn
61	service_ftp_data	71	service_klogin	81	service_netstat
62	service_gopher	72	service_kshell	82	service_nnsp
63	service_harvest	73	service_ldap	83	service_nntp
64	service_hostnames	74	service_link	84	service_ntp_u
65	service_http	75	service_login	85	service_other
66	service_http_2784	76	service_mtp	86	service_pm_dump
67	service_http_443	77	service_name	87	service_pop_2
68	service_http_8001	78	service_netbios_dgm	88	service_pop_3
69	service_imap4	79	service_netbios_ns	89	service_printer

Tablo Ek. 4: NSL-KDD Veriseti 90-122 Arası Özellik Tablosu

No	Özellik	No	Özellik	No	Özellik
90	service_private	100	service_systat	110	service_whois
91	service_red_i	101	service_telnet	111	flag_OTH
92	service_remote_job	102	service_tftp_u	112	flag_REJ
93	service_rje	103	service_tim_i	113	flag_RSTO
94	service_shell	104	service_time	114	flag_RSTOS0
95	service_smtp	105	service_urh_i	115	flag_RSTR
96	service_sql_net	106	service_urp_i	116	flag_S0
97	service_ssh	107	service_uucp	117	flag_S1
98	service_sunrpc	108	service_uucp_path	118	flag_S2
99	service_supdup	109	service_vmnet	119	flag_S3
				120	flag_SF
				121	flag_SH
				122	Class

ÖZGEÇMİŞ

Oğuzhan TAŞ

A) EĞİTİM

1995-1999 : Elektronik ve Bilgisayar, Teknik Eğitim Fak., Fırat Üniversitesi.

2000-2002 : Bilgisayar Müh., Mühendislik Fakültesi, Fırat Üniversitesi.

2016- : Bilgisayar Bil. ve Müh., İstanbul Sabahattin Zaim Üniversitesi.

B) MESLEKİ DENEYİM

Kurumlarda Verilen Dersler

2000-2002 : Araştırma Görevlisi, Bilgisayar Müh., Fırat Üniversitesi

Dersler: Yazılım Mühendisliği, Bilgisayar Mühendisliğindeki Gelişmeler, Nesne Yönelimli Programlama (Delphi, C++ ve Java ile)

1999-2022 : Bilişim Teknolojileri Öğrt, Bilişim Teknolojileri Bölümü, Millî Eğitim Bakanlığı.

Dersler: Açık Kaynak İşletim Sistemleri (Linux), C# ile Nesneye Yönelik Programlama, Veritabanı Yönetim Sistemleri, PHP ile Web Programlama, Gelişmiş İnternet Uygulamaları (ASP.net), Python ile Programlamaya Giriş, Arduino Programlama (Standart C ile), Flutter ile Mobil Programlama.

2019-2021 : Öğretim Görevlisi (Yarı Zamanlı), Bilgisayar Programcılığı ve Yazılım Mühendisliği, İstinye Üniversitesi.

Dersler: PHP ile Web Programlama, Unix Programlama I ve II, Web Tasarım (HTML, CSS, Bootstrap, JavaScript), JavaScript (REACT), C++ ile Nesneye Yönelik Programlama, Veritabanı Yönetim Sistemleri

2021-2022 : Öğretim Görevlisi (Yarı Zamanlı), Bilgisayar Programcılığı, Beykent Üniversitesi.

Dersler: Java ile Nesne Yönelimli Programlama (10 Saat-3 grup)

C) YAYINLAR

Hakemli Dergilerce Kabul Edilen Yayınlar

- Taş, O., Kiani F. (2021). Nesnelerin İnterneti (IoT) ve Kablosuz Algılayıcı Ağların Güvenliğine Yapılan Saldırıların Tespit Edilmesi ve Önlenmesi. Politeknik Dergisi, 24(1), 219-235. Doi: 10.2339/politeknik.627825
- Taş, O. & Kiani, F. (2018). Blok Zinciri Teknolojisine Yapılan Saldırılar Üzerine bir İnceleme. Bilişim Teknolojileri Dergisi, 11 (4), 369-382. Doi: 10.17671/gazibtd.451695

Gönderilen Dergi Yayınları

- Taş, O. & Yahyaoui A. (2022), Intrusion Detection System for IoT Environment using XgBoost Ensemble Algorithm and Grey Wolf Optimization Algorithm, Expert Systems with Applications.

Konferanslarda Sunulan Çalışmalar

- Taş, O. & Yahyaoui A. (2022), Machine Learning based Intrusion Detection System using Grey Wolf Optimization for Feature Selection, Springer CMES 2022, 6th International Conference on Computational Mathematics and Engineering Sciences.
- Taş, O., Mısır M. (2021), Anomali Tabanlı Saldırıların Tespit Edilmesi için Makine Öğrenmesi Algoritmalarının Kullanılması, İZÜ Fen ve Mühendislik Kongresi
- Taş, O., Mısır M. (2021), Saldırı Tespit Sistemleri üzerine Yapılan İncelemelerin, Veri Setlerinin ve Yaklaşımların Karşılaştırılması, İZÜ Fen ve Mühendislik Kongresi

- Taş, O., Kiani F. (2018). Blockchain Solutions for IoT(Keynote Speakers), ICCIE 2018: 20th International Conference on Computer and Information Engineering.