SİBER SALDIRILARIN TESPİTİNDE YAPAY ZEKÂ TABANLI ALGORİTMA TASARIMI

2023
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Ahmet Nusret ÖZALP

Tez Danışmanı
Dr. Öğr. Üyesi Zafer ALBAYRAK

SİBER SALDIRILARIN TESPİTİNDE YAPAY ZEKÂ TABANLI ALGORİTMA TASARIMI

Ahmet Nusret ÖZALP

Tez Danışmanı
Dr. Öğr. Üyesi Zafer ALBAYRAK

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Doktora Tezi
Olarak Hazırlanmıştır

KARABÜK
Nisan 2023

Ahmet Nusret ÖZALP tarafından hazırlanan "SİBER SALDIRILARIN TESPİTİNDE YAPAY ZEKÂ TABANLI ALGORİTMA TASARIMI" başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Dr. Öğr. Üyesi Zafer ALBAYRAK .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 28/04/2023

Ünvanı, Adı SOYADI (Kurumu)	İmzası
Başkan : Dr. Öğr. Üyesi Zafer ALBAYRAK (SUBU)	.....
Üye : Prof. Dr. Necmi Serkan TEZEL (KBÜ)	.....
Üye : Prof. Dr. Numan ÇELEBİ (SAU)	.....
Üye : Doç.Dr. Yüksel ÇELİK (KBÜ)	.....
Üye : Doç.Dr. Salih GÖRGÜNOĞLU (KÜ)	.....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Müslüm KUZU .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Ahmet Nusret ÖZALP

ÖZET

Doktora Tezi

SİBER SALDIRILARIN TESPİTİNDE YAPAY ZEKÂ TABANLI ALGORİTMA TASARIMI

Ahmet Nusret ÖZALP

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Zafer ALBAYRAK

Nisan 2023, 84 sayfa

İnternet ağlarındaki hızlı genişleme bilgisayar ağlarında güvenliğin sağlanmasını zorlaştırmaktadır. Siber saldırıların tespiti ve önlenmesi zorlu bir süreçtir. Bu saldırıların tespiti ve önlem alınması için Saldırı Tespit Sistemleri (IDS'ler) ve Saldırı Önleme Sistemleri (IPS'ler) geliştirilmiştir. Farklı güvenlik çözüm önerilerinin çeşitlendirilmesine rağmen IDS'lerin hala düşük algılama doğruluğu (Acc), Yanlış Negatifler (FN) ve Yanlış Pozitifler (FP) gibi bazı zayıf yanları bulunmaktadır. Bu sorunların giderilmesinde ise izinsiz giriş tespit doğruluğunu artırmaya yardımcı olan ve yanlış negatif oranını ve yanlış pozitif oranını büyük ölçüde azaltan Yapay Zeka (YZ) çözümleri ve Makine Öğrenimi (ML) teknikleri kullanılmaktadır. Siber uzaydaki tehditlerin belirlenmesinde yapay zekâ yöntemleriyle birleştirilerek siber saldırılara karşı yeni yöntemler önerilmektedir. Bu araştırmalar siber güvenlik alanında makine öğrenmesi ve derin öğrenme yöntemlerinin geleneksel kural tabanlı algoritmalara karşı daha başarılı sonuçlar ortaya çıkardığını göstermiş olsada; hala geliştirilen saldırı

modellerinden IDS'lerin düşük algılama doğruluğu, Yanlış Negatifler (FN) ve Yanlış Pozitifler (FP) gibi saldırıların tespit edilmesinde eksik yönleri bulunmaktadır.
Bu tez çalışmasının ilk bölümünde bilgisayar ağlarına yönelik siber saldırıların tespitinde veri setindeki özelliklerin frekans etkileri incelenmiştir. İlk olarak veriseti içindeki özniteliklerin frekansları belirlenmiştir. Belirlenen özniteliklerin yüksek frekans özelliklerinin siber saldırıları tespit etmedeki etkisi, yaygın olarak kullanılan makine öğrenme algoritmaları Random Forest (RF), J48, Naive Bayes (NB) ve Multi-Layer Perceptron (MLP) ile incelenmiştir. Her bir algoritmanın performansı Kesinlik (P), FP, Doğruluk (Acc) ve Gerçek Pozitif (TP) Oranı istatistikleri dikkate alınarak değerlendirilmiştir. NSL-KDD veri setindeki farklı tipteki siber saldırıların tespiti makine öğrenmesi algoritmaları ile analiz edilmiştir. Saldırı tespitinde makine öğrenmesi algoritmalarının başarı kriterleri olarak P, Receiver Operating Characteristic (ROC), F1 skoru, hatırlama ve doğruluk istatistikleri seçilmiştir. Sonuçlar, yüksek frekansa sahip özelliklerin saldırıları tespit etmede etkili olduğunu göstermiştir.

Tez çalışmasının ikinci bölümünde yaygın olarak görülen siber saldırıların tespit edilmesinde iki farklı veriseti kullanılarak 4 farklı saldırı tespit modeli önerilmiştir. Tasarlanan saldırı tespit modelleri NSL-KDD ve CICIDS2018 verisetleri ile test edilmiştir. Önerilen modellerde ilk olarak verisetleri ön işlem ile normalize edilmiştir. Daha sonra hibrit modelde kullanılan Long Short-Term Memory (LSTM) ve Convolutional Neural Network (CNN) algoritmaları ile normalize edilmiş verilerden öznitelik çıkarım işlemi yapılmıştır. Son aşamada LightGBM ve XGBoost algoritmalarının saldırıları tespit edilmesi için sınıflandırma algoritması olarak kullanılmıştır. Önerilen modellerin test edilmesinde doğruluk, kesinlik, recall ve F1 score parametreleri kullanılmıştır. Yapılan deneysel çalışmada SQL Injection, Brute Force ve Denial-of-Service Attack (DoS) atakların tespitinde XGBoost algoritmasının daha başarılı sonuçlar elde ettiği görülmüştür. Önerilen bir diğer modelde ise CNN-LSTM/LightGBM algoritmaları ile yapılan testlerde, yaygın olarak görülen 14 saldırı türünün tespitinde başarılı olduğu görülmüştür.

Anahtar Sözcükler : Saldırı Tespiti, Siber Güvenlik, IDS, Öznitelik seçimi, Makine Öğrenimi, Derin Öğrenme

Bilim Kodu **:** 92403

ABSTRACT

Ph.D. Thesis

AI-BASED ALGORITHM DESIGN IN DETECTION OF CYBER ATTACKS

Ahmet Nusret ÖZALP

Karabuk University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assist April 2023, 84 pages

The rapid expansion in internet networks makes it difficult to provide security in computer networks. Detection and prevention of cyber attacks are challenging processes. Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPS) have been developed to detect and take action against these attacks. Despite the diversification of different security solution proposals, IDSs still have some weaknesses such as low detection accuracy (Acc), False Negatives (FN), and False Positives (FP). Artificial Intelligence (AI) solutions and Machine Learning (ML) techniques are used to solve these problems, which helps to increase the intrusion detection accuracy and greatly reduces the false negative rate and false positive rate. New methods are proposed against cyber attacks by combining them with artificial intelligence methods in identifying threats in cyber space. Although these studies have shown that machine learning and deep learning methods in the field of cyber security are more successful than traditional rule-based algorithms; IDSs, which are still developed attack models, have shortcomings in detecting attacks such as low detection accuracy, False Negatives (FN) and False Positives (FP).

In the first part of this thesis, the frequency effects of the features in the data set were examined in the detection of cyber attacks against computer networks. First, the frequencies of the features in the dataset were determined. The effect of the high frequency properties of the determined features in detecting cyber attacks was investigated by widely used machine learning algorithms Random Forest (RF), J48, Naive Bayes (NB) and Multi-Layer Perceptron (MLP). The performance of each algorithm was evaluated considering the Precision (P), FP, Accuracy (Acc) and True Positive (TP) Ratio statistics. The detection of different types of cyber attacks in the NSL-KDD dataset was analyzed by machine learning algorithms. P, Receiver Operating Characteristic (ROC), F1 score, recall and accuracy statistics were chosen as the success criteria of machine learning algorithms in attack detection. The results showed that features with high frequency are effective in detecting attacks.

In the second part of the thesis, 4 different attack detection models were proposed by using two different datasets to detect common cyber attacks. The designed intrusion detection models were tested with NSL-KDD and CICIDS2018 datasets. In the proposed models, firstly, the datasets were normalized by preprocessing. Then, feature extraction was performed from the data normalized with the Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) algorithms used in the hybrid model. In the last stage, LightGBM and XGBoost algorithms were used as classification algorithms to detect attacks. Accuracy, precision, recall and F1 score parameters were used to test the proposed models. In the experimental study, it was seen that the XGBoost algorithm achieved more successful results in detecting SQL Injection, Brute Force and Denial-of-Service Attack (DoS) attacks. In another proposed model, in tests with CNN-LSTM/LightGBM algorithms, it was found to be successful in detecting 14 common attack types.

Key Word **:** Intrusion Detection, Cybersecurity, IDS, Feature Selection, Machine Learning, Deep Learning

Science Code : 92403

TEŞEKKÜR

Doktora tez çalışmamda her türlü özeni gösteren ve tezimi bitirmemde büyük emeği geçen danışman hocam Sayın Dr. Öğretim Üyesi Zafer ALBAYRAK'a öncelikli olarak şükranlarımı sunuyorum. Tez çalışmam esnasında katkılarından dolayı değerli hocalarım Prof. Dr. Necmi Serkan TEZEL ve Doç. Dr. Yüksel ÇELİK'e teşekkür ederim.

Beşikten mezara kadar devam eden, gelişim ve değişim çıktıları ile insana değer katan öğrenme faaliyetlerinin uzun ve zor sürecinin tamamlanmasında emeği geçen değerli anne ve babama; bu zor süreçte sabrı ve bitmeyen desteği için canım eşime teşekkür ederim.

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

z : çıktı

w : ağırlık

a : eğilim

M_s : k adet öznitelik içeren S altkümesinin fayda değeri

rcf : sınıf etiketi ile ilgili öznitelik arasındaki korelasyon

rff : özniteliklerin birbirleri arasındaki korelasyon

n : Verisetindeki öznitelik sayısı

o_i : i'inci öznitelik için gözlenen frekans değeri

e_i : i'inci öznitelik için beklenen frekans değeri

h : Sınıflandırıcı

θk : rastgele vektör

x_i : ağaç sınıf etiketi

$P(c|x)$: Arka Olasılık

$P(x|c)$: Olasılık

$P(c)$: Sınıf Öncelikli Olasılık

$P(x)$: Predictor Önceki Olasılık

m : Bir önceki katmandaki nöron sayısı

x : giriş değeri

b : rastgele sapma

$L(x,y)$: Çapraz entropi

Obj^m : XGBoost fonksiyonu

w_j : XGBoost ağırlık

$V_j(d)$: Bilgi kazancı

r_i : Gradyan mutlak değer

D : Training data

$L(y,\theta)$: Loss function

M : Veri örnekleme uzayı

U_{normal} : Giriş

$U_{optimal}$: Çıkış

KISALTMALAR

IDS	: Saldırı Tespit Sistemleri
IPS	: Saldırı Önleme Sistemleri
FN	: Yanlış Negatif
FP	: Yanlış Pozitif
TP	: Gerçek Pozitif
TPR	: Gerçek Pozitif Oranı
FPR	: Yanlış Pozitif Oranı
ML	: Makine Öğrenimi
DL	: Derin Öğrenme
RF	: Random Forest
NB	: Naive Bayes
MLP	: Multi-Layer Perceptron
P	: Kesinlik
Acc	: Doğruluk
LSTM	: Long Short-Term Memory
CNN	: Convolutional Neural Network
RNN	: Tekrarlayan Sinir Ağı
DNN	: Derin Sinir Ağı
DoS	: Denial-of-service attack
ROC	: Receiver Operating Characteristic
HIDS	: Host Tabanlı Saldırı Tespit Sistemi
NIDS	: Ağ Tabanlı Saldırı Tespit Sistemi
R2L	: Remote to Local
U2R	: User to Root
CBS	: Correlation-Based Self-Attribute Selection
CS	: Chi-square
1-R	: One-R
SUC	: Symmetrical Uncertainty Coefficient

IGA : Information Gain
GRF : Gain Rate
APT : Advanced Persistent Threat
DDoS : A Distributed Denial-of-Service
IoT : Internet of Things
MITM : Man-In-The-Middle
MıTMo : Man-In-The-Mobile
URL : Uniform Resource Locator
NAT : Ağ Adresi Çevirisi
IP : Internet Protocol
STS : Saldırı Tespit Sistemi
SÖS : Saldırı Önlem Sistemi
CRC : Döngüsel Yedeklilik Denetimi
TCP : Transmision Control Protocol
NIPS : Network-Based Intrusion Prevention
HIPS : Host-Based Intrusion Prevention
WIPS : Wireless Intrusion Prevention Systems
NBA : Network Behavior Analysis
CPU : Merkezi işlem birimi
GPU : Grafik işlem birimi
SVM : Destek Vektör Makineleri
USB : Universal Serial Bus
RFC : Request For Comments
SYN : Synchronize
OSI : Open Systems Interconnection
SQL : Structured Query Language
TLD : Top Level Domain
DNS : Domain Name Server
SIEM : Security Information and Event Management
RAM : Rastgele Erişebilir Bellek
CIA : Confidentiality Integrity Availability
MD5 : Message-Digest algorithm 5
SHA-1 : Secure Hash Algorithm 1

SHA-256	: Secure Hash Algorithm 256
SHA-512	: Secure Hash Algorithm 512
HTTP	: Hypertext Transfer Protocol
ICMP	: Internet Control Message Protocol
UDP	: User Datagram Protocol
TCP	: Transmission Control Protocol
NIC	: Network Interface Cart
HOIC	: High Orbit Ion Cannon
LOIC	: Low Orbit Ion Cannon
XSS	: Cross-Site Scripting
SSH	: Secure Shell Protocol
FTP	: File Transfer Protocol
MDPCA	: Modified Density Peak Clustering Algorithm
DBN	: Deep Belief Network
PCA	: Principal Component Analysis
ReLU	: Rectified Linear Unit
DMTK	: Distributed Machine Learning Toolkit)

BÖLÜM 1

GİRİŞ

Bilgisayar ağlarındaki cihaz sayısı arttıkça üretilen ağ trafiği de artmaktadır. Ağdaki güvenlik tehditleri ve olası siber saldırılar veri bütünlüğü, veri gizliliği ve veri erişilebilirliği gibi kavramları ortaya çıkarmıştır. Siber saldırıların tespiti ve önlenmesi karmaşık bir sorundur. Saldırıları tespit ve önlemek amacıyla Saldırı Tespit Sistemleri (IDS) ve Saldırı Önleme Sistemleri (IPS) gibi sistemler geliştirilmiştir. Yapılan birçok çalışmaya rağmen IDS'lerin düşük algılama doğruluğu, Yanlış Negatifler (FN) ve Yanlış Pozitifler (FP) gibi saldırıların tespit edilmesinde zayıf yönlerine bulunmaktadır [2]. Geleneksel kural tabanlı tespit sistemlerinde önceden belirlenen güvenlik ilkeleri ile tespit gerçekleştirilir. Belirlenen güvenlik kuralları dışında bir durum gerçekleştiğinde, saldırının tespiti ve yanıt süresi noktasında problemler ortaya çıkmaktadır. Bu gibi durumlara karşı saldırı tespit doğruluğunu artırmaya yardımcı olan, yanlış negatif oranını ve yanlış pozitif oranını azaltacak Makine Öğrenimi (ML) yaklaşımları önerilmektedir. Yapılan araştırmalarda bu yaklaşımlarla geliştirilen modellerde farklı yapay zekâ algoritmaları birleştirilerek siber saldırılara karşı daha etkin müdahale yöntemleri geliştirilmektir [2-3]. Bu araştırmalar siber güvenlik alanında makine öğrenmesi ve derin öğrenme yöntemlerinin geleneksel kural tabanlı algoritmalara karşı daha üstün olduğunu göstermiştir [4]. Saldırı tespit, kötü amaçlı yazılım analizi ve spam tespitinde yapay zekâ tekniklerinin daha güçlü olduğu gözlemlenmiştir [5]. Siber saldırıların algılanması ve önlenmesi ile ilgili yapılan araştırmalar güncel veri setleri üzerinden yapılmalı ve seçilen veri setlerinin güncel ve farklı saldırı tiplerini içermelidir. Farklı saldırı tiplerini içeren verisetleri ile makine öğrenmesi ve derin öğrenme algoritmaları ile saldır tespit performansları karşılaştırılarak doğruluk analizleri yapılmaktadır.

Yapılan çalışmalar incelendiğinde, kullanılan algoritma ve yöntemlere göre saldırı tespitindeki elde edilen doğruluk oranlarında farklılıklar olduğu görülmüştür.

Doğruluk oranını etkileyen faktörlerin başında öznitelik seçim yöntemi gelmektedir [1]. Makine öğrenimi ve Derin öğrenme algoritmaları kullanarak geliştirilen yöntemlerde yaklaşımın aşırı öğrenmesi, ilişkisel bağı olmayan öznitelikler algoritmanın sınıflandırma performansını azaltan ve desteklenmeyen bellek gereksinimine neden olmaktadır. Literatür incelemesi sonucu elde ettiğimiz bu sonuçlar bizi bu alanda çalışmaya motive etmiştir. Mevcut makine öğrenimi ve derin öğrenme algoritmalarının performanslarına göre Kesinlik (P), Yanlış Pozitif Oranı (FPR), Gerçek Pozitif Oranı (TPR), Doğruluk (Acc) metrikleri incelenmiştir. Yapılan literatür araştırmalarında XGBoost, LightGBM, LSTM ve CNN saldırı tespit performansları incelendiğinde saldırıların tespitinde kullanıldığı görülmüştür [6].

Bu çalışmada, literaturden farklı olarak dört farklı model önerilmiştir. Dört modelde de öncelikle farklı boyut ve özniteliklere sahip verisetleri normalize edilmiştir. Dengesiz ve gereksiz veriler temizlenerek, modelin gereksiz verileri tutması engellenmiştir. Karmaşık verisetleri kullanılarak, ağ trafiğindeki olası saldırıları tespiti için dört farklı model tasarlanmıştır. Tasarlanan modellerde Derin Öğrenme ve Makine öğrenimi algoritmaları ile dengesiz dağılan veriler azaltılarak, saldırıların tespitinde sınıflandırma modelinin etkin kullanılması sağlanmıştır. Çalışmanın başlıca katkıları aşağıda özetlenmiştir:

- Saldırı tespit sistemlerinde kullanılan güncel CICIDS2018 ve NSL-KDD verisetleri tercih edilerek önerilen dört model ile 2 veriseti üzerinde test edilmiştir. Verisetleri ön işlem aşamasında dengesizliklik oranları azaltılarak, sınıf dengesizliği minimize edilmiştir.
- Verisetlerinde öznitelik seçimi için derin öğrenme algoritmaları, sınıflandırma için makine öğrenimi algoritmaları seçilmiştir.
- Derin öğrenme algoritmaları ile anomali durum tespiti yapılarak önerilen modellerde CNN ve LSTM algoritmalarıyla 32 çekirdekli 2 gizli katman oluşturulmuştur. Bu çekirdeklerin görevi eğitim sırasında öğrenilen verilerin kayıt edilmesidir. Girdilerin normalleşmesi için gizli katmanlar kullanılıp; bu yöntem sınıflandırma sırasında doğruluğu arttırmak için tercih edilmiştir.
- Ağdaki saldırıların tespiti için Makine öğrenimi algoritmaları ile sınıflandırma yapılmıştır. Sınıflandırma algoritmaları seçilirken hesaplama kaynaklarına bağımlılık yüzdesi düşük ve yüksek performans gösteren XGBoost ve

LightGBM algoritmaları seçilmiştir. LightGBM ile verisetlerinde bulunan 14 farklı saldırıyı tespit etmek için kullanılmıştır.

- Çalışma sonunda iki farklı veriseti üzerinde Kesinlik, Doğruluk, recall ve F1 score performansları ölçülmüştür.Çalışma sonunda 2 farklı veriseti üzerinde 4 farklı saldırı tespit modeli önerilmiştir. Önerilen modeller literatürdeki çalışmalarla karşılaştırıldığında saldırıların tespitinde yüksek doğruluk göstermiştir.

1.1. ÇALIŞMANIN AMACI

Yukarıda belirtilen problemlerin çözümüne ilişkin, her geçen gün artan siber saldırılara karşı bir tespit sisteminin geliştirilerek, bu modelin saldırıları doğru ve hızlı şekilde tespitinin toplanan veriler üzerinden yapılması bu tez çalışmasının konusunu oluşturmaktadır. Önerilen 4 modelde, saldırıların tespitinde yüksek doğruluk göstererek, saldırıların önlenmesine yönelik yeni dört model sunulmuştur. Ayrıca verisetleri üzerinden yapılan saldırı tespitinde yüksek frekanslı özniteliklerin saldırıların tespitinde önemli bir etki ettiği de gösterilmektedir.

1.2. LİTERATÜR İNCELEMESİ

Bu bölümde siber saldırıların tespiti ve önlenmesine ilişkin yapılan literatür çalışmasına yer verilmiştir. Siber saldırıları tespit edilmesi, saldırılara karşı önlem alma ve sistemlerin güvenliğini sağlamak için derin öğrenme ve makine öğrenimi algoritmalarından yararlanılmaktadır [1]. Saldırı tespit sistemleri hakkında yapılan literatür taramasında makine öğrenmesi, veri madenciliği ve derin öğrenme algoritmaları ile yapılan uygulamalar görülmektedir. Ağ trafiğindeki anormal durumların tespit edilmesinde, veri madenciliği ve makine öğrenmesi teknikleri kullanılmaktadır [2]. Trafiğin sınıflandırılmasında ise makine öğrenmesi tekniklerinden faylanılmakta [3–5] ve bu teknikleri kullanılarak saldırı tespit uygulamalarında doğruluk, pozitif doğruluk oranı, algılama süresi gibi performanslar değerleri test edilmiştir [3,6,7]. Ağ trafiği üzerinden saldırıların tespit edilmesinde makine öğrenmesi tekniklerinin doğruluk oranının yüksek olduğu görülmektedir [7]. Bu durum makine öğrenme algoritmaları kullanarak yapılan saldırı tespit yaklaşımlarının diğer yöntemlere göre daha yüksek başarı sağladığı görülmektedir [8].

Bu yöntemlerde bilgisayar ağları üzerinden toplanan veriler işlenerek, elde edilen bilgiler anlamlı hal getirilir. Bu yöntemleri çok fazla öznitelik ve çok fazla veri barındıran veri kümeleri üzerinde uygulamak zaman alıcı bir süreçtir ve bu durum bilginin doğruluğunu etkilemektedir. Ağ tabanlı saldırı tespit sistemlerindeki öznitelik seçimi aşaması, ağ üzerinden gelen büyük veri yığınının anlamlı hale getirilmesi noktasında zor bir süreçtir [9]. Gereksiz ve önemsiz trafikten elde edilen öznitelikler algoritmanın sınıflandırma performansını azaltan ve desteklenmeyen bellek gereksinimine neden olmaktadır. Bununla birlikte saldırı tespit sistemi modellerinin eğitim ve test aşamalarında ihtiyaç duyulan hesaplama kaynağını arttırmaktadır [10].

Çizelge 1.1'de saldırı tarama yöntemlerinin saldırı tespit sistemleri üzerindeki etkisi gösterilmiştir. Çizelgedeki verilerde ilk olarak, gönderilen paketlerin erişim kontrol listeleri ve güvenlik duvarı üzerinden geri dönen paketler tespit edilmektedir. Elde edilen bilgiler saldırılar için önemli bir parametre oluşturur. Port tarama saldırılarına karşı kontrol listeleri oluşturulur. Güvenlik duvarının doğru şekilde port tarama işlemlerini tespit etmesi istenmektedir. Yönlendirme ve filtreleme işlemi yapan cihazlar belirli kaynak portları by-pass eder ve kullanılması istenmeyen portlar için izinsiz girişi engelleyen özel kurallar yazılmalıdır. Yazılan bu kurallara göre ağ trafiği üzerinde veri toplanır. Toplanan verilerle ağdaki trafiğin durumu gözlenmektedir. Olağandışı durum tespit edildiğinde ise saldırı yönteminin belirlenmesi istenmektedir.

Çizelge 1.1.1. Ağ Tarama Saldırıları

Tarama Teknikleri	**Giden Paket**	**Port Durumu**	**Gelen Paket**	**3'lü el sıkışma**	**IDS Durum**
TCP Connect/Full Open Scan	TCP	Açık	RST	Evet	Evet
Stealth Scan/Half-open Scan	TCP	Açık	RST	Evet	Evet
Inverse TCP Flag Scanning	FIN, URG, PSH,	Açık	RST	Evet	Evet
Xmas Scan (Xmas Taraması)	FIN, URG, PSH, TCP	Açık	Inverse TCP	Evet	Evet
ACK Flag Probe Scanning	TCP / ACK	Açık	RST	Evet	Evet
IDLE/IPID Header Scan	TCP, SYN	Açık	RST/SYN ACK/ RST	Evet	Evet

Ağ trafiği analizi ve doğrulaması konusunun önemi nedeniyle, son zamanlarda yayınlanan birçok çalışmada bu konu ele almıştır. Gerçek zamanlı paket analizi ile anomali tespit edilirken performans, doğruluk gibi parametrelerle tespit etmek zordur [3]. Bu durum cihazlardaki aşırı enerji kullanımı ve bellek yetersizlikleri veri setlerinin kullanılmasını gerektirir. Makine öğrenmesi ve derin öğrenme teknikleriyle anomali tespiti için yapılan çalışmalarda, veri setleri eğitim ve test verisi olarak kullanılır. Eğitilen veriler saldırıların oluştuğu anı tespit ederek, alınacak tedbirler hakkında bilgi verir. Eğitim verisine eklenen her bilginin analizi ve kontrolü gerekmektedir [8]. Böylece etiketlenen verilerle öğrenme sağlanır. Kullanıcı ve cihaz sayısındaki artışlar, gerçek zamanlı saldırıların tespit edilmesinde yaşanan zorluklar, cihazların saldırıları tespit etmesindeki donanımsal yetersizlikler ve maliyet gibi sebepleri ortaya çıkarmaktadır. Bu bölümde güncel verisetleri olan NSL-KDD ve CIC-IDS2018 kullanılarak makine öğrenimi ve derin öğrenme algoritmaları ile yapılan çalışmalar incelenmiştir.

Jin ve arkadaşları [9] KDD Cup 99 veri kümesi üzerinde CatBoost ve K-en Yakın Komşular (KNN algoritmalarının avantajları ile karma bir model önermiştir. Bu model çoklu sınıflandırma performansı ortaya konulmaktadır. Önerilen model duyarlılık performans metriği açısından DoS (%99.4), Probing (%94.14), U2R (%50) ve R2L (%55,23) doğruluk (accuracy) oranı sonuçlarını elde etmiştir.

Khraisat ve arkadaşları [11] tarafından NSL-KDD ve Avusturalya Savunma Kuvvetleri Akademisi (ADFA) veri kümesi üzerinde C5 karar ağacı algoritması ve Tek Sınıf Destek Vektör Makinesinin birleştirildiği karma bir model önerilmiştir. Önerilen modelin ikili sınıflandırma performansı ortaya konmaktadır. Önerilen model NSLKDD veri kümesi üzerinde %83,24 doğruluk (accuracy) oranı elde etmiştir.

Kasongo ve Sun [12] saldırı tespiti için derin öğrenme tabanlı bir sistem önermişlerdir. Önerilen sistemde özelliklerin çıkarımı için entropiye dayalı bir algoritma kullanılmıştır. Özellik çıkarımı işleminden sonra geleneksel makine öğrenme yöntemleri ile önerilen ileri beslemeli derin sinir ağı yöntemi karşılaştırılmıştır. Çalışmada eğitim aşamasında KDDTrain veri setinin %75'i seçilmiştir. Test aşamasında KDDTest+ verisinde %87,74 alındığı görülmüştür.

Hu ve arkadaşları [13] saldırı tespitini önlemek için yine derin öğrenme tabanlı hibrit bir yöntem önermişlerdir. Çalışmada NSL-KDD veri setindeki saldırı türlerinin dağılımını dengelemek için Adasyn algoritmasını kullanmışlardır. Veri setindeki ön işleme aşamasının ardından geliştirilmiş CNN modeli tasarlanmıştır. Önerilen model, kanallar arası bilgi fazlalığının model eğitimi üzerindeki etkisini ortadan kaldırabilen bölünmüş evrişim modülüne dayanmaktadır. KDDTest+ verisinde %84,08, KDDTest21 verisinde ise %72,54 başarı oranı elde edildiği belirtilmiştir [13].

Latah ve Toker [14] KNN, aşırı öğrenme makinesi ve hiyerarşik aşırı öğrenme makinesi sınıflandırma yöntemlerine dayalı çok katmanlı bir yaklaşım önermişlerdir. Önerilen yöntemde her katmanda eğitim veri seti belirli bir saldırı türü (DoS, Probe vb.) ve normal bir trafiği ya da sonraki katman tarafından tespit edilmesi gereken bir saldırıyı temsil eden "diğer" kategorisi olmak üzere iki kategoriye ayrılmıştır. Modelde her bir katman başına bir sınıflandırıcı kullanılmıştır. Sonuçlar incelendiğinde KDDTest+ verisinde %84,29 başarı oranı elde edildiği görülmüştür.

Wang ve arkadaşları [15] saldırı tespit sistemi için makine öğrenmesi temelli bir çerçeve önermişlerdir. Önerilen yöntem veri setindeki önemli özellikleri çıkararak, özelliklerin değerleri ile belirli saldırı türleri arasındaki ilişkileri keşfetme temeline dayanmaktadır. Önerilen yöntem ile KDDTest+ verisinde %80,60 başarı oranı elde edildiği belirtilmiştir.

Su ve arkadaşları [16] iki aşamalı derin öğrenme tabanlı bir sistem önermişlerdir. Birinci aşamada trafik verilerinin yerel özelliklerini yakalamak için evrişimli katman kullanılmıştır. İkinci aşamada ise, veri paketindeki zaman serisi özelliğini öğrenmek için Bidirectional Long Short-Term Memory (BLSTM) modeli önerilmiştir. Çalışma geleneksel derin öğrenme yöntemleri ile karşılaştırılmış olup, önerilen sistemin diğer yöntemlere göre KDDTest+ verisinde %84,25, KDDTest-21 verisinde %69,42 başarı oranı ile daha iyi sonuçlar elde edildiği belirtilmiştir. Mohammed ve Grashi [17] çalışmalarında özellikleri seçmek için özyinelemeli özellik eleme kullanarak (IDS) uygulaması ve sınıflandırma için Derin Sinir Ağı (DNN) ve Tekrarlayan Sinir Ağı (RNN) kullanması önerdiler. Önerilen model %94'e ulaşan yüksek doğruluk oranıyla iyi sonuçlar verdiğini gördüler [17].

Çizelge 1.2'de NSL-KDD veriseti kullanılarak yapılan çalışmalarda doğruluk oranlarını gösterilmiştir. Deneysel çalışmada Random Forest (RF) algoritmasının diğer yöntemlere göre doğruluk yüzdesinin yüksek olduğu sonucuna ulaşılmıştır. Ayrıca hibrit olarak geliştirilen yöntemlerde de modellerin performans ve doğruluk değerlerinin de arttığı görülmektedir.

Çizelge 1.2 NSL-KDD ile yapılan çalışmalarda elde edilen doğruluk değerleri

Yapılan araştırmalar	**Veriseti**	**Doğruluk (%)**	**Kullanılan Teknikler**
El Boujnouni ve ark. [18]	NSL-KDD	77,5	SSPVSVDD
Gao ve ark. [19]	KDDTest	84,54	RT
Tama ve ark. [20]	KDDTest	85,00	RF
Yang ve ark. [21]	KDDTest	82,02	MDPCA-DBN
Liu ve ark. [22]	KDDTest	81,30	CNN
Wu ve ark. [23]	KDDTest	85,73	DBN-SVM
Haggag ve ark. [24]	KDDTest	85,44	MLP, RNN, LSTM

CIC-IDS2018 veriseti güncel internet trafiği modeline en yakın verisetidir. 80 öznitelik ve internet trafiğine ilişkin veriler ağ trafik akışına dayalı olarak üretilmiştir. Ferrag [8] ve arkadaşları CIC-IDS2018 ve BoT-IoT gerçek verisetlerinin kullarak farklı derin öğrenme tekniklerini karşılaştırmıştır.

Çizelge 1.3 CIC-IDS 2018 veriseti ile yapılan çalışmalardaki doğruluk değerleri

Yapılan araştırmalar	**Veriseti**	**Doğruluk (%)**	**Kullanılan Teknikler**
Ferrag ve ark. [10]	CIC-IDS 2017/2018	97,28	DBM, RNN, DNN, RBM, DBFN, CNN
Karataş ve ark. [25]		92,63	RF, DT, KNN
Zhou ve Pezaros [26]		91,36	DL
Kanimozhi ve Jacob [28]		96,00	ML
Yu ve ark. [29]		99	PBCNN
Basnet ve ark. [30]		99	RF-DL
Kim ve ark.[31]		92,56	CNN
Vinayakumaret [32]		96,2	ANN

Çalışma sonuçları değerlendirildiğinde RNN ile XSS saldırılarında en yüksek oranda tespit ettiler. Karataş [25] ve arkadaşları SMOTE tekniği ile CIC-IDS2018 verisetini kullanarak dengesiz verilerin oranını azaltmış ve Adaboost, RF,DT,KNN gibi algoritmaları ile saldırıların tespiti üzerine çalışmıştır. Pezaros[26] ve arkadaşları [27] zeroday saldırılarına yönelik olarak CIC-IDS2018 veri seti kullanarak farklı derin öğrenme teknikleri kullanmış, %96 oranında doğrulukla saldırıları tespit etmişlerdir.

Kim ve arkadaşları CNN modelinde CIC-IDS2018 verisetini kullanarak 78 özellik içeren bir görüntü boyutu oluşturarak saldırıları tespit etmiştir. Çizelge 1.3'de CIC-IDS2018 veriseti ile yapılan çalışmalarda elde edilen doğruluk oranları gösterilmiştir. Burada %99 oranında derin öğrenme teknikleri ile saldırıların yüksek doğruluk oranı ile tespit edildiği görülmektedir.

1.3. TEZE GENEL BAKIŞ

Hazırlanan tez çalışması altı bölüm şeklinde düzenlenmiştir. Birinci bölümde, literatür taraması ve teze genel bir giriş yapılarak tez çalışmasının amacı açıklanmıştır.

İkinci bölümde, siber güvenlik kavramı açıklanmış, siber güvenlik yaklaşımları hakkında bilgi verilmiş, saldırı türleri, yaygn görülen saldırılar ve saldırı tespit sistemleri hakkında bilgi verilmiştir.

Üçüncü bölümde, verisetleri ve veri ön işlemleri hakkında bilgi verilmiş, performans kriterleri açıklanmıştır.

Dördüncü bölümde, önerilen modellerde yapılan ön veri işlemleri, kullanılan algoritmalar ve modelin test sonuçları hakkında bilgi verilmiştir.

Beşinci bölümde, önerilen saldırı tespit sistem modelleri detayları açıklanmış ve deneysel çalışma sonucu elde edilen sonuçlar değerlendirilmiştir.

Altıncı bölümde, çalışma sonucunda elde edilen veriler yorumlanmış ve tartışılmıştır.

BÖLÜM 2

BİLGİSAYAR AĞLARI VE SİBER GÜVENLİK

Birçok cihazın birbiri ile iletişim kurarak oluşturduğu bilgisayar ağları, hayatın bir parçası haline gelmiştir. Bu ağlar bilgi akışını sağlayarak günlük yaşamın her yerinde karşımıza çıkmaktadır. Finansal, tıbbi ağlar başta olmak üzere yüksek kapasiteli veri akışı, bu ağların önemini arttırmaktadır. Genelde yönetimsel yapılar depoladıkları verileri depolama, paylaşma ve işleme noktasında bu ağları kullanmaktadır. Her geçen gün artan veriye bağlı olarak ortaya çıkan veri trafiği bilgisayar ağlarının güvenliğinin sağlanmasının önemini arttırmaktadır. Bu noktada siber güvenlik kavramı, bu verilerin yetkisiz kullanıcılara karşı korunma ve zarar görmesine karşı korumayı amaçlayan bir disiplin olarak açıklanabilir. Güvenlik kişisel düzeyde başlayarak verilerin cihaz ve kullanıcı bazlı korunmasını gerektirir. Kurumsal büyüklükte düşünüldüğünde ise, bilgi güvenliği kavramının ekonomik bir karşılığının da olduğunu göstermektedir. Devletlerin bekasını sürdürmesi, sosyo-ekonomik kalkınmalarını sağlamasında da siber uzaydaki güvenlik gereksinimlerini yerine getirmesi gerekmektedir. Kullanıcı bazlı oluşturulması gereken güvenlik ilkeleri ile çevrimiçi ortamdaki verilerin güvenli akışı sağlanabilir. Bu platformlarda paylaşılan verilerin sınırlı olması kullanıcı bazlı oluşturulacak ilkelerin başında gelir. Kişisel verilerden yola çıkılarak birçok veriye ulaşılabilmesi, siber uzayda paylaşılan verilerin önemini arttırmaktadır.

Son yıllarda çok daha değerli hale gelen veri kavramı siber güvenliğin merkezi oluşturmaktadır. Özellikle kişisel bilgileri içeren tıbbi bilgiler; ekonomik bilgileri içeren finansal veriler toplumları doğrudan etkileyen veriler olmuştur. Değerli bir verinin olduğu noktaların kötü niyetli kullanıcılar tarafından tehdit unsuru olması siber güvenlik kavramının önemini arttırmaktadır. Konvensiyonel veri olarak adlandırılan finansal, kurumsal, kişisel verilerin güvenliğinin sağlanması için izlenen stratejiler günümüzde yetersiz kalmaktadır. Artan cihaz sayısı ile ortaya çıkan zafiyetler bu verilerin korunmasını zorlaştırmaktadır. Büyük veri (big data), sanallaştırma yoluyla oluşturulan yüksek kapasiteli depolama alanlarını ortaya çıkarmıştır. Nesnelerin

İnterneti (IoT) bu büyük veri yığınlarının ortaya çıkmasındaki temel faktördür. IoT ağları, geleneksel ağ yapısının sensör verileri sayesinde daha yüksek verilerin akışının olduğu yapılar haline getirmiştir [33]. Verilerin çeşitliliği, genişleme hızları, sahip olduğu hacimleri güvenlik kavramının önemini başka bir boyuta getirmiştir. Siber güvenlik kavramında bilgilerin gizliliği, bütünlüğü ve uygunluğu Confidentiality (Gizlilik), Integrity (Bütünlük), Availability (Erişilebilirlik) CIA üçlüsü olarak adlandırılan yapıyı ortaya çıkarmıştır. Güvenlik için bu üç gereksinim karşılanması gerekmektedir.

Gizlilik

Verilerin öncelikli olarak önem ve güvenlik düzeylerine göre kategorize edilmesi gerekir. Böylece bilgilere erişim ilkeleri yetkili/yetkisiz kullanıcıları ortaya çıkarır. Verilerin şifrelenmesi, parola/şifre oluşturma, kimlik doğrulama politikaları ile gizliliğin sağlanmasında kullanılan yöntemlerdendir.

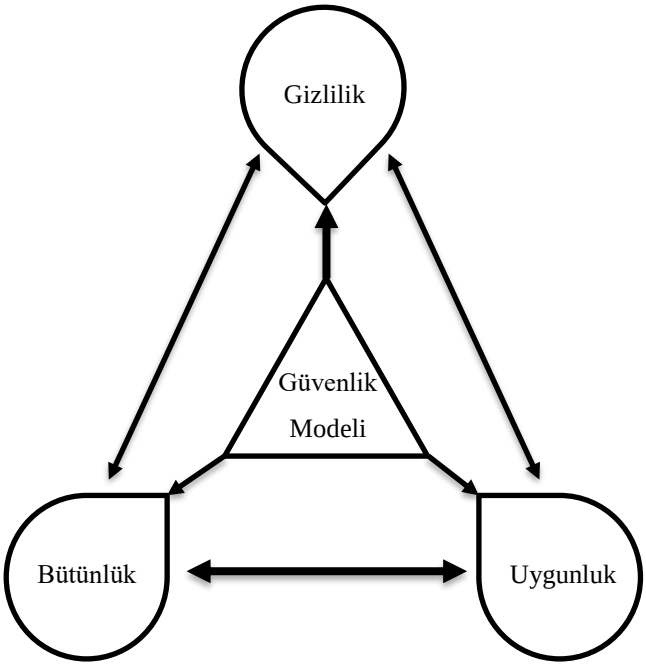
Bütünlük

Verilerin tutarlılığı, doğruluğu ve güvenilirliği bütünlük kavramını oluşturur. Bilgisayar ağları üzerindeki verilerin yetkisiz kullanıcılar tarafından erişilemez, değiştirilemez olmalıdır. Checksum Hashing, bütünlüğün doğrulanması için kullanılır. Message-Digest algorithm 5 (MD5), Secure Hash Algorithm 1(SHA-1), Secure Hash Algorithm 256, Secure Hash Algorithm 512 şeklinde çeşitleri vardır. Veri aktarımı sırasında bütünlüğün sağlanması önemlidir. Hash değerleri üzerinden aktarım sırasında bir sorunun olup olmadığı bilindiğinden, hash haline getirilen bilgiler kullanılarak, gerçek veriye doğrudan erişilemez. Genelde parola şifre unutma şeklinde ortaya çıkar ve şifre/parola sıfırlama işlemi yapılmaktadır.

Uygunluk

Donanımsal alt yapının yanında yazılımsal anlamda da sorunların ortadan kaldırılması, yetkilendirme gibi kavramlar uygunluk olarak ortaya çıkmaktadır. IDS yada firewall (güvenlik duvarı) gibi yapılar, hizmet reddi yada DoS gibi yaygın olarak görülen saldırı ve ihlallere karşı koruma sağlar.

Bu üç kavram düşünüldüğünde bir sistemi farklı türlerde yapılan siber saldırılardan korumanın teknik olarak mümkün olmadığı görülmektedir. Her an artan verilerin yer aldığı bilgisayar ağlarının korunması yüksek maliyeti de beraberinde getirmektedir. Ortaya çıkan güvenlik zaafiyetlerinin tespiti ve önlenmesi sırasında olası veri kayıpları, sistemin belli bir süre kesintiye uğraması gibi durumların da düşünülmesi gerekmektedir. Çevrimiçi verilerin yaşam döngüsüne bakıldığında, ortaya çıkan olası veri ihlali sonucu farklı kaynak ve platformlarda uzun süre erişime açık olduğu görülmektedir. Bu da kişisel ve kurumsal verilerin güvenliğinin sağlanmasının önemini arttırmaktadır. Güvenlik ihlalleri bazen yüksek finansal maliyeti, donanımsal kayıpları,kurumsal itibarın kayıp edilmesi ve ulusal güvenlik sorunlarını ile ortaya çıkmaktadır. Tüm bu sebepler siber güvenlik kavramının kişisel ve kurumsal anlamda bir yaklaşım olarak ele alınması gerektiği göstermektedir.



Şekil 2.1. CIA Üçlüsü

Siber saldırı senaryoları incelendiğinde mutlaka bir hazırlık aşamasının olduğunu görüyoruz. Motivasyon kaynakları farklı saldırgan tipleri, sistemli bir şekilde organize olarak saldırı senaryolarını uygulamaktadır. Saldırılara karşı geliştirilecek savunma stratejilerinde ise bu motivasyon kaynaklarını doğru analiz ederek önlemlerin alınması gerekmektedir. Araştırmacılar bu noktada farklı model ve yaklaşımlar üzerine çalışmaktadır. Bu modellerden biri Mandiant tarafından geliştirilen Attack Life Cycle

diğeri ise Locheed Martin tarafından geliştirilen Cyber Kill Chain (Siber Ölüm Zinciri) yaklaşımıdır. Attack Life Cycle, başarılı şekilde saldırıyı gerçekleştiren saldırganın sistem üzerinde yapmış olduğu işlemleri inceler. Burada saldırganın izlediği yöntem ve işlemler üzerinde yoğunlaşılır.

2.1. CYBER KILL CHAIN (SİBER ÖLÜM ZİNCİRİ)

Cyber Kill Chain 7 aşamalı bu yaklaşımda siber güvenlik politikalarının geliştirmesi için önerilen zincirin her aşamasında yapılanlar ve bunlara karşı alınacak önlemler ortaya koyulur.

2.1.1.Keşif (Reconnaissance)

Bilgi toplama aşamasıdır. Sistem analiz edilerek, farklı kaynaklardan toplanan veriler sızma (Penetration) amaçlı kullanılır. Aktif yada pasif kaynaklardan toplanan veriler saldırı stratejisini şekillendirir.

2.1.2. Silahlanma (Weaponization)

Toplanan veriler sonrası yapılan analizler sonucunda saldırı yapılacak noktalar belirlenir. Bu aşamada saldırı için kullanılacak zararlı tasarlanır, test edilir ve sisteme infekte olması için gerekli parametreler oluşturulur.

2.1.3. İletme (Delivery)

Bir önceki aşamada hazırlanan zararlı bu aşamada saldırı yapılacak sisteme gönderilir. Gönderme noktasında iletim yöntemi bu aşamada hazırlanır. İletim bazen çevrimiçi olarak gönderilirken, bazen de pasif yöntemlerle USB bellek vb yollarla yapılabilir. Son yıllarda yaygın olarak e-posta yoluyla iletimin yapıldığı görülmektedir.

2.1.4. Sömürme (Exploitation)

Oluşturulan zararlının sisteme bulaştırılarak, verilere erişim aşamasıdır. Zararlı hedef sistemde çalıştırılır ve istenen işlemlerin yapılması sağlanır. Burada işletim sistemi zafiyetlerinden yararlanılır. İletim aşamasında gönderilen zararlının çalıştırılması ile bu aşama başarılı bir şekilde yapılır.2.1.5. Yükleme (Installation)

Zararlı yazılımın hedef sisteme yüklenerek, mümkün olan en uzun süre sistemde kalmasının sağlandığı aşamadır. İşletim sistemi ve güvenlik kontrolleri aşarak sisteme kurulması sağlanan zararlı ile sistemin istenen kontrolleri sağlanmasına hazırlık yapılır.

2.1.6. Komuta-Kontrol (Command-Control)

Şifreli bir protokol üzerinden seçilen yöntemlerle hedef sistemin komuta kontrol edildiği aşamadır. Bu aşama ile artık hedef tamamen kontrol edilebilir şekilde kullanılabilir. Seçilen güvenli haberleşme yöntemi işletim sistemi ve güvenlik yazılımları zafiyetleri gözönüne alınarak aktif edilir.

2.1.7. Eylem (Action on Objectives)

Bir önceki aşamada güvenli bir protokol ile kontrol edilen sistem üzerinde artık saldırgan istediği işlemleri gerçekleştirebilir. Eğer geçici hizmet durdurma gibi eylemler planlanıyorsa botnet oluşturma yoluyla senaryolar aktif edilebilir. Günümüzde yaygın olarak görülen ramsomware saldırıları için de bu aşamada strateji geliştirilebilir.

2.2. MANDIANT ATTACK LIFE CYCLE

Siber ölüm zincirine göre farklı bir yaklaşım olan Attack Life Cycle modelinde yapılan keşif ile hedef sistem üzerinde ayrıntılı bir çalışma yapılma amacı ile hareket edildiği görülmektedir.

2.2.1. Başlangıç Keşfi (Initial Recon)

Aktif ve pasif bilgi toplama araçları başta olmak üzere, hedef üzerinde ayrıntılı bir çalışma yapılır. Sosyal ağlar bu noktada aktif olarak kullanılır. Saldırı vektörleri oluşturularak yapılacak saldırı modeli seçilir.

2.2.2. İlk Hareket (Initial Comprimise)

Seçilen hedefin oltalama gibi yöntemlerle sızma işlemleri yapılır.Bazen e-posta yoluyla yapılan bu teknikler zararlının bulunduğu ortama yönlendirme amacıyla kullanılır. Kullanıcı sayısının milyarları aştığı sosyal platformlar oltalama için sıkça

tercih edilmektedir. Eğer bir önceki aşamada toplanan veri miktarı yeterli aşamada ise, bu işlemlerin yapılması daha başarılı olacaktır.

2.2.3. Yerleşme (Establish Foothold)

Özellikle arka kapı (backdoor) açma yoluyla yapılan bağlantı aşamasıdır. Genel olarak arka kapı yazılımlarının saldırgan tarafından özel olarak tasarlandığında sorunsuz şekilde yerleşme işlemi yapılır. Shell olarak adlandırılan komut satırı üzerinden yapılan bağlantıların sıkça tercih edildiği görülmektedir.

2.2.4. Yetki Yükseltme (Escalate Priviliges)

Sisteme istenen erişim yetkileri ile erişmek için yapılan işlemlerin aşamasıdır. Erişim sağlanan hedef üzerinde değerli olarak adlandırılan verilere erişim için mümkün olan en üst düzeyde ayrıcalığa sahip olmak gerekir. Kullanıcı şifre/parolalarına erişim için hash formatındaki verilere erişilerek kullanıcı bilgileri üzerinden ayrıcalık kazanılır.

2.2.5. İç Keşif (Internal Recon)

Sistem üzerinde tanımlı yetkili bir kullanıcı ayrıcalıklarına erişim sağlanması ile ağdaki diğer cihaz ve kullanıcı bilgilerine erişim sağlanabilir. Böylece bilgisayar ağı üzerindeki tüm verilerin toplanır. İç ağda yapılan keşif çalışmaları genel olarak kullanılan e-posta sunucusu, dosya transfer sunucusu gibi servisler üzerinden yapılır.

2.2.6. Yayılma (Move Laterally)

Yapılan keşif çalışmasının ardından istenen derinlikte bir bilgiye ulaşılmadığı durumlarda ağa bağlı diğer cihazlar üzerinden yapılan uygulama kurma çalışmaları ile yayılma işlemi yapılır.Kritik bir aşamadır. Bu aşamada saldırgan sistem üzerinde uzun zaman periyotları geçirdiği için tespiti yapılabilir.

2.2.7. Yerini Sağlamlaştırma (Maintain Presence)

Bu aşama ile saldırgan sistem üzerine kuracağı diğer uygulamalarla erişim ve yetkilerini arttırabilir. Bu aşamada kurulan zararlı yazılımların sayısının arttığı ve saldırının tespit edilmesi durumunda kurulan birçok zararlı ile erişimin sürdürülebilir olduğu görülmektedir.

2.2.8. Görevi Tamamlama (Complete Mission)

Saldırganın temel amacı verilere erişim ve akan veriler hakkında bilgi sahibi olmaktır. Saldırgan bu aşamada ulaştığı verileri farklı formatlar üzerinden yedekleme yada uzak sunuculara aktarma işlemi yapar. Günümüzde yapılan bilgi sızma vakalarında sıkıştırılmış ve şifrelenmiş verilerin bu aşamada düzenlendiği görülmektedir.

2.3. SİBER GÜVENLİĞİN ÖNEMİ

İnsanoğlu varolduğundan bu yana hayatını devam ettirebilmesi için güvenlik kavramını merkeze almıştır. Güvende olmanın da ilk şartının haberdar olmak (Özalp & Asker, 2017) tezi bunu desteklemektedir. Siber alanda yapılan iş ve işlemler sırasında veri akışının gözlemlenebilmesi, verilerin ölçeklenerek yedeklenmesi sırasındaki kayıp ve müdahaleleri engellemesinin tespit ve önlem neticesinde olması gerektiğini göstermektedir. Yapılan saldırıların tür ve etkilerinin her geçen gün arttığı bu alanda, saldırı gruplarının doğru tespit edilmesi gerekir. Farklı motivasyonlara sahip saldırganlar, motivasyon kaynaklarını oluşturan değerli şeylere karşı farklı stratejiler geliştirir. Bazen sadece merak ile başlayan saldırılar, gerekli güvenlik önlemleri alınmayan sistemler üzerinde kalıcı hasarlar oluşturabilmektedir. Amatör olarak görülen bu tür saldırganlar, internet ortamından sağladıkları araçlarla deneme yanılma yöntemi ile saldırılar yapabilmektedir. Bu tür açık kaynak araçların yıkıcı etkileri birçok sistem tarafından bilinerek önlemi alınsa da kişisel verilere karşı yıkıcı etkiler oluşturabilmektedir. Hacker olarak adlandırılan grupların sistemlere yönelik yaptıkları saldırı amaçlarına göre siyah,gri ve beyaz şapkalı olarak sınıflandırıldığı görüyoruz.Zafiyetler testlerini yapılması, izinli şekilde sistemlerin kontrol edilmesinde görev alan beyaz şapkalıların sayısının artması siber güvenlik için önemlidir. Bu türdeki hackerların gri alana kayma olasılığının artması güvenlik zafiyetlerinin artmasına neden olur. Siyah şapkalı olarak adlandırılan gruplar ise kişisel ve kurumsal verilere erişmek için farklı motivasyonlarla hareket eder. Son yıllarda organize olarak hareket eden, bazen devlet destekli olan saldırgan grupları artmaktadır. Kendilerini hacktivist olarak da adlandıran bu gruplar finansal gücü ele geçirme veya kontrol odaklı motive olurlar. Devlet destekli olan grupları istihbarat faaliyetlerinde bulunur. Son yıllarda ülkelerin siyasi iktidarların belirlenmesi için yapılan seçimlerin manüpile edilmesinde rol aldıklarını görüyoruz.

Siber Güvenlik stratejilerinin geliştirilmesi için öncelikle tehdit olgusunun netleştirilmesi gerekir. Bir sistem veya organizasyon için saldırıların iç ve dış tehdit olarak kategorize edildiğini görüyoruz. Yapılan saldırılar incelendiğinde büyük bir kısmının kurum-kuruluşu, sistemi veya organizasyonu oluşturan kaynaklardan ortaya çıktığını görüyoruz. Bilgi güvenliği döngüsü içinde hareket etmeyen bilinçli yada bilinçsiz şekilde yapılan iş ve işlemler tehditlerin artmasına neden olmaktadır. Verilerin gizliliğine dikkat edilmemesi, sahip olunan bilgisayar ağ alt yapısına karşı tehdit oluşturabilecek kullanımlar, kötü amaçlı ve lisanslı yazılımların sistem içinde kullanılması gibi nedenler iç tehditleri oluşturmaktadır. Bu tehditlerin verdiği zararın yüksek olması, kullanıcıların daha dikkatli ve bilgili olmasını gerektirmektedir. Dış tehditler ise genel de bilgisayar ağının oluşturan cihaz ve yazılımların güvenlik açıklarından yararlanarak yapılan saldırıları oluşturur. Bu tür tehdit vektörlerinin oluşturulması sırasında sosyal mühendislik faaliyetlerin yapıldığını görüyoruz.

Siber uzayda ortaya çıkan bir güvenlik açığı yazılım yada donanım kaynaklı olabilir. Saldırganlar bu açıkları tespit ederek, tehdit vektörleri oluşturur. Bir güvenlik açığından yararlanmak için tasarlanmış kod bloklarına exploit; güvenlik açığına karşı yapılan faaliyete saldırı denir. Saldırı motivasyonun da ise veri kaynaklarına erişim isteği vardır. Donanımsal güvenlik açıkları tasarım ve mimariden kaynaklı sorunlardır. Bu tür açıkların başında Rastgele Erişebilir Bellek (RAM) bellek kaynaklı zafiyetler gelmektedir. Rowhammer adı verilen bu istismar türünde aynı adrese sahip verilerin tekrarlı şekilde belleğe yazılarak komşu adresteki verilere erişim söz konusudur. Donanım kaynaklı zafiyetlerin cihaz bazlı olması olasu saldırıların etkisini arttırmaktadır. Bu nedenle cihazların sahip oldukları gömülü yazılımların güncel tutulması önemlidir. Yazılım kaynaklı zafiyetler ise genel olarak işletim sistemi yada uygulama kaynaklı ortaya çıkar. Servis paketi, yama olarak adlandırılan düzeltme yazılımları ile bu açıklar kapatılabilir. Sürekli yama yada güncelleme ile açıklarını kapatan sistemlerin kritik bölgelerde kullanılması da risk oluşturur.

Yaygın olarak görülen güvenlik zafiyetlerin birçoğunun yazılım kaynaklı olduğu kabul edildiğinde öncelikli olarak bu noktaların belirlenmesi gerekmektedir. Ara bellek taşması olarak adlandırılan açıklarda, veriler arabellek adreslerinin dışına yazılır. Bu durum uygulamaların diğer bellek alanlarına erişimini sağlar. Böylece veriler güvenlik

riskleri ile karşı karşıya gelir. Yarış koşulları adı verilen güvenlik açıklarında, program çalışma zamanında zamana bağlı olarak sıralı sonuçlar üretmesi beklenirken olağandışı çıktılar üretmeye başlar. Bu anda güvenlik açıkları ortaya çıkabilir. Uygulamaların olağandışı çalışmasına yönelik çalışmasına zorlanması doğrulanmamış girdilere neden olur. Program ve uygulamalar çalışması sırasında geçersiz arabellek çıktıları ürettiğinde açıklar ortaya çıkar.Kullanıcıların yetkileri doğrultusunda erişim ve denetim yetkilerinin kontrol edilmesi gerekir. Eğer erişim denetimleri doğru kurgulanmazsa birçok güvenlik açığı ortaya çıkar. Kritik noktalardaki donanımlara yönelik fiziksel olarak erişim söz konusu ise, verilerin zarar görme oranı da yüksektir. Kurumsal olarak tasarlanmış bilgisayar ağlarında yetkilendirme, kimlik doğrulama ve şifreleme gibi güvenlik senaryolarındaki zafiyetler büyük tehdit oluşturur. Güvenlik denetimi için kullanılan Security Information and Event Management (SIEM) çözümlerinin yapılandırılması sırasındaki eksikliklerde bu tehditi arttırmaktadır.

2.3.1. Zararlı Yazılım Türleri

Güvenlik tehditlerinin başında kötü amaçlı yazılımlar gelmektedir. Kötü amaçlı Yazılım (malware) sistemlere kalıcı veya geçici hasarlar vermek, verilerin içerikleri bozmak, kullanıcı erişim ve denetim yetkileri manipüle etmek için kullanılabilir. Yaygın olarak görülen kötü amaçlı yazılım türleri aşağıdaki gibidir.

2.3.1.1. Adware

Reklam yazılımları (adware), genel olarak online platform yada uygulama sürümleri ile birlikte sistemlere bulaşır.Birçoğu sadece reklam amaçlı kullanılsa da, sisteme bulaşarak casus yazılım özelliği olan sürümleri de vardır.

2.3.1.2. Ransomware

Fidye yazılımı (Ransomware), son yıllarda yaygın olarak görülen bulaştığı sistemi esir alarak, istenen ödeme yapılıncaya kadar sistemi şifreli şekilde bırakan zararlılardır. Kaynağı doğrulanmamış yada bilinmeyen ortamlardan indirilen dosya yada uygulamalarla bulaşan zararlı, bulaştığı platformlardan bağlı olduğu diğer alanlara da kendini kopyalama özelliğine de sahip olabilir.

2.3.1.3. Casus Yazılım

Bulaştığı sistem üzerinden kullanıcı etkinliklerini takip ve izleme amaçlı tasarlanan zararlıdır. Genel olarak keylogger olarak adlandırılan tuş bazlı veri toplama tekniğini kullanır.Sistemlerin güvenlik politikalarını da değiştirme yeteneğine de sahiptir.

2.3.1.4. RootKit

Bulaştığı sistemin yeniden yapılandırılmasına neden olan rootkit yazılımları, arka kapı (backdoor) oluşturarak saldırının bu noktalar üzerinden yapılmasını sağlar. Yetki yükseltme, işletim sistemi ayarlarını değiştirme gibi işlemleri yapar.

2.3.1.5. Bot

Çevrimiçi platformlar üzerinden sisteme bulaşan botlar, otomatik olarak birçok işlemi yapmak üzerine tasarlanmıştır. İlk bakışta zararsız gibi gözükse de saldırıların bulaştıkları sistemleri kullanarak yapılmasını sağlar. Botnet saldırıları sistemlere bulaşan botlarla yapılır. Günümüzde yaygın olarak görülen ve çoğu kullanıcının sistemine bulaştığından haberi olmadığı zararlı türüdür.

2.3.1.6. Virus

Her zararlının virüs olarak adlandırılarak yapılan hatanın sonucu olarak virüs kavramı günümüzde genel bir tanım olarak kullanılmaktadır. İlk planda zararsız olarak görülen program yada uygulamalar üzerine eklenir.Kritik sistemlerin bir ağa bağlandığı kendi içinde güvenli bir bilgisayar ağında çalıştığı sistemlerde dış kaynak verilerinin Universal Serial Bus (USB) bellek gibi ekipmanlarla virüslerin bulaştığı görülmektedir.

2.3.1.5. Truva Atı

Kullanıcı bazlı olarak atanan ayrıcalıklara sahip olmak için ilk planda zararsız gibi gözüken işlemler üzerinden bulaşan zararlıdır. Çevrimiçi olarak paylaşılan dosyalar üzerinde bulaşarak virüs gibi sistemlerde kalıcı hasarlar bırakır.

2.3.1.6. Korku Yazılımları (Scareware)

Korku yazılımları(scareware), işletim sistemlerine bulaşarak kullanıcıyı rahatsız ederek korku yoluyla ikna etme şeklinde kurgulanmış zararlılardır. Açılan bilgi pencereleri ile kullanıcının sahip olduğu ayrıcalıkları kendi eliyle paylaşması sağlanır. Sahte iletilerle kullanıcı bilgilerine erişim sağlanarak yazılımın sisteme yüklenmesi sağlanır.

2.3.1.7. Solucanlar (Worms)

Solucanlar(worm), kendi kendini çoğaltarak saldırı senoryaları geliştiren zararlılardır. Temel amaçları sistemleri gereksiz meşgul ederek yanıt verme sürelerini uzatmaktır. Böylece sistemin çalışması engellenecek, hızlı bir şekilde kendini sistem üzerinde çoğaltacaktır. Son yıllarda en yıkıcı etkileri bu yollarla verildiği görülmektedir. Bu zararlı algoritması sistemleri meşgul eden başka saldırı modellerine ilham kaynağı olmuştur.

2.3.2. Siber Saldırı Yöntemleri

Son yıllarda artan siber saldırıların temel amacı genel olarak hedef sisteme erişim sağlayarak bilgiye erişmek veya sistemde kalıcı hasarlar bırakmaktır. Bazen erişilen bilgiyi manipüle ederek, sistemi güvenilmez hale getiren bu saldırılar farklı yöntemlerle yapılmaktadır. Aktif saldırı adı verilen sistem kaynaklarına yönelik yapılan saldırılarda temel amaç verileri bozmaktır. Pasif saldırı olarak adı verilen bir diğer saldırı türünde ise amaç sadece bilgiye erişim ve bu bilgiyi kullanmaktır. Yaygın olarak yapılan siber saldırılar aşağıda açıklanmıştır.

2.3.2.1. APT (Advanced Persistent Threat)

Advanced Persistent Threat (APT), gelişmiş sürekli tehdit belirlenen hedefe yönelik yapılan gizli, farklı aşamalı ve uzun vadeli olan bir işlemdir. Kişisel verilere erişimden öte kurumsal kaynaklara ve ülkelere yönelik tehdit oluşturmak için kullanılır. Temel hedefleri arasında finansal saldırılar yoluyla kalıcı hasarlar bırakma da vardır. Bilgisayar ağları üzerinden yapılan bu saldırılarda hedefin geçici veya sürekli işlem dışı kalması, uzun vadede aşamalı olarak saldırı şiddetini arttıran senaryoları aktif etmektir.

2.3.2.2. DoS (Denial-of-Service Attack)

Denial-of-Service (DoS) saldırısı öncelikli olarak kullanıcı,cihaz ve uygulamalara yönelik yapılan ağ saldırı türüdür. Temel amaç bir ağ üzerinden hizmet veren yada çevrimiçi bir hizmeti yöneten sistemleri bu hizmeti vermesine engel olmaktır.Kaynaktan çıkan paketler doğrudan hedefe gönderilir. Genel olarak tek kaynak üzerinden tek hedef alınır. İki farklı türde yapılan DoS saldırılarında, hedef olarak alınan bilgisayar, bir servis, bir uygulama yada bilgisayar ağının hizmet veremeyecek düzeyde veri gönderilmesi şeklinde yapılır.Böylece hedefin hizmet vermesi engellenmiş olur. Diğer DoS saldırı türünde ise, zararlı içerik olarak paketlenmiş verileri hedefin işleyememesidir. Bu durumda ise paket kaynaklı hatalar yine sistemin çalışmasında yavaşlama ve sistemin çökmesine neden olacaktır. Yapılması için gelişmiş araç ve bilgi birikime ihtiyaç olmadığı için yapılması oldukça kolaydır.

2.3.2.3. DDoS (A Distributed Denial-of-Service)

A Distributed Denial-of-Service (DDoS) DoS'a benzer yönleri olmakla birlikte daha organize bir sistem tarafından yapılan saldırı türüdür. Öncelikle saldırı öncesinde BotNeT adı verilen, zombi adı verilen zararlıların bulaştığı bir bilgisayar ağı oluşturulur. Bu zararlı içeren hostlar sürekli olarak ağdaki üye sayısını arttırarak hizmet reddi saldırısının şiddetini arttırmak ister. İstenen düzeye ulaşan zombi host sayısı sonrasında ise saldırı aktif edilir. Küresel anlamda günümüzde sıkça rastlanan saldırı türüdür. Yaygın olarak görülen DDoS saldırı türleri aşağıdaki Çizelge 2.1'de belirtilmiştir.

Çizelge 2.1. DDoS Saldırı Türleri

Saldırı Seviyesi	**Saldırı Türü**	**İzlenen Yol**
7.Katman-Uygulama	DNS flooding	DNS çözümlemesini bozma.
7.Katman-Uygulama	HTTP flooding	Post-GeT kullanarak yapılır.
Protokol tabanlı	Smurf	ICMP-IP zafiyetleri ile yapılır
Protokol tabanlı	SYN flooding	Sürekli SYN gönderilerek yapılır.
Protokol tabanlı	Fraggle	Sahte UDP ile yapılır.
Hacim arttırma	ICMP flooding	ICMP paketleri ile yapılır.
Hacim arttırma	UDP flooding	UDP ile yapılır.
Hacim arttırma	Packet snoofing	Sahte IP adresleri üzerinden yapılır.

2.3.2.4. Harmanlanmış Saldırılar

Saldırganın birden fazla teknik kullanarak yapmış olduğu saldırı türüdür. Saldırgan kodlamış olduğu zararlı yazılım türüne bağlı olarak teknik geliştirir. Son kullanıcı başta olmak üzere veri güvenliğini tehlikeye atan bu saldırı türünde, sahte web siteleri, içeriği manipüle edilmiş e-posta iletileri kullanılmaktadır. Kimlik avı içeren bu zararlı içeriklerin infekte edilmesi için DDoS saldırıları da kullanılmaktadır.

2.3.2.5. Man-In-The-Middle (MiTM)

Ortadaki adam (Man-In-The-Middle) (MiTM) saldırısı ile sistem üzerinde kullanıcının bilgisi olmadan yetki alma şeklinde gerçekleşir. Saldırıda hedef kurbanın bilgileri karşı tarafa ulaşmadan önce saldırı gerçekleşme noktasına gelir. Bu bilgiler saklanabilir, manipüle edilebilir yada silinebilir. Özellikle bankacılık gibi finansal alanlarda sıkça görülen saldırı modelidir.

2.3.2.6. Man-In-The-Mobile (MiTMo)

Man-In-The-Mobile (MiTMo) mobil cihazlarına yönelik erişim yetkisi kazanma amaçlı yapılan zararlı türüdür. Ortadaki adamın algoritmasına benzer bir çıktı üretir. Özellikle ikili doğrulama seçeneklerine karşı bilgilerin aktarılması sağlar.

2.3.2.7. Sosyal Mühendislik (Social Engineering)

Günümüzde saldırı öncesinde hedef ile ilgili yapılan aktif ve pasif bilgi toplama yöntemlerinin başında sosyal mühendislik faaliyetleri gelmektedir. Kullanıcıların zayıf yönlerini tespit etme yoluyla bilgilerinin toplanır ve erişim alternatifleri yaratılır. Tailgating adı ile yapılan sosyal mühendislik türünde saldırgan yetkili kullanıcılar takip edilir. Pretexting türü sosyal mühendislik yönteminde veri doğrulama gibi yöntemlerle ayrıcalıklı verilere ulaşım sağlanır. Quid pro quo'da (bir iyilik için bir iyilik) ücretsiz olarak sağlandığı vurgulanan hediye yoluyla kullanıcı verilerine erişim istenir. Kablosuz ağlarda ve IoT ağlarında cihazlarda tanımlanan parolalara erişim için bu yöntemler tercih edilir. Öntanımlı kullanıcı adlarının silinmediği sistemlerde yada kullanıcı adının belirlendiği sistemlerde kaba kuvvet saldırıları ile oluşturulan sözlük şifreleri ile deneme yoluyla parolalara erişim sağlanmaya çalışılır.Bu işlem için

kullanılan açık kaynak araçlar uzun zaman aralıklarında tahmin yoluyla parolara erişim sağlayabilir.

2.3.2.8. Oltalama Saldırıları (Pishing Attacks)

Yaygın olarak görülen oltalama saldırılarında kurban olarak seçilen kullanıcılara yönelik sistematik bir senaryo ile hareket edilir. Saldırgan finansal verilere ulaşmak gibi temel motivasyonla hareket ettiği için kullanıcı parola ve şifreler gibi hassas bilgilere erişmek ister.

2.3.2.9. SQL Injection Saldırıları (SQL Injection Attacks)

Open Systems Interconnection (OSI) referans modeli uygulama katman seviyesinde gerçekleştirilen saldırı türüdür. Özellikle web tabanlı dinamik yapıların sahip olduğu verilere yönelik yapılır. Saldırılar Structured Query Language (SQL) sorgulama dili komut ve özellikleri ile geliştirilen tekniklerle gerçekleştirilir. Veritabanlarının içerdiği tablo ve tabloların içeriklerine doğrudan zarar verme amacı vardır. Web tabanlı uygulamalara eklenen kötü amaçlı sorgu komutları ile hassas verilere erişim sağlanır.2.3.2.10. U2R, R2L ve Probe Saldırıları (DNS Tunneling)

User to Root Attack (U2R) saldırganın temel amacı kullanıcı hesap erişim yetkileri üzerinden en yetkili kullanıcı bilgilerine erişmektir. Root to Local (R2L) saldırılarında uzak bağlantı ile yetkisi olmayan cihazlarda yetki elde ederek paket göndermek şekilde gerçekleşir. Probe saldırılarında bağlantı sağlanan ağdaki cihazlara ağ kartları üzerinden bilgi edinerek yapılır. Bu şekilde zafiyet taraması, güvenlik açıkları belirlenir.

2.3.2.11. DNS Tünelleme Saldırıları

Domain Name Server (DNS) IP adreslerini alan adına; alan adını IP adresine dönüştüren bir servistir. DNS kayıtları tutulan serviste talep edilen IP adresine karşılık gelen alan adı varsa yapılan sorguya yanıt verilerek web sitesine erişim sağlanmaktadır. Eğer istenen kayıt bulunamazsa kök sunucularına adres sorulur. Buradan Top Level Domain (TLD) sunucularına bilgi gönderilmesiyle işlem tamamlanır. Yapılan bu sorgu sırasında çalıştırılarak saldırı için kullanılan kötü amaçlı araçlar ile DNS tünelleme saldırıları yapılmaktadır.

2.4. SALDIRI TESPİT SİSTEMLERİ

Saldırı Tespit Sistemleri (STS), bir sunucu, ağ cihazı yada güvenlik duvarı gibi yapılarda ortaya çıkabilecek olası kötü amaçlı trafiği, bünyesindeki kural, veritabanı yada imzaları kullanarak tespit eden sistemlerdir. Bu şekilde olağandışı bir durum tespit edildiğinde STS bir uyarı oluşturarak kayıt düşer. Tespit sonrasında sistem harekete geçmez, engelleme yapamaz. Sadece raporlama, loglama ve algı uyarısı yapan bir görevi vardır. STS eğer ağda bir tarama yapıyorsa bu durum ağda geçikme süresi oluşturur. Bunu engellemek için STS'ler genel olarak çevrimdışı olarak yapılandırılır. Ağdaki veriler kopyalanarak eş zamanlı olarak STS'ye iletilir. Farklı işletim sistemlerinde çalışabilen açık kaynak birçok araç bu amaçla kullanılmaktadır. STS mimari ve işlev olarak değerlendirildiğinde aşağıda belirtilen görevleri vardır.

1. Ağdaki paket akışının sorunsuz şekilde sürekliliği sağlar.
2. Belirlenen kötü amaçlı paketleri işaretler.
3. Saldırı yada olağandışı durumda uyarır.
4. Olağandışı durumun belirlenen kaynağından gelen trafiği engeller.
5. DDoS vb. bir durum tespitinde bağlantının sıfırlanması sağlar.
6. CRC durumunda hatanın düzeltilmesini sağlar.
7. TCP'de segmentlerin sıralanmasını sağlar.

STS güvenlik yöntemleri olarak kategorize edildiğinde iki şekilde tasarlanmaktadır.

2.4.1. Host Tabanlı Saldırı Tespit Sistemi

Host Intrusion Detection Systems (HIDS), yapısında bulunan ajanlar ile özellikle işletim sistemini kontrol ederek, verileri loglar. HIDS genel olarak tüm ağı izleyemez. Bilgisayar ağ mimarisinde kritik bölgelerde bulunan sunucular üzerinde oluşan trafikteki olağandışı durumların tespiti için kullanılır. Birden fazla bilgisayardaki trafiğin izlenmesi bu şekilde zordur. Sistemde oluşan alarm neticesinde sonuç alınmadığı takdirde sistem devre dışı kalabilir.

2.4.2. Ağ Tabanlı Saldırı Tespit Sistemi

Network Intrusion Detection Systems (NIDS), iki farklı ağ cihazından oluşur. Farklı modlarda çalışan bir ethernet kartı (NIC) ile yönetilebilir bir ağ cihazına sahiptir. Bağlı olduğu ağdaki tüm trafiği izleyerek alt ağlarda analiz yapar. Tespit edilen anormal

durumlarda alarm oluşturarak bilinen saldırıları hızlı bir şekilde tespit eder. Sıfırıncı gün türündeki saldırın tespitinde başarı oranı düşüktür.

2.4.3. Tespit Yaklaşımına göre Saldırı Tespit Sistemleri

STS, algılama yaklaşımına göre 6 başlıkta incelenir.

2.4.3.1. İmza Tabanlı STS (Signature-based IDS)

Signature-Based IDS, zararlı faaliyetlerin "bilinen örüntülerini (imza)" aramaya dayanmaktadır. İmza tabanlı tespit yönteminde, her saldırı benzersiz tanımlanan bir imza ile sözlük (wordlist) oluşturularak kayıt edilir. Tespit edilen her yeni saldırı bu sözlükte saklanır. Böylece bilinen ve keşif edilen saldırılar üzerine bir savunma sistemi oluşturulur. Tek yapması gereken, bilinen saldırı imzalarının listesini aramak ve eğer bir eşleşme raporu bulduysa bunu kullanıcı/kuruma vb. bildirmektir. Sadece gördükleri ile önceden belirlenmiş bir kural arasında bir karşılaştırma yaptığı için hızlıdır. Olumsuz olarak ,yeni bir saldırı gerçekleştirildiğinde, kendi veritabanından herhangi bir örüntüyle eşleşmeyeceği için koruma yapamayacaktır. Saldırılar, mesajları bölerek kendisini kamufle edilebilir. Yeni bir saldırı kaydedildikten sonra, veri dosyalarının ağ güvenli hale gelmeden güncellenmesi gerekir.

2.4.3.2. Anomali Tespit Tabanlı STS (Anomaly-based IDS)

Anomali tespit yönteminde, ağdaki trafikten paketler alınarak, olağandışı bir durumun olup olmadığını değerlendirir. Olağandışı bir durum tespit edildiğinde saldırı önleme sistemi devreye girer. İmza tabanlı tespit sistemlerinin tespit edemediği saldırıları, anomali tespit tabanlı sistemler tespit edebilir. Tespit başarısını arttırmak için bu iki yaklaşımı birleştiren hibrit sistemler geliştirilmiştir. Hibrit saldırı tespit sistemleri kullanım alanlarına göre ikiye ayrılır: Birincisi anomali tespit tabanlı (sequence-based) hibrit saldırı tespit sistemi, ikincisi ise; paralel tabanlı saldırı tespit sistemidir. Anomali tabanlı STS, zararlı aktivitenin bilinmeyen benzersiz davranış modelini izlemeyi temel alır. Yeni bir saldırıya veya bilinmeyen tehditlere karşı koruma sağlar. Anormal ve potansiyel olarak zararlı davranışları tanımlamak için ağ trafiğine bakılır ve karşılaştırılır. Olumsuz olarak false positive çok fazla yakalar çünkü NIDS'ler, davranış kalıplarına dayalı bir sistemi izler. Doğru kişinin çok fazla kaynak veya kaynak kullanması, bir anormallik gibi tespit edilebilir ve imza haline gelebilir.

2.4.3.3. Desen Eşleştirme Tabanlı STS (Pattern Matching)

Desen Eşleştirme STS, tek bir paket içinde sabit bir bayt dizisi arar. Trafik incelemesini filtrelemek için, model genellikle belirli bir hizmet ve kaynak veya hedef bağlantı noktasıyla ilişkilendirilir. Bununla birlikte, birçok protokol ve saldırı iyi bilinen bağlantı noktalarından yararlanmaz ve bu nedenle Desen Eşleştirme bu tür saldırıları tespit etmede zorluk çeker. Ayrıca eşleşme, benzersiz olmayan bir patterne dayanıyorsa, çok sayıda false-positive sonuç ortaya çıkabilir.

2.4.3.4. Durum Bilgili Model Eşleştirme Tabanlı STS (Stateful Pattern Matching)

Durum bilgili model eşleştirme STS analizini tek bir pakete dayandırmaktan ziyade kurulan oturumun bağlamını hesaba kattığı için biraz daha karmaşık bir yaklaşım sunar. Bir akış içinde birkaç paket arasında dağıtılabilen stringler arayarak şablon eşlemesini yapar. Örnek verecek olursak paket içerisinde "za" stringi algılanırsa ve bir sonraki pakette "rarlı" algılanırsa, bu durum için alarmı tetikleyerek, desen eşleştirme iyileştirilebilir. Bununla birlikte çok fazla bellek ve sistem kaynağı kullanması sebebiyle desen eşleştirmeye göre yavaştır. Desen eşleştirme kadar olmasa da çok sayıda yanlış-pozitif sonuç üretebilir.

2.4.3.5. Protokol Kod Çözme Tabanlı STS (Protocol Decode-Based)

Protokol kod çözme tabanlı STS, desen eşleştirme'nin akıllı bir uzantısı olarak düşünebilirsiniz. Bu tür bir imza ile STS, Request For Comments (RFC) tarafından tanımlanan protokol ihlallerini arar ve belirli bir alan için pattern eşleşmelerini içerebilir. Bu yöntem, iyi tanımlanmış protokoller için false-positive azaltmada etkili olmasına rağmen, protokol belirsiz veya gevşek tanımlanmış ise STS tarafından kolayca gözden kaçırılır.

2.4.3.6. Sezgisel Tabanlı Analiz STS (Heuristic-Based Analysis)

Sezgisel tabanlı analiz yapan STS bir alarmın tetiklenip tetiklenmeyeceğini belirlemek için algoritma kullanır. Bu tür bir analizin ve uyarının bir örneği, belirli bir ana makinede eşik değerlerin eşik sayısı taranırsa bir alarm veren bir imza olacaktır. İmza ayrıca, örneğin bir çevre yönlendiricisi gibi belirli bir kaynaktan gelen Synchronize (SYN) paketleri ile sınırlandırılabilir. Sezgisel tabanlı imzalar, belirli saldırı türlerini tespit etmenin tek yolu olsa da, benzersiz ağ ortamlarına daha iyi uyum sağlamaları

için ayarlama ve modifikasyon gerektirir. Bununla birlikte belleği CPU'yu ve sistem kaynağını çok fazla kullanır.

2.5. SALDIRI ÖNLEM SİSTEMLERİ

Saldırı Önleme Sistemleri (SÖS), STS aksine imza eşleştirme ve pozitif kural bazlı trafiği reddetme ve engelleme özelliğine sahiptir. Gerçek zamanlı port ve trafik analizi yapabilir. İçerik arama ve eşleştirme özelliği vardır. Periyodik olarak raporlama ve analiz yaparak sistem üzerine entegre çalışan diğer güvenlik mimarileri ile eş zamanlı çalışabilir. Genel olarak saldırı önleme sistemleri dört sınıfa ayrılır.

1. Network-Based Intrusion Prevention (NIPS): Tüm ağdaki şüpheli durumları tespit etmek için protokol analizi yöntemini kullanarak izler.
2. Wireless Intrusion Prevention Systems (WIPS): Kablosuz ağ üzerindeki şüpheli durumları tespit etmek için kablosuz ağ protokolü ile izler.
3. Network Behavior Analysis (NBA): Ağ trafiğindeki davranışların analizinde kullanılır.
4. Host-Based Intrusion Prevention (HIPS): Ana bilgisayara yönelik saldırıları önlemek için kullanılır.

2.6. GÜVENLİK DUVARLARI

Bilgisayar ağ mimarisinde kullanılan bu kavram, ağın belli bir bölümünde olası saldırı yada olağandışı bir durumu ağın diğer alanlarına yayılmasını önlemek için tasarlanmış bir mimaridir. Ağda izin verilen iletişim protokolleri, aygıt bazlı kontrol listeleri, filitre ve denetim listeleri güvenlik duvarları ile sağlanır. Güvenlik duvarı bağımsız şekilde bir donanım olabildiği gibi bir iş istasyonu yada istemciye de kurulabilir. Ağ tabanlı bir güvenlik duvarı kullanılacaksa tamamen bağımsız bir cihazda mimaride aktif olarak kullanılır. Genişleyen ağ ve artan kullanıcı sayıları tehdit vektörlerinin artmasına ve çeşitlenmesine neden olmuştur. Bu sebeple farklı türde güvenlik duvarları yapılandırılarak güvenliğin sağlanması amaçlanmaktadır. Ana bilgisayar tabanlı Güvenlik Duvarı, yapılandırılan tek bir bilgisayar ile sistemin gelen paket ve talepleri filitrelemesi ile oluşturan sistemdir. Ağ Katmanı Güvenlik Duvarı, trafik oluşturan hedef ve kaynak IP adreslerine göre yapılan filitrelemedir. Proxy Sunucu ile ağda trafik oluşturan medya, Uniform Resource Locator (URL) tabanlı isteklerin filitrelenmesidir. Ağ Adresi Çevirisi (NAT) Güvenlik Duvarı, iç ağdaki IP adreslerini

maskeleyerek gizlenmesini sağlar. Aktarım Katmanı Güvenlik Duvarı, hedef ve kaynak arasındaki bağlantı noktaları, portlara göre filitreleme yapan sistemdir. Ters Proxy sunucuları, genel olarak web sunucularının önüne yapılandırılır. Böylece web ataklarına karşı sunucunun üzerine düşen olası yükü azaltır. Uygulama Katmanı Güvenlik Duvarı, hizmet, uygulamalara göre filitreleme yapan sistemlerdir. Bağlam Duyarlı Uygulama Güvenlik Duvarı, uygulama türleri, cihaz, kullanıcı profili ve tehdit unsurlarına göre filitreleme yapan sistemlerdir.

2.7. SALDIRILARIN GERÇEK ZAMANLI OLARAK TESPİT EDİLMESİ

Hızla genişleyen bilgisayar ağları, gerçek zamanlı yapılan saldırılarında artmasına neden olmuştur. Özelikle işletim sistemi ve yaygın olarak kullanılan birçok uygulama saldırganların hedefi haline gelmektedir. Yazılım mimarisi yada kod yapısından kaynaklı hataların düzeltilmesinden önce gerçekleşen saldırılara sıfırıncı gün (ZeroDay) saldırısı adı verilmektedir. Bu tür saldırıların tespitinin zor olması, kurumsal ağlara yönelik riskleri arttırmaktadır. Gerçek zamanlı olarak tespit edilmesinde ortaya çıkacak olan zafiyet kurumlar uğradıkları sıfırıncı gün saldırılarına uzun zaman sonunda tespit edebilmektedir. Bu sebeple saldırıların anlık ve hızlı bir şekilde ağın aktif olduğu zaman aralığında tespiti büyük öneme sahiptir. DDoS saldırılarına karşı anlık tespitin verilmesi ve önlemin alınması oldukça zordur. Yüksek sayıda saldırganı barındıran botnet ağları, başlangıçta olağan trafik gibi görünse de düzenli olarak yapılan saldırılar neticesinde birçok sunucu ve ağın kullanılmasını engellemektedir. Geçici bile olsa hizmet vermeyen sunucu ve hizmetler kurumlar için maddi ve prestij kayıplarına sebep olmaktadır. Gerçek zamanlı bu tür saldırılara yanıt vermek bu nedenle önemlidir. Uç noktadan gerçek zamanlı algılamada, Saldırı Tespit Sistemi (STS) ve Saldırı Önlem Sistemi (SÖS) aktif olarak kullanılmalıdır. Saldırıların tespit edilerek anlık olarak önlem alınmasında güvenlik duvarlarının yükünü de azaltmaktadır. Birbiri ile eş zamanlı çalışabilen bu sistemler zararlı yazılımlarının tespiti için de kullanılmaktadır. Ayrıca saldırı türüne bağlı olarak bilgisayar ağında ortaya çıkan anomali trafiklerinin tespiti içinde STS'ler kullanılmaktadır.

2.6. YAPAY ZEKA

Yapay zeka kavramı 1900'lü yılların ortasında ilk olarak Alan Mathison Turing ve John McCarthy tarafından atılmıştır. Claude Elwood Shannon ve Alan Turing II.Dünya Savaşı'nda geliştirdikleri makine şifreleme, problem çözme süreçlerine büyük katkı sağlamıştır. 1957 yılında Frank Rosenblatt tarafından "Perceptron" kavramı ortaya atılmıştır. Perceptron en basit tek katmanlı sinir ağı modeli olarak tanımlanır. Eğitilebilecek yapay sinir hücresine sahiptir. 1959 yılında Marvin Lee Mainsky tarafından ilk yapay zeka laboratuvarı kurulmuştur. 1969 yılında Papert ve Minsky tarafından tek katmanlı olarak tasarlanan topolojilerin XOR problemlerini çözemeceğini gösterildi. 1986 yılında çok katmanlı perceptron yapısı Geoffrey Hinton, David Rumelhart ve Ronald Williams tarafından önerilmiştir. Bu gelişmeler büyük veri (big data) kavramının önemini arttırmıştır. Destek Vektör Makineleri (SVM) 1995 yılında Vladimir Vapnik ve Corinna Cortes tarafından literatüre kazandırılmıştır. Bu yöntem "Yapay Öğrenme (Makine Öğrenmesi)" kavramını ortaya çıkarmıştır. 1998 yılında gradyan temelli bir teori Yann LeCun tarafından Evrişimsel Sinir Ağları (CNN) sunulmuştur. Aslında bu yaklaşım "Makinelere Öğretme" fikrinin başlamasına neden olmuştur. 2010 yılına kadar örüntü, nesne tanıma, sınıflandırma, görüntü işleme alanlarındaki yaklaşımlar donanımlara ek yükler getirmiştir. Merkezi işlem birimi (CPU) yerine grafik tabanlı işlemcilerin (GPU) bu donanımsal yükleri alması bu alandaki gelişimi hızlandırmıştır. 2012 yılında Geoofrey Hinton, Ilya Sutskever ve Alex Krizhevsky tarafından AlexNet oluşturulan CNN modeli ile birçok soruna çözüm üretmiştir. Res-Net ve Inception mimarileri bu alana "Derin Öğrenme" kavramını katmıştır. Artık birden fazla katman içeren CNN ağları, boyutlandırma işlemleri, optimizasyon,filitreler, performans parametreleri kullanılmaya başlanmıştır. 2014 yılında Goodfellow tarafından "Çekişmeli Üretici Ağlar" çalışması günümüz yapay zeka teknolojilerine yön vermiştir. Yapay Zeka günümüzün bilgi iletişim teknolojileri başta olmak üzere birçok alanın temel tanımını oluşturmaktadır. Bilgisayar ve makinelerin düşünme, bilişsel süreçlerde rol alması ve karar verebilmesi noktasında insan gibi davranma yeteneğine sahip olması şeklinde tanımlanabilir. Bunun sonucu olarak anlama ve öğrenme faaliyetleri ile bilgi işlem süreçlerinin yapay zeka teknolojileri ile yönetilmesi söz konusu olmaktadır. Başlangıçta düşünce üretim kapasiteni arttırma ve insan gibi yüksek doğrulukla karar vermeye yardımcı mekanizmalar üretilmesi amacı, günümüzde birçok alanda yapay zeka kavramının

kullanılmasını sağlamıştır. Mevcut gelişim aşaması ve kullanım alanları göz önüne alındığında yapay zeka üç alanda incelenmektedir.

Yapay Dar Zeka

Sınırlı ve belirgin şekilde belirtilen görevleri yerine getiren sistemlerin tanımlandığı alandır. Otonom araçlar, görüntü işleme, akıllı cihazlar üzerinde kullanılan oyun tabanlı uygulamalarda yaygın olarak yapay dar zeka örnekleri görülmektedir.

Yapay Genel Zeka

İnsan gibi düşünen, insanın sahip olduğu yeteneklere sahip olan makinelerin kendi başına karar verme süreçlerinde yer aldığı yapay zeka alanıdır. Günümüzde bu alandaki çalışmaların giderek somutlaştırıldığı özellikle doğal dil işleme gibi alanlarda örneklenen çalışmalar görülmektedir. Yapay Sinir Ağları bu noktada insanın sahip olduğu sinir sistemini model alan bir model ortaya koymaktadır.

Yapay Süper Zeka

İnsan zekanın makinelerdeki zeka kavramına katması istenen temel özelliklerin başında yargıda bulunma özelliği gelmektedir. Öğrenme, sistematik olarak ilişki kurma gibi insana özgü olan özelliklerin makineler tarafından yapılabilmesi ilkesine dayanır. Alanda yapılan çalışmalar gelecekte insan zekasına ulaşması amaçlanan araştırmaların yapıldığını göstermektedir.

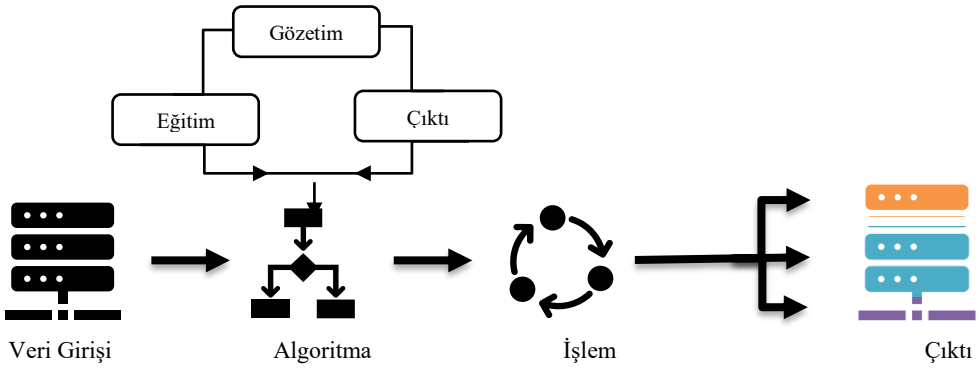
Şekil 2.2. Yapay Zeka Kapsamı

2.6.1. Makine Öğrenmesi

Makine öğrenmesi, yapay zekanın bir alt kümesi olmasının yanında büyük veri (big data) ile değerli hale gelen veri kavramı sonucu anlamlı hale gelmiştir. Büyük veri, üretilen, depolanan ve ihtiyaç duyulan verinin artması sonucu ortaya çıkmıştır. Makine öğrenmesi, dışarıdan bir komut yada müdahale olmazsızın önceden tanımlanmış bir görevin tamamlanmasında kullanılan modeldir. İstatistiki bazı teorem ve kabullere dayanır. Karar vermek için öngörü oluşturarak istatistiki verilere göre hareket eder. Büyük veri içinden değerli olarak kabul edilen verilerin alınması işlemi "veri madenciliği" olarak tanımlanmaktadır. Bu yöntem incelendiğinde makine öğrenmesi altındaki bir çalışma alanı olduğu görülür. Son yıllarda makine öğrenmesi yaklaşımları kullanılarak görüntü işleme, siber güvenlik, doğal dil işleme gibi alanlarda çalışmalar yapılmaktadır. Yapısal olarak bakıldığında makine öğrenmesi bulanık mantık, yapay sinir ağları ve derin öğrenme şeklinde alt dallara ayrılmaktadır. Makine öğrenmesi denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli (takviyeli) öğrenme şeklinde sınıflandırılır.

2.6.1.1. Denetimli Öğrenme

Denetimli Öğrenme ile sisteme girilen veriler, önceki çıktılar ile karşılaştırılır. Öğrenme bu şekilde gerçekleşir.Her girilen veri aslında giderek öğrenilmiş verinin sonucu olur.

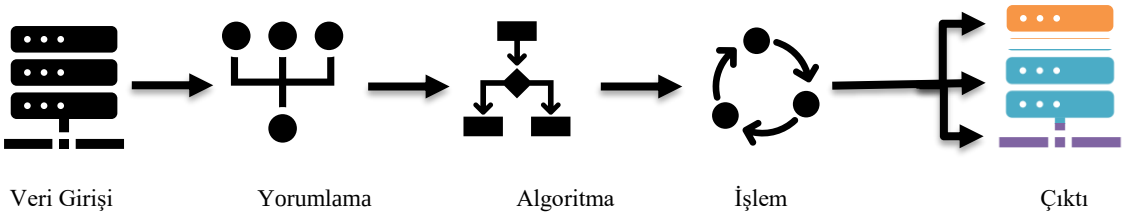


Şekil 2.3. Denetimli Öğrenme

2.6.1.2. Denetimsiz Öğrenme

Denetimsiz Öğrenmede çıktı olarak alınan verileri ile bağlantı kesilerek, giriş verileri dikkate alınır. Bunun amacı giriş verilerinin birbiri ile karşılaştırılarak aralarında

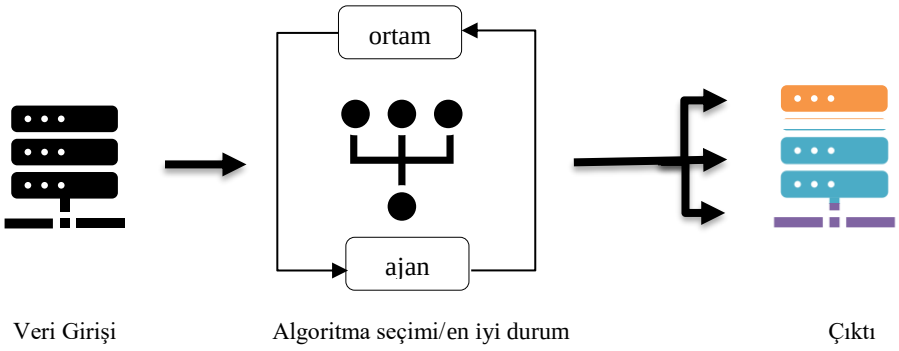
anlamlı bir ilişki kurma çabasıdır. Böylece birbirine yakın verilerin kümelenmesi sağlanır.



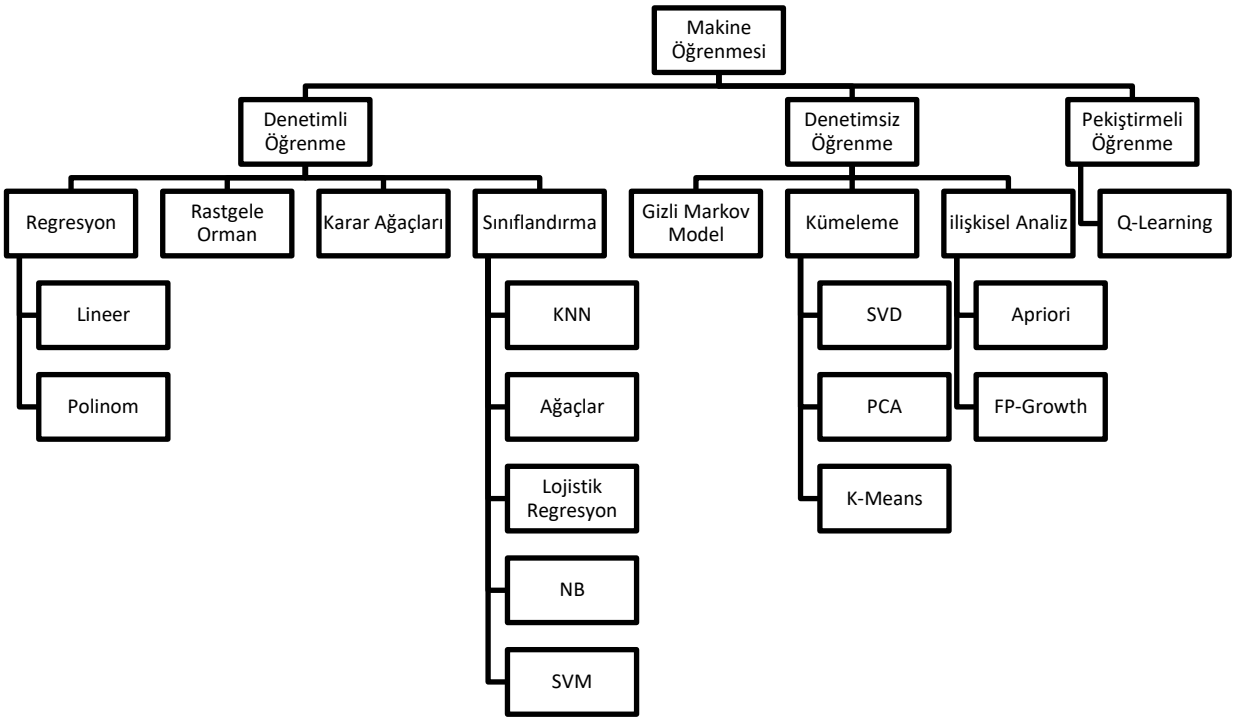
Şekil 2.4. Denetimsiz Öğrenme

2.6.1.3. Pekiştirmeli (Takviyeli) Öğrenme

Pekiştirmeli (Takviyeli) Öğrenmede, elde edilen sonuç verilerinin giriş verileri ile karşılaştırılıp, doğru-yanlış, iyi-kötü şeklinde ölçüt elde edilir. Q-Learning Algoritması bilinen en iyi pekiştirmeli öğrenme algoritmasıdır.



Şekil 2.5. Pekiştirmeli(Takviyeli) Öğrenme



Şekil 2.6. Makine Öğrenmesi Algoritmaları

2.6.1.4. Topluluk Öğrenme

Literatürdeki çalışmalar incelendiğinde makine öğrenmesi algoritmalarının yanında kullanılan tekli sınıflandırma modellerinde yüksek performans elde edildiği görülmektedir (Bilgin 2018; Subasi 2020). Topluluk öğrenme bu mantıkla ortaya çıkmıştır. Tasarlanan modellerin birlikte yüksek doğruluklu kararlar vermesi amaçlanır. Modelin eğitilmesi için birden fazla öğrenme alternatifleri kullanılır. Bu yöntem sınıflandırma ve regresyon problemlerinde kullanılabilir. Torbalama, Güçlendirme ve Yığınlama olarak kullanılan üç topluluk öğrenme yöntemi vardır.

2.6.1.4.1. Torbalama (Bagging)

Breiman tarafından geliştirilmiştir. Regresyon ve sınıflandırma yaklaşımları ile doğruluğun arttırılması şeklinde bir yöntem kullanılır. Torbalama yönteminde veri yığınlarıyla rastgele modeller oluşturularak aynı anda birden fazla eğitim gerçekleştirilir. RF torbalama yöntemine örnek bir algoritmadır. RF sonucunda oylama bir araya getirilerek orman oluşturulur.

2.6.1.4.2. Güçlendirme (Boosting)

Sıralı olarak eğitim şeklinde yapılan güçlendirme yönteminde eğitim seti rastgele seçilir. Her modelin bir önceki model üzerinden iyileştirilmesi ile doğru yanlış sınıflandırma örneklerinin ağırlık değerleri değiştirilir. Buradaki amaç zayıf bir öğrenici durumunu bu şekilde güçlü hale getirmektir.Sıkça kullanılan güçlendirme yöntemleri Gradient Boosting, CatBoost ve AdaBoost (Adaptive Boosting) örnek verilebilir.

2.6.1.4.3. Yığınlama (Stacking)

Farklı algoritmalarla elde edilen modeller sonucu yapılan tahminlerin birleştirilmesi yoluyla yapılan yöntemdir. İki kısımdan oluşur. İlk bölüm veri yığınının farklı modellerin eğitim sonuçlarını ikinci kısım ise veri yığını ile birleştirerek genel öğrenme için kullanılır.Böylece tek bir çatı model oluşturulur.

2.6.2. Derin Öğrenme

Derin öğrenme, makine öğrenmesinin bir alt alanı olarak ele alınmaktadır. Yapısal olarak tahmin eden birden fazla katman içerir. 1990'lı yıllarda yapay zeka alanında yapılan çalışmaların sınırlı kalmasının birçok nedeni vardır. Donanımsal yetersizlikler ve yetersiz veri yığınları bu gelişimi olumsuz etkilemiştir. Derin öğrenme makine öğrenmesinden olduğu gibi denetimli, denetimsiz yada pekiştirmeli olarak tasarlanabilir. Farkı veri girişlerine göre tekil özellikleri kendi kendine öğrenir. Girdi sayısının yüksek olması öğrenme başarısını olumlu yönde etkiler. Girdiler birden fazla katmandan geçerek, öğrenme işlemi yapılır. Genel olarak üç derin öğrenme modeli vardır.

1. Çok Katmanlı Algılayıcılar (MLP)
2. Konvolüsyonel Sinir Ağları (CNN)
3. Tekrarlayan Sinir Ağları (RNN)

Günümüzde her geçen gün artan veri yığınları, araştırma alanlarında verisetlerinin çeşitlenmesine neden olmuştur. Sağlıktan, finansa, güvenlik sistemlerinden otonom teknolojilere kadar oluşturulan verisetleri teknolojilerin gelişmesine katkı

sağlamaktadır. Derin öğrenme yaklaşımlarının kullanıldığı alanlar aşağıda belirtilmiştir.

1. Görüntü işleme ve iyileştirme araştırmaları
2. Sağlıkta anomali araştırmaları
3. Ses ve yüz tanıma teknolojileri
4. Otonom sistem araştırmaları
5. Tavsiye sistemleri
6. Siber Güvenlik analiz ve tehdit algılama sistemleri

Sinir ağı mimarisi olarak tanımlanan model, yinelenen ve evrişimsel sinir ağlarını kapsar. Sinir ağı aşağıdaki gibi formulize edilir.

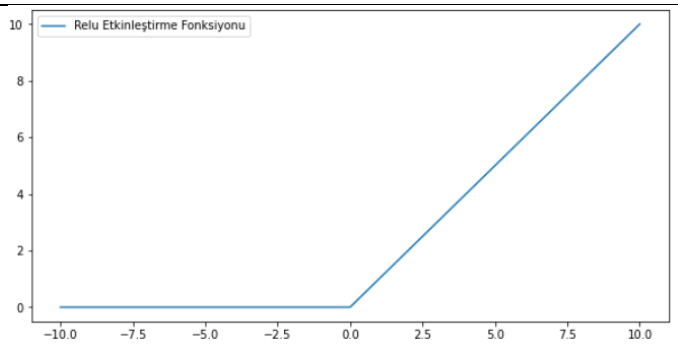
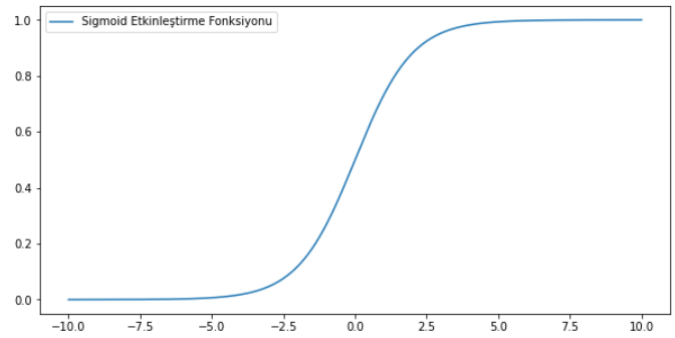
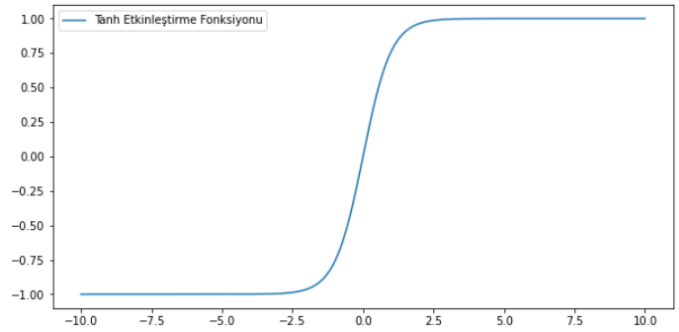
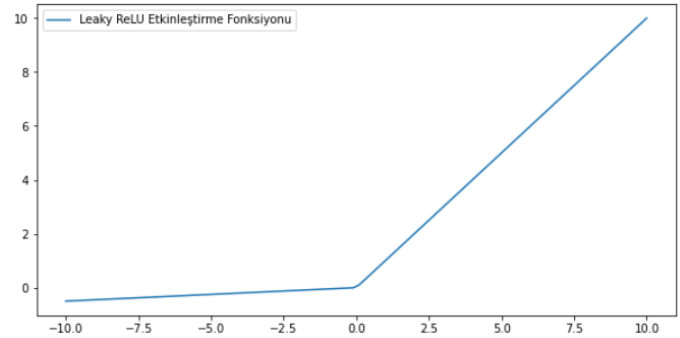
$$z_j^{[i]} = w_j^{[i]^T} x + a_j^{[i]} \tag{2.1}$$

z=çıktı, w=ağırlık, a=eğilim

Çizelge 2.2. Aktivasyon Fonksnksiyon ve Formülleri

Etkileştirme Fonksiyonu	**Formülasyon**
ReLU	$g(z) = \frac{1}{1 + e^{-z}}$ (2.2)
Sigmoid	$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{z}}$ (2.3)
Tanh	$g(z) = \max(0, z)$ (2.4)
Leaky ReLU	$g(z) = \max(\in z, z) \in \ll 1$ (2.5)

Çizelge 2.3. Aktivasyon Fonksiyon Kod ve Grafikleri

Etkinleştirme Kod Örneği	**Fonksiyon Grafiği**
x = np.linspace(-10, 10, 1000) y = np.maximum(0, x) plt.figure(figsize=(10, 5)) plt.plot(x, y) plt.legend(['Relu']) plt.show()	
x = np.linspace(-10, 10, 1000) y = 1 / (1 + np.exp(-x)) plt.figure(figsize=(10, 5)) plt.plot(x, y) plt.legend(['sigmoid function']) plt.show()	
x = np.linspace(-10, 10, 1000) y = (2 / (1 + np.exp(-2*x))) - 1 plt.figure(figsize=(10, 5)) plt.plot(x, y) plt.legend(['Tanh Etkinleştirme Fonksiyonu’]) plt.show()	
def leaky_ReLU(x): data = [max(0.05*value,value) for value in x] return np.array(data, dtype=float) x_data = np.linspace(-10,10,100) y_data = leaky_ReLU(x_data) plt.plot(x_data, y_data) plt.title('Leaky ReLU Etkinleştirme Fonksiyonu') plt.show()	

Derin öğrenme modellerinde, gizli katmanların sonunda modele lineer olmayan karmaşıklık eklemek için etkinleştirme fonksiyonları kullanılır. Yaygın olarak

kullanılan etkinleştirme fonksiyonları aşağıdaki Çizelge 2.2'de verilmiştir. Entropi genel olarak sistemler üzerindeki düzensizliğin tanımlanması için kullanılır. Rastgelelik ve değişkenler için dağılımların belirtilmesi için tanımlanır. Çapraz entropi ise iki olasılık dağılımı arasındaki farkın belirtilmesidir. Sinir ağı içinde yaygın olarak kullanılan çapraz entropide bir olasılığın dağılımı x, diğer dağılım y olarak kabul edildiğinde aşağıdaki gibi formulize edilir. Elde edilen sonuç metrik bir değer değildir. L(x,y) ile L(y,x) aynı olmadığı durumlar olabilir.

$$L(x, y) = -\sum_{i} p(a_i) * \log(y(a_i)) \tag{2.6}$$

Çapraz entropi çok sınıflı problemlerde hata parametresi olarak kullanılır. Performansı 0-1 arasında olasılık değeri olarak ölçer. Çapraz entropi kaybı arttıkça tahmin değeri düşer. İdeal bir sistemin kaybı 0'dır. Epoch, verilerin model tarafından bir defa işlenmesine Epoch denir. Öğrenme oranı modeldeki ağırlıkların hangi aralıklarda güncellendiğini gösterir. Öğrenme oranı eğitimden önce belirlenmesi gereken bir parametredir. Öğrenme oranı eğer yüksek seçilirse minimum noktalar atlanacağından optimum değere ulaşılamaz. Öğrenme oranı düşük seçilirse eğitim süresi otomatik olarak uzar ve epoch sayısı da artar. Seyreltme (dropout) değeri tam bağlı katmanlarda belli eşik değeri altında kalan düğümlerin gözardı edilmesini sağlar. Bu şekilde öğrenme hızlanır ve zayıf bilgilerin gereksiz yere hafızada tutulmasını önlenir. Sinir ağındaki ağırlıkları güncellemek için geri yayılım kullanılır.

Evrişimsel Sinir Ağları

Girdi verisi üzerinden sınıflandırma, benzerlik, kümeleme ve nesne bazlı tanıma işlevlerine sahip derin sinir ağıdır. Alınan girdiler matris formatında olmalıdır. Bununla birlikte bir filitre matrisi alır. Giriş matrisi ile birlikte çekirdek matrisi değerleri kullanılarak çıkış matrisi oluşturulur. Bu işleme konvolüsyon denir. aşamadan sonra aktivasyon fonksiyonları kullanılır. Sonraki aşamada daha küçük matrisler elde etmek için pooling yapılır.

Yinelenen Sinir Ağları

Düğümler arasındaki iletim bağlantılarının bir döngü oluşturularak model oluşturulur. Bu şekilde dinamik bir yapı oluşur. Ardışık bilgileri kullanmak için tercih edilir.

Katmanlarda önceki giriş ve durumlar üzerinden ilişkilendirme yapılır. Yapısal olarak bir döngüye sahiptir. Çok sayıda sinir ağı kopyalanması şeklinde bir model şeklindedir. Her bir sinir ağı bir sonrakine bilgi aktaracak şekilde tasarlanır. Uzun kısa vadeli hafıza (LSTM) kısa ve uzun zaman aralıklarında hafıza özelliğine sahiptir. Sürekli olarak önceki bilgiler üzerinden yeni çıkış verileri üretir.

2.7. SALDIRI TESPİT SİSTEMİNDE ÖZNİTELİK SEÇİMİ

Ağ tabanlı saldırıların tespitinde veri yığınları içinden elde edilen verilerin anlamlı hale getirmek karmaşık bir süreçtir. Büyük veri yığınlarının üzerinden daha anlamlı hale getirilen verisetleri üzerinden çıkarımda bulunmak daha kolay bir işlemdir. Bunun için makine öğrenmesi,veri madenciliği ve derin öğrenme yöntemleri tercih edilmektedir. Verisetleri içinde bulunan öznitelikler çoğu zaman gereksiz ve alakasız veriler içerir. Bu durum algoritma ve yaklaşımların performanslarını olumsuz etkiler. Veri boyutunun azaltılması farklı yöntemlerle yapılır. Öznitelik seçimi ve öznitelik dönüşüm ile ilgili ve ilgisiz öznitelikler birbirinden ayrılır. Öznitelik seçimi, veriseti içeriğinin daha verimli kullanmak ve analiz sırasında gereksiz performans kayıplarını önlemek için yapılır. Gereksiz öznitelikler verisetinden kaldırılır. Öznitelik dönüştürme işleminde ise özniteliklerin yapısında bozulmalar olmaktadır. Boyutsal sorunları ortadan kaldırmak için yapılan bu işlem sonrasında sonucu etkileyen öznitelikler kayıp edilebilir. Kullanılan algoritmaların sınıflandırma, tahmin ve seçim gibi performansları doğrudan etkileyen öznitelik seçim işlemi, geliştirilen modeller test edilmesinde veriseti içinde daha anlamlı veriler ile işlem yapılmasını sağlar. Tasarlanan modelin eğitim ve öğrenme süresini etkileyen bu işlemler algoritmaların aşırı öğrenmesine de engel olmaktadır.

BÖLÜM 3

VERİ SETLERİ VE DENEYSEL ÇALIŞMA

Saldırıların tespit edilmesi için tasarlanan modellerde açık kaynaklı veri kümeleri kullanılmaktadır. Farklı saldırı tipleri, farklı platformlar, farklı veri formatlarında kullanılan bu verisetleri eğitim seti olarak kullanılarak geliştirilen modeller test edilmektedir. Siber saldırı tespit sistemleri alanında yapılan araştırmalarda KDDCup99, NSL-KDD, DARPA, CIC-IDS2017/2018, ISCX2012, UNSW-NB15, KDDCUP 1999 verisetleri kullanılmaktadır. Bu verisetleri dışında alanda ticari olarak faaliyet gösteren firmalarında kullandığı diğer verisetleri de vardır. Verisetinin içerdiği saldırı tiplerinin güncel olması, verisetindeki öznitelik, veriseti boyutu gibi özelllikler verisetinin kullanılabilirliğini etkilemektedir. Bu çalışmada kullanılan NSL-KDD ve CIC-IDS2018 verisetleri makine öğrenimi ve derin öğrenme algoritmaları ile geliştirilen IDS yaklaşımlarında yaygın olarak kullanılmaktadır. Aşağıdaki bu verisetleri içerik ve özellikleri gösterilmiştir.

3.1. DENEYSEL ÇALIŞMADA KULLANILAN VERİSETLERİ

2009 yılında KDD Cup99 veriseti üzerinden tasarlanmıştır. Gereksiz kayıtlar silinerek, test ve eğitim setleri gereksiz tekrarları içermez. NSL-KDD verisetinde bulunan öznitelikleri 4 farklı kategoride değerlendirebiliriz. 1-9 nolu öznitelikler temel ağ bağlantı özelliklerini; 10-22 nolu öznitelikler trafik özelliklerini; 23-31 nolu öznitelikler zamana bağlı olarak oluşan trafik özellikleri; 32-41 nolu öznitelikler bilgisayar tabanlı trafik özellikleri şeklindedir. İçerdiği saldırı türleri U2L, DoS, R2L ve Probe şeklinde kategorilere ayrılmıştır. NSL-KDD verilerinin farklı format ve platformlarda kolay ve hızlı bir şekilde kullanıldığı için saldırı tespit sistemleri için geliştirilen modellerin testleri için sıkça kullanılmaktadır. Çalışmada kullanılan veriseti içeriğinde 77054 normal; 54487 DoS; 14077 Probe; 3880 R2L ve 119 U2R bulunmaktadır. Çizelge 3.1'de çalışmada kullanılan eğitim ve test veri sayıları gösterilmiştir. Literatür taramasında incelenen çalışmalarda %80 eğitim; %20 test

verisi olarak kullanıldığında elde edilen başarım oranlarının yüksek olduğu görülmüştür. Çalışmada bu oranlar kullanılmıştır.

Çizelge 3.1. NSL-KDD veriseti farklı saldırı davranışlarının dağılımı

NSL-KDD	**Veriseti içindeki Saldırı Tipleri**					**Toplam**
	Normal	**DoS**	**Probe**	**R2L**	**U2R**	
KDDTest+	15408	10677	2815	776	23	29999
KDDTrain+	61646	42408	11262	3106	96	118518
Toplam	77054	53385	14077	3882	119	148517

Veriseti Çizelge 3.2'de görüldüğü gibi 24 farklı ağ saldırısı ile 4 kategoride gösterilmiştir. DoS, ağa erişimi engellemek için yapılan saldırılara verilen addır. Probe (araştırma), hedefteki güvenlik açıklarının tespiti için yapılan IP ve portların taranmasıdır. Remote to Local (R2L)'de saldırgan oturum açma yetkisine sahip değildir fakat hedefe paket gönderebilir. User to Root (U2R)'da saldırgan erişim sağlamak için parola girişlerini izler. Standart kullanıcı modunda erişim hakkı olsa bile yetkili kullanıcı ile erişim hakkı elde etmeye çalışır.

Çizelge 3.2. NSL-KDD Saldırı Tipleri

	Saldırı Tipleri			
	U2R	**Probe**	**Dos**	**R2L**
Eğitim Saldırı Tipi	buffer_overflow loadmodule rootkit perl,	Ipsweep Nmap Portsweep Satan	Back Land Neptune Pod, Smurf Teardrop	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy,
Test Saldırı	buffer_overflow Loadmodule Rootkit Perl Sqlattack Xterm	Satan Ipsweep Nmap Portsweep Mscan Saint	Back, Land, Neptune, Pod Smurf, Teardrop, Apache2, Udpstorm,	Guess_Password, Ftp_write,Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop,

Tipi	Ps		Processtable, Worm	Snmpguess, Snmp Getattack,HttptnnelSendmail, Named

Veri kümesinde bulunan 41 öznitelikler tek tek değerlendirildiği gibi saldırı türlerine göre de 4 başlıkta değerlendirilebilir. Bunlar TCP bağlantı özelliklerine göre yapılan saldırılar, zaman etiketli iki saniyelik hesaplanan saldırılar, iki saniyeden fazla süren saldırılar ve içerik bilgisine göre yapılan saldırı özniteliklerdir [17,34]. Kanada Siber Güvenlik Enstitüsü tarafından oluşturulan en güncel verisetlerinden biridir. Gereksiz tekrar içeren öznitelik sayısı çok düşüktür. Belirsiz olarak nitelendirilen öznitelik sayısı minimize edilmiştir. Csv formatında hazır olarak hizmete sunulması büyük avantaj sağlamaktadır. CIC-IDS2018 verisetinde bulunan öznitelikleri 8 farklı kategoride değerlendirilir. Çizelge 3.3'de CIC-IDS2018'de bulunan saldırı türleri ve örnek sayıları görülmektedir.

Çizelge 3.3. CIC-IDS2018 veriseti farklı saldırı davranışlarının dağılımı

No	**Saldırı Tipi**	**Eğitim+**	**Test+**	**Toplam**
1	Bening	10787767	2696941	13484708
2	DoS Attack-HOIC	1348810	337202	1686012
3	DoS Attack -LOIC HTTP	1260953	315238	1576191
4	DoS Attack-Hulk	1169530	292382	1461912
5	Bot	228953	57238	286191
6	FTP-Brute Force	154688	38672	193360
7	SSH-Brute Force	150072	37517	187589
8	Infiltration	129648	32286	161934
9	SlowHTTPTest	111912	27978	139890
10	DoS Attack-GoldenEye	33207	8301	41508
11	DoS Attack-Slowloris	7926	1982	9908
12	DoS Attack-LOIC-UDP	1730	346	1384
13	Brute Force Web	550	110	440
14	SQL Injection	68	17	82
15	Brute Force XSS	181	46	227
	Toplam:	**15386459**	**3846484**	**19232943**

1-4 arası öznitelikler temel ağ bağlantı özelliklerini; 5-16 arası öznitelikler ağ paket özelliklerini; 17-22 arası öznitelikler ağ akış özelliklerini; 23-45 arası öznitelikler ağ istatistiklerini; 46-63 arası öznitelikler trafik çeşitlerini; 64-67 arası öznitelikler alt ağ akış özelliklerini; 68-79 genel ağ trafik özelliklerini; 80-83 arası öznitelikler ağ bağlantı özellikleri göstermektedir.

3.2. VERİ ÖN İŞLEME

Her iki veriseti de incelendiğinde verilerin dengeli bir şekilde dağılmadığı görülmektedir.Verisetlerinin işlenmesi sırasında doğruluk ve kesinlik gibi metriklerin hesaplanmasında bu dengesiz dağılım olumsuz etkiler. Bu oran aşağıdaki gibi hesaplanabilir.

$$Dengesizlik\ oranı = \frac{max_i\ \{C_i\}}{min_i\{C_i\}} \tag{3.1}$$

Denklemde belirtilen C_i parametresi veriseti içindeki veri boyutunu belirtmektedir. Veriseti içindeki maksimum örnek sayısının minimum örnek sayısına oranı olarak açıklanabilir. Saldırılara yönelik geliştirilen modellerin verimliliğini arttırmak için dengesizlik oranının azaltılması gerekmektedir. Bu çalışmada kullanılan NSL-KDD veriseti için dengesizlik oranı 64,8; CIC-IDS2018 veriseti için dengesizlik oranı 53,887 olarak hesaplanmıştır.

3.3. ÖZNİTELİK SEÇİMİ

Öznitelik seçim yöntemleri arasında sıklıkla Information Gain (IGA) tercih edilmektedir. Bunun sebebi bilgi kazancı yöntemindeki fazla veriye sahip olmamasına rağmen çeşitliliği fazla veri kümeleri lehine karar vermesidir [34–36]. Bu durum büyük verisetlerinde başka seçim yöntemleri ile desteklenmesi ihtiyacını ortaya çıkarmaktadır [37–39]. Aşağıdaki çalışmada kullanılan öznitelik seçim yöntemleri açıklanmıştır.

3.3.1. Korelasyona Dayalı Öz Nitelik Seçimi (Correlation-Based Self-Attribute Selection (CBS))

Yöntem, birbiri ile doğrudan ilişkisi olmayan kümelerin belirlenmesi temeline dayanır. Filitreleme mantığı vardır. Düşük korelasyon gösteren öz nitelikler elenerek, frekansı yüksek veriler üzerinden gidilir.

$$\mathrm{Ms} = \frac{\mathrm{krcf}}{\sqrt{\mathrm{k} + \mathrm{k}(\mathrm{k}-1)\mathrm{rff}}} \tag{3.2}$$

M_s= k adet öznitelik içeren S altkümesinin fayda değeri

rcf= Sınıf etiketi ile ilgili öznitelik arasındaki korelasyon

rff= Özniteliklerin birbirleri arasındaki korelasyon

3.3.2. Ki-Kare (Chi-square (CS))

İstatiksel bir yöntemdir. Sınıflara göre gözlemlenen ilk değerler, X2 istatistiği baz alınarak hesaplanır. Sonraki aşamada belirlenen önem durumuna göre veriseti içindeki öz nitelik sayısının 1 eksiğine göre seçim yapılır. Beklenen frekans değerini uygunluğu durumunda X2 sıfıra yaklaşır. Aksi durumda uyumsuzluğu işaret eder.

$$\mathrm{x}^2 = \sum_{\mathrm{i}=1}^{\mathrm{n}} \frac{(\mathrm{o_i} - \mathrm{e_i})^2}{\mathrm{e_i}} \tag{3.3}$$

n: veri setindeki öznitelik sayısı

o_i: i'inci öznitelik için gözlenen frekans değeri

e_i; i'inci öznitelik için beklenen frekans değeri

3.3.3. Bir-R (One-R (1-R))

Eğitim için ayrılan verisetindeki her öz nitelik belirlenen kurala göre sınıflandırılır. Oluşan hata oranına göre sıralama yapılarak en sık rastlanan öz niteliklere göre sıralama yapar. Çalışma mantığı; tüm tahmin için, yapılan tahminin her değeri için bir kural tanımlanır; sınıfta bulunan her bir değerin sıklığı sayılır. Frekansı en yüksek olan sınıf bulunur. Kurala bu tahmini ekleyip,toplam hatayı hesaplar, hata oranı en düşük olanı seçilir [38].

3.3.4. Simetrik Belirsizlik Katsayısı (Symmetrical Uncertainty Coefficient (SUC))

Bilgi kazancı yönteminde ortaya çıkan olumsuz durumu ortadan kaldırma adına X ve Y şeklinde örneklenen öz niteliklerin entropilerinin toplanması ile normalize edilir [39].

3.3.5. Bilgi Kazancı (Information Gain (IGA))

Bilgi kazancı, simetrik belirsizlik kazancındaki olumsuz kısımları normalize etmenin yanında entropiye dayanan bir yoldur. X özelliği ile Y özelliği birbirine bağlı olarak değişim gösterir. Yöntemdeki en büyük problem, fazla veriye sahip olmamasına rağmen çeşitliliği fazla veri kümeleri lehine karar verebilir olmasıdır [40].

$$Bilgi\ kazancı = H(Y) - H(\frac{Y}{X}) \tag{3.4}$$

3.3.6. Kazanç Oranı (Gain Rate (GRF))

Bilgi kazancı yöntemini normalize etmek için kullanılır. Amaç ortaya çıkan çeşitliliği minimize etmektir [41].

$$Kazanç\ Oranı = \frac{Bilgi\ Kazancı}{H(X)} \tag{3.5}$$

3.4. PERFORMANS METRİKLERİ

Bir bilgisayar ağındaki izinsiz girişlerin etkin bir şekilde tespit edilebilmesi için bazı metrikler kullanılır. Bunlar kesinlik, doğruluk, f1 score, recall, Gmean, FPR (false positive rate, DR (tespit oranı), AUC(area under the ROC curve) and ROC (receiver operating characteristic curve)'dir. Burada confusion matrix kullanılarak bu performans ölçüleri hesaplanır. True Positive (TP), 1 olarak sınıflandırdığımız ve gerçekten de 1 olan değerlerin sayısıdır. False Positive (FP) , 1 olarak sınıflandırdığımız fakat gerçekte 0 olan değerlerin sayısıdır. True Negative (TN), 0 olarak sınıflandırdığımız ve gerçekten de 0 olan değerlerin sayısıdır. False Negative (FN), 0 olarak sınıflandırdığımız fakat gerçekte 1 olan değerlerin sayısıdır. Çizelge 3.4'de karışıklık matrisi gösterilmiştir.

Çizelge 3.4. Ağ İzinsiz Giriş Tespiti için Karışıklık Matrisi

Sonuçlar	**Tahmini Trafik**	
	Normal Trafik	Saldırı Trafiği
Normal durum (ND)	Doğru Negatif (TN)	Yanlış Pozitif (FP)
Saldırı durumu (SD)	Yanlış Negatif (FN)	Doğru Pozitif (TP)

Doğru Negatif Oranı (True Nagative Rate) (TNR) yüksek ise sınıflandırma modelinin performansı yüksektir.

$$TNR = \frac{TN}{FP + TN} \tag{3.6}$$

Yanlış Pozitif Oranı (False Positive Rate) (FPR) düşük ise sınıflandırma modelinin performansı yüksektir.

$$FPR = \frac{FP}{FP + TN} \tag{3.7}$$

F score, Kesinlik ve Recall değerlerinin harmonik ortalamasını verir.

$$f\ score = 2x\frac{PrecisionxRecall}{Precision + Recall} \tag{3.8}$$

$$f\ score = \frac{2xTP}{2xTP + FP + FN} \tag{3.9}$$

Recall, sadece pozitif değerlerden doğru sınıflandırılanların oranını verir. DR yüksek ise sınıflandırma modelinin performansı yüksektir.

$$DR = recall = \frac{TP}{TP + FN} \tag{3.10}$$

Kesinlik, doğru sınıflandırılan verilerin oranını verir.

$$Kesinlik = \frac{TP}{TP + TN} \tag{3.11}$$

Doğruluk, doğruluk oranı sınıflandırıcı tarafından yapılan doğru tahmin sayısının tüm veri setindeki veri sayısına oranıdır. Doğruluk oranı ile sınıflandırıcının ne sıklıkla doğru bir tahminde bulunduğu ölçülmektedir. Doğruluk oranı 0 ile 1 arasında bir değere sahiptir. 0 en kötü oranı, 1 ise en iyi oranı ifade etmektedir. Doğruluk oranı denklem 4'de gösterildiği gibi hesaplanmaktadır.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.12}$$

Gmean kesinliğin geometrik ortalamasıdır. Gmean yüksek ise sınıflandırma modelinin performansı yüksektir.

$$Gmean = \sqrt{\frac{TP}{TP + FN}} x \frac{TN}{TN + FP} \tag{3.13}$$

$$Gmean = \sqrt{TPRxTNR} \tag{3.14}$$

ROC performans göstergesi olarak kullanılan yaygın bir grafiktir. Olası tüm eşik değerleri üzerinden bir sınıflandırma yapar. İki boyutlu bir düzlem üzerine çizilen eğri ile gösterilir. AUC ise ROC eğrisi altında kalan alan olarak tanımlanır. Alan değeri yüksek ise modelin performansı yüksektir. AUC değeri genel olarak 0,5-1 aralığındadır.

$$AUC = \int_0^1 \frac{TP}{TP + FN} d \frac{FP}{TN + FP} \tag{3.15}$$

ROC: ROC Eğrisi Örnek Kod bloğu

```
def roc_curve(y, prob):
    tpr_list = []
    fpr_list = []
    threshold = np.linspace(1.2, 0, 10)
    for t in threshold:
        y_pred = np.zeros(y.shape[0])
        y_pred[prob >= t] = 1
```

```
        TN = y_pred[(y_pred == y) & (y == 0)].shape[0]
        TP = y_pred[(y_pred == y) & (y == 1)].shape[0]
        FP = y_pred[(y_pred != y) & (y == 0)].shape[0]
        FN = y_pred[(y_pred != y) & (y == 1)].shape[0]
        FPR = FP / (FP + TN)
            TPR = TP / (TP + FN)
        fpr_list.append(FPR)
            tpr_list.append(TPR)
    return fpr_list, tpr_list, threshold
prob = predict_proba(x)
fpr,tpr,threshold = roc_curve(y, prob)
plt.plot(fpr, tpr, 'b')
plt.plot([0,1],[0,1], 'r--')
plt.xlabel("False Positive Rate", fontsize=10)
plt.ylabel("True Positive Rate", fontsize=10)
plt.show()
```

Öznitelik seçim yöntemleri, özellikle saldırı tespit sistemlerinin tasarlanması sırasında özellik ölçeklendirmenin önemini göstermektedir. Verisetlerinde bulunan verilerin normal dağılım dağılmadığının tespiti algoritmaların performans ve çalışma zamanlarını doğrudan etkilemektedir. Bu durum önerilen ve tasarlanan saldırı modellerinin doğruluk ve kesinlik performanslarının sonuçları etkilemektedir.Veriseti içindeki özniteliklerin ölçek farklılıklarının dengesiz olması, çoğu zaman gereksiz bellek kullanımı yanında aşırı öğrenme gibi istenmeyen durumları da ortaya çıkarmaktadır. Bu dengesiz, baskın yada gereksiz durumları ortadan kaldırma adına standardizasyon veya normalizasyon yöntemleri geliştirilmiştir. Çizelge 3.5'de saldırı tespit sistemlerinde kullanılan veri yığınları üzerine yapılan bu işlemler için kullanılan metotlar gösterilmiştir.

Çizelge 3.5. Normalizasyon Metotları

Metot	**Formül**
MinMax Scaling	$\frac{X - X_{min}}{X_{max} - X_{min}}$ (3.16)
Robust Scaler	$\frac{X - Q1(X)}{Q3(X) - Q1(X)}$ (3.17)
MaxAbs Scaler	$\frac{X}{max(abs(X))}$ (3.18)
Standardizasyon	$\frac{X - u}{s}$ (3.19)
Power Transformer	

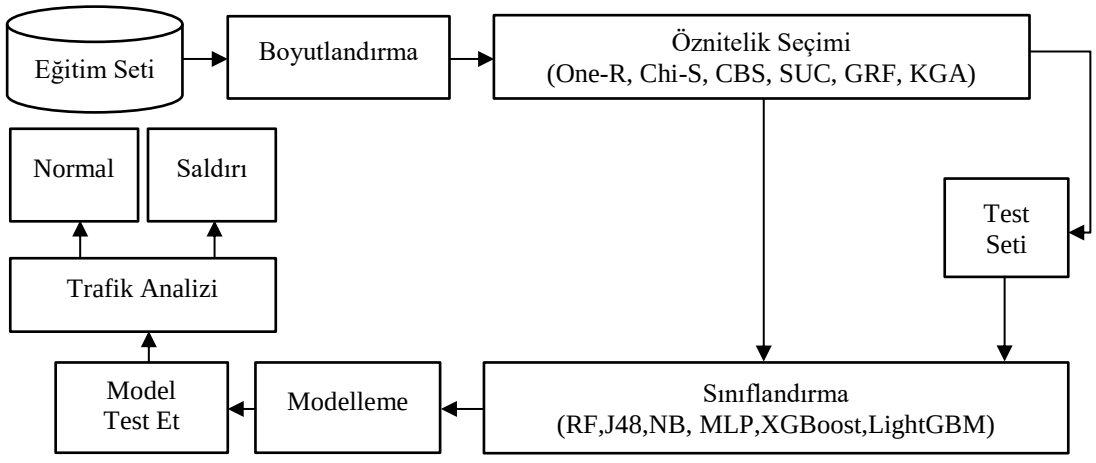
Min-Max scaling verinin 0 ile 1 arasında değerler aldığı bir durumdur. Burada 'outlier' denilen dışta kalan verilere karşı hassasiyet durumu vardır, bu yüzden bu değerlerin fazla olduğu bir durumda iyi bir performans gösteremeyebilir. Robust Scaler normalizasyon ile benzer şekilde çalışır. Yine veri dağılımı ile benzerlik gösterir ancak aykırı değerler dışarıda kalır. MaxAbs scaler medyan değeri sonradan kullanılmak üzere elenir ve değerler 1.ve 3. aralığına oturtulur. Her özelliğin maksimum mutlak değeri 1 olacak şekilde her özelliği ayrı ayrı ölçeklendirir ve dönüştürülür.Standardizasyon ortalama değerin 0, standart sapmanın ise 1 değerini aldığı, dağılımın normale yaklaştığı bir metottur. Formülü şu şekildedir, elimizdeki değerden ortalama değeri çıkartıyoruz, sonrasında varyans değerine bölüyoruz. Power transformer varyansı stabilize etmek ve çarpıklığı en aza indirmek için en uygun ölçeklendirme faktörünü bulur. Yine ortalama değerin 0, standart sapmanın ise 1 değerini aldığı bir metottur.

BÖLÜM 4

ÖZNİTELİK SEÇİMİ İÇİN ÖNERİLEN YÖNTEM

4.1. VERİ İŞLEME YÖNTEMİ

Veri işlemede ilk olarak öznitelik belirleme yöntemi olarak kazanç oranı, Korelasyon tabanlı öz nitelik seçimi, bilgi kazancı ki-kare, simetrik belirsizlik katsayısı, One-R seçilmiştir. Şekil 4.1'deki işlem basamaklarında görüldüğü gibi eğitim setinin boyutunun yeniden belirlenerek uygun değerlendirme aralığına getirilmelidir. Bu aşamanın önemi en iyi özniteliklerin belirlendiği aşama olmasıdır. Özellikle anomali tespitinde toplanan verilerin boyutlarının küçültülmesi saldırı tespiti sırasındaki yükü azaltacaktır. Bilinen saldırı türlerinden elde edilen özniteliklerle sistemin başarısı da artar. Alınan veri seti rastgele ve komple arama gibi yöntemlerle alt kümelere ayrılır. Böylece değerlendirilecek bir alt veriseti oluşturulur. Seçime göre bağımlı yâda bağımsız kriterler belirlenir. Bu aşamada en iyi alt küme belirlemek amacı ile yeterince alt küme oluşturulana kadar devam edilir. Belirsizlik söz konusu olduğu için öznitelik seçimi sırasında entropi kullanılır. Öznitelik seçiminde sıralama ölçütüne göre NSL-KDD veriseti için 10 adet öznitelikleri seçildi. Seçilen öznitelikler 4 farklı algoritma ile sınıflandırıldı. Test için alınan bu öznitelikler modele aktarıldı. Alınan veri setleri alt kümelere indirgenerek, öznitelik seçim yöntemleri ile sıralanarak NSL-KDD için 10 öznitelik seçimi yapılmıştır.



Şekil 4.1. Öznitelik Seçim Modeli

4.2. ÖZNİTELİK SEÇİM YÖNTEMİ

Çalışmada NSL-KDD veriseti 6 farklı öznitelik belirleme yöntemi kullanılmıştır. Her yöntemde en başarılı 10 öznitelik başarım sıralamasına göre seçilmiştir. NSL-KDD veriseti ile yapılan çalışmalar incelendiğinde seçilen öznitelik sayısının, öznitelik seçim yöntemi ve sınıflandırma yaklaşımlarının farklı olduğu görülmektedir. Çizelge 4.1'de NSL-KDD verisetinde ile yapılan öznitelik seçiminde One-R ve CBS kullanılarak elde edilen sonuçlar gösterilmektedir. Burada yüksek frekanslı özniteliklerin tespit işlemi yapılmıştır.

Çizelge 4.1. NSL-KDD'de One-R ve CBS ile frekans ve seçim sonuçları

One-R			**CBS**		
No	**Öznitelik Adı**	**Frekans**	**No**	**Öznitelik Adı**	**Frekans**
5	src_bytes	96.374	3	service	0.747
3	service	91.558	5	src_bytes	0.725
6	dst_bytes	90.994	12	logged_in	0.695
4	flag	88.098	4	flag	0.692
30	diff_srv_rate	87.380	6	dst_bytes	0.691
29	same_srv_rate	87.324	29	same_srv_rate	0.634
34	dst_host_same_srv_rate	85.426	30	diff_srv_rate	0.595
33	dst_host_srv_count	85.015	25	serror_rate	0.576
35	dst_host_diff_srv_rate	83.929	26	srv_serror_rate	0.563
12	logged_in	82.947	33	dst_host_srv_count	0.531

Çizelge 4.2'de NSL-KDD verisetinde ile yapılan öznitelik seçiminde GRF ve CS kullanılarak elde edilen sonuçlar gösterilmektedir. Burada yüksek frekanslı özniteliklerin tespit işlemi yapılmıştır.

Çizelge 4.2. NSL-KDD'de GRF ve CS ile frekans ve seçim sonuçları

GRF			**CS**		
No	**Öznitelik Adı**	**Frekans**	**No**	**Öznitelik Adı**	**Frekans**
12	logged_in	0.418	5	src_bytes	0.968
26	srv_serror_rate	0.373	3	service	0.932
4	flag	0.339	6	dst_bytes	0.876
25	serror_rate	0.332	4	flag	0.756
39	dst_host_srv_serror_rate	0.332	30	diff_srv_rate	0.740
30	diff_srv_rate	0.267	29	same_srv_rate	0.738
38	dst_host_serror_rate	0.264	33	dst_host_srv_count	0.669
6	dst_bytes	0.258	34	dst_host_same_srv_rate	0.670
5	src_bytes	0.231	35	dst_host_diff_srv_rate	0.628
29	same_srv_rate	0.224	12	loggen_in	0.600

Çizelge 4.3'de NSL-KDD verisetinde ile yapılan öznitelik seçiminde SUC ve IGA kullanılarak elde edilen sonuçlar gösterilmektedir. Burada yüksek frekanslı özniteliklerin tespit işlemi yapılmıştır.

Çizelge 4.3. NSL-KDD'de SUC ve IGA ile frekans ve seçim sonuçları

SUC			**IGA**		
No	Öznitelik Adı	Frekans	No	Öznitelik Adı	Frekans
12	loggen_in	0.411	5	src_bytes	0.816
4	flag	0.411	3	service	0.671
26	srv_serror_rate	0.377	6	dst_bytes	0.633
6	dst_bytes	0.367	4	flag	0.519
39	dst_host_srv_serror_rate	0.362	30	diff_srv_rate	0.518
25	serror_rate	0.360	29	same_srv_rate	0.509
5	src_bytes	0.360	33	dst_host_srv_count	0.475
30	diff_srv_rate	0.353	34	dst_host_same_srv_rate	0.438
38	dst_host_serror_rate	0.320	35	dst_host_diff_srv_rate	0.410
29	same_srv_rate	0.311	38	dst_host_serror_rate	0.405

Öznitelik seçim yöntemleri sonucu elde edilen 10 öznitelik adı, verisetinde numarası ve öznitelik frekansı aşağıdaki tabloda gösterilmiştir. 4,5,6,12,29,30 numaralı

özniteliklerin frekansının yüksek olduğu görülmüştür. 6 farklı öznitelik seçim yöntemleri ile elde edilen frekansı yüksek bu özniteliklerin özellikle anomali tespitinde etkili olduğu görülmektedir.

Çizelge 4.4. Yapılan sıralamada NSL-KDD öznitelik frekanslar

Öznitelik No	**Öznitelik Adı**	**Frekansı**
3	service	4
4	flag	6
5	src-bytes	6
6	dst-bytes	6
12	Logged-in	5
25	Serror-rate	3
26	Srv-serror-rate	3
29	same srv-rate	6
30	diff-srv-rate	6
33	dst-host-srv-count	3
34	dst-host-same-srv-rate	3
35	dst-host-diff-srv-rate	3
38	dst-host-serror-rate	3
39	dst-host-srv-serror-rate	2

Öznitelik alt kümeleri sırasıyla Random Forest, Naive Bayes, J48 ve Multi-Layer Perceptron (MLP) gibi 4 farklı sınıflandırma algoritması ile analiz edilmiştir. Çizelge 4.4 görüldüğü gibi yapılan çalışmalarda saldırının tespit edilmesi noktasında pozitif doğru algılama oranı, yanlış algılama oranı, yanlış pozitif oranı, pozitif yanlış oranı, gösterge çizelgesi altında kalan alan ve doğruluk oranı gibi parametreler kullanılmaktadır [59]. Güncel saldırı tiplerinin sınıflandırılması ve modellenmesi ile karar verildiği için spesifik bir saldırı tespit referans ölçüsü yoktur. Çalışmamızda saldırıyı kesinlik, doğruluk, saldırı tespit oranı ve yanlış pozitif oranı ölçütleri olarak alınmıştır. Verisetleri üzerinde algoritmaların saldırıların tespit etme başarılarını ortaya koymak için tespit matrisi oluşturulmuştur. Bu matris Çizelge 4.4'de belirtilen değerlendirme kriterlerini göstermektedir [60].

4.3. ÇALIŞMADA KULLANILAN SINIFLANDIRMA ALGORİTMALARI

Optimizasyon, makine öğrenmesi ve derin öğrenme için kayıp işlevini en aza indirmenin bir yolunu sağlasa da, optimizasyon ve derin öğrenmenin hedefleri temelde farklıdır. İlki esas olarak bir hedefi en aza indirmekle ilgilenirken, ikincisi sınırlı miktarda veri verildiğinde uygun bir model bulmakla ilgilidir. Optimizasyon algoritmasının amaç işlevi genellikle eğitim veri setine dayalı bir kayıp işlevi olduğunda eğitim hatasını azaltmaktır. Bununla birlikte, istatistiksel çıkarımın amacı genelleme hatasını azaltmaktır. Bu durum saldırı tespit sistemlerinde makine öğrenmesi ve derin öğrenme yaklaşımlarının kullanılabilir olduğunu gösterir. Saldırı tespit sistemleri yapısal olarak siber saldırılara karşı hızlı yanıt vermek zorundadır [28,29,30]. Makine öğrenmesi algoritmalarının saldırı tespit sistemlerinde kullanımını sınıflandırma, ölçekleme ve her iki yöntem kullanılarak değerlendirebiliriz. Veri setinden seçilen öznitelikler sınıflandırması aşamasında makine öğrenmesi algoritmaları ile işlenir. Sınıflandırma aşamasında kullanılan teknikler uygun model oluşturulması sağlar.

4.3.1.Random Forest Algorithm

Random Forest algoritması siber saldırıların tespit edilmesine yönelik yapılan çalışmalarda kullanılan sınıflandırma algoritmasıdır [42]. Karar ağacı sınıflandırması olarak tercih edilir. Ağacı oluşturan her bir düğüm ile alt kümelere ayrılan özniteliklerin karşılaştırılarak, yeni dallar oluşturulur. Ağacın sahip olduğu yapraklar sınıf olarak ifade edilir. En büyük avantajı diğer algoritmalara göre daha az parametreye sahiptir. Elde bulunan veriler içindeki anormal veriler karşısında gereksiz işlem yapmaz. Daha düşük yüklerle çalışır [42]. Bu özelliği ile bilgisayar ağlarındaki anomali tespitinde başarılı sonuçlar verir [42,43,44].

$\{h(x,\Theta k),k=1,2,\ldots i\ldots\}$ J48 (4.1)

h: Sınıflandırıcı Θk*: rastgele vektör* x*: ağaç sınıf etiketi*

4.3.2. Naive Bayes Algorithm

Naive Bayes algoritması, sınıflandırma için elde bulunan her bir öznitelik çift birbirinden bağımsız olarak işlenen bir bayes yaklaşımıdır. Naive Bayes, verileri birbirinden bağımsız değerlendirir. Amaç her bir parametrenin sonuca eşit ağırlıkta

etki etmesidir. Yapısal olarak oldukça basit ve hızlı bir algoritmadır. Siber saldırıların tespit edilmesinde tercih edilen yöntemlerdendir[43].

$$P(c \mid x)=(P(x \mid c)P(c))/(P(x))$$
$$P(c \mid X)=P(x\_1 \mid c)*P(x\_2 \mid c)*\ldots*P(x\_n \mid c)*P(c) \quad (4.2)$$

P(c|x): Arka Olasılık *P(x|c): Olasılık*

P(c): Sınıf Öncelikli Olasılık *P(x): Predictor Önceki Olasılık*

Örnekleme noktasında daha az eğitim verisine ihtiyaç duyduğu için kısa sürede sonuç üretebilir. NB, sınıfları koşullu marjinal yoğunlukları bulmak için ayırıcı sınıflar problemini azaltır, bu da belirli bir örneğin olası hedef sınıflardan biri olma olasılığını temsil eder. NB, birbirleri ile ilişkili girdiler içermedikçe diğer algoritmalara karşı iyi performans göstermektedir [44]. Kısa sürede sonuç üretebilmesi, verileri birbirinden bağımsız değerlendirmesi gibi avantajlarından dolayı sınıflandırma için seçilmiştir.

4.3.3. J48 Algorithm

J48 Ross Quinlan tarafından geliştirilen bir algoritmadır. ID3 algoritmasının devamı olarak nitelendirilir. Karar ağacı oluşturulabildiği için yapısal anlamda istatiksel sınıflandırıcı olarak kullanılır [46]. Bu sınıflandırıcıda, ağaç modeli şeklinde akış şeması oluşturularak problem çözümü tahmine dayalı çözülmeye çalışılır. Veri kazancı hesaplanarak öznitelikler alt kümelere bölünür. Bölünen her bir sınıf bir karar düğümü geliştirir. Karar ağacında bulunan düğümler giriş için alınan örnekleri, ağaç yaprakları da bu girişe bağlı yapılan tahminleri gösterir [47].

4.3.4. Multi-Layer Perceptron- Convolutional Neural Network

Multi-Layer Perceptron, ileri beslemeli bir yapay sinir ağıdır. Giriş katmanı, gizli katman ve çıktı katmanı olarak en az üç katmandan oluşur. Giriş düğümleri hariç, her düğüm doğrusal olmayan bir aktivasyon işlevi kullanır. Eğitim için geri beslemeli denetimli öğrenme tekniği kullanır.

$$f(x)=(\sum\_(i=1)^n w\_i *x\_i)+b \quad (4.3)$$

m: Bir önceki katmandaki nöron sayısı

w:rastgele ağırlık x:giriş değeri b:rastgele sapma

Bu yönüyle doğrusal olmayan veriler içinde, istenen verilerin ayırt edilmesinde kullanılabilir [45]. Saldırı tespit sistemleri için kullanılan verisetleri de doğrusal

olmayan veriler içermektedir. Verisetindeki özniteliklerin yüksek doğrulukla ayırt edilebilmesi için bu algoritma seçilmiştir.

4.4.5. Extreme Gradient Boosting (XGBoost)

Model performansı ve hız özellikleri ile ön plana çıkan algoritma özellikle eğitim sırasında paralel hesaplama yöntemini kullanır. Toplu öğrenme yöntemi ile doğru tahminlerin yapılmasındaki performansı özellikle siber saldırıların tespitinin yapılmasında önemlidir.

$$Obj^m = \sum_i \Omega(y_i, y_i)^{(m-1)} + \sum_k \Phi(f_k) \tag{4.4}$$

$$w_j = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} = -\frac{G_j}{H_j + \lambda} \tag{4.5}$$

$$Obj^m = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{4.6}$$

Çizelge 4.5. XGBoost Algoritması Pseudo kodu

Algoritma:

Giriş**:** $U_{normalized}$ $(f_i^{norm} \ldots\ldots f_n^{norm})$

Çıkış: $U_{optimal}$: Seçilen özellik vektörü maksimum

1: Normalleştirilmiş özellik vektörünü yükle

2: Puanları kaydetmek için boş bir S oluşturun

3: Bir GradientBoostingClassifier örneğini şu şekilde oluşturun: f

4: f yerleştir

5: Fıs oluştur

6**:** FI eşiğini belirle FI_p

for n from $U_{normalized}$ **do**

if $(FI(x^i) \geq FI_p)$ **then**

$FI(x^i)$ S'ye ekle

end if

end for

5: $U_{optimal}$ için S'deki elde edilen sonuçları kullan

XGBoost algoritması Gradient Boosting algoritmasının optimize edilmiş bir türüdür. Önceki versiyonlara göre sağladığı avantajları XGBoost kullanımının yaygınlaşmasındaki en önemli nedendir. XGBoost, ağacı oluştururken maksimum derinlik değerini kullanır. Oluşturulan ağaç aşağı yönde aşırı ilerleme gösterirse, budama gerçekleştirilir. Aşırı öğrenmenin önüne geçilir. Gradient Boosting algoritması, kayıp fonksiyonun hesaplanmasında birinci dereceden fonksiyon kullanırken, XGBoost bu hesaplamaları ikinci dereceden fonksiyonlar kullanarak gerçekleştirir. Paralel çalışma özelliği, diğer algoritlamalara göre sonuca daha kısa sürede ulaşılmasını sağlar.

4.4.6. Long and Short Time Memory (LSTM)

Tekrarlayan Sinir Ağı (RNN) olan LSTM 1997'de Schmidhuber ve Hochreiter tarafından önerilmiştir. Modelde gates, cell state yardımıyla hatırlanacak veri bir sonraki aşamaya aktarılır yada önemsiz olarak etiketlenerek unutulur. Bir önceki katmandan gelen veri sigmoid fonksiyon üzerinden geçerek 0 ve 1'e yakınlık durumuna göre işlem görür. Uzun bağımlılık ilişkisi olan verisetleri üzerindeki başarısı sebebiyle LSTM algoritması seçilmiştir.

4.4.7. Gradient Boosting Decision Tree (LightGBM)

LightGBM, Microsoft DMTK (Distributed Machine Learning Toolkit) projesi kapsamında 2017 yılında geliştirilmiş bir boosting algoritmasıdır. Diğer boosting algoritmaları ile karşılaştırıldığında yüksek işlem hızı, büyük verileri işleyebilmesi, daha az kaynak (RAM) kullanımı, yüksek tahmin oranı, paralel öğrenme ve GPU öğrenimini desteklemesi gibi avantajları vardır. Modelin tanıtıldığı "LightGBM: A Highly Efficient Gradient Boosting Decision Tree" makalesine göre, yapılan çalışmalarda LightGBM'in diğer modellere göre 20 kat daha hızlı olduğu sonucuna ulaşılmıştır [12].

Çizelge 4.6. LightGBM Algoritması Pseudo kodu

Algoritma:
Giris:
training data: $D=\{(x_1,y_1),(x_2,y_2)\ldots(x_n,y_n)I\ x_i \in x_n \subseteq R,\ y_i \in \{-1,1\}$ loss function:$L(y,\theta(x))$;
iterasyon:

M; büyük gradyan veri örnekleme oranı:a; hafif gradyan veri örnekleme oranı:b;

1: xi,I={1…N}'nin birbirini dışlayan özelliklerini (özellik hiçbir zaman aynı anda sıfır olmayan değerleri kabul etmez) dışlayıcı özellik demetleme tekniği ile birleştirilmesi;

2: Set θ_0 (x)=arg $\min_c \sum_i^N L(y_i, c)$;

for m=1 to M **do**

3: Gradyan mutlak değerlerini hesaplama:

$$r_i = \left|\frac{\partial L(y_i,\theta(x_i)}{\partial \theta(x_i)}\right|_{\theta(x)=\theta_{m-1(x)}}, \text{ i=\{1…,N\}} \quad (4.7)$$

4: Degrade tabanlı tek taraflı örnekleme kullanarak veri kümesini yeniden örnekleme **islem:**

topN=a *len(D); randN=b*len(D);

sorted=GetSortedIndices(abs(r));

A=sorted[1:topN]; B=RandomPick(sorted[topN;len(D)],randN;

D=a+b;

Bilgi kazancı hesaplama:

$$V_j(d) = \frac{1}{n}\left(\frac{(\sum_{x_i\in A_l} r_i + \frac{1-a}{b}\sum_{x_i\in B_l} r_i)^2}{n_l^j(d)} + \frac{(\sum_{x_i\in A_T} r_i + \frac{1-a}{b}\sum_{x_i\in B_T} r_i)^2}{n_T^j(d)}\right) \quad (4.8)$$

5: D^i setinde yeni bir karar ağacı $\theta_m(x)^I$ geliştirme:

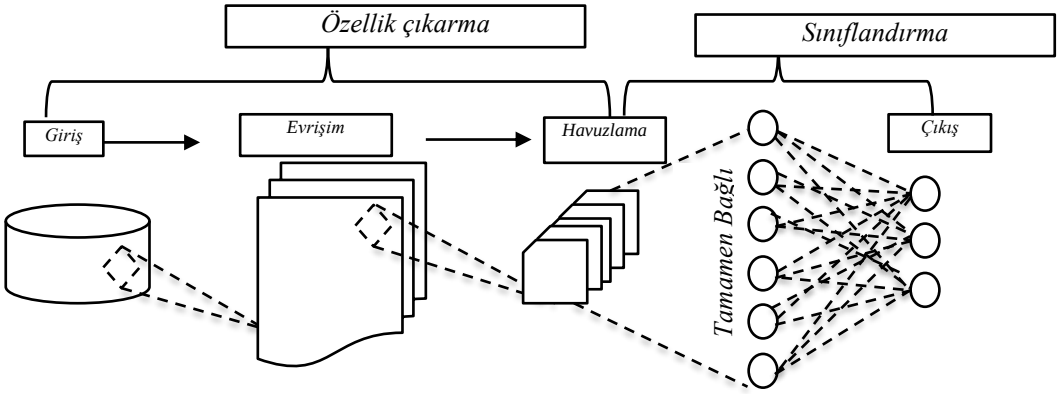
6: update $\theta_m(x)= \theta_{m-1(x)}+ \theta_m x$

end for

return $\theta(x)= \theta_m(x)$

4.4.8. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN), çok katmanlı algılayıcıların bir türüdür. Denetimli öğrenme şeklinde temel olarak iki ana katmandan oluşur. Öznitelik çıkarımında konvolüsyon ve alt örnekleme şeklinde çalışır. Sınıflandırma algoritmasında çok katmanlı algılayıcılar gibi ilerler.



Şekil 4.2. CNN Algoritma Modeli

4.4. ÖZNİTELİK SEÇİM YÖNTEMİ TEST SONUÇLARI

Çalışma sonucu elde edilen veriler Tablo 7 ve Tablo 8'de gösterilmiştir. Sınıflandırma algoritmalarının P, FPR, Acc ve ADR kriterlerine göre anomali tespit performansları karşılaştırılmıştır. Random Forest ile J48 algoritmalarının yüksek sınıflandırma oranı, öznitelikler üzerinden anomali tespitini de etkilediği görülmüştür. Çizelge 4.5'te random forest algoritması ile frekansı en yüksek olarak belirlenen on özniteliğin performans sonuçları görülmektedir.

Çizelge 4.7. RF ile sınıflandırma sonucu elde edilen performans sonuçları

Sınıflandırma Doğruluğu		**RF**				
		99.9174 %				
Öznitelik	**Metot**	**Süre (sn)**	**P %**	**FAR %**	**ACC %**	**ADR %**
5,3,6,4,30,29,34,33,35,12	One-R	2,90	78,65	3,23	84,95	95,43
12,26,4,25,39,30,38,6,5,29	GRF	4,11	74,82	5,43	81,23	91,23
5,3,6,4,30,29,33,34,35,12	CS	3,01	76,78	3,21	84,65	93,54
5,3,6,4,30,29,33,34,35,38	IGA	3,05	76,71	3,28	84,89	93,87
12,4,26,6,39,25,5,30,38,29	SUC	3,98	78,42	6,40	85,54	91,20
3,5,12,4,6,29,30,25,26,33	CBS	5,43	76,43	6,54	76,54	89,65
Tüm öznitelikler	-	6,78	75,43	3,45	89,45	99,76

Çizelge 4.6'da J48 algoritması ile frekansı en yüksek olarak belirlenen on özniteliğin performans sonuçları görülmektedir.

Çizelge 4.8. J48 ile sınıflandırma sonucu elde edilen performans sonuçları

Sınıflandırma Doğruluğu		**J48**				
		99.7817 %				
Öznitelik	**Metot**	**Süre (sn)**	**P %**	**FAR %**	**ACC %**	**ADR %**
5,3,6,4,30,29,34,33,35,12	One-R	3,34	76,86	4,23	83,56	93,52
12,26,4,25,39,30,38,6,5,29	GRF	4,11	73,20	6,58	80,23	89,42
5,3,6,4,30,29,33,34,35,12	CS	3,01	74,52	4,23	83,46	91,12
5,3,6,4,30,29,33,34,35,38	IGA	3,05	74.56	4,74	83,20	90,36
12,4,26,6,39,25,5,30,38,29	SUC	3,98	76,59	7,52	83,76	89,93
3,5,12,4,6,29,30,25,26,33	CBS	5,43	74,86	7,41	75,47	88,63
Tüm öznitelikler	-	6,78	73,47	4,69	89,45	98,45

Çizelge 4.7'da Naive Bayes algoritması ile frekansı en yüksek olarak belirlenen on özniteliğin performans sonuçları görülmektedir.

Çizelge 4.9. NB ile sınıflandırma sonucu elde edilen performans sonuçları

Sınıflandırma Doğruluğu		**NB**				
		90.4178 %				
Öznitelik	**Metot**	**Süre (sn)**	**P %**	**FAR %**	**ACC %**	**ADR %**
5,3,6,4,30,29,34,33,35,12	One-R	5,20	75,38	5,89	82,58	89,36
12,26,4,25,39,30,38,6,5,29	GRF	4,78	71,24	8,23	79,56	88,26
5,3,6,4,30,29,33,34,35,12	CS	3,97	72,28	5,27	80,26	89,78
5,3,6,4,30,29,33,34,35,38	IGA	3,98	72,56	5,23	80,29	89,87
12,4,26,6,39,25,5,30,38,29	SUC	4,21	74,36	8,54	80,56	88,25
3,5,12,4,6,29,30,25,26,33	CBS	6,23	72,57	9,23	72,14	87,41
Tüm öznitelikler	-	8,30	70,29	4,69	89,45	93,34

Çizelge 4.8'de MLP algoritması ile frekansı en yüksek olarak belirlenen on özniteliğin performans sonuçları görülmektedir.

Çizelge 4.10. MLP ile sınıflandırma sonucu elde edilen performans sonuçları

Sınıflandırma Doğruluğu		**MLP**				
		98.4354 %				
Öznitelik	**Metot**	**Süre (sn)**	**P %**	**FAR %**	**ACC %**	**ADR %**
5,3,6,4,30,29,34,33,35,12	One-R	19,65	69,29	4,17	80,86	83,50
12,26,4,25,39,30,38,6,5,29	GRF	30,43	77,41	7,12	84,52	91,47
5,3,6,4,30,29,33,34,35,12	CS	18,40	68,86	5,21	80,14	83,20
5,3,6,4,30,29,33,34,35,38	IGA	19,97	68,276	5,74	81,01	82,98
12,4,26,6,39,25,5,30,38,29	SUC	23,43	65,98	7,51	79,52	81,26
3,5,12,4,6,29,30,25,26,33	CBS	22,50	75,58	6,27	80,74	80,37
Tüm öznitelikler	-	45,86	72,41	4,12	81,38	91,34

Seçilen 10 öznitelik ile yapılan sınıflandırmalarda Random Forest %99,76; J48 %98,45; Naive Bayes %93,34 ve MLP-CNN %91,34 oranında başarım göstermiştir. Çizelge 4.9'de veriseti içinde bulunan test ve eğitim verilerindeki normal ve anomali trafik bilgileri görülmektedir.

Çizelge 4.11. Veriseti içinde bulunan özniteliklerin durumu

Durum	**Eğitim Seti**	**Test Seti**
Anomali	58630	12833
Normal	67343	9711

Eğitim seti içinde bulunan 58630 anomaly DoS, U2R, R2L ve Probe attack içermektedir. Anomaly içeren verisetinin %78,33'ü DoS saldırıları, %0,08 U2R saldırıları, %1,69 R2L saldırıları ve %19,88 Probe saldırılarından oluşmaktadır. NSL-KDD veriseti içerdiği DoS, Probe, R2L ve U2R atak tipleri Random Forest, Naive Bayes, J48 ve Multi-Layer Perceptron (MLP)-CNN gibi 4 farklı sınıflandırma algoritması ile analiz edilmiştir. Makine öğrenmesi algoritmalarının saldırıları tespit etme başarım ölçütü olarak P, ROC, F1 Score, Recall ve doğruluk (Accuracy) kriterleri alınmıştır. Çizelge 4.10'da yaygın olarak görülen probe attack saldırılarının dört farklı Algoritma kullanılarak saldırı tespit performansları görülmektedir.

Çizelge 4.12. Probe saldırılarının algoritmalar ile tespit analizi

Saldırı Adı	Algoritmalar	P	ROC	F1-Score	Re-call	Accuracy (%)
Probe Attack	Multi-Layer Perceptron (MLP)-CNN	0.954	0.996	0.998	0.998	98.510
	Naive Bayes	0.986	0.976	0.961	0.971	90.398
	Random Forest	**0.999**	**1.000**	**1.000**	**1.000**	**99.952**
	J48	0.994	0.999	0.999	1.000	99.951

Çizelge 4.11'da yaygın olarak görülen DoS saldırılarının dört farklı Algoritma kullanılarak saldırı tespit performansları görülmektedir.

Çizelge 4.13. DoS saldırılarının algoritmalar ile tespit analizi

Saldırı Adı	Algoritmalar	P	ROC	F1-Score	Re-call	Accuracy (%)
DoS Attack	Multi-Layer Perceptron (MLP)-CNN	0.954	0.841	0.948	0.998	95.752
	Naive Bayes	0.979	0.909	0.951	0.914	94.178
	Random Forest	**1.000**	**0.999**	**0.999**	**0.999**	**99.842**
	J48	0.995	0.667	**0.999**	**0.999**	99.774

Çizelge 4.12'da yaygın olarak görülen R2L saldırılarının dört farklı Algoritma kullanılarak saldırı tespit performansları görülmektedir.

Çizelge 4.14. R2L saldırılarının algoritmalar ile tespit analizi

Saldırı Adı	Algoritmalar	P	ROC	F1-Score	Re-call	Accuracy (%)
Remote to Local Attack	Multi-Layer Perceptron (MLP)-CNN	0.997	0.996	0.992	0.992	99.814
	Naive Bayes	0.999	0.957	0.935	0.889	98.928
	Random Forest	**0.999**	**0.999**	**0.999**	**1.000**	**99.999**
	J48	0.998	0.995	0.998	0.999	99.997

Çizelge 4.13'da yaygın olarak görülen U2R saldırılarının dört farklı Algoritma kullanılarak saldırı tespit performansları görülmektedir.

Çizelge 4.15. U2R saldırılarının algoritmalar ile tespit analizi

Saldırı Adı	Algoritmalar	P	ROC	F1-Score	Re-call	Accuracy (%)
User Root Attack	Multi-Layer Perceptron (MLP)-CNN	0.995	0.995	0.995	0.995	99.210
	Naive Bayes	0.999	0.949	0.961	0.943	88.859
	Random Forest	0.999	**0.998**	0.997	**0.998**	**99.859**
	J48	**1.000**	0.937	**0.998**	**0.998**	99.674

BÖLÜM 5

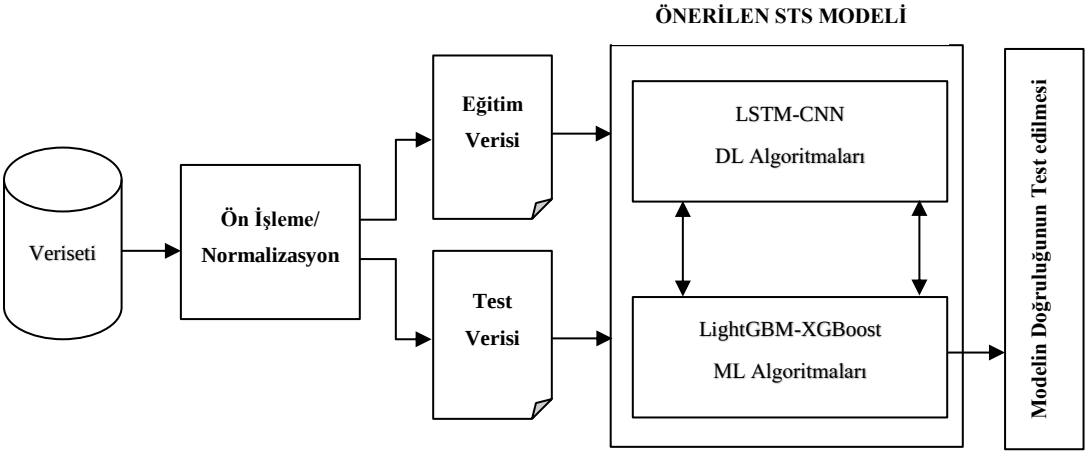
SİBER SALDIRILARIN TESPİTİ İÇİN ÖNERİLEN MODELLER

Siber saldırıların tespiti için öncelikle gerçek zamanlı veri toplamak gerekir. Verisetleri saldırı tespit sistemlerinin tasarımında yaygın olarak kullanılmaktadır. Gerçek zamanlı veriler toplanarak oluşturulan verisetlerinin güncel ve olası tüm saldırı tiplerini içermesi istenir. Güncel ve yeteri kadar veriye sahip verisetinin seçilmesinin ardından bu verilerin anlamlı hale getirilmesi gerekir. Bu çalışmada önerilen model Şekil 5.1'de gösterilmiştir. Öncelikle verisetlerinin öznitelikleri kategorik olarak sahip oldukları veriler analiz edildi. Literatürdeki çalışmalar incelendiğin kategorik kodlama için Dummy Encoding yada One-Hot Encoder kullanılmaktadır. Verilerin dağılımındaki olası dengesizlikler algoritmaların çalışmasını etkilemektedir. Özniteliklerin ölçeklendirme işlemi sonucu normalize edilmiş veriler haline gelmesiyle tahminleyici algoritmaların doğruluk (Acc) ve kesinlik (P) değerleri olumlu yönde etkilenir. Önerilen modelin test edilmesinde kullanılan NSL-KDD ve CIC-IDS2018 verisetleri 20/80 test ve eğitim verisi olarak ayrılmıştır. Verisetlerinden öznitelik çıkarım işlemi için deep learning algoritmaları kullanılmıştır. Conv1D'ler özniteliklerin çıkarımı için kullanılmıştır. Eğitim sonucu önem derecesi daha düşük olan veriler Pooling1D katmanında tutulmaktadır. Bırakma katman değeri olarak 0.2 seçilmiştir. Bırakma katmanı (Dropout Layer), gereksiz tekrarları engellemek için kullanılmıştır.Bırakma katmanlarını tek bir vektör değeri haline gelecek şekilde birleştirilerek makine öğrenimi algoritmasının giriş verisi olarak verilmiştir.

Çizelge 5.1. Çalışmada kullanılan platformun teknik bilgileri

Donanım/Yazılım	**Özellikleri**
CPU	Intel Core™ i7-12700H CPU 2.6 Ghz
GPU	NVIDIA RTX 3060 6 GB
Memory	32GB DDR3, 4800 MHz
Storage	512GB NVMe SSD
OS	Windows 10 Pro, 64 bits
Software	Python 3.7.6, Tensorflow 2.1.0, MongoDB 4.2,Apache Spark 3.x,
	NVIDIA Cuda

Çizelge 5.1'de önerilen modelin test edildiği donanım ve yazılım ortamı ile ilgili bilgiler verilmiştir. Verisetlerinin boyutlarının yüksek olmasının ile kullanılan donanımın sınırlılıkları modelin testi sırasında zaman verisini olumsuz etkilemiştir.



Şekil 5.1. Önerilen Saldırı Tespit Sistem Modeli

Saldırıların tespiti sırasında eğitim süreci önemlidir. Bu kısımda eğitim sürecindeki elde edilen sonuçların ve öğrenme sırasında modelin öğrenmesini kolaylaştırmak için düzeltilmiş doğrusal aktivasyon kullanılmıştır. Rectified Linear Unit (ReLU) fonksiyon yapısal olarak iyi bir tahmin edici olması saldırıların tespitindeki başarım yüzdesini arttırdığı görüldü. Yapılan testlerde alternatif olarak kullanılan Softmax yapısal olarak tahminleri 0 ve 1 arasına sıkıştırır. Burada tahmin olasılıklarının önem noktasında değişiklik yapmaması özellikle saldırı tespitlerinde özniteliklerin frekanslarının tespitinde hataları ortaya çıkarmıştır. Bu sebeple ReLU seçilmiştir. Saldırıların tespitinde özellikle verisetleri üzerindeki testlerde karşılaşılan sorunların başında modelin aşırı öğrenmesi (overfitting) gelmektedir. Bunu minimize etmek için Dropout layers tercih edildi. CNN ve LSTM algoritmaları için seçilen parametre ve değerler Çizelge 5.2'de belirtilmiştir. Yapılan testlerde dropout 0.1 ve 0.2 olarak seçildi. Diğer derin öğrenme algoritması testlerinde hücre sayısının 100 olarak seçilmesi gerektiği testler sonunda görüldü. Gürültülü verisetleri için daha uygun olan Adam Optimization Algorithm 0.02 gibi sabit bir oran ile kullanıldı. Burada Adam Optimization Algorithm hesaplama noktasında verimli sonuçlar ortaya çıkardı. Önerilen ilk modelde CNN ve LSTM XGBoost kullanılmıştır. 32 çekirdekli 1B dizisi iki gizli katmandan oluşan bir mimaride tasarlanmıştır. Çekirdek boyutları 2 olarak verilmiştir. One-hot encoding yöntemi ile öznitelikler vektör formatına çevrildi.

Glorot ve Xavier weight Initializater olarak bilinen sigmoid ve tanh aktivasyonları ağırlıkların başlatılması için kullanıldı.

Çizelge 5.2. Seçilen algoritmaların sınıflandırma parametreleri

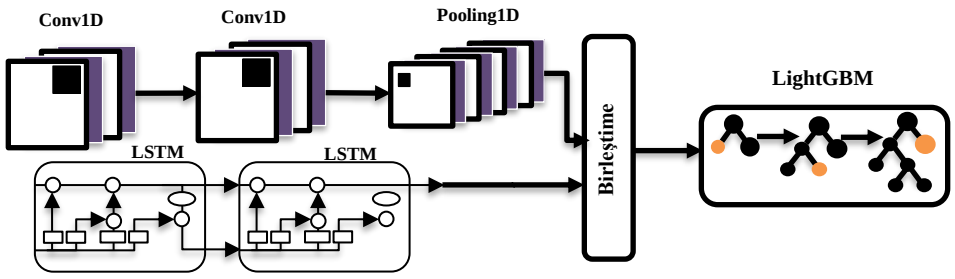
ML Model Sınıflandırma	**Parametreleri**	**DL-Modeli**	**Parametreleri**
XGBoostClassifier	cV=10 learning_rate : [0.05,0.10,0.15,0.20] max_depth: [3, 4, 5, 6] gamma": [0.0, 0.25, 0.50]	CNN-LSTM	Cell=100 →Dropout=0.1 Epoch=100
LightGBMClassifier	bagging_fraction=0.01 bagging_req=20 max_bin=32 learning_rate" : [0.05,0.10,0.15,0.20]	CNN-LSTM	Conv1D →Filter=32 Kernel=2 Dropout=0.2 Conv1D →Filter=32 Kernel=2 Dropout=0.2 RELU

Çizelge 5.3. Xaiver Uniform Algoirtması

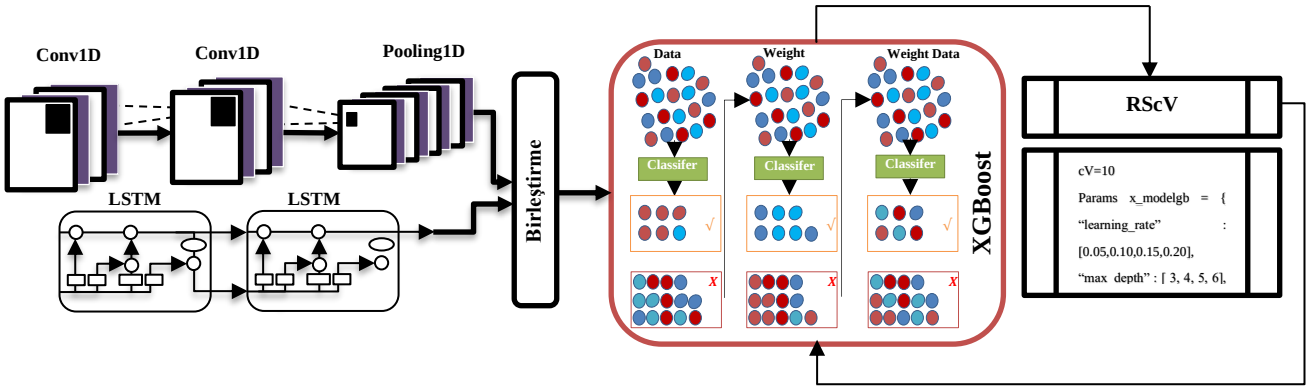
Algoritma *Pseudo kodu*

```
# önceki katmandaki düğüm değeri
n = 20
# ağırlık aralıkları
lwr, uppr = -(1.0 / sqrt(n)), (1.0 / sqrt(n))
# rastgele üretiliyor
numbers = rand(1000)
# ölçekleme işlemi
olcek = lwr + numbers * (uppr - lwr)
```

Önerilen modeli sonuçlarını genellemek için ise RScV(Randomize Search CV) özellikle XGBoost algoritmasını beslemek için kullanıldı. İkinci modelde CNN ve LSTM LightGBM kullanılmıştır.



Şekil 5.2. DL ve LightGBM ile Saldırı Tespiti



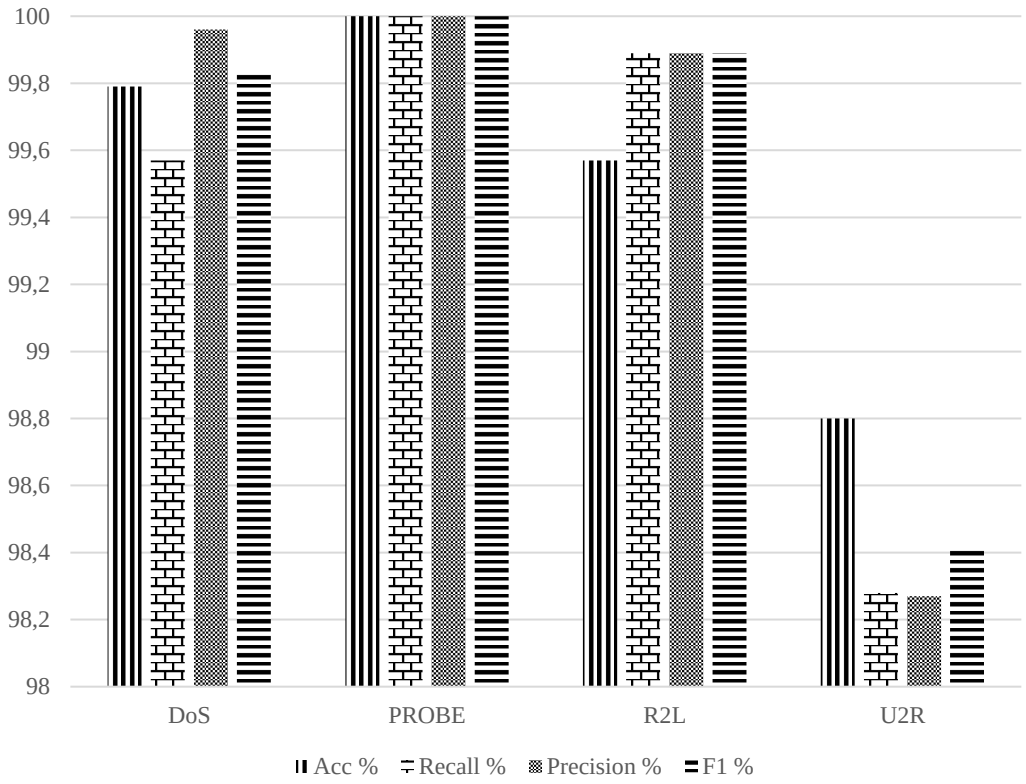
Şekil 5.3. DL ve XGBoost ile Saldırı Tespiti

5.1. MODELLERDE KULLANILAN VERİSETLERİ İLE ELDE EDİLEN SONUÇLAR

Önerilen iki model farklı saldırı tiplerini içeren NSL-KDD ve CIC-IDS2018 verisetleri üzerinde test edilmiştir. Testte karışıklık matrisi üzerinden elde edilen değerler, modellerin performans kriterlerini belirlemiştir. Bu değerler doğruluk, kesinlik, Recall ve F1-score 'un hesaplanması için kullanılmıştır. Çizelge 5.3'te NSL-KDD veri setinde elde edilen XGBoost algoritması ile test edilerek doğruluk, kesinlik, recall ve F1 puan değerleri elde edilmiştir. Tabloda önerilen bu saldırı tespit modelinde yaygın olarak görülen DoS saldırılarının %99,79 oranında başarı olduğu görülmektedir.

Çizelge 5.4. Derin Öğrenme ile XGBoost test sonuçları

Saldırı Tipi	**DL ve XGBoost**			
	Doğruluk %	**Recall %**	**P %**	**F1 %**
DoS	**99.79**	**99.57**	**99.96**	**99.83**
PROBE	100	100	100	100
R2L	99.57	99.89	99.89	99.89
U2R	98.80	98.28	98.27	98.41

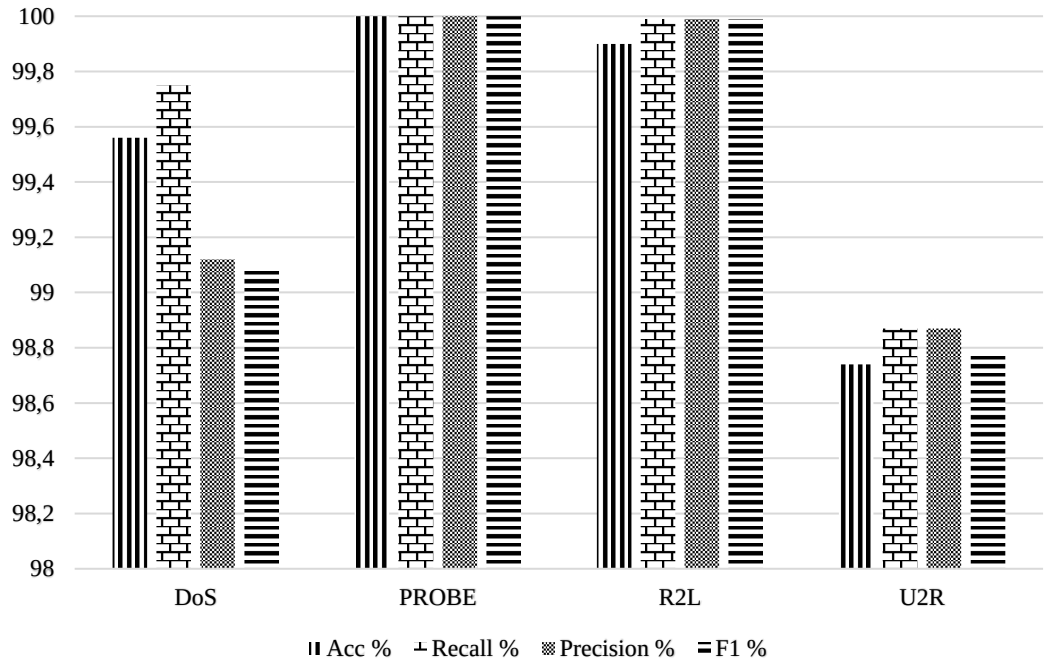


Şekil 5.4. NSL-KDD'de DL ve XGBoost ile Performans sonuçları

Çizelge 5.4'de NSL-KDD veri setinde elde edilen LightGBM algoritması ile test edilerek doğruluk, kesinlik, recall ve F1 puan değerleri elde edilmiştir. Tabloda önerilen bu saldırı tespit modelinde yaygın olarak görülen DoS saldırılarının %99,56 oranında başarı olduğu görülmektedir. LightGBM algoritması eğitim süresinin kısa sürede yapılmasıyla kaynak kullanımının da minimal düzeye indiği görülmüştür. LightGBM algoritmasının tahmin gücü yüksek ve öğrenim süresinin de kısa olması önerilen modelin saldırıların tespitindeki başarıyı arttırmıştır.

Çizelge 5.5. Derin Öğrenme ile LightGBM test sonuçları

Saldırı Tipi	**DL ve LightGBM**			
	Acc %	**Recall %**	**P %**	**F1 %**
DoS	**99.56**	**99.75**	**99.12**	**99.09**
PROBE	100	100	100	100
R2L	99.9	99.99	99.99	99.99
U2R	98.74	98.87	98.87	98.78

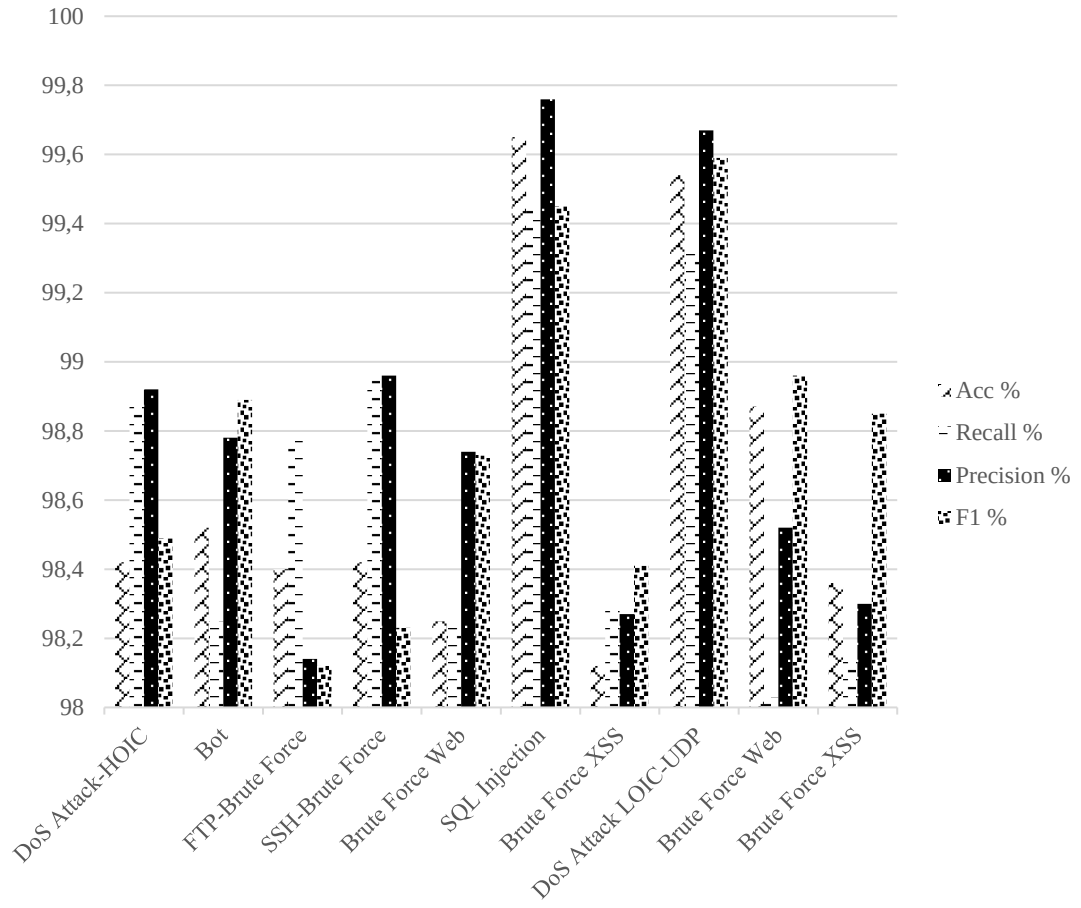


Şekil 5.5. NSL-KDD'de DL ve LightGBM ile Performans sonuçları

Çizelge 5.5'te CIC-IDS2018 veri setinde elde edilen XGBoost algoritması ile test edilerek Precision, Recall ve F1 puan değerleri elde edilmiştir. Çizelgede SQL Injection türündeki saldırıların %99,65 doğruluk oranı ile tespit edildiği görülmektedir. Burada XGBoost algoritmasının saldırıların sınıflandırılmasındaki başarısını learning rate oranın da düşük hassasiyet oranlarının etkilediği görülmüştür.

Çizelge 5.6. Derin Öğrenme ile XGBoost CIC-IDS2018 test sonuçları

Saldırı Tipi	**DL ve XGBoost**			
	Acc %	**Recall %**	**P %**	**F1 %**
DoS Attack-HOIC	98.42	98.87	98.92	98.49
Bot	98.52	98.25	98.78	98.89
FTP-Brute Force	98.40	98.78	98.14	98.12
SSH-Brute Force	98.42	98.96	98.96	98.23
Brute Force Web	98.25	98.23	98.74	98.73
SQL Injection	99.65	99.45	99.76	99.45
Brute Force XSS	98.12	98.28	98.27	98.41
DoS Attack LOIC-UDP	99.54	99.32	99.67	99.59
Brute Force Web	98.87	98.03	98.52	98.96
Brute Force XSS	98.36	98.14	98.30	98.85



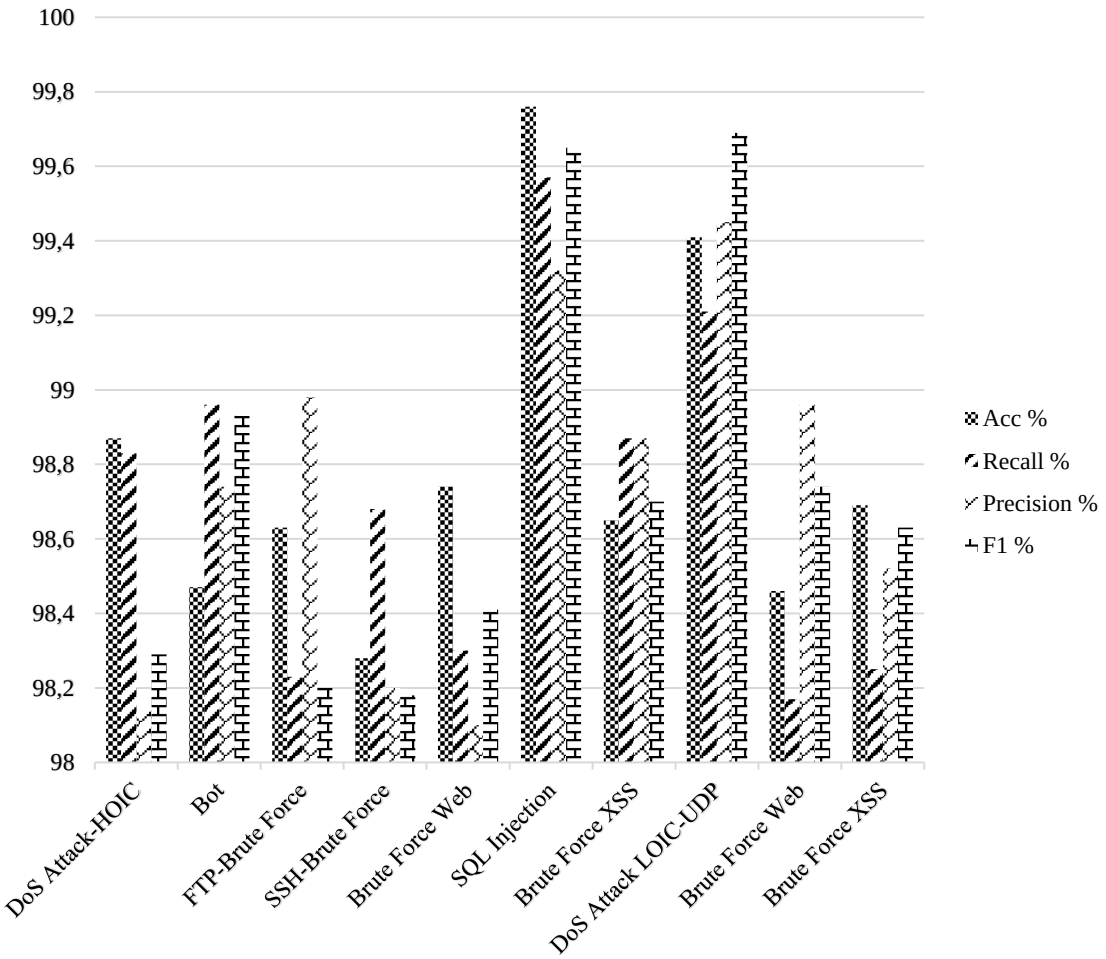
Şekil 5.6. CIC-IDS2018'de DL ve XGBoost performans sonuçları

Çizelge 5.6’da CIC-IDS2018 veri setinde elde edilen LightGBM algoritması ile test edilerek Precision, Recall ve F1 puan değerleri elde edilmiştir. SQL Injection türündeki saldırıların %99,76 doğruluk oranı ile tespit edildiği görülmektedir.

Çizelge 5.7. Derin Öğrenme ile LightGBM CIC-IDS2018 test sonuçları

Saldırı Tipi	**DL ve LightGBM**			
	Acc %	**Recall %**	**P %**	**F1 %**
DoS Attack-HOIC	98.87	98.83	98.14	98.29
Bot	98.47	98.96	98.74	98.93
FTP-Brute Force	98.63	98.23	98.98	98.20
SSH-Brute Force	98.28	98.68	98.20	98.18
Brute Force Web	98.74	98.30	98.10	98.41
SQL Injection	99.76	99.57	99.32	99.65
Brute Force XSS	98.65	98.87	98.87	98.7
DoS Attack LOIC-UDP	99.41	99.21	99.45	99.69
Brute Force Web	98.46	98.17	98.96	98.74
Brute Force XSS	98.69	98.25	98.52	98.63

Şekil 5.7. CIC-IDS2018'de DL ve LightGBM performans sonuçları



5.2. ÖNERİLEN MODELLERİN SALDIRI TESPİT SONUÇLARI

Çizelge 5.8. Önerilen STS modellerinin mevcut çalışmalara göre doğruluk performans karşılaştırılması I

Yapılan Çalışma Adı	**Kullanılan Yöntem**	**Doğruluk (%)**
New Intrusion Detection System Based on Support Vector Domain Description with Information Gain Metric (2018)[18]	SSPVSVDD	77,5
A Distributed Network Intrusion Detection System for DDoS in Vehicular Ad Hoc Network (2019) [35]	RF	98,95
Feature data processing: Making medical data fit deep neural networks (2019) [22]	DNN-RNN	86,86
Improving the Classification Effectiveness of IDS by Using Improved Conditional Variational Autoencoder and DL (2019) [36]	MDPCA and DBN	82,02
Fast Binary Network Intrusion Detection based on Matched Filter Optimization (2020) [37]	J48, NB, RF, MLP and SVM	76,67
A Hybrid Modified Grasshopper Optimization Algorithm and Genetic Algorithm to Detect and Prevent DDoS Attacks (2020) [38]	NB, J48, RF, KNN, MLP and SVM	99,67
Intrusion Detection Model of Algorithm Based on Mean Control (2020) [39]	CNN-BiLSTM	99,10
Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform (2020) [24]	MLP, RNN, LSTM	79,80,81
A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system (2023) [40]	Bagging GBM	94,66
An in-depth experimental study of anomaly detection using gradient boosted machine (2019) [20]	RF	85,00
Anomaly Detection Using XGBoost Ensemble of Deep Neural Network Models (2021) [41]	CNN	99,50
IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning Methods and NSL-KDD (2022) [34]	LSTM BILSTM	98
Deep Learning for Improving Attack Detection System Using CSE-CICIDS2018 (2022) [42]	CNN, LSTM, RNN	98,31
Denial-of-Service (DoS) Threat Detection Using Supervised Machine Learning Algorithms on CICIDS2018 (2022) [43]	NB, RF, J48	92,63
Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework (2022) [44]	DNN-LSTM	99.36
Hybrid IDS using MapReduce based Black Widow Optimized Convolutional LSTM Neural Networks (2021) [45]	LSTM	98,67

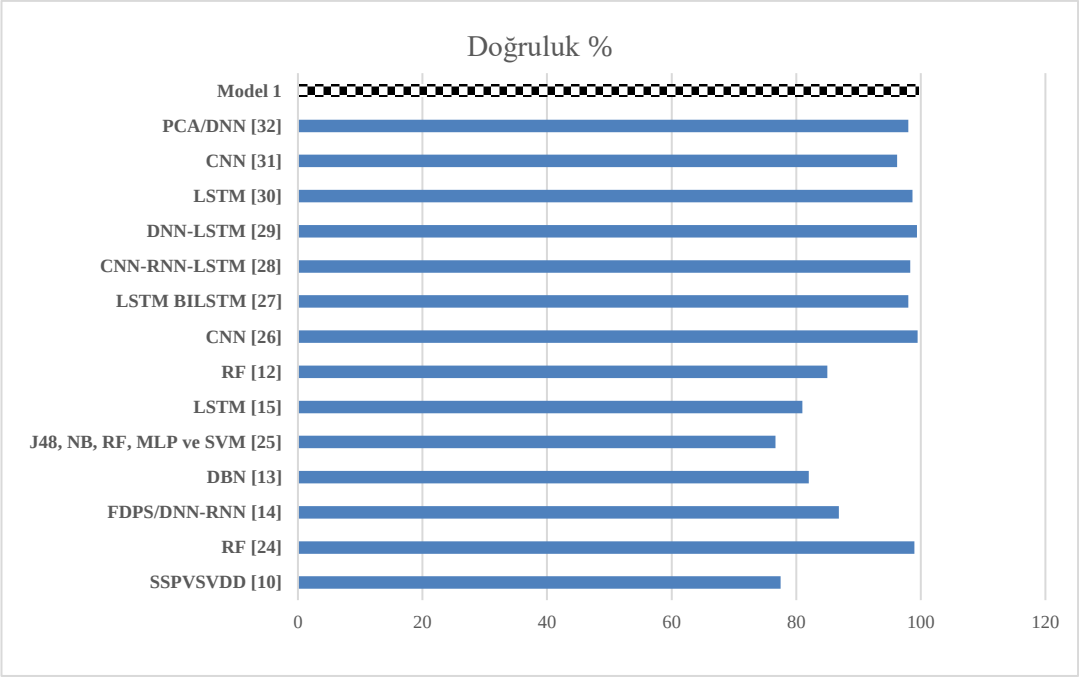
Çizelge 5.9. Önerilen STS modellerinin mevcut çalışmalara göre doğruluk performans karşılaştırılması II

Yapılan Çalışma Adı	**Kullanılan Yöntem**	**Doğruluk (%)**
Proposing a Model for Detecting Intrusion Network Attacks Using Machine Learning Techniques (2022) [46]	CNN	96,20
Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior (2022) [47]	PCA-DNN	98
Önerilen Model 1 (NSL-KDD)	**CNN-LSTM/ XGBoost**	**99.56**
Önerilen Model 2 (NSL-KDD)	**CNN-LSTM/LightGBM**	**99.79**
Önerilen Model 3 (CIC-IDS2018)	**CNN-LSTM/ XGBoost**	**98,42**
Önerilen Model 4 (CIC-IDS2018)	**CNN-LSTM/LightGBM**	**98.87**

Çalışma sonucunda elde edilen verilere ait model bazlı karşılaştırma sonuçları çizelge ve şekillerle gösterilmiştir. Elde edilen sonuçlar ile önerilen modeller arasında en iyi doğruluk sonucunu veren Model 2'nin olduğu görülmektedir.

Çizelge 5.10. 1. STS Modeli'nin literatürde kullanılan yöntemlere doğruluk perfomansı

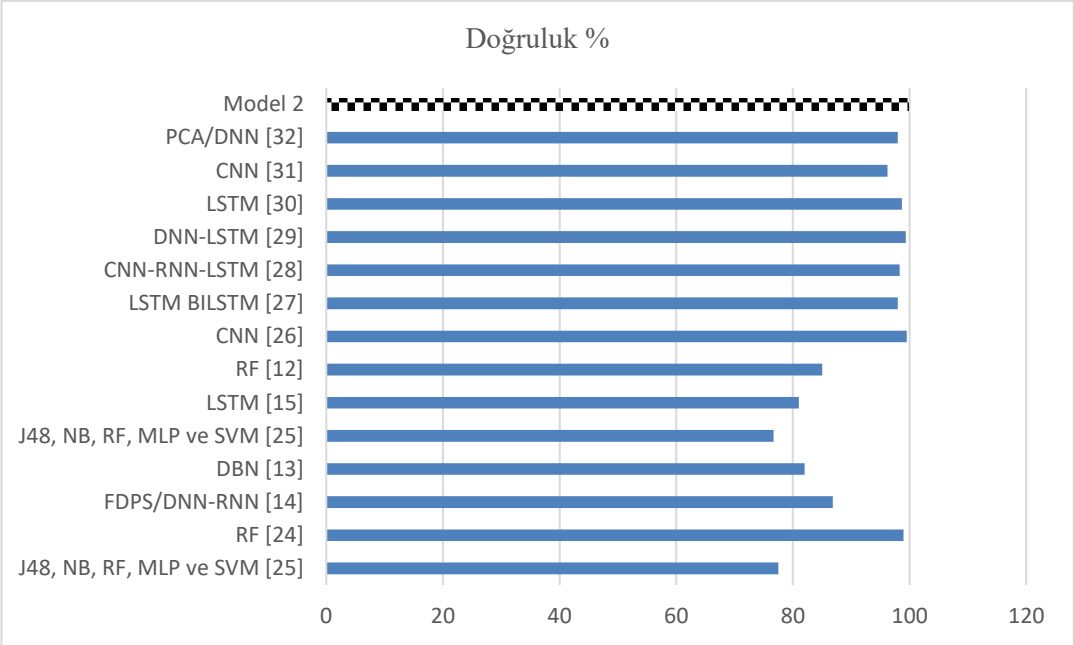
Kullanılan Yöntemler	**Doğruluk (%)**
SSPVSVDD [48]	77,5
RF [49]	98,95
FDPS/DNN-RNN [22]	86,86
DBN [21]	82,02
J48, NB, RF, MLP and SVM [37]	76,67
LSTM [24]	81
RF [20]	85
CNN [41]	99,50
LSTM BILSTM [34]	98
CNN-RNN-LSTM [42]	98,31
DNN-LSTM [50]	99,36
LSTM [45]	98,67
CNN [46]	96,20
PCA/DNN [47]	98
Önerilen Model 1	**99,56**



Şekil 5.8. 1. STS Modelinin mevcut modellere göre doğruluk performansı

Çizelge 5.11. 2. STS Modeli'nin literatürde kullanılan yöntemlere doğruluk perfomansı

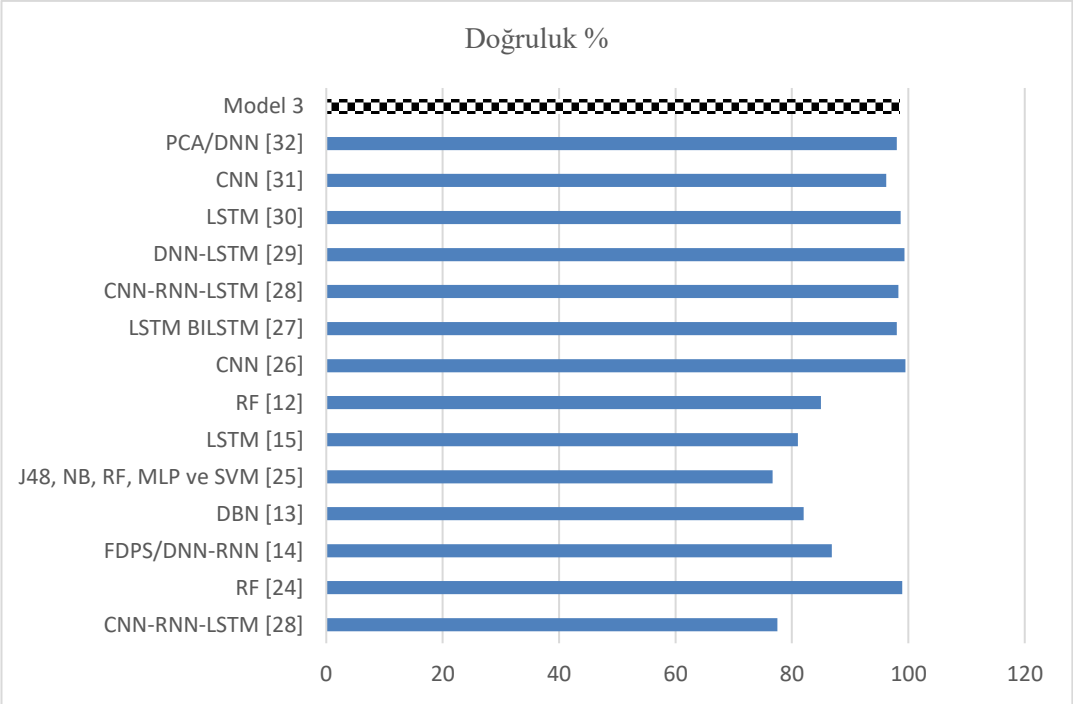
Kullanılan Yöntemler	**Doğruluk (%)**
SSPVSVDD [48]	77,5
RF [49]	98,95
FDPS/DNN-RNN [22]	86,86
DBN [21]	82,02
J48, NB, RF, MLP and SVM [37]	76,67
LSTM [24]	81
RF [20]	85
CNN [41]	99,50
LSTM BILSTM [34]	98
CNN-RNN-LSTM [42]	98,31
DNN-LSTM [50]	99,36
LSTM [45]	98,67
CNN [46]	96,20
PCA/DNN [47]	98
Önerilen Model 2	**99,79**



Şekil 5.9. 2. STS Modelinin mevcut modellere göre doğruluk performansı

Çizelge 5.12. 3. STS Modeli'nin literatürde kullanılan yöntemlere doğruluk perfomansı

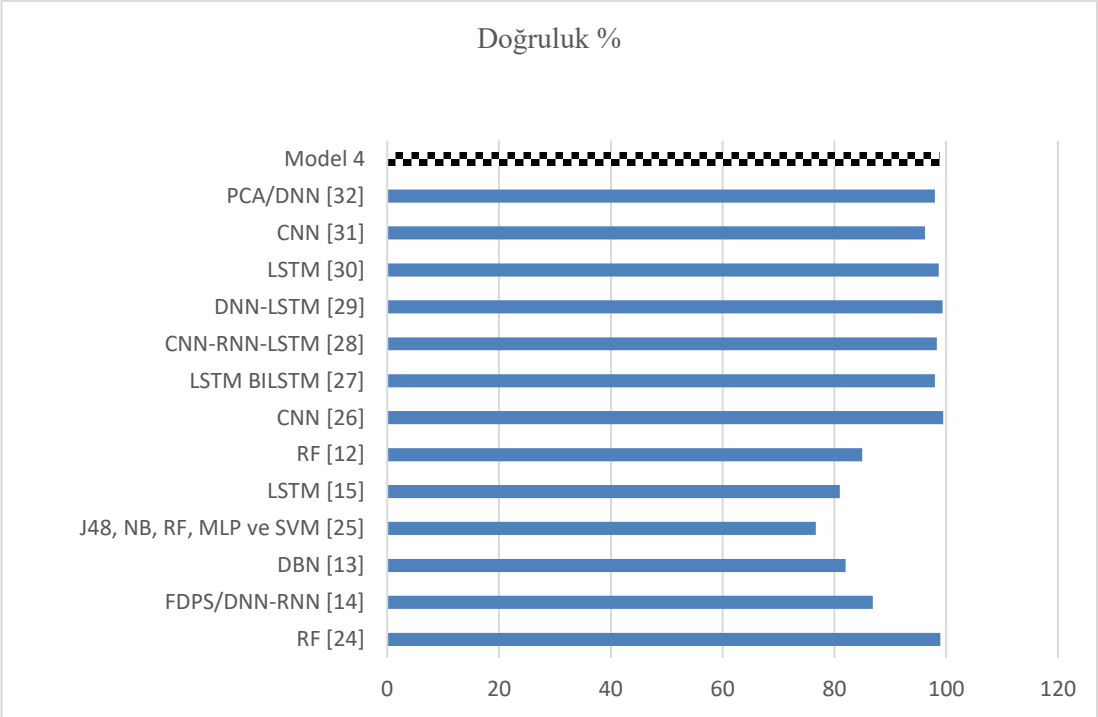
Kullanılan Yöntemler	**Doğruluk (%)**
SSPVSVDD [48]	77,5
RF [49]	98,95
FDPS/DNN-RNN [22]	86,86
DBN [21]	82,02
J48, NB, RF, MLP and SVM [37]	76,67
LSTM [24]	81
RF [20]	85
CNN [41]	99,50
LSTM BILSTM [34]	98
CNN-RNN-LSTM [42]	98,31
DNN-LSTM [50]	99,36
LSTM [45]	98,67
CNN [46]	96,20
PCA/DNN [47]	98
Önerilen Model 3	**98,42**



Şekil 5.10. 3. STS Modelinin mevcut modellere göre doğruluk performansı

Çizelge 5.13. 4. STS Modeli'nin literatürde kullanılan yöntemlere doğruluk perfomansı

Kullanılan Yöntemler	**Doğruluk (%)**
SSPVSVDD [48]	77,5
RF [49]	98,95
FDPS/DNN-RNN [22]	86,86
DBN [21]	82,02
J48, NB, RF, MLP and SVM [37]	76,67
LSTM [24]	81
RF [20]	85
CNN [41]	99,50
LSTM BILSTM [34]	98
CNN-RNN-LSTM [42]	98,31
DNN-LSTM [50]	99,36
LSTM [45]	98,67
CNN [46]	96,20
PCA/DNN [47]	98
Önerilen Model 4	**98,87**



Şekil 5.11. 4. STS Modelinin mevcut modellere göre doğruluk performansı

BÖLÜM 6

SONUÇLAR VE TARTIŞMA

Yapılan tez çalışmasında iki aşamalı sonuçlar elde edilmiştir. Çalışmanın birinci aşamasında öznitelik seçiminin önemi ve saldırıların tespitine etkisi araştırılmıştır. Bu aşama sonunda bir siber saldırının tespit edilmesi için, NSL-KDD veri setinden elde edilen öznitelikler Random Forest, Naive Bayes, J48 ve MLP algoritmalarıyla sınıflandırılmıştır. Sınıflandırma sonuçları literatürde belirtilen kriterler incelenerek, P, FAR, Acc, ve ADR 'ye göre değerlendirilmiştir. Makine öğrenmesi yöntemleri, ağdaki anomalinin tespit edilmesi için yanlış alarm oranı, doğruluk ve tespit oranı kullanılarak karşılaştırılmıştır. Seçilen 10 öznitelik ile yapılan anomali tespitinde Random Forest Algoritması %99,76; J48 Algoritması %98,45; Naive Bayes Algoritması %93,34 ve MLP Algoritması %91,34 oranında başarım göstermiştir. NSL-KDD verisetinin içerdiği DoS, Probe, R2L ve U2R siber atak tipleri Random Forest, Naive Bayes, J48 ve Multi-Layer Perceptron sınıflandırma algoritmaları ile analiz edilmiştir. Makine öğrenmesi algoritmalarının saldırıları tespit etme başarım ölçütü olarak P, ROC, F1 Score, Recall ve doğruluk kriterleri alınmıştır. Random Forest Algoritması Probe attack tespitinde %99,952; DoS attack tespitinde %99,842; Remote to Local attack tespitinde %99,99 ve user root attack tespitinde %99,859 doğruluk göstermiştir. Random Forest Algoritması, karşılaştırılan diğer algoritmalara göre farklı türdeki siber saldırıların tespitinde %1,7 oranında doğruluk göstermiştir. Öznitelik seçim yöntemi ve seçilen öznitelik sayısı saldırı tespit sistemleri için önemli bir parametre olduğu görülmüştür.

Çalışmanın ikinci aşamasında saldırı tespitine yönelik bir hibrit model üzerine çalışma yapılmıştır. Son aşamada ise derin öğrenme ve makine öğrenimi algoritmaları kullanarak tasarlanan hibrit bir saldırı tespit sistemi önerilmiştir. İki derin öğrenme algoritması ve iki makine öğrenmesi algoritması ile dört farklı model ortaya çıkarılmıştır. Önerilen modellerde yaygın olarak kullanılan NSL-KDD ve CIC-IDS2018 verisetleri kullanılarak 14 farklı tehdit türünü tespit etmeyi başarmıştır.

Saldırı tespit performansı doğruluk, kesinlik, recall ve F1 score parametreleri ile gösterilmiştir. Önerilen hibrit model özellikle SQL Injection, Brute Force ve DDoS saldırılarında doğruluk noktasında yüksek performans göstermiştir. Önerilen modeller mevcut çalışmalarla karşılaştırılarak performans karşılaştırılması yapılmıştır. Seçilen özniteliklerin seçim yöntemleri modelin performans ölçümleri üzerinde etkisinin yüksek olduğu görülmüştür. Gelecek çalışmalarda parametreleri otomatik olarak seçen algoritmaların geliştirilerek, saldırı tespit süreleri düşürülecek modeller geliştirilebileceği öngörülmektedir. Ayrıca önerilen bu saldırı tespit modelleri kullanılarak, yaygın olarak görülen sıfırıncı gün saldırılarına karşı yeni yaklaşımlar geliştirileceği düşünülmektedir.

KAYNAKLAR

1. Özalp, A. N. and Albayrak, Z., "Detecting Cyber Attacks with High-Frequency Features using Machine Learning Algorithms", *Acta Polytechnica Hungarica*, 19 (7): 213–233 (2022).
2. Nguyen, M. T. and Kim, K., "Genetic convolutional neural network for intrusion detection systems", *Future Generation Computer Systems*, 113: 418–427 (2020).
3. Dwivedi, S., Vardhan, M., and Tripathi, S., "Distributed Denial-of-Service Prediction on IoT Framework by Learning Techniques", *Open Computer Science*, 10 (1): 220–230 (2020).
4. Ahmad, J., Farman, H., and Jan, Z., "Deep Learning Methods and Applications", SpringerBriefs in Computer Science, (2019).
5. Gamage, S. and Samarabandu, J., "Deep learning methods in network intrusion detection: A survey and an objective comparison", *Journal Of Network And Computer Applications*, 169 (August): 102767 (2020).
6. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., and Karimipour, H., "Cyber intrusion detection by combined feature selection algorithm", *Journal Of Information Security And Applications*, 44: 80–88 (2019).
7. Lakshminarayana, D. H., Philips, J., and Tabrizi, N., "A survey of intrusion detection techniques", (2019).
8. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., and Janicke, H., "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study", *Journal Of Information Security And Applications*, 50: 102419 (2020).
9. Jin, D. and Lu, Y., "KC-IDS : Multi-layer Intrusion Detection System", (2020).
10. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., and Janicke, H., "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study", *Journal Of Information Security And Applications*, 50: (2020).

11. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., and Alazab, A., "Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine", *Electronics (Switzerland)*, 9 (1): (2020).
12. Kasongo, S. M. and Sun, Y., "A deep learning method with filter based feature engineering for wireless intrusion detection system", *IEEE Access*, 7: 38597–38607 (2019).
13. Hu, Z., Wang, L., Qi, L., Li, Y., and Yang, W., "A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network", *IEEE Access*, 8: 195741–195751 (2020).
14. Latah, M. and Toker, L., "Minimizing false positive rate for DoS attack detection: A hybrid SDN-based approach", *ICT Express*, 6 (2): 125–127 (2020).
15. Jiang, K., Wang, W., Wang, A., and Wu, H., "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network", *IEEE Access*, 8: 32464–32476 (2020).
16. Su, T., Sun, H., Zhu, J., Wang, S., and Li, Y., "BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset", *IEEE Access*, 8: 29575–29585 (2020).
17. Mohammed, B. and Gbashi, E., "Intrusion Detection System for NSL-KDD Dataset Based on Deep Learning and Recursive Feature Elimination", *Engineering And Technology Journal*, 39 (7): 1069–1079 (2021).
18. el Boujnouni, M. and Jedra, M., "New intrusion detection system based on support vector domain description with information gain metric", *International Journal Of Network Security*, 20 (1): 25–34 (2018).
19. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., and Wang, C., "Machine Learning and Deep Learning Methods for Cybersecurity", *IEEE Access*, (2018).
20. Tama, B. A. and Rhee, K. H., "An in-depth experimental study of anomaly detection using gradient boosted machine", *Neural Computing And Applications*, 31 (4): 955–965 (2019).

21. Yang, Y., Zheng, K., Wu, C., and Yang, Y., "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network", *Sensors (Switzerland)*, 19 (11): (2019).
22. Xu, Y., Liu, Z., Li, Y., Hou, H., Cao, Y., Zhao, Y., Guo, W., and Cui, L., "Feature data processing: Making medical data fit deep neural networks", *Future Generation Computer Systems*, 109: 149–157 (2020).
23. Jiang, K., Wang, W., Wang, A., and Wu, H., "Network Intrusion Detection Combined Hybrid Sampling with Deep Hierarchical Network", *IEEE Access*, 8: 32464–32476 (2020).
24. Haggag, M., Tantawy, M. M., and El-Soudani, M. M. S., "Implementing a deep learning model for intrusion detection on apache spark platform", *IEEE Access*, 8 (Dl): 163660–163672 (2020).
25. Karatas, G., Demir, O., and Sahingoz, O. K., "Deep Learning in Intrusion Detection Systems", (2019).
26. Zhou, Q. and Pezaros, D., "Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection -- An Analysis on CIC-AWS-2018 dataset", (July): (2019).
27. Kim, T. and Pak, W., "Hybrid Classification for High-Speed and High-Accuracy Network Intrusion Detection System", *IEEE Access*, 9: 83806–83817 (2021).
28. Kanimozhi, V. and Jacob, T. P., "Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing", *ICT Express*, 5 (3): 211–214 (2019).
29. Yu, L., Dong, J., Chen, L., Li, M., Xu, B., Li, Z., Qiao, L., Liu, L., Zhao, B., and Zhang, C., "PBCNN: Packet Bytes-based Convolutional Neural Network for Network Intrusion Detection", *Computer Networks*, 194 (January): 108117 (2021).
30. Basnet, R. B., Shash, R., Johnson, C., Walgren, L., and Doleck, T., "Towards detecting and classifying network intrusion traffic using deep learning frameworks", *Journal Of Internet Services And Information Security*, 9 (4): 1–17 (2019).

31. Kim, T. and Pak, W., "Hybrid Classification for High-Speed and High-Accuracy Network Intrusion Detection System", *IEEE Access*, 9: 83806–83817 (2021).
32. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., and Venkatraman, S., "Deep Learning Approach for Intelligent Intrusion Detection System", *IEEE Access*, (2019).
33. Ozalp, A. N., Albayrak, Z., Cakmak, M., and Ozdogan, E., "Layer-based examination of cyber-attacks in IoT", *HORA 2022 - 4th International Congress On Human-Computer Interaction, Optimization And Robotic Applications, Proceedings*, (2022).
34. Esmaeili, M., Goki, S. H., Hajipour, B., Masjidi, K., Sameh, M., Gharagozlou, H., and Mohammed, A. S., "ML-DDoSnet : IoT Intrusion Detection Based on Denial-of-Service Attacks Using Machine Learning Methods and NSL-KDD", 2022: (2022).
35. Gao, Y., Wu, H., Song, B., Jin, Y., Luo, X., and Zeng, X., "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network", *IEEE Access*, 7: 154560–154571 (2019).
36. Yang, Y., Zheng, K., Wu, C., and Yang, Y., "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network", *Sensors (Switzerland)*, 19 (11): (2019).
37. Alsaadi, H. S., Hedjam, R., Touzene, A., and Abdessalem, A., "Fast Binary Network Intrusion Detection based on Matched Filter Optimization", *2020 IEEE International Conference On Informatics, IoT, And Enabling Technologies, ICIoT 2020*, 195–199 (2020).
38. Mohammadi, S. and Babagoli, M., "A hybrid modified grasshopper optimization algorithm and genetic algorithm to detect and prevent DDoS attacks", *International Journal Of Engineering, Transactions A: Basics*, 34 (4): 811–824 (2021).
39. Zhang, L., Huang, J., Zhang, Y., and Zhang, G., "Intrusion Detection Model of CNN-BiLSTM Algorithm Based on Mean Control", *Proceedings Of The IEEE International Conference On Software Engineering And Service Sciences, ICSESS*, 2020-Octob: 22–27 (2020).

40. Louk, M. H. L. and Tama, B. A., "Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system", *Expert Systems With Applications*, 213 (PB): 119030 (2023).
41. Ikram, S. T., Cherukuri, A. K., Poorva, B., Ushasree, P. S., Zhang, Y., Liu, X., and Li, G., "Anomaly Detection Using XGBoost Ensemble of Deep Neural Network Models", *Cybernetics And Information Technologies*, 21 (3): 175–188 (2021).
42. Hagar, A. A., "Deep Learning for Improving Attack Detection System Using CSE-CICIDS2018", (August): (2022).
43. Saleh, A. J. M. and Adnan, N., "Denial-of-Service (DoS) Threat Detection Using Supervised Machine Learning Algorithms on CICIDS2018 Dataset", *Lecture Notes In Networks And Systems*, 437 (October): 519–533 (2022).
44. al Razib, M., Javeed, D., Khan, M. T., Alkanhel, R., and Muthanna, M. S. A., "Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework", *IEEE Access*, 10 (3): 53015–53026 (2022).
45. Kanna, P. R. and Santhi, P., "Hybrid Intrusion Detection using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks", *Expert Systems With Applications*, 194 (October 2021): 116545 (2022).
46. Ali, T. A. J. and Jawhar, M., "Proposing a Model for Detecting Intrusion Network Attacks Using Machine Learning Techniques", *Journal Of Education And Science*, 31 (3): 99–109 (2022).
47. Al-Fawa'reh, M., Al-Fayoumi, M., Nashwan, S., and Fraihat, S., "Cyber threat intelligence using PCA-DNN model to detect abnormal network behavior", *Egyptian Informatics Journal*, 23 (2): 173–185 (2022).
48. el Boujnouni, M. and Jedra, M., "New intrusion detection system based on support vector domain description with information gain metric", *International Journal Of Network Security*, 20 (1): 25–34 (2018).
49. Gao, Y., Wu, H., Song, B., Jin, Y., Luo, X., and Zeng, X., "A distributed network intrusion detection system for distributed denial of service attacks in vehicular ad hoc network", *IEEE Access*, 7: 154560–154571 (2019).

50. al Razib, M., Javeed, D., Khan, M. T., Alkanhel, R., and Muthanna, M. S. A., "Cyber Threats Detection in Smart Environments Using SDN-Enabled DNN-LSTM Hybrid Framework", *IEEE Access*, 10: 53015–53026 (2022).

ÖZGEÇMİŞ

Ahmet Nusret ÖZALP, Ankara Yenimahalle Teknik Lisesi, Bilgisayar bölümünden mezun olduktan sonra 1999 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi bölümüne girdi. 2003 yılında mezun oldu. 2012 yılında Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde yüksek lisansını tamamladı. 2015 yılında Anadolu Üniversitesi İktisat Fakültesi Kamu Yönetimi bölümünü tamamladı. 2017 yılında Karabük Üniversitesi Kamu Yönetimi bölümünde yüksek lisansını tamamladı. 2021 yılında lisans olarak Karabük Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünü tamamladı. 2003 yılından bu yana Milli Eğitim Bakanlığı'nda Bilişim Teknolojileri Öğretmeni olarak görevini sürdürmektedir.

YAYINLAR

1. **Özalp, A. N**., & Albayrak, Z. (2022). Detecting Cyber Attacks with High-Frequency Features using Machine Learning Algorithms. **Acta Polytechnica Hungarica**, 19(7).

2. **ÖZALP, A. N.,** Albayrak, Z., Çakmak, M., & Özdoğan, E. (2022, June). Layer-Based Examination of Cyber-Attacks in IoT. In 2022 **International Congress On Human-Computer Interaction, Optimization And Robotic Applications (Hora**) (pp. 1-10). IEEE.

3. Altamimi, M., Albayrak, Z., Çakmak, M., & **ÖZALP, A. N.** (2022). BGP Anomaly Detection Using Association Rule Mining Algorithm. **Avrupa Bilim ve Teknoloji Dergisi**, (42), 134-139.

4. Uluer, A. F., Albayrak, Z., **ÖZALP, A. N.,** Çakmak, M., & Altunay, H. C. (2022, May). BGP Anomali Tespitinde Hibrit Model Yaklaşımı. In 2022 **30th Signal Processing and Communications Applications Conference** (SIU) (pp. 1-4). IEEE.

5. Altunay, H. C., Albayrak, Z., **ÖZALP, A. N.,** & Çakmak, M. (2021, June). Analysis of anomaly detection approaches performed through deep learning methods in SCADA systems. In 2021 **3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA**) (pp. 1-6). IEEE.

6. **ÖZALP, A. N**., Albayrak, Z., (2019). International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES). Classification of Solution Analyzes for the elimination of security requirements in IoTs operating in Cyber Physical System (CPS).

7. **ÖZALP, A.N**., & FINDIK, O.(2018) Steganaliz Tekniklerinde Verimlilik Analizleri. Akademik Bilişim Konferansı 2018.

8. **ÖZALP, A. N.,** Albayrak, Z., & Zengin, A. (2017, September). Expansion of Wireless Networks using IEEE 802.3 af Protocol in Protected Areas. In **5th International Symposium on Innovative Technologies in Engineering and Science** 29-30 September 2017 (ISITES2017 Baku-Azerbaijan).