T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ ALGORİTMALARININ HİBRİT YAKLAŞIMI İLE AĞ ANOMALİSİ TESPİTİ

YÜKSEK LİSANS TEZİ

Feyza ÖZGER

Enstitü Anabilim Dalı	**:**	**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**
Tez Danışmanı	**:**	**Doç. Dr. Halit ÖZTEKİN**

Ekim 2023

T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ ALGORİTMALARININ HİBRİT YAKLAŞIMI İLE AĞ ANOMALİSİ TESPİTİ

YÜKSEK LİSANS TEZİ

Feyza ÖZGER

Enstitü Anabilim Dalı : ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ

Bu tez 04/10/2023 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

JÜRİ	BAŞARI DURUMU
Jüri Başkanı: Doç. Dr. Halit ÖZTEKİN	BAŞARILI
Üye: Dr. Öğr. Üyesi Ekin EKİNCİ	BAŞARILI
Üye: Dr. Öğr. Üyesi Kayhan AYAR	BAŞARILI

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim

Feyza ÖZGER

04/10/2023

TEŞEKKÜR

Tez çalışmam boyunca bana rehberlik eden, olumlu tavrıyla beni cesaretlendiren, bilgi birikimi ile çalışmama destek veren Sayın Doç. Dr. Halit ÖZTEKİN'e danışmanlığı için çok teşekkür ederim. Yüksek Lisans sürecinde ve hayatımın her anında maddi ve manevi katkılarıyla, sabır ve anlayışlarıyla çalışmalarımdaki başarıma büyük katkısı olan aileme ve çalışmam süresince küçük veya büyük desteğini esirgemeyen herkese teşekkürü borç bilirim.

İÇİNDEKİLER

KISALTMALAR

ABC	: Artificial Bee Colony (Yapay Arı Kolonisi)
AE	: Auto Encoder
AIS	: Artificial Immune Systems
ARP	: Address Resolution Protocol
AUC	: Area Under the Curve
CICIDS2017	: Canadian Institute for Cybersecurity Intrusion Detection System
CNN	: Convolutional Neural Network (EvrişimliSinirAğı)
CSV	: Comma Separated VariabIes
DBN	: Deep Belief Networks
DBSCAN	: Density-based spatial clustering of applications with noise
DDoS	: Distributed Denial of Service
DNS	: Domain Name System
Dos	: Denial of Service
DRDoS	: Distributed Reflection Denial of Service
DT	: Decision Tree
FS	: Forward Selection
FPR	: False Positif Rate - YanlışPozitif Oran
GSA	:GravitasyonelAramaAlgoritması
ICA	: Independent component analysis
ICMP	: Internet Control Message Protocol-İnternet KontrolMesajProtokolü
IDS	: Intrusion Detection System
IP	: Internet Protocol
KDD Cup	: Annual Data Mining and Knowledge Discovery competition
KNN	: K-nearest Neighbor
LR	: Logistic Regression
LSTM	: Long Short-Term Memory

LVQ	: ÖğrenmeVektörüNicemleme
MAC	: Media Acsess Control
MANET	: Mobile Ad-hoc Networks (Mobil GeçiciAğ)
MITM	: Man-in-the-Middle
MLP	: Multilayer Perceptron
MÖ	:MakineÖğrenmesi
NB	: Naive Bayes
P2P	: Peer ToPeer
PCA	: Principal Component Analysis (TemelBileşenAnalizi)
PSO	: ParçacıkSürüOptimizasyonu
R2L	: Remote to User
RF	: Random Forest
RFE	: Recursive Feature Elemination (ÖzyinelemeliÖzellikSeçimi)
RNN	: Recurrent Neural Network
ROC	: Receiver Operating Characteristic
SDN	: Software Defined Network
SQL	: Structured Query Language
SVD	:Singular Value Decomposition
SVM	: Support Vektör Machine (DestekVektörMakinleri)
SYN	: Synchronize
TCP	: Transmission Control Protocol
TPR	: True Positive Rate
U2R	: User to Root
UDP	: User Datagram Protocol
UNSW-NB15	: University of New South Wales - Network-based 15
URL	: Uniform Resource Locator
WPT	: Wavelet Packet Transmission (Wavelet PeaketDönüşümü)
WSN	: KablosuzSensörAğlarında
XSS	: Cross-Site Scripting

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

MAKİNE ÖĞRENMESİ ALGORİTMALARININ HİBRİT YAKLAŞIMI İLE AĞ ANOMALİSİ TESPİTİ

ÖZET

İnternet, insanların iletişim kurması, bilgiye erişimi sağlaması, ticaret yapması ve birçok günlük aktivitesini gerçekleştirmesi için hayati bir öneme sahiptir. Ancak, bu artış beraberinde siber saldırılar ve tehditlerin de artmasına neden olmuştur. Siber saldırganlar her geçen gün daha sofistike yöntemler geliştirerek kişisel verileri çalmak, sistemlere zarar vermek veya hizmetleri engellemek gibi kötü niyetli eylemlerde bulunmaktadır. Bu durum, siber güvenlikte tespit sistemlerinin önemini daha da artırmıştır. Özellikle network anomali tespiti gibi sistemler, ağ trafiğindeki normal davranışları öğrenerek beklenmeyen veya anormal aktiviteleri tespit edebilmektedir. Bu sayede saldırıların erken aşamada tespit edilmesi ve önlenmesi sağlanmaktadır. Bu teknolojiler, bireylerin ve organizasyonların bilgilerini koruyarak dijital saldırıların potansiyel hasarını minimize etmeye yardımcı oluyor. Bu nedenle, siber güvenlik algılama sistemlerine yönelik araştırmalar kritik bir değere sahiptir.

Hibrit modeller ile siber saldırı tespitinin yüksek başarıyla yapıldığı gözlemlenmiştir. Ağ anamolisinde kullanılan makine öğrenmesi algoritmalarının performansları genellikle KDD Cup 1999 veri kümesi üzerinde değerlendirilmiştir. Araştırmada, genellikle yüksek doğruluk seviyeleri gösterdikleri ve literatürde sıkça tercih edildikleri için Karar Ağacı (DT), Lojistik Regresyon (LR), Naive Bayes (NB), Rastgele Orman (RF) ve En Yakın Komşu (KNN) makine öğrenimi yöntemleri test edilmiştir. Veri madenciliği ve makine öğrenimi tekniklerinin ağ güvenliği alanındaki etkinliğini değerlendirmek amacıyla iki farklı hibrit öznitelik indirgeme yöntemi olan PCA + RFECV ve RFECV + FS yöntemleri karşılaştırılmıştır. PCA + RFECV yönteminde, temel bileşen analizi ile boyut indirgeme yapılmış ve ardından Recursive Feature Elimination with Cross-Validation (RFECV) yöntemi ile en iyi öznitelikler seçilmiştir. Değerlendirme metrikleri olarak, Çapraz Doğrulama ve ROC eğrileri tercih edilmiştir; bu metriklerin seçimi, algoritmaların performansının kapsamlı ve objektif bir şekilde analiz edilmesini sağlaması amacıyla yapılmıştır. Öznitelik indirgeme uygulanmadan, RF sınıflandırıcısı %98,15 ile en yüksek doğrulukta iken, KNN %96,31 doğruluk, %97,41 kesinlik, %95,24 duyarlılık ve %96,31 F1 skoru ile dikkat çekmiştir. PCA + RFECV uygulamasında, KNN'nin metrikleri benzer kalmış fakat NB sınıflandırıcısında %61,93 doğruluk ile büyük bir düşüş gözlemlenmiştir. RFECV + FS kullanıldığında, KNN %96,68 doğruluk, %97,76 kesinlik, %95,64 duyarlılık ve %96,69 F1 skoru ile öne çıkmıştır, bu da öznitelik indirgeme yöntemlerine duyarlılığını vurgulamaktadır. Sonuçlar, öznitelik seçiminin sınıflandırma performansındaki kritik rolünü vurgulamakta olup, veri kümesinin boyutunu azaltma, anlamlı öznitelikleri seçme ve hibrit yöntemler kullanma stratejilerinin sınıflandırma performansını artırabileceğini ortaya koymaktadır.

Anahtar Kelimeler: Ağ güvenliği, denetimli öğrenme, makine öğrenmesi, metasezgisel algoritmalar, KDD Cup 1999.

NETWORK ANOMALY DETECTION WITH A HYBRID APPROACH OF MACHINE LEARNING ALGORTHMS

ABSTRACT

The Internet has become an essential medium for people to communicate, access information, conduct business, and carry out many daily activities. However, this surge has also led to a rise in cyberattacks and threats. Cyber adversaries are increasingly devising sophisticated methods to steal personal data, harm systems, or disrupt services with malicious intent. This escalation underscores the critical importance of cybersecurity detection systems. Particularly, network anomaly detection systems, which learn normal behavior patterns in network traffic, can identify unexpected or anomalous activities. This facilitates the early detection and mitigation of attacks. Strengthening and developing cybersecurity detection systems is of paramount importance in today's digital landscape. These systems safeguard individual users and institutions by protecting their data and information, thereby minimizing potential damages from cyberattacks. Moreover, they prevent service interruptions, ensuring the seamless operation of the Internet. In this context, research and studies on cybersecurity detection systems hold immense significance.

The process of anomaly detection, aimed at identifying unexpected or deviant behaviors in datasets, is conducted through various techniques, prominently including machine learning, statistical methods, and data analysis techniques. This detection primarily focuses on identifying values that are either above or below the norm and holds critical importance across various domains. Anomaly detection techniques are primarily categorized into three main types: Point Anomaly, which defines situations where a single data point significantly deviates from the rest, such as an unexpected high transaction amount in a bank account. Contextual Anomaly pertains to the identification of a data point that is abnormal in relation to other data within a specific context; this could involve an unexpected change in network traffic. Collective Anomaly refers to the deviation of a combination of multiple features or attributes from the general behavior pattern; employee performance evaluations serve as an example for this kind of analysis. Applications of these detection methods span a wide range, from optimizing business processes to identifying potential threats.

Network Anomaly Detection is utilized to identify unexpected behaviors in computer networks. It operates based on three main methods: Signature-Based, which detects pre-defined patterns; Behavior-Based, which distinguishes between normal and abnormal behaviors using statistical parameters; and Machine Learning-Based, which classifies new and unknown anomalies through trained models. These techniques are of critical importance for optimizing the security and performance of networks.

Network attacks refer to threats aimed at computer networks and connected devices. The objective of these attacks is to engage in malicious activities such as seizing network resources, stealing data, causing service disruptions, or crashing the system. For instance, DDoS attacks aim to disrupt the service by flooding the network with excessive traffic. UDP Flood attacks can exhaust target system resources by persistently sending a large amount of UDP traffic. Smurf attacks cause service interruptions by bombarding network devices with deceptive ICMP Echo Request messages. Teardrop attacks induce crashes in the target machine by using faulty fragment information. Botnet attacks orchestrate infected devices to create service disruptions. Clickjacking permits malicious actions without the user's knowledge, while DRDoS attacks amplify the attack impact using reflection techniques. On the other hand, malware attacks seize devices with malicious software, and Man-in-the-Middle attacks monitor and alter communication within the network. Ransomware attacks demand payment from users, while password cracking attacks aim to decode passwords. Social engineering attacks target the theft of personal information; whereas SQL injection, XSS, and phishing attacks target websites. ARP, DNS, and IP spoofing attacks misdirect network traffic with the intention of stealing or monitoring information. Ping of Death and SYN Flood attacks target disrupting network services.

Popular test datasets used for network anomaly detection include NSL-KDD, UNSW-NB15, CICIDS2017, DARPA 1998, and KDDCUP99. NSL-KDD is an improved version of the KDD Cup 1999 dataset and contains 41 different network attack types. UNSW-NB15 consists of real network traffic data and is a detailed labeled set with 49 features; CICIDS2017 has 80 million event records encompassing 15 network attack types. DARPA1998 includes attacks conducted on a real network along with normal traffic, while KDDCUP99 is based on 5 million data samples obtained from a real computer network and has an imbalanced structure. These datasets serve as significant tools for network security and the training of machine learning algorithms.

In this research endeavor, we employed the Anaconda distribution and crafted the code using Python in the Jupyter Notebook environment. Anaconda serves as both a package handler and an environment, encompassing Python and various tools tailored for data manipulation and machine learning. Key libraries like Sklearn, Numpy, and Pandas played a role in our study. The machine used for this research runs a 64-bit Microsoft Windows OS and boasts suitable technical features.

- CPU: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- RAM: 16 GB

In this study, an initial overview of network attacks was presented, followed by an extensive literature review. Conducted researches were thoroughly evaluated, and frequently used supervised machine learning algorithms and metaheuristic algorithms for network anomaly detection were identified. Studies where machine learning and metaheuristic algorithms were used in a hybrid manner were also analyzed. The performance of machine learning algorithms was tested on the KDD Cup 1999 dataset. During the data preprocessing phase, data related to attacks and

normal traffic in the dataset were distributed evenly, with attack data labeled as 1 and normal traffic data labeled as 0. Missing values in the dataset were filled using the calculated median value. Categorical attributes (such as protocol_type, flag, service) were digitized using the one-hot encoding method. To scale the data and ensure they are on the same scale, the columns src_bytes and dst_bytes underwent normalization. Furthermore, a correlation matrix was calculated to measure the relationship between the features in the dataset, and highly correlated values were identified. In this case, the PCA method was employed to minimize the relationship between the features.

In the study, classification was performed using machine learning algorithms such as Decision Tree (DT), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), and K-Nearest Neighbors (KNN). Cross-Validation and ROC curves were employed as evaluation metrics. Additionally, to evaluate the role of data mining and machine learning methods in the field of network security, two distinct hybrid feature reduction methods, namely PCA + RFECV and RFECV + FS, were compared.

In the PCA + RFECV method, dimensionality reduction was conducted using principal component analysis, followed by the selection of the best features through the Recursive Feature Elimination with Cross-Validation (RFECV) method. In this method, the Random Forest (RF) classifier was observed to achieve the highest accuracy results, while the K-Nearest Neighbors (KNN) classifier was successful in terms of the precision metric.On the other hand, with the RFECV + FS method, important features were first identified with RFECV, followed by the selection of the best features using the Forward Selection (FS) method. In this method, the KNN classifier stood out with the highest accuracy, precision, sensitivity, and F1 metrics.

The results highlight the critical role of feature selection in classification performance, demonstrating that strategies of reducing dataset size, selecting meaningful features, and using hybrid methods can enhance classification performance. This study will be a valuable resource for academics researching network security and industrial organizations.

Keywords: Network security, supervised learning, machine learning, metaheuristic algorithms, KDDCup 1999.

BÖLÜM 1. GİRİŞ

Günümüzde web erişiminin yaygınlaşması, dijital risklerin de paralel bir şekilde çoğalmasına yol açmaktadır. Bu durum, kurum ve kuruluşların siber güvenlik önlemlerini artırmalarını zorunlu hale getirmiştir. Bu bağlamda, siber saldırganların ağa yaptığı anomali davranışlarının tespit edilmesi, siber güvenlik açısından büyük önem taşımaktadır.

Anomali tespiti, veri kümesindeki normal davranıştan farklı olan verileri tespit etme sürecidir ve genellikle veri analizi, makine öğrenmesi veya istatistiksel yöntemler kullanılarak gerçekleştirilir[1].

Anomali tespiti yöntemleri çeşitli faktörlere göre sınıflandırılabilir. Bu faktörlerden bazıları şunlardır[2]:

- Nokta Anomali: Bir veri setindeki belli bir veri öğesi, diğerlerine kıyasla belirgin bir biçimde ayrışabilir. Bu ayrışma, söz konusu veri öğesinin diğerlerinden çok daha yüksek ya da düşük bir değere sahip olması, alışılmadık bir niteliğe sahip olması veya diğer verilere göre oldukça farklı bir yapıya sahip olması şeklinde görülebilir. Örneğin, bir finansal veri kümesinde, normalde 1000 TL olan bir işlem için 100.000 TL ödenmesi veya normalde sadece 5 işlem yapılan bir hesapta birdenbire binlerce işlem yapılması gibi durumlarda nokta anomalisi oluşabilir[1].
- Bağlamsal Anomali: Bir veri noktasının normal davranıştan sapması durumunda ortaya çıkar. Bu tür bir anomali, veri kümesindeki diğer verilerin davranışlarına göre belirlenir ve bu nedenle veri noktasının anormal olup olmadığı, veri kümesinin geri kalanına göre karşılaştırılarak belirlenir. Bağlamsal anomali tespiti için, veri noktalarının doğal dağılımlarını ve beklenen davranışlarını belirlemek üzere istatistiksel yöntemler ve makine öğrenmesi teknikleri kullanılır. Bağlamsal anomali, özellikle ağ güvenliği ve siber güvenlik

alanlarında kullanılır. Örneğin, bir ağ üzerindeki bir cihazın normal işlevlerinden sapması durumunda, cihazın anomali olarak tanımlanması ve bir saldırı belirtisi olarak kabul edilmesi mümkündür. Benzer şekilde, bir web sitesinin ziyaretçi trafiğinin normal seyrinden sapması da bir bağlamsal anomali olarak kabul edilebilir ve potansiyel bir siber saldırının belirtisi olarak yorumlanabilir[1].

- Toplu Anomali: birden fazla özellik veya öznitelik üzerinde yapılan analizler sonucu, normal davranış modellerinden farklı bir şekilde hareket eden küme veya grupları tespit etmek için kullanılan bir yöntemdir. Toplu anomali tespiti, nokta anomali ve bağlamsal anomali tespit yöntemlerinden farklı olarak, birden fazla değişkenin bir arada ele alındığı veri setleri için kullanılır. Örneğin, bir şirketin çalışanlarının performans değerlendirme verilerini ele alalım. Performans değerlendirme verileri bir dizi özellikten oluşur, örneğin, hedeflerin tamamlanma oranı, müşteri geri bildirimleri, işe devam süresi vb. Bu verileri toplu anomali tespiti için kullanabiliriz. Toplu anomali tespiti, bu özellikleri bir araya getirerek, her bir çalışanın diğerlerine göre nasıl bir performans sergilediğini değerlendirir. Böylece, normal davranış modellerinden farklı bir şekilde performans sergileyen çalışanları belirleyebilir ve bu durumu araştırmak için adımlar atılabilir [1].

1.1. Ağ Anomali Tespiti

Ağ Anomali Tespiti, bilgisayar ağlarındaki anomali veya hatalı davranışların tespit edilmesi için kullanılan bir dizi teknik ve yöntemdir. Ağ anomali tespiti, ağ trafiği analizi, sistem davranış modelleme, veri madenciliği, istatistiksel analiz gibi çeşitli disiplinlerin kesişiminde oluşan bir alan olarak kabul edilebilir [3].

Ağ anomali tespiti genellikle aşağıdaki gibi üç ana kategoride incelenir:

- İmza Temelli (Signature-Based) Anomali Tespiti: Bu yöntemde, ağ trafiği için önceden tanımlanmış olan bilinen örüntü veya imzalar kullanılır. Eğer bir trafik paketi, tanımlanmış bir imza ile eşleşiyorsa, o zaman bu paket normal kabul edilir. Ancak eğer trafik paketi tanımlanmamış bir imza taşıyorsa, bu paket anomali olarak kabul edilir. İmza temelli yaklaşımın en büyük dezavantajı, sadece önceden tanımlanmış imzaları tespit edebilmesidir. Yani, bilinmeyen veya yeni bir anomali tespit edilemez [3, 4].

- Davranış Temelli (Behavior-Based) Anomali Tespiti:Bu yöntemde, ağ trafik verilerinin ve sistem davranışlarının normal davranış modelleri oluşturulur. Normal davranışlar, önceden belirlenmiş istatistiksel parametrelere göre ölçülür ve hesaplanır. Bu modellerin üzerinde yapılan istatistiksel analizler sayesinde, bilinen veya bilinmeyen anomali davranışlarının tespit edilmesi mümkündür. Davranış temelli yaklaşım, bilinen ve bilinmeyen anomali davranışlarını tespit edebilmesi nedeniyle, imza temelli yaklaşıma göre daha avantajlıdır [3].
- Makine Öğrenmesi (Machine Learning) Tabanlı Anomali Tespiti:Bu yöntemde, bir makine öğrenmesi modeli, normal davranışları öğrenmek için eğitilir ve ardından yeni gelen trafik verileri bu modele verilerek normal veya anomali olarak sınıflandırılır. Bu yöntemde, imza temelli ve davranış temelli yaklaşımlara göre daha esnek bir yapıya sahip olduğu için, bilinmeyen veya yeni türdeki anomali davranışları da tespit edebilir [3, 5].

1.2. Ağ Saldırıları

Ağ saldırıları, bilgisayar ağlarına veya internete bağlı cihazları hedef alan bir tür siber saldırıdır. Bu saldırılar, ağ kaynaklarını ele geçirmek, kullanıcı verilerini çalmak, hizmet kesintisine neden olmak, ağa zararlı yazılım bulaştırmak veya ağın normal işleyişini bozmak için yapılmaktadır [6].

Ağ saldırıları, birçok farklı şekilde gerçekleştirilebilir. Bunlardan bazıları şunlardır:

- DDoS Saldırıları: Hizmet reddi saldırıları (DDoS), hedef ağa çok sayıda bilgisayar veya cihazdan yapılan trafik yükü saldırısıdır. Bu saldırı türü, ağdaki kaynakların tükenmesine neden olur ve ağa erişimi engeller.
- Malware Saldırıları:Kötü amaçlı yazılımlar (malware), ağa sızmak için kullanılır ve saldırganların ağdaki cihazları ele geçirmesine izin verir. Bu tür saldırılar, bilgisayar korsanlarının ağa zararlı yazılım yüklemelerine ve sistemi ele geçirmelerine izin verir [7].
- Man-in-the-Middle (MITM) Saldırıları: Bu tür saldırılar, saldırganların ağdaki iletişimi dinlemesine ve hatta değiştirmesine izin verir. Bu tür saldırılar, ağdaki güvenlik açıklarından yararlanarak gerçekleştirilir [8].
- Fidye Yazılım Saldırıları: Fidye yazılımı saldırıları, bilgisayar korsanlarının bir cihazı ele geçirmesine ve ardından cihazın kullanıcısından para talep etmesine izin

verir. Bu tür saldırılar genellikle e-posta yoluyla veya yanıltıcı bir web sitesi aracılığıyla yayılır [9].

- Şifre Kırma Saldırıları: Bu tür saldırılar, saldırganların bir cihazın şifresini kırmak için farklı teknikler kullanmasına izin verir. Bu tür saldırılar, genellikle ağa erişimi olan bir cihazın ele geçirilmesinden sonra gerçekleştirilir [10].
- DoS Saldırıları: Hizmet reddi saldırıları (DoS), hedef ağa çok sayıda talebin gönderilmesiyle gerçekleştirilir. Bu tür saldırılar, ağın kaynaklarının tükenmesine neden olur ve hizmetlerin kullanılamaz hale gelmesine yol açar.
- Sosyal Mühendislik Saldırıları: Sosyal mühendislik saldırıları, saldırganların insanları yanıltarak bilgilerini elde etmelerine izin verir. Bu tür saldırılar genellikle e-posta veya telefon yoluyla gerçekleştirilir ve kullanıcılardan özel bilgilerini veya giriş bilgilerini paylaşmaları istenir [7].
- SQL Injection Saldırıları: Bu tür saldırılar, kötü amaçlı kişilerin web sitelerinde bulunan form alanlarına veya veritabanı sorgularına kötü amaçlı kodlar yerleştirmesine izin verir [11].
- Cross-Site Scripting (XSS) Saldırıları:Bu tür saldırılar, bir web sitesinde bulunan açıklardan yararlanarak, kullanıcılara kötü amaçlı bir kod yüklenmesine izin verir. Bu kod, kullanıcıların tarayıcılarında çalıştığında, saldırganın istediği eylemleri gerçekleştirebilir [12].
- Phishing Saldırıları: Bu tür saldırılar, sahte web siteleri, sahte e-postalar veya sahte mesajlar kullanarak, kullanıcıları kişisel bilgilerini ifşa etmeye veya kötü amaçlı bir yazılımı indirmeye ikna etmeye çalışır [13].
- Password Attack Saldırıları: Bu tür saldırılar, saldırganların bir kullanıcının şifresini çözmek için farklı teknikler kullanmalarını içerir. Bu teknikler arasında brute force saldırıları, sözlük saldırıları ve rainbow table saldırıları yer alır[10].
- ARP Spoofing Saldırıları: Ağdaki cihazların iletişimini engellemek veya izinsiz veri erişimi sağlamak amacıyla ARP (Address Resolution Protocol) protokolünün açığından yararlanılarak gerçekleştirilen bir saldırıdır. Bu saldırı türünde, saldırgan ağdaki diğer cihazların IP adresleri ile kendine ait MAC adresini eşleştirir ve bu sayede veriler saldırganın kontrolündeki bir araç üzerinden geçerek hedef cihaza iletilir. Bu sayede saldırgan, hedef cihaza ait ağ trafiğini takip edebilir ve hatta manipüle edebilir. ARP spoofing saldırıları genellikle ağda bulunan diğer cihazların

arasına sızarak gerçekleştirilir ve saldırgan, ağda bulunan cihazlar arasındaki trafiği izlerken bilgileri değiştirebilir veya çalabilir[14].

- DNS Spoofing Saldırıları: Ağ trafiğini yönlendirerek hedef kullanıcının doğru IP adresine bağlanmasını engelleyerek, sahte bir IP adresine bağlanmasına neden olan bir saldırıdır. Saldırganlar, DNS sunucusuna sahte bir alan adı ve IP adresi çifti iletilerek, hedef kullanıcının istediği web sitesine gitmek istediğinde, saldırganların sahte IP adresine yönlendirilmesini sağlarlar. Bu şekilde saldırganlar, hedef kullanıcının internet trafiğini izleyebilir, verilerini çalabilir veya sahte web siteleri oluşturarak kimlik avı yapabilirler. Bu saldırı genellikle kötü amaçlı yazılımlar veya bir ağda zayıf güvenlik önlemleri olduğunda gerçekleştirilir[15].
- IP Spoofing Saldırıları: Saldırganın IP paketlerinde yanıltıcı bilgiler sağlayarak bir başka kişinin veya sistemin kimliğini taklit etmesidir. Saldırgan, IP paketlerinin kaynak adresini değiştirerek, paketlerin doğru kaynaktan geldiği yanılsamasını yaratabilir. Bu saldırı türü, kimlik doğrulama protokollerinin ve güvenlik duvarlarının atlatılmasına yardımcı olabilir ve genellikle DoS (Denial of Service) saldırılarında kullanılır[16].
- Ping of Death Saldırıları: Bir ağa ICMP ping paketleri göndererek hedef sistemdeki ICMP protokolünün hatalı bir şekilde işlenmesine neden olmak suretiyle gerçekleştirilen bir saldırı türüdür. Bu saldırı, gönderilen ping paketinin normalden çok daha büyük boyutta olması nedeniyle oluşur. Normalde bir ping paketi, 64 byte boyutundadır, ancak bir Ping of Death saldırısında, 65.535 byte boyutundan daha büyük olabilen bir paket gönderilir.Bu saldırı, hedef sistemdeki ağ yığını veya işletim sistemi yazılımı tarafından sağlanan ICMP protokolünün hatalı bir şekilde işlenmesine neden olabilir. Bu, hedef sistemde çökme veya işlevsizlik gibi ciddi sonuçlara yol açabilir. Ping of Death saldırıları genellikle, saldırganın hedef sistemi tamamen devre dışı bırakmak istediği durumlarda kullanılır[16].
- SYN Flood Saldırıları: Bir sunucunun veya bilgisayarın erişimini engellemek için TCP/IP protokolünün zayıflıklarını kullanarak gerçekleştirilen bir DoS (Denial of Service) saldırısıdır. Saldırgan, hedef sistem üzerinde TCP bağlantıları kurmaya çalışarak yanıt almak için SYN paketleri gönderir. Ancak, SYN paketi alındığında, normal bir bağlantı kurmak için gereken ACK paketi gönderilmez. Bunun yerine, saldırgan SYN paketlerini çok sayıda gönderir, fakat ACK paketleri göndermez, bu nedenle hedef sistem, yanıtlanmayan çok sayıda SYN paketiyle boğulur ve gerçek

kullanıcıların bağlantı kurmasını engeller. Bu tür saldırılar, kaynak IP adresini değiştirerek ve sahte IP adresleri kullanarak gerçekleştirilebilir. SYN Flood saldırıları, ağ kaynaklarının tüketilmesine, performans sorunlarına ve hizmet kesintilerine neden olabilir[17].

- UDP Flood Saldırıları: Bir hedef sistemdeki ağ bağlantı noktalarına sürekli olarak büyük miktarda UDP trafiği göndererek, sistem kaynaklarının tükenmesine ve hizmetin kesintiye uğramasına neden olan bir DDoS saldırısı türüdür. UDP protokolü, TCP protokolüne kıyasla daha az güvenlik özellikleri içerir ve bu nedenle UDP Flood saldırıları daha kolay gerçekleştirilebilir. Saldırganlar, genellikle botnet adı verilen birçok bilgisayarı kontrol ederek UDP trafiği gönderirler. Bu bilgisayarlar genellikle, saldırganın kontrolü altında olan kötü amaçlı yazılımlarla enfekte edilmiş veya hacklenmiş sistemlerdir. Saldırılar, hedef sistemin ağ bağlantı noktalarını tıkanarak, normal trafiği engeller ve hizmetin kesintiye uğramasına neden olur [18].
- Smurf Saldırıları: ICMP Echo Request mesajlarının gönderildiği bir ağa bir saldırgan tarafından gönderilen yanıltıcı ICMP Echo Request mesajlarıyla gerçekleştirilen bir DoS saldırı türüdür. Bu saldırı türünde, saldırgan bir ağdaki IP adreslerini taklit ederek, hedef ağdaki tüm cihazları yanıltıcı ICMP mesajları ile sürekli olarak isteklerle bombardımana tutar. Böylece hedef ağ, yüksek miktarda trafik ile aşırı yüklenerek hizmet veremez hale gelir. Smurf saldırıları, internetin erken dönemlerinde yaygın bir şekilde kullanılmıştır. Ancak bugün, pek çok ağ güvenlik önlemi bu saldırı türüne karşı koruma sağlamaktadır[19].
- Teardrop Saldırıları: Teardrop saldırısı, hedefe gönderilen IP paketlerindeki fragment (parçalar) bilgilerinin hatalı olması sonucunda hedef makinenin işlem yapamaz hale gelmesi ile gerçekleştirilen bir saldırı türüdür. Bu saldırıda, saldırgan bir IP paketini daha küçük parçalara ayırarak hedefe gönderir ve bu paketlerin fragment bilgilerinde hatalar meydana getirir. Hedef makine, bu hatalı bilgileri işleyemeyerek çökme durumuna gelir.Teardrop saldırıları, özellikle Windows işletim sistemleri için tasarlanmıştır ve 1997 yılında keşfedilmiştir. Bu saldırı türü, ICMP (Internet Control Message Protocol) protokolünün Fragmentation Needed (Type 3, Code 4) mesajlarının hedefe yönlendirilmesiylegerçekleştirilir. Teardrop saldırıları, ağdaki tüm sistemleri etkileyebilir ve hizmet kesintilerine neden olabilir[20].
- Botnet Saldırıları: Büyük ölçekli DDoS saldırıları yapmak için enfekte edilmiş bir dizi cihazın (bot) kullanılmasıdır. Botnet'ler genellikle kötü amaçlı yazılım veya

fidye yazılımı aracılığıyla kurban cihazlara bulaşır ve ardından komut ve kontrol (C&C) sunucusu aracılığıyla yönlendirilirler. Botnet'ler genellikle birkaç yüz bin hatta milyonlarca cihazdan oluşabilir ve bu nedenle hedef ağa çok büyük bir yük yaratabilirler. Bu tür saldırılar, hedef ağın çalışmasını engelleyebilir ve hizmet kesintilerine neden olabilir. Botnet'ler genellikle sıfır gün açıklarını veya zayıf şifreleme yöntemlerini hedef alarak saldırganların bir ağa bulaşmasına neden olur[20].

- Clickjacking Saldırıları: Kullanıcının farkında olmadan kötü amaçlı bir işleme izin vermek için tasarlanmış bir tür sosyal mühendislik saldırısıdır. Bu tür saldırılar, web sayfalarındaki tıklama alanlarının, kullanıcının gördüğünden farklı bir yerde gösterilmesiyle gerçekleştirilir.Örneğin, bir saldırgan hedef web sitesinde bir tıklama alanı (button) oluşturur ve bu alanı kullanıcının tıklama yapacağı bir yere yerleştirir. Ancak bu tıklama alanı, görünen yerine farklı bir konumda yer alır ve kullanıcının gerçekte hangi işlemi yapacağını anlamadan, kötü amaçlı bir işlem gerçekleştirilir. Bu tür saldırılar, kullanıcının kimlik avı (phishing) sayfalarına yönlendirilmesi, kötü amaçlı yazılım yükleme veya kullanıcının bilgilerinin ele geçirilmesi gibi sonuçlar doğurabilir. Clickjacking saldırılarına karşı alınacak önlemler arasında web sayfalarının güvenliği için gerekli kontrollerin yapılması ve kullanıcı eğitimlerinin verilmesi yer alabilir[21].
- Dağıtılmış Yansıma Hizmet Reddi (DRDoS) Saldırıları: DDoS saldırılarına benzer, ancak saldırganlar bu saldırıda genellikle yansıtma saldırılarını kullanırlar. Bu saldırıda, saldırganlar bir yansıtma saldırısı gerçekleştirerek, bir hedef IP adresine hizmet talepleri gönderirler. Ancak saldırganlar bu isteklerin kaynak IP adresini değiştirerek, hedef makinenin yanıtı, gerçekte saldırana yönlendirilir.Böylece, yansıtma saldırısı aracılığıyla, saldırganlar tek bir makineyle değil, birçok farklı makineyle saldırı yapabilirler. Bu da saldırının etkisini artırır ve daha zor bir şekilde tespit edilmesine neden olabilir[7].

1.3. Veri Kümeleri

Ağ anomali tespiti için oluşturulan test veri kümeleri, büyük miktarda normal ve kötü niyetli ağ trafiği içermektedir. Bu veri setleri, ağ trafiğindeki anomali davranışlarını, yani normal davranıştan sapmaları ve ağa yönelik saldırılar gibi kötü niyetli davranışları simüle etmektedir. Bu veri setleri, gerçek ağ trafiğini kullanma zorluğunu aşmak ve ağ

anomalileri ile izinsiz girişlerin tespitinde kullanılan makine öğrenmesi algoritmalarının eğitilmesi ve test edilmesi için önemli bir araçtır. Bu veri kümeleri, farklı ağ topolojilerinde, protokollerinde ve hedeflerindeki anomali ve saldırı tiplerini simüle etmektedir. Bu veri setleri arasında aşağıdaki gibi popüler veri kümeleri yer almaktadır.

1.1. NSL- KDD

NSL-KDD, ağ anomali tespiti için kullanılan bir veri kümesidir. Bu veri kümesi, 1998 Darpa Intrusion Detection Evaluation Programı'nda kullanılan KDD Cup 1999 veri kümesinin yeniden işlenmiş bir sürümüdür. NSL-KDD veri kümesi, ağ tehditlerine karşı korunma amacıyla geliştirilen siber güvenlik uygulamalarını değerlendirmek adına sıkça tercih edilmektedir[22].

NSL-KDD veri kümesi, TCP/IP protokolü üzerinden gerçekleştirilen 41 farklı ağ saldırısı türünü içermektedir. Bu saldırı türleri, 4 ana kategoriye ayrılır: Denial of Service (DoS), Remote to User (R2L), User to Root (U2R) ve Probe.

Veri kümesi, normal ağ trafiği örneklerinin yanı sıra, farklı sayıda ve tipte saldırı örneklerini de içermektedir. Bu örneklerin tamamı, bazı özelliklerine göre kategorize edilmiştir. Toplamda, 22 ayrı özellik türü vardır ve her bir özellik türü, saldırıları belirlemek için kullanılan farklı tekniklerden elde edilen verileri ifade eder.

NSL-KDD veri kümesi, ağ saldırısı tespiti için kullanılan makine öğrenmesi algoritmalarını eğitmek ve test etmek için yaygın olarak kullanılmaktadır. Veri kümesi, bilgisayar ağlarını savunmak için kullanılan siber güvenlik sistemlerinin geliştirilmesinde de önemli bir araçtır.

1.2. UNSW-NB15

UNSW-NB15 (University of New South Wales - Network-based 15) veri kümesi, ağ güvenliği için oluşturulmuş bir veri kümesidir. Bu veri kümesi, 2015 yılında UNSW Canberra Cyber tarafından oluşturulmuş ve ağ trafiği analizi, tehdit tespiti, bilgi güvenliği ve siber güvenlik alanlarında kullanılmak üzere tasarlanmıştır.

Veri kümesi, gerçek bir ağdan toplanan ağ trafiği verilerinden oluşmaktadır ve ayrıntılı bir şekilde etiketlenmiştir. Toplamda 49 özellik (feature) ve 5 sınıf (class) içermektedir. Sınıflar şunlardır: normal, saldırı, saldırı-benzeri-aktivite, kötü amaçlı ve diğer.

Veri kümesi, toplamda 2.540.044 ağ trafiği paketinden oluşmaktadır. Bu paketler, gerçek bir ağda yer alan bir dizi farklı sistem, uygulama ve hizmetler arasındaki etkileşimleri yansıtmaktadır. Paketler, 1 saniyelik aralıklarla 30 gün boyunca kaydedilmiştir[23].

Veri kümesi, saldırıların çeşitli tiplerini ve saldırı-benzeri aktiviteleri kapsamaktadır. Örneğin, DoS (Denial of Service), R2L (Remote to Local), U2R (User to Root), probing (keşif) gibi saldırılar ve benzeri aktiviteler yer almaktadır. Bu çeşitlilik, veri kümesinin ağ güvenliği alanında kullanılabilmesini sağlar.

UNSW-NB15 veri kümesi, ağ güvenliği için kullanılan popüler veri setlerinden biridir ve birçok araştırmada kullanılmıştır. Ayrıca, veri kümesi, siber güvenlik alanında eğitim amaçlı kullanılmak üzere de uygun bir kaynak olarak kabul edilir.

1.3. CICIDS2017

CICIDS2017 (Canadian Institutefor Cybersecurity Intrusion Detection System) veri kümesi, ağ saldırılarını tespit etmek için kullanılan bir veri kümesidir. Bu veri kümesi, 2017 yılında yapılan Uluslararası Siber Güvenlik Konferansı'nda yayınlanmıştır. Veri kümesi, gerçek dünya ağ trafiğine dayanmaktadır ve geniş kapsamlı ağ saldırılarına karşı test edilmiştir.

CICIDS2017, 15 ayrı ağ saldırısı türü içeren 80 milyon etkinlik kaydından oluşmaktadır. Bu saldırı türleri arasında botnet, DDoS, port tarama, teşebbüs edilmiş bağlantı, kötü amaçlı yazılım ve kötü amaçlı URL gibi çeşitli saldırı türleri bulunmaktadır. Veri kümesi, ayrıca normal ağ trafiği de içermektedir[24].

Veri kümesi, toplamda 7 ayrı CSV dosyasından meydana gelir ve 82 farklı nitelik barındırır. Bu nitelikler içerisinde ağ protokol türü, kaynağın IP adresi, hedefin IP adresi, kaynak ve hedefin bağlantı noktaları, veri paketinin büyüklüğü ve zamanı gibi ağ hareketine dair detaylar mevcuttur. Veri kümesi, eğitim ve deneme olmak üzere iki bölüme ayrılmıştır.

CICIDS2017 veri kümesi, gerçek dünya ağ trafiğine dayandığı için, gerçek dünya senaryolarına yakın sonuçlar elde etmek için kullanılabilir. Ayrıca, geniş kapsamlı bir saldırı türü yelpazesini kapsadığı için, farklı saldırı türleri için ayrı ayrı test edilerek performans değerlendirmesi yapılabilir.

1.4. DARPA 1998

DARPA1998 veri kümesi, 1998 yılında yapılan DARPA Intrusion Detection Evaluation Programı (IDEVAL) kapsamında oluşturulmuş bir veri kümesidir. Bu veri kümesi, ağ güvenliği alanında ilk kez yaygın olarak kullanılan bir veri kümesidir ve birçok akademik çalışmada referans olarak kullanılmıştır.
DARPA1998 veri kümesi, gerçek bir ağda gerçekleştirilen ağ saldırıları ve normal ağ trafiği örneklerini içerir. Toplamda 7 farklı saldırı türü içeren 4 farklı veri kümesi sunulmuştur[25]:

a. Training set: Bu set, normal ağ trafiği ve saldırılar içerir. Toplamda 2,1 milyon örneği içermektedir.
b. Validation set: Bu set, normal ağ trafiği ve saldırılar içerir. Toplamda 312.000 örneği içermektedir.
c. Test set 1: Bu set, normal ağ trafiği ve 7 farklı saldırı türü içeren örneklerden oluşur. Toplamda 311.029 örneği içermektedir.
d. Test set 2: Bu set, normal ağ trafiği ve 7 farklı saldırı türü içeren örneklerden oluşur. Toplamda 62.201 örneği içermektedir.

Saldırı türleri arasında DOS (Denial of Service), Probe, U2R (User toRoot) ve R2L (Remote to Local) saldırıları yer almaktadır. Veri kümesi, özellik seçimi ve veri işleme işlemleri için de uygun bir veri kümesidir. Ancak, günümüzdeki ağ güvenliği tehditlerini yansıtmaktan uzaktır ve daha modern veri setleri tercih edilmektedir.

1.5. KDDCUP99

KDDCUP99, ağ anomali tespiti için kullanılan bir veri kümesidir ve 1998 yılında düzenlenen KDD Cup yarışması için oluşturulmuştur. Bu veri kümesi, gerçek bir bilgisayar ağından elde edilen verileri içermektedir.

Veri kümesi, bağlamsal anomali tespiti amacıyla oluşturulmuştur ve farklı protokoller üzerinden yapılan ağ trafiğiyle ilgilidir. Veri kümesi, 41 ayrı öznitelik içermektedir ve bu öznitelikler ağ trafiğinin çeşitli özelliklerini tanımlamaktadır. Özellikle, bu öznitelikler arasında paket sayısı, zaman damgası, kaynak IP adresi, hedef IP adresi,

kaynak port numarası, hedef port numarası, paket boyutu ve protokol tipi bulunmaktadır.

KDDCUP99 veri kümesi, normal trafiğin yanı sıra saldırı trafiği de içermektedir. Saldırı trafiği, DoS (Hizmet Dışı Bırakma), R2L (Uzaktan Yerel Giriş), U2R (Yetkisiz İçeriye Giriş) ve Probing (Tarama) gibi çeşitli tiplerdeki saldırıları içermektedir.

Veri kümesi, yaklaşık 5 milyon veri örneği içermektedir. Bu örneklerin yaklaşık %20'i normal trafiği, geri kalan %80'si ise saldırı trafiğini içermektedir. Bu nedenle, veri kümesi dengesiz bir veri kümesidir. KDDCUP99, ağ güvenliği alanındaki araştırmalar ve makine öğrenmesi algoritmalarının eğitimi ve test edilmesi için yaygın olarak kullanılan bir veri kümesidir [11].

BÖLÜM 2. LİTERATÜR TARAMASI

Ağ trafiği analizi, bilgi korunması ve ağ performansının devamlılığı için hayati bir rol taşımaktadır. Giderek karmaşıklaşan ağ yapıları ve siber tehditler, ağ trafiği analizi konusunda çeşitli zorlukları beraberinde getirmekte ve bu zorlukların üstesinden gelmek için yeni ve etkili yöntemlerin araştırılmasını zorunlu kılmaktadır. Bu bağlamda yapılan çalışmalarda, ağ trafiğindeki anormallikleri tespit etme ve bu anormalliklerin sebep olduğu problemleri çözme konusunda birçok yaklaşım sunmaktadır. Bu çalışmalar, ağ trafiği analizinde karşılaşılan spesifik problemleri tanımlayarak ve bu problemlere yönelik önerilen çözümleri sunarak alandaki bilgi birikimine katkıda bulunmaktadır. Örneğin, ağ trafiği verilerinin büyüklüğü ve karışıklığına ilişkin sorun, PCA (Temel Bileşen Analizi) gibi tekniklerle öznitelik azaltma yaklaşımıyla ele alınıp etkin bir yanıt geliştirilmiştir. Bu literatür incelemesinde, ağ trafiği analizi hakkında yapılan kilit araştırmalar, yaşanan zorluklar ve sunulan çözüm yolları kapsamlı bir biçimde ele alınarak alandaki bilgi seviyesine derinlik kazandırılmıştır. Eigen-dekompozisyon tekniklerinin ağ trafiği analizindeki kullanımı Hirose ve ark. tarafından yararlı bulunmuşken, Sheyner O. ve arkadaşları istatistiksel analiz teknikleriyle ağ trafiği verilerindeki anormalliklerin nasıl tespit edilebileceğine değinmiştir [26, 27]. Özellikle, veri madenciliği ve istatistiksel analiz kombinasyonunu ele alan Last M. ve ark., One-Class SVM ve Autoencoder algoritmalarını bu kapsamda incelenmiştir [28]. Chandola ve ark. ise yapay sinir ağlarının, ağ trafiğindeki anormallikleri tespitte geleneksel yöntemlere göre üstün olduğunu belirtmiştir [29].

Mukkamala ve ark. makine öğrenimi algoritmaları, özellikle YSA ve SVM, üzerinden ağ saldırılarının tespitini ele almıştır [4]. Bu alandaki zaman ve frekans domain analizlerine dair yaklaşımları M. Thottan ve C. Ji sunmuştur [1]. Özellikle, Liu ve arkadaşları, k-means, BIRCH ve DBSCAN algoritmalarının anormallik tespitindeki etkiliğini vurgulamışken, gerçek zamanlı ağ veri tespitindeki NaiveBayes algoritmasının etkinliğini Zhao ve arkadaşları incelemiştir [30,31]. Derin öğrenme yaklaşımıyla ise Convolutional Neural Networks (CNN) ve LSTM tabanlı modellerin üstünlüğüne vurgu

yapılmıştır [32]. Ancak, Yazılım Tanımlı Ağlar (SDN) kapsamında Deep Belief Networks (DBN) temelli yaklaşımların, CNN ile kıyaslandığında daha üstün sonuçlar verdiği ifade edilmiştir[33].

Zhang ve ark. istatistiksel trafik izlerini P2P botnet aktivitesinin tespiti için nasıl kullanabileceğini incelerken, Tan ve ark. TCP ve UDP protokollerindeki hizmet reddi saldırılarının tespiti için çok değişkenli korelasyon analizini önermiştir [34,35]. Öte yandan, wavelet tabanlı bir sinir ağı önerisi sunan Limthong ve ark. ve TCP/IP ağ trafiği üzerinden bir veri madenciliği yaklaşımını tanıtan Bloedorn ve ark. tarafından gerçekleştirilen çalışmalarda bu alanda dikkat çekici yöntemler sunulmuştur [36,37].

Lakshman, örnek alma ve oyun teorisi kullanarak ağ saldırılarının tespitinin verimliliğini incelerken[38], Boughaci ve ark. otonom ajanlar yaklaşımının ağ tehditlerini tespit kapasitesini incelemiştir [39]. Ayrıca, Jain ve Abouzakhar, Gizli Markov Modeli ve Destek Vektör Makinesi kombinasyonunun avantajlarını ve Shyu vd. ise PCA ve K-NN sınıflandırıcısını birleştiren bir anormallik tespit yöntemi önermiştir [40,41]. Garcia-Teodoro vd. dağıtık bir ağ anormallik tespiti yöntemi sunarken [42], G. Poojitha vd. yapay sinir ağlarıyla ağ saldırıları için veri setleri üzerinde çalışmıştır [43].

Ağ anomalisi ve anormallik tespiti konularında, literatürde birçok öznitelik indirgeme yöntemi ve sınıflandırma tekniği bulunmaktadır. Özellikle PCA (Principal Component Analysis), Chandola ve ekibinin [29] gerçekleştirdiği araştırmalar, bu konuda başarılı bir yaklaşımı işaret etmektedir. PCA, yüksek boyutlu veri setlerini sıkıştırarak öznitelikler arasındaki korelasyonları azaltır.

Ancak literatürün de işaret ettiği gibi, öznitelik seçimi ve indirgemenin tek bir yöntemle sınırlı kalmamalıdır. Bu kapsamda, PCA'nın lineer dönüşüm yeteneklerini RFECV (Recursive Feature Elimination with Cross-Validation) ile birleştirerek özelliklerin sınıflandırma performansını daha spesifik bir şekilde değerlendirmeyi amaçladık. RFECV, Mukkamala ve ark. tarafından da belirtildiği üzere [4], özniteliklerin sınıflandırma performansını iteratif olarak değerlendirebilen güçlü bir yöntemdir.

Öznitelik indirgemenin etkinliği, doğru özelliklerin seçilmesiyle doğru orantılıdır. RFECV tek başına oldukça başarılı bir özellik seçimi yöntemi olmasına rağmen, bu öznitelik seçiminin daha da rafine edilmesi gerekliliğinden yola çıkarak, Feature Selection (FS) algoritması ile birleştirilmiştir. RFECV ile oluşturulan öznitelik

sıralaması, FS ile daha da optimize edilerek deney sayısını artırma ve sonuçların doğruluğunu teyit etme amacı güdülmüştür. Bu yaklaşımın literatürde özellikle KDDCUP 99 veri kümesi üzerinde kullanılmadığını gözlemledik ve bu eksiği doldurmayı hedefledik. Bu hibrit yaklaşım, literatürdeki diğer metodolojilerle kıyaslandığında, öznitelik indirgemesi konusundaki yenilikçi yaklaşımımızı ve avantajlarını göstermektedir.

Mukkamala ve ark. [4] gibi literatürdeki birçok çalışmada, farklı sınıflandırma algoritmalarının ağ trafiği analizindeki başarısı belgelenmiştir. Bu algoritmaların arasında SVM, Naive Bayes, Decision Tree (DT), Logistic Regression (LR), Random Forest (RF) ve K-Nearest Neighbors (KNN) gibi yöntemler bulunmaktadır. Bu algoritmaların her biri, farklı matematiksel ve istatistiksel temellere dayandığı için, birçok yaklaşımı kapsamlı bir şekilde değerlendirme olanağı sunmaktadır.

Seçilen öznitelik indirgeme ve sınıflandırma yöntemleri, literatürde yer alan çalışmaların sonuçlarına ve bu alandaki bilimsel gelişmelere dayanarak belirlenmiştir. Özellikle KDDCUP 99 veri kümesi üzerinde PCA+RFECV ve RFECV+FS hibrit yöntemlerinin karşılaştırmalı olarak kullanılmadığını gözlemleyerek, bu eksikliği doldurmayı ve literatüre katkıda bulunulması hedeflemiştir.

BÖLÜM 3. MAKİNE ÖĞRENMESİ ALGORİTMALARININ VE PERFORMANS ÖLÇÜTLERİNİN İNCELENMESİ

Ağ trafiği değerlendirmesi, bilgi koruma ve ağ performansının istikrarı için hayati bir rol taşımaktadır. Bu analizlerin başarılı bir şekilde gerçekleştirilmesi, doğru algoritma seçiminin kritik bir rol oynadığı bir gerçektir. Karmaşık ağ yapıları ve siber tehditlerle başa çıkmak için uygun makine öğrenmesi algoritmalarını belirlemek, bu alandaki araştırmalarda merkezi bir konu olmuştur. Bu bağlamda, KDDCUP99 veri kümesi üzerinde, özellikle ağ anomali tespitinde sıkça kullanılan ve öznitelik indirgeme yöntemlerine karşı farklı tepkiler gösteren KNN, NB, LR, DT ve RF gibi makine öğrenmesi algoritmaları seçilmiştir. Bu algoritmaların seçilmesinin arkasındaki temel neden, bu algoritmaların ağ trafiği analizinde yaygın olarak kullanılması ve öznitelik indirgeme yöntemlerine karşı duyarlılıklarının farklılık göstermesidir. Ön işleme süreçlerinden geçirilen veri kümesi, bu yöntemlerle model oluşturma adımlarından geçirildi ve sonrasında sonuçlar kıyaslandı. Bu çalışma, farklı makine öğrenmesi algoritmalarının KDDCUP99 veri kümesi üzerindeki performansını ortaya koyarak, özellikle öznitelik indirgeme yöntemlerine tepkilerini ve algoritma seçimindeki kritik faktörleri vurgulamaktadır.

Bu çalışma, Windows 10 işletim sisteminde yürütülmüştür. Programlama dili olarak Python 3.10.9 sürümü tercih edilmiştirBilimsel işlemler için Numpy, veri inceleme için Pandas, makine öğrenimi işlerinde Scikit-learn ve görsel sunumlar için Matplotlib kütüphanelerinden yararlanılmıştır. Donanım olarak, bu araştırma HP ProBook 450 G8 model bir cihazda, Intel® Core™ i5-1135G7 merkezi işlemci, 16 GB bellek ve Intel® Iris® Xe Graphics serisi grafik birimiyle yürütülmüştür.

Makine öğrenimi algoritmalarının performansını doğru bir şekilde değerlendirmek, modelin genel başarısını ve uygulanabilirliğini anlamak için kritik bir adımdır. Bu değerlendirme, modelin gerçekte ne kadar iyi çalıştığını, hangi noktalarda başarılı olduğunu ve hangi alanlarda iyileştirmeye ihtiyaç duyduğunu gösterir. Özellikle

sınıflandırma problemleri için, bir modelin sadece doğru tahmin yapmadığı, aynı zamanda yanlış tahminlerde de nasıl performans gösterdiği önemlidir. Bu nedenle, bu çalışmada makine öğrenimi modelinin performansını değerlendirmek için çok yönlü bir yaklaşım benimsenmiştir. Seçilen metrikler, modelin başarısını farklı açılardan ölçerek, modelin genel performansını, hassasiyetini, duyarlığını ve tahmin olasılıklarının doğruluğunu değerlendirmeye yardımcı olmaktadır. Aşağıda, bu metriklerin her birinin detaylı bir açıklaması ve bu metriklerin bu çalışmada neden seçildiği sunulmaktadır.

Sınıflandırma problemlerinde, özellikle dengesiz veri kümelerinde, basit doğruluk metriği genellikle yetersiz kalmaktadır. Modelin pratikteki durumlarda ne kadar başarılı olacağını görmek için daha geniş çaplı bir inceleme gerçekleştirmek önemlidir. Hassasiyet, duyarlılık gibi metrikler, modelin pozitif ve negatif sınıflarını nasıl tahmin ettiğini ayrıntılı olarak göstererek, yanıltıcı sonuçlardan kaçınmamıza yardımcı olur. Özellikle ağ trafiği, finans ve tıbbi teşhis gibi kritik uygulama alanlarında yanlış negatif veya yanlış pozitif sonuçların maliyeti yüksek olabilir. Bu çalışmada, modelin tam bir resmini elde etmek için çoklu değerlendirme metrikleri kullanılmıştır. F1 skoru, hassasiyet ve duyarlığın birleşik bir ölçümüdür ve modelin dengeli bir performans sergileyip sergilemediğini gösterir. ROC eğrisi ve AUC, sınıflandırıcının performansını farklı eşik değerlerinde değerlendirmemize olanak tanır, bu da optimal eşik değerini belirlememize yardımcı olabilir. Log Kaybı ise modelin tahminlerinin ne kadar güvenilir olduğuna dair bir ölçüm sunar. Bu kapsamlı değerlendirme yöntemi, algoritmanın tüm yönleriyle nasıl performans gösterdiğini anlamamızı sağlar ve bu da modelin gerçek dünyada nasıl tepki vereceğine dair daha iyi bir anlayış kazanmamıza yardımcı olur.

Bu çalışmada makine öğrenmesi algoritmalarının performanslarını değerlendirmek için birçok farklı metrik kullanılmıştır.

Doğruluk (Accuracy): Doğru tahmin edilen örneklerin toplam örnek sayısına oranıdır. Doğruluk oranı, tüm sınıflar için eşit ağırlıklı olarak hesaplanır.

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \tag{3.1}$$

TP (True Positive) doğru pozitif tahmin sayısı,
TN (True Negative) doğru negatif tahmin sayısı,
FP (FalsePositive) yanlış pozitif tahmin sayısı ve

FN (FalseNegative) yanlış negatif tahmin sayısıdır.

Hassasiyet (Precision): Pozitif olarak tahmin edilen örneklerin gerçekten pozitif olan örneklerin oranıdır.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{3.2}$$

Duyarlılık (Recall/Sensitivity): Gerçek pozitif örneklerin doğru pozitif tahmin edilen örneklerin oranıdır.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{3.3}$$

F1 Skoru: Hassasiyet ve duyarlılık metriklerinin harmonik ortalamasıdır.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision } + \text{Recall}} \tag{3.4}$$

ROC Eğrisi (Receiver Operating CharacteristicCurve): Yanlış pozitif oranın (FPR) doğru pozitif orana (TPR) karşı hesaplanan bir eğridir. ROC eğrisi, sınıflandırıcının farklı kesme noktalarındaki performansını gösterir.

AUC (Area Under theCurve): ROC eğrisinin altındaki alanın (AUC) hesaplanmasıdır. AUC, sınıflandırıcının performansını tek bir sayısal değerle ölçer.

Log Kaybı (LogLoss): Sınıflandırıcının olasılık tahminlerinin gerçek sınıf etiketleriyle ne kadar uyuştuğunu ölçer.

$$\text{Log Loss} = -\left(\frac{1}{\text{n}}\right) \times \sum(\text{y}_\text{i} \times \log \text{p}_\text{i} + (1 - \text{y}_\text{i}) \times \log(1 - \text{p}_\text{i})) \tag{3.5}$$

Burada n örnek sayısı, y_i gerçek sınıf etiketi, p_i tahmin edilen olasılık ve log( ) doğal logaritmadır.

Bu çalışmada Naive Bayes, Karar Ağacı, Lojistik Regresyon, K-Nearest Neighbour, Rastgele Orman makine öğrenmesi algoritmaları kullanılmıştır.

a. Naïve Bayes

NaïveBayes algoritması sınıflandırma problemleri için kullanılan bir olasılık temelli makine öğrenimi algoritmasıdır. Özellikle doğal dil işleme gibi sektörlerde sıkça başvurulur.

NaïveBayes algoritması, BayesTeoremi'nin bir uygulamasıdır. Bayes Teoremi, koşullu olasılık hesaplamalarında kullanılır ve aşağıdaki formül ile ifade edilir:

$$P(A|B) = P(B|A) \times P(A) / P(B) \quad (3.6)$$

Bu formülde, A ve B birer olaydır. P(A) ve P(B) sırasıyla A ve B olaylarının marjinal olasılıklarını temsil eder. P(B|A), A olayı gerçekleştiğinde B olayının olasılığıdır. P(A|B) ise, B olayı gerçekleştiğinde A olayının olasılığıdır.

NaïveBayes algoritması, verilen özelliklerin sınıf etiketini tahmin etmek için kullanılır. Örneğin, bir e-postanın spam olup olmadığını belirlemek için kullanılabilir. Bu durumda, e-posta özellikleri mesajın içeriğindeki belirli kelimeleri içerebilir.

Algoritmanın çalışma mantığı şu şekildedir;

Öncelikle, verilen veri kümesi üzerinde sınıf etiketlerinin olasılıkları hesaplanır. Bu, veri kümesindeki her sınıfın sayısının toplam veri sayısına oranı şeklinde hesaplanır.

Daha sonra, verilen bir örnek ve sınıf etiketleri arasındaki olasılık hesaplanır. Bu, Bayes Teoremi kullanılarak hesaplanır. Verilen örnek için sınıf etiketinin olasılığı, sınıf etiketinin verilen özelliklere göre olasılıklarının çarpımına eşittir.

Son olarak, verilen örneğin sınıf etiketi, olasılığı en yüksek olan sınıf etiketi olarak tahmin edilir.

NaïveBayes algoritması, "Naïve" yani "saf" olarak adlandırılır, çünkü özellikler arasında bağımsızlık varsayımı yapar. Yani, özellikler arasındaki ilişkileri dikkate almaz ve her özniteliği diğer özniteliklerden bağımsız olarak ele alır. Bu varsayım sayesinde, hesaplamalar çok daha basit hale gelir ve algoritma çok daha hızlı çalışır.

b. Karar Ağacı

Karar Ağacı, sınıflandırma veya regresyon problemlerinde kullanılan bir makine öğrenmesi algoritmasıdır. Amacı, veri kümesindeki özniteliklerin doğrusal olmayan birleşimlerini kullanarak karar verme kurallarını öğrenmektir. Bu karar verme kuralları, bir ağaç yapısı şeklinde ifade edilir ve her bir düğüm bir özniteliği temsil eder.

Algoritmanın çalışma mantığı, veri kümesindeki her bir öznitelik için bir ayrıştırma kriteri belirlemeye dayanmaktadır. Bu ayrıştırma kriterleri, genellikle "bilgi kazancı"

adı verilen ölçütlere dayanarak belirlenmektedir. Bilgi kazancı, özniteliklerin sınıflandırma doğruluğundaki etkisini ölçer.

Öncelikle, veri kümesindeki özniteliklerin bilgi kazançları hesaplanır. Bilgi kazancı, veri kümesindeki bir özniteliğin sınıflandırma doğruluğundaki değişimi ölçen bir ölçüttür. Yani, özniteliklerin sınıflandırmada ne kadar önemli olduğunu belirlemeye yarar.

En yüksek bilgi kazancına sahip öznitelik, ağacın ilk düğümü olarak seçilir. Bu düğüme göre veri kümesi alt kümelere ayrılır. Bu alt kümeler, aynı işlemi, yani bilgi kazançlarına göre en önemli olan özniteliği seçerek ayrıştırılır. Bu işlem, alt kümelerin sınıflarına ayrılana kadar devam eder.

Ağaç, bu şekilde alt kümelerin sınıflarına göre bölünerek ilerler. Ağacın yaprak düğümleri, son kararları veren düğümlerdir. Her yaprak düğümü, veri kümesindeki bir sınıfı temsil eder. Yeni bir veri noktası, ağaçtaki karar kurallarını izleyerek bir yaprak düğümüne ulaşır ve bu yaprak düğümündeki sınıf etiketi, veri noktasının sınıf etiketi olarak atılır.

Karar ağacının oluşturulması sırasında ayrıştırma kriterlerinin belirlenmesi ve alt kümelere ayrılacak verilerin seçimi, ayrıştırma kriterlerinin nasıl belirleneceği gibi konular, algoritmanın uygulandığı farklı versiyonlarına göre değişebilir. Ancak, temel olarak, Karar Ağacı algoritması veri kümesindeki özniteliklerin ayrıştırma kriterlerine göre ağaç yapısı şeklinde bir karar modeli oluşturur.

c. Lojistik Regresyon

Lojistik Regresyon yöntemi, bir hedef değişkenin iki ya da birden fazla kategoriye bölünmesinde tercih edilen bir sınıflama tekniğidir. Bu algoritma, bağımsız değişkenlere dayanarak hedef değişkenin olasılığını tahmin etmeye dayalıdır.

Lojistik Regresyon algoritması, Sigmoid fonksiyonu (ya da Lojistik fonksiyon) olarak bilinen bir matematiksel fonksiyon kullanır. Bu fonksiyon, verilen bir girdi değerinin 0 ile 1 arasında bir çıktıya dönüştürülmesini sağlar. Bu çıktı, girdinin olasılık değerine karşılık gelir. Lojistik fonksiyon aşağıdaki gibidir:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{3.7}$$

Burada $\sigma(z)$, bağımsız değişkenlerin ağırlıkları ve sabit terimin bir fonksiyonudur. Modelin amacı, bu ağırlıkları öğrenmek ve girdi değişkenlerinin bir fonksiyonu olarak bağımlı değişkenin olasılık dağılımını tahmin etmektir.

Verilen bir gözlem için, modelin tahmini olasılığı aşağıdaki şekilde hesaplanır:

$$p\ (y = 1|\mathrm{x}) = \sigma\ (\beta_1 x_1 + \cdots + \beta_n x_n) \tag{3.8}$$

Burada p(y) bağımlı değişkenin sınıfı (0 veya 1), $\sigma(x)$ise bağımsız değişkenler ve$\sigma(\beta)$ise bu değişkenlere ait ağırlıkların vektörüdür.

Modelin eğitimi, verilerin log-olasılık fonksiyonuna göre maksimize edilmesi ile yapılır. Bu, ağırlıkların maksimum olasılık tahminine göre ayarlanması anlamına gelir.

Lojistik Regresyon, sınıflandırma yapmak için kullanılırken, modelin çıktısı, bir eşik değeri (threshold) ile karşılaştırılır. Eğer çıktı eşik değerinden büyükse, gözlem pozitif sınıfa (1) ait olarak sınıflandırılır. Eğer çıktı eşik değerinden küçükse, gözlem negatif sınıfa (0) ait olarak sınıflandırılır.

d. En Yakın Komşu (KNN)

K-En Yakın Komşu (KNN) algoritması, sınıflandırma ve regresyon problemleri için kullanılan bir makine öğrenmesi algoritmasıdır. Temel prensibi, yeni bir örneğin sınıflandırılması veya değerlendirilmesi için komşu noktalara bakarak tahmin yapmaktır.

Algoritmanın formülü şu şekildedir:

2. Veri kümesindeki tüm örneklerin benzerlik ölçüleri hesaplanır.
3. Yeni örnek için benzerlik ölçüleri hesaplanır.
4. En yakın K komşu seçilir.
5. K seçimine göre, sınıflandırma için en sık kullanılan sınıf veya regresyon için en yaygın olan değer atanır.

Benzerlik ölçüsü, örneğin öklid mesafesi veya Manhattan mesafesi gibi bir ölçü kullanılarak hesaplanabilir. K, en yakın komşuların sayısını belirler ve genellikle tek bir değerdir.

KNN algoritması, veri kümesinin boyutu büyüdükçe hesaplama karmaşıklığı artar. Ayrıca, veri kümesindeki dengesizlikler, aykırı değerler ve gürültü gibi faktörler,

algoritmanın performansını etkileyebilir. Bu nedenle, KNN algoritması veri kümesinin doğru şekilde ölçeklendirilmesi ve temizlenmesi gerektiğinde daha iyi sonuçlar verir.

e. Rastgele Orman :

Rastgele Orman yöntemi, bir topluluk tekniği olarak bilinir ve çoklu karar ağaçlarını birleştirerek daha etkili bir model meydana getirir. Bu karar ağaçları, öğrenme sürecinde veri kümesinin rastgele alt kümeleriyle eğitilir.

Rastgele Orman algoritmasının formülü şu şekildedir:

1. Veri kümesinden rastgele alt kümeler oluşturulur.
2. Her alt küme için bir karar ağacı oluşturulur.
3. Karar ağaçları, eğitim verilerindeki özniteliklerin rastgele alt kümesi ile eğitilir.
4. Karar ağaçlarının her biri, test verilerindeki girdilere dayalı bir tahmin yapar.
5. Tahminler, bir oylama yöntemi kullanılarak birleştirilir ve sonuç oluşturulur.

Rastgele Orman algoritması, veri kümesindeki birçok özellik arasında karşılaştırma yaparak, en önemli özellikleri belirlemek için de kullanılabilir. Bu, özelliklerin sıralamasına dayalı bir öznitelik önem sıralaması oluşturarak yapılabilir.

Rastgele Orman algoritması, sınıflandırma ve regresyon problemleri için kullanılabilir ve yüksek doğruluk, az varyans ve düşük öğrenme hatası ile bilinir.

BÖLÜM 4. UYGULAMA

Makine öğrenimi ve veri bilimi araştırmalarında, kullanılan araçlar ve teknolojiler, modelin başarısını ve uygulanabilirliğini doğrudan etkileyebilir. Doğru programlama dili, kütüphaneler ve çalışma ortamının seçilmesi büyük önem taşır. Bu çalışma kapsamında, veri işlemeden model eğitimine kadar olan süreçte endüstri standardı araçlar ve teknolojiler tercih edilmiştir. Bu amaçla Anaconda dağıtımı kullanılmış, kodlar Jupiter Notebook üzerinde Python ile yazılmıştır ve Sklearn, Numpy ve Pandas gibi kütüphaneler kullanılmıştır. Bu araştırma, 64-bit'lik bir Microsoft Windows platformunda bulunan bir PC üzerinde yürütülmüştür. Son yıllarda makine öğrenimi teknikleri birçok alanda başarıyla kullanılmış olup, bilgisayar ağlarındaki saldırı tespitinde de etkili sonuçlar elde edilmiştir. Bu nedenle, sürekli evrilen saldırı türlerine karşı, bu çalışmada KDDCUP99 veri kümesi üzerinde anomali tespiti yaklaşımı benimsenerek saldırıların belirlenmesi amacıyla bir model oluşturulmuştur. Bu model, bilgisayar ağı güvenliği alanındaki yenilikçi çözümler için bir temel oluşturabilir. Bu model, ileriki çalışmalarda araştırmacılara fikir verebilecek ve bilgisayar ağı güvenliği alanında yenilikçi çözümler sunabilecektir. Temel aşamaları şu şekilde sıralanabilmektedir.

- Veri kümesinin ön işleme aşamasından geçirilmesi
- Veri kümesi analiz edilerek sınıflandırma modeli belirlenmesi
- Sınıflandırma için uygun özniteliklerin belirlenmesi
- Sonuçların değerlendirilmesi

Çalışma sürecinde yapılan tüm deneyler “Python” yazılım dilinde “Jupiter Notebook” geliştirme ortamında gerçekleştirilmiştir. Bu çalışmada kullanılan araçlar hakkında bilgi Tablo 4.1. ‘ de gösterilmektedir.

Tablo 4.1 : Yazılım ve donanım platformu

No	Araç	Açıklama
1	Windows 10	İşletim sistemi
2	Python.3.10.9	Programlama Dili
3	Numpy	Bilimsel hesaplama kütüphanesi
4	Pandas	Veri analizi kütüphanesi
5	Scikit-learn	Makine öğrenmesi kütüphanesi
6	Matplotlib	Görselleştirme kütüphanesi
7	System	HP proBook 450 G8
8	CPU	IntelCore™ i5-1135G7 @ 2.40GHz 2.42 GHz
9	RAM	16 GB
10	GPU	Intel® Iris® Xe Graphics Family

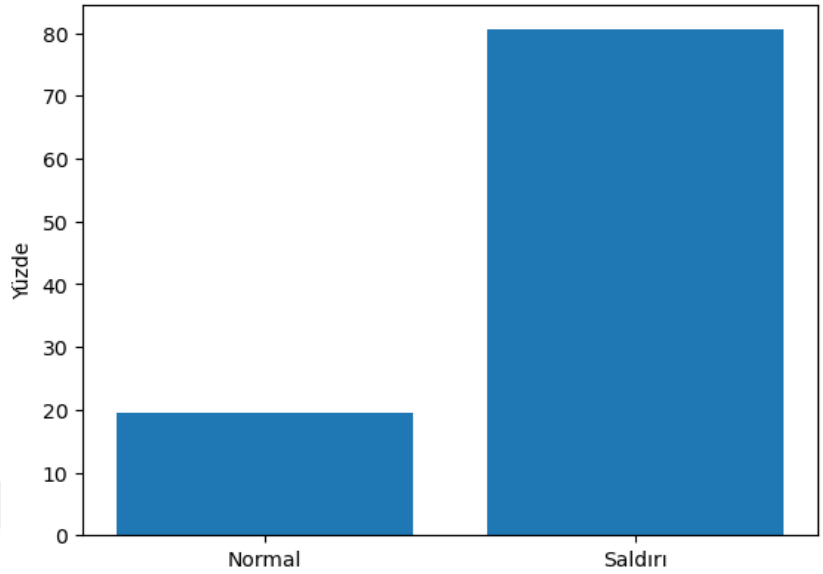
4.1. Ön işleme

Veri ön işleme, bir veri kümesinin analiz edilebilir hale getirilmesi için yapılan işlemler bütünüdür. Bu bölümde, verilerin doğruluğu, eksiklikleri, tutarsızlıkları ve gürültüleri tespit edilip düzeltilmiştir. Bu işlemler, veri kümesinde sınıflandırıcının doğru analiz yapabilmesi için orantılı hale getirildi, standart sapmanın dışında ve boş kalan kayıtlar kaldırılmış, kategorik verilerin sayısallaştırılması sağlandı. Son olarak veri seti normalleştirildi ve MÖ yöntemleri için uygun temiz bir veri seti oluşturuldu.

Veri ön işleme sürecinde, "attack_type" kolonu incelenerek veri kümesindeki saldırı ve normal trafik verileri ayırt edilmiştir. Saldırı verileri için 1, normal trafik için 0 etiketlemesi yapılmıştır. Veri kümesi içerisindeki eksik değerlerin olup olmadığına bakıldı ve eksik değer bulunan yerlere medyan değer ile tamamlama yapılmıştır.

Kategorik değerli öznitelikler (örneğin, protocol_type, flag, service) one-hot encoding yöntemi kullanılarak sayısallaştırılmıştır. Bunun için pandas kütüphanesinin get_dummies fonksiyonu kullanılmıştır. Veri kümesindeki sayısal değerlerin ölçeklendirilmesi adına, 'src_bytes' ve 'dst_bytes' sütunları MinMaxScaler kullanılarak normalizasyon işlemine tabi tutulmuştur. Bu işlem, bu iki öznitelik için yeni 'src_bytes_normalized' ve 'dst_bytes_normalized' sütunları oluşturarak gerçekleştirilmiştir. Veri kümesinden 'duration' ve 'attack_type' sütunları çıkartılmıştır. Son olarak, 'success_pred' sütunu etiket olarak belirlenmiş ve özellik matrisi (X) ile

etiket vektörü (y) oluşturulmuştur. Bu matris ve vektör daha sonra model eğitimi için kullanılmak üzere ayrılmıştır.

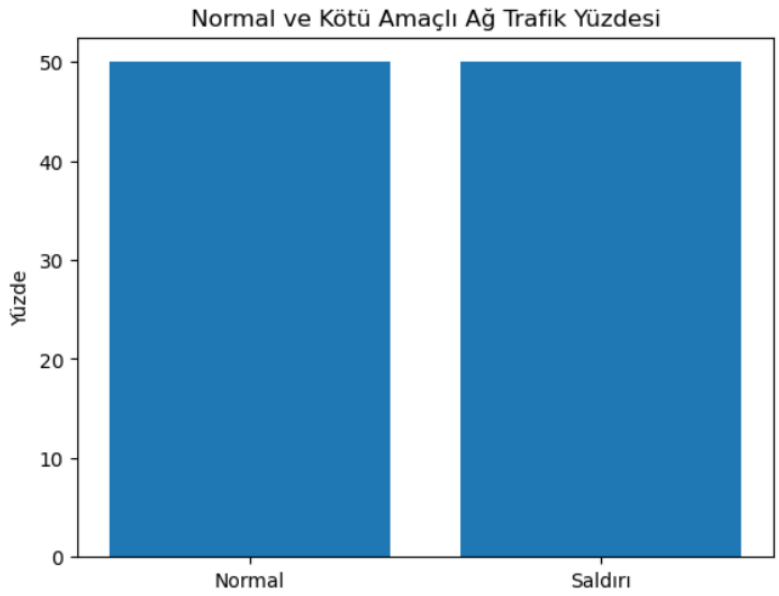


Şekil 4.1 : Normal ve kötü amaçlı ağ trafik yüzdesi.

Şekil 4.1, veri kümesinin normal ve kötü amaçlı ağ trafiğinin oranını yansıtmaktadır. Tablo 4.1'den de anlaşılacağı üzere, bu veri kümesi dengesiz bir dağılıma sahiptir. Bu, sınıflandırma algoritmalarını eğitilirken karşılaşılabilecek önemli bir zorluktur. Saldırıyı temsil eden veri sayısı, normal trafiği temsil eden veri sayısından çok daha fazladır. Bu, 250,436 saldırı (kötü amaçlı) verisi ve 60,593 normal trafik verisiyle sonuçlanmıştır. Dengesiz veri setleriyle çalışmanın bazı zorlukları vardır. Özellikle sınıflandırma problemlerinde, modelin çoğunluk sınıfını aşırı öğrenmesi ve azınlık sınıfını ihmal etmesi riskiyle karşı karşıya kalınabilmektedir. Bu tür bir dengesizlik, modelin nadir sınıfı, bu durumda normal trafiği, doğru bir şekilde öğrenmesini zorlaştırabilmektedir. Bu, potansiyel olarak zararlı bir trafik öğesinin "Normal" olarak sınıflandırılması riskini taşır ve bu, güvenlik ihlallerine yol açabilmektedir.

Bu nedenle, veri kümesi dengeli bir yapıya getirmiştir. Bunun için, pandas kütüphanesini kullanılarak saldırıları ve normal trafiği temsil eden veri noktalarını ayrı ayrı seçilmiştir. Daha sonra, saldırıları temsil eden veri noktalarından rastgele bir şekilde, normal trafiği temsil eden veri sayısı kadar örnek alınmıştır. Böylece, her iki sınıfta da eşit sayıda veri noktası elde edilmiştir. Son adımda, bu dengelenmiş veri kümesi bir bar grafiğiyle görselleştirilmiştir. Bu, her iki sınıfın da veri kümesinde eşit bir yüzdede temsil edildiğini net bir şekilde göstermektedir.

Şekil 4.2’ de veri kümesinin normal ve kötü amaçlı dengelenmiş ağ trafik yüzdesini göstermektedir.



Şekil 4.2 : Normal ve Kötü Amaçlı Ağ Trafik Yüzdesi

KDD Cup 1999 veri kümesindeki özniteliklerin anlamları aşağıdaki gibidir:

- duration: saldırının sürdüğü süre
- protocol_type: ağ protokolü türü (örn. TCP, UDP, ICMP)
- service: hedef hizmetin tipi flag: ağ trafiği durumu
- src_bytes: kaynak bilgisayar tarafından hedef bilgisayara iletilen bayt sayısı
- dst_bytes: hedef bilgisayar tarafından kaynak bilgisayara iletilen bayt sayısı
- land: kaynak ve hedef IP adresleri aynı mı?
- wrong_fragment: bozuk veya eksik fragment sayısı
- urgent: acil durum işareti
- hot: çeşitli servislerin sayısı
- num_failed_logins: başarısız giriş denemesi sayısı
- logged_in: kullanıcının oturum açtığı durum (1 = oturum açtı, 0 = aksi)
- num_compromised: saldırgan tarafından kompormize edilmiş bilgisayar sayısı
- root_shell: kök kullanıcı erişimi
- su_attempted: root erişimi için su (set user) komutu kullanıldı mı?
- num_root: root hesabı ile gerçekleştirilen işlemlerin sayısı

- num_file_creations: oluşturulan dosya sayısı num_shells: açılan kabuk (shell) sayısı
- num_access_files: değiştirilen erişim dosyası sayısı
- num_outbound_cmds: gerçekleştirilen çıkış komutlarının sayısı
- is_host_login: saldırganın hedef bilgisayara doğrudan giriş yapmış olması durumu
- is_guest_login: kullanıcının misafir olarak oturum açtığı durum count: hedef bilgisayara yapılan bağlantı sayısı
- srv_count: hedef servise yapılan bağlantı sayısı
- serror_rate: hedef serviste oluşan hata oranı
- srv_serror_rate: hedef serviste yapılan bağlantılarda oluşan hata oranı
- rerror_rate: hedef bilgisayardan kaynaklanan hata oranı
- srv_rerror_rate: hedef serviste yapılan bağlantılardan kaynaklanan hata oranı
- same_srv_rate: aynı servise yapılan bağlantı oranı
- diff_srv_rate: farklı servislere yapılan bağlantı oranı
- srv_diff_host_rate: farklı hostlara yapılan hedef servis bağlantı oranı dst_host_count: hedef bilgisayara yapılan bağlantıların kaynak sayısı dst_host_srv_count: hedef servise yapılan bağlantıların kaynak sayısı
- dst_host_same_srv_rate: hedef bilgisayara yapılan bağlantıların aynı servise yapılan oranı
- dst_host_same_src_port_rate: Hedef bilgisayarın, kaynak bilgisayardan gelen bağlantıların kaçının aynı kaynak portunu kullandığını gösteren bir orandır.
- dst_host_srv_diff_host_rate: Hedef bilgisayarın, aynı servisi kullanan bağlantıların kaçının farklı kaynak bilgisayarlardan geldiğini gösteren bir orandır.
- dst_host_serror_rate: Hedef bilgisayarda, istemciye hizmet verirken bir hata oluştuğu durumlarda (bağlantı talebinin reddedilmesi, zaman aşımına uğraması, vb.) bu hataların tüm bağlantılara oranıdır.
- dst_host_srv_serror_rate: Hedef bilgisayarda, belirli bir servis için istemciye hizmet verirken bir hata oluştuğu durumlarda (bağlantı talebinin reddedilmesi, zaman aşımına uğraması, vb.) bu hataların tüm bağlantılara oranıdır.

- dst_host_rerror_rate: Hedef bilgisayarda, istemci tarafında bir hata oluştuğu durumlarda (bağlantı talebi gönderilememesi, bağlantı kesilmesi, vb.) bu hataların tüm bağlantılara oranıdır.
- dst_host_srv_rerror_rate: Hedef bilgisayarda, belirli bir servis için istemci tarafında bir hata oluştuğu durumlarda (bağlantı talebi gönderilememesi, bağlantı kesilmesi, vb.) bu hataların tüm bağlantılara oranıdır.
- attack_type: Gözlemlenen ağ saldırısının türünü belirten etiketlenmiş verilerdir. Toplam 22 farklı saldırı türü mevcuttur.
- success_pred: Veri kümesinin her bir örneği için, örnekteki ağ trafiğinin normal veya saldırı olup olmadığını tahmin etmek için oluşturulan bir etiketlenmiş veridir. 1, normal trafiği; 0, saldırı trafiğini ifade eder.

Veri kümesinde bulunan saldırı tiplerninden bir kısmı Tablo 4.2'de sunulan bilgiler, EK-1 bölümünde tam olarak bulunmaktadır.

Tablo 4.2 : Veri kümesindeki saldırı verilerinin dağılımı

No	Saldırı Adı	Kayıt Sayısı
1	smurf	164091
2	normal	60598
3	neptune	58001
4	snmpgetattack	7741
5	mailbomb	5000
6	guess_passwd	4367
7	satan	1633
8	warezmaster	1602
9	back	1098
10	mscan	1053
11	apache2	794
12	processtable	759
13	saint	736
14	portsweep	354

4.2. Öznitelik seçimi

Makine öğreniminde, veri kümesini hazırlarken özellik seçiminin model performansı üzerinde kritik bir rolü vardır. Veri kümesindeki tüm özellikler, modelin amacını tahmin

ederken bir giriş olarak işlenir. Ancak, tüm özniteliklerin kullanılması bazen gereksiz ve hatta zararlı olabilmektedir.

Özellik seçimi, modelin daha az özellikle daha etkili çalışmasını sağlayarak, lüzumsuz ya da olumsuz etki yaratan özelliklerin veri kümesinden kaldırılmasına yardımcı olur. Bunun yanı sıra, öznitelik seçimi sayesinde veri kümesindeki gürültülü ve yetersiz özniteliklerin de model tarafından kullanılmaması sağlanarak, modelin daha doğru ve güvenilir sonuçlar üretmesi hedeflenmektedir.

Ayrıca, öznitelik seçimi sayesinde, modelin eğitilmesi ve tahminde bulunma süresi ve hesaplama kaynaklarının da azaltılması sağlanabilmektedir. Bu nedenle, uygun bir öznitelik seçimi yöntemi kullanarak veri kümesindeki en önemli öznitelikleri seçmek, modelin daha yüksek performans göstermesini sağlayabilmekte ve aynı zamanda eğitim sürecini hızlandırabilmektedir.

Bu çalışmanın başlangıcında kullanılan veri kümesi 311,029 satır ve 43 sütundan oluşmaktaydı. Ancak, veri ön işleme işlemi sonrasında bu veri kümesi 121,186 satır ve 117 sütundan oluşan bir yapıya dönüşmüştür. Bu nedenle, 117 öznitelik, sınıflandırma işlemi daha etkili ve çabuk kılmak amacıyla azaltılması gereken bir boyuttur. Bu çalışmanın temel amacı, etkili bir saldırı tespit sistemi geliştirmek için ağ trafiği analizi yapmaktır. Ön işleme sonrasında elde edilen 117 öznitelik, sınıflandırma sürecinin etkinliğini artırmak ve işlem süresini kısaltmak için anlamlı özniteliklerin korunması hedeflenmiştir. Veri hazırlama aşamasında ilk olarak 'success_pred' sütunu üzerinden hedef değişken oluşturulmuştur. Bu sütun, normal trafik durumlarını 0, saldırı durumlarını ise 1 olarak temsil edecek şekilde ayarlanmıştır. Bu değişiklik, sınıflandırma modelinin istenilen saldırıları doğru bir şekilde tanımlayabilmesini sağlamak amacıyla yapılmıştır.

Sonrasında, 'duration' sütununun sonuçları üzerindeki etkisinin zayıf olduğu düşünüldüğünden bu sütun veri kümesinden çıkarılmıştır. Böylece, işlem yapısı daha sadeleştirilmiş ve gereksiz bilgilerin model üzerindeki etkisi azaltılmıştır.

Veri kümesi üzerinde eksik veri kontrolü yapılmış ve eksik veri tespit edilmediği için veri kümesi eksiksiz bir şekilde kullanılmıştır. Kategorik değişkenler olan 'protocol_type', 'service' ve 'flag' sütunlarına One-Hot-Encoding yöntemi uygulanmıştır. Bu dönüşüm, kategorik verileri sayısal formata çevirerek makine öğrenimi modellerinin daha iyi çalışmasını sağlamıştır. Sınıflandırma işlemi için hedef değişken olarak

'success_pred' kullanıldığından, bu aşamada 'attack_type' sütunu veri kümesinden çıkarılmıştır. Bu değişiklik, modelin hedeflediğimiz sınıflandırma problemine odaklanmasını sağlamıştır. 'src_bytes' ve 'dst_bytes' sütunlarına MinMaxScaler uygulanarak bu öznitelikler normalize edilmiş ve 'src_bytes_normalized' ile 'dst_bytes_normalized' adında yeni öznitelikler oluşturulmuştur. Bu normalizasyon, farklı ölçeklerdeki verilerin model performansını etkilemesini engellemek amacıyla yapılmıştır.

Bu çalışmada, veri kümesimizin boyutunu azaltarak sınıflandırma işleminin verimliliğini artırmayı amaçlayan bir öznitelik seçimi stratejisi olarak PCA + RFECV yöntemini hibrit bir yaklaşım olarak benimsemeyi tercih ettik. Bu stratejinin nedenleri ve uygulama aşamaları aşağıda detaylı bir şekilde açıklanmıştır. İlk olarak, boyut indirgeme amacıyla PCA (Principal Component Analysis) yöntemini uygulanmıştır. Bu adımın temel amacı, veri kümesindeki yüksek boyutluluğu ve öznitelikler arasındaki yüksek korelasyonu ele alarak verinin özünü yakalamaktır. PCA, özdeğer analizi kullanarak öznitelikleri yeni bileşenlere dönüştürerek varyansın büyük bir kısmını korumaktadır. Böylece, veri kümesinin boyutunu daha düşük bir uzayda temsil etmek amaçlanmıştır. %95 varyans seviyesini seçilmesinin nedeni, veri kümesinin karmaşıklığı azaltılırken önemli bilgi kaybının en aza indirilmesidir.

PCA'nın ardından RFECV (Recursive Feature Elimination with Cross-Validation) yöntemi uygulanmıştır. Bu yöntem, veri kümesindeki öznitelikleri sırayla ekleyerek veya çıkararak sınıflandırma modelinin performansını değerlendirmektedir. Bu aşamada bir sınıflandırıcı olarak Logistic Regression modelini tercih edilmiştir. Logistic Regression, sınıflandırma işlemi için kullanılan temel bir algoritma olup, modelin öğrenmesini ve özniteliklerin sınıflandırmadaki etkisini anlamayı kolaylaştırmaktadır. max_iter parametresini 1000 olarak ayarlanarak, yeterli iterasyon sayısı sağlandığında uygun bir sonuç elde etmek amaçlanmıştır.

RFECV, cross-validation (çapraz doğrulama) kullanılarak her iterasyonda hangi özniteliklerin modelin performansını artırdığı değerlendirilmiştir. Bu sayede, en az etkili öznitelikler çıkarılarak, modelin daha hızlı ve etkili çalışması hedeflenmiştir. Elde edilen sonuçlar, en iyi öznitelikleri seçerek veri kümesini güncellenmiş ve X_train_selected ve X_test_selected veri setlerinde ifade edilmiştir.

Sonuç olarak, PCA + RFECV yöntemi ile öznitelik seçimi, boyut indirgeme ve sınıflandırma performansını bir araya getirerek veri kümesinin karmaşıklığını azaltırken, sınıflandırma modelinin etkinliğini artırmayı amaçlamıştır. Bu hibrit yaklaşım, daha küçük boyutlu ve anlamlı özniteliklerle oluşturulan veri kümesi üzerinde sınıflandırma modelinin daha hızlı ve etkili çalışmasını sağlamıştır.

Deney sayısını arttırmak ve iki hibrit yöntemi karşılaştırmak amacıyla, bir diğer öznitelik indirgeme yöntemi olan RFECV (Recursive Feature Elimination with Cross-Validation) yöntemi ile İleri Yönelimli Seçim (Forward Selection) yöntemini bir araya getiren bir hibrit yaklaşım kullanılmıştır. Bu yaklaşım, veri kümesindeki anlamsız özniteliklerin çıkarılmasını sağlarken, aynı zamanda en iyi özniteliklerin seçilerek sınıflandırma performansının artırılmasını amaçlamaktadır.

İlk adımda, DecisionTreeClassifier kullanılarak RFECV yöntemi uygulanmıştır. Bu adım, veri kümesi üzerindeki özniteliklerin ayrı ayrı çıkarılarak sınıflandırma modelinin performansının değerlendirildiği bir adımdır. Her adımda bir öznitelik çıkarılarak cross-validation (çapraz doğrulama) ile modelin başarısı ölçülmüştür. Scoring parametresi "accuracy" olarak belirlenmiş ve cv parametresi 5 olarak ayarlanmıştır. Bu sayede modelin performansı kıyaslanarak, en az etkili öznitelikler çıkartılmıştır.

RFECV sonrasında, get_support() yöntemi kullanılarak en önemli özniteliklerin belirtilen sütunlarının indeksleri alınmıştır.. X veri kümesindeki bu önemli öznitelikleri içeren yeni bir veri kümesi oluşturulmuştur. Bu yeni veri kümesinin sınıflandırma performansını daha da artırmak amacıyla İleri Yönelimli Seçim yöntemi uygulanmıştır.

Yeni oluşturulan X_important veri kümesinden "success_pred" sütunu çıkarılarak, yalnızca özniteliklerin bulunduğu bir veri kümesi olan Y_important elde edilmiştir. Daha sonra, öznitelik seçiminde daha fazla odaklanılmış ve daha fazla deney yapılarak en iyi sonuçların elde edilmesi amaçlanmıştır. Bu amaç doğrultusunda, RandomForestClassifier kullanılarak gerçekleştirilen SequentialFeatureSelector yönteminde n_features_to_select parametresi dikkate alınmıştır.

n_features_to_select parametresi, en iyi öznitelik sayısını belirlemek için kullanılmıştır. Farklı n_features_to_select değerleri denenerek, her bir değerin model performansı üzerindeki etkisi değerlendirilmiştir. Bu deneyler sonucunda elde edilen sonuçlar incelenmiş ve n_features_to_select parametresinin 20 olarak belirlenmesinin en optimal sonuçları sağladığı görülmüştür. Dolayısıyla, sınıflandırma performansını artırmanın ve

veri kümesini en iyi şekilde temsil etmenin en uygun değerinin 20 olduğuna karar verilerek bu değer tercih edilmiştir.

Bu adımların amacı, veri kümesinin boyutunu etkili bir şekilde azaltarak işlem süresini optimize etmek ve aynı zamanda sınıflandırma performansını en üst düzeye çıkarmaktır. Bu nedenle, n_features_to_select parametresinin 20 olarak belirlenmesi, dikkatli bir değerlendirme ve deney süreci sonucunda yapılan bir tercihtir.

SequentialFeatureSelector sonrasında, get_support() yöntemi kullanılarak en iyi özniteliklerin indeksleri elde edilmiştir. Bu indeksler kullanılarak, Y_important veri kümesinde sadece seçilen en iyi özniteliklerin bulunduğu yeni bir veri kümesi (X_forward) oluşturulmuştur.

Bu hibrit yöntem sayesinde, veri kümesi içerisindeki anlamsız özniteliklerin çıkarılması ve en iyi özniteliklerin seçilerek sınıflandırma performansının artırılması hedeflenmiştir. Bu adımlar, deney sayısını artırmak ve modelin daha hızlı ve etkili çalışmasını sağlamak amacıyla bir araya getirilmiştir.

4.3. Sınıflandırma

Öznitelik seçimi ve veri ön işleme adımlarının tamamlanmasının ardından, oluşturulan özniteliklerle sınıflandırma işlemi gerçekleştirilmiştir. Bu aşamada, sınıflandırma modelinin eğitimi ve performans değerlendirmesi için veri kümesindeki kayıtlar ayrılmıştır. Veri kümesindeki kayıtların %80'i eğitim için kullanılmış ve modelin öğrenmesi amaçlanmıştır. Kalan %20'lik kısım ise modelin test edilmesi ve performansının değerlendirilmesi için ayrılmıştır.

Bu çalışmada, sınıflandırma için ikili sınıflandırma yöntemi tercih edilmiştir. İkili sınıflandırma, sonuçları yorumlamak ve sunmak için daha basit bir yapıya sahip olduğundan, bu alandaki karmaşık veri setleri için ideal bir seçenektir. Özellikle, KDDCUP 99 veri kümesi gibi dengesiz veri setleriyle çalışırken, ikili sınıflandırma, sınıflar arası dengesizlikleri daha etkili bir şekilde ele alarak, sınıflandırma performansını optimize etmeye yardımcı olmaktadır.

Dahası, teknik olmayan paydaşlarla sonuçların paylaşılması durumunda, ikili sınıflandırmanın sağladığı sezgisel ve anlaşılır sonuçlar oldukça değerlidir. İkili sınıflandırma, çok sınıflı sınıflandırmaya kıyasla eğitim ve değerlendirme süreçlerini

hızlandırabilmektedir. Ayrıca, bu yaklaşımın kullanılması, performans metrikleri gibi değerlendirme araçlarının daha etkili bir şekilde uygulanmasına olanak tanımaktadır.

Saldırı tespiti gibi kritik görevlerde, bir saldırının var olup olmadığını hızla belirlemek esastır. Bu nedenle, ilk aşamada saldırıların tespiti, onların türlerinden daha kritik bir öneme sahip olabilmektedir. İkili sınıflandırma bu amacı doğrudan destekler ve bu çalışmanın temel ihtiyaçlarına hizmet etmektedir.

Çalışmada, farklı sınıflandırıcı algoritmaları kullanılarak model performansı değerlendirilmiştir. Bu algoritmalar arasında Decision Tree (DT), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF) ve K-Nearest Neighbors (KNN) yer almaktadır. Her bir sınıflandırıcı algoritması, belirlenen öznitelikler kullanılarak eğitilmiş ve eğitilen modelin test verisi üzerindeki performansı değerlendirilmiştir.

Bu süreçte, PCA + RFECV ve RFECV + FS hibrit öznitelik indirgeme yöntemleri kullanılarak elde edilen özniteliklerin sınıflandırma modellerine nasıl etki ettiği incelenmiştir. Bu yöntemlerin kullanılmasıyla, veri kümesinin boyutu etkili bir şekilde azaltılmış ve model performansı optimize edilmiştir. Bu adımlar sayesinde, hem sınıflandırma doğruluğu artırılmış hem de işlem süresi kısaltılarak verimlilik elde edilmiştir.

4.4. Araştırma Bulguları

Deney sonuçlarının değerlendirilmesinde Karmaşıklık Matrisi (Confusion Matrix) ile Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) ve F1 metriklerinden yararlanılmıştır. Çalışmanın ilk aşamasında, öznitelik indirgeme yöntemi uygulanmadan sınıflandırma algoritmalarının performans metrikleri değerlendirilmiştir. Bu temel değerlendirme, algoritmanın orijinal veri kümesi üzerinde nasıl bir performans gösterdiğini ortaya koymak amacıyla gerçekleştirilmiştir.

Bu temel değerlendirmenin ardından, hibrit bir yaklaşım olan PCA + RFECV yöntemi uygulanmıştır. PCA + RFECV yönteminde, ilk olarak temel bileşen analizi (PCA) ile boyut indirgeme işlemi gerçekleştirilmiştir. Bu aşamada, veri kümesindeki öznitelikler arasındaki yüksek korelasyon azaltılarak daha az sayıda bileşen elde edilmiştir. PCA sonrasında, RFECV yöntemi uygulanarak en iyi öznitelikler belirlenmiştir.

Bu adımlarla, orijinal veri kümesi üzerinden elde edilen temel değerlendirme sonuçları ile öznitelik indirgeme sonrası elde edilen sonuçlar karşılaştırılmıştır. Böylelikle, öznitelik seçimi ve indirgeme yöntemlerinin sınıflandırma performansına olan etkisi değerlendirilmiştir.

Öznitelik indirgeme yöntemi uygulamadan sınıflandırmaya ait elde edilen değerler Tablo 4.3 ve Tablo 4.4'te gösterilmektedir.

Tablo 4.3 : Öznitelik indirgeme yöntemi kullanılmadan yapılan sınıflandırmaya ait test sonuçları.

Sınıflayıcı	Doğruluk (Accuracy Score)	Kesinlik (Precision Score)	Duyarlılık (Recall Score)	F1 Score
KNN	0,9631	0,9741	0,9524	0,9631
NB	0,9518	0,9532	0,9518	0,9517
LR	0,9405	0,9747	0,9059	0,9391
DT	0,9806	0,9964	0,9652	0,9806
RF	0,9815	0,9965	0,9668	0,9806

Tablo 4.4 : Öznitelik indirgeme yöntemi kullanılmadan yapılan sınıflandırmaya ait doğrulama sonuçları.

Sınıflayıcı	Karmaşıklık Matrisi $\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$	Çapraz Doğrulama
KNN	$\begin{bmatrix} 11653 & 310 \\ 584 & 11691 \end{bmatrix}$	0,9664
NB	$\begin{bmatrix} 11712 & 251 \\ 917 & 11358 \end{bmatrix}$	0,9539
LR	$\begin{bmatrix} 11675 & 288 \\ 1154 & 11121 \end{bmatrix}$	0,9440
DT	$\begin{bmatrix} 11921 & 42 \\ 426 & 11849 \end{bmatrix}$	0,9826
RF	$\begin{bmatrix} 11922 & 41 \\ 407 & 11868 \end{bmatrix}$	0,9833

Birinci öznitelik indirgeme yönteminin sonuçlarına göre sınıflandırmaya ait elde edilen değerler Tablo 4.5 ve Tablo 4.6'da gösterilmektedir.

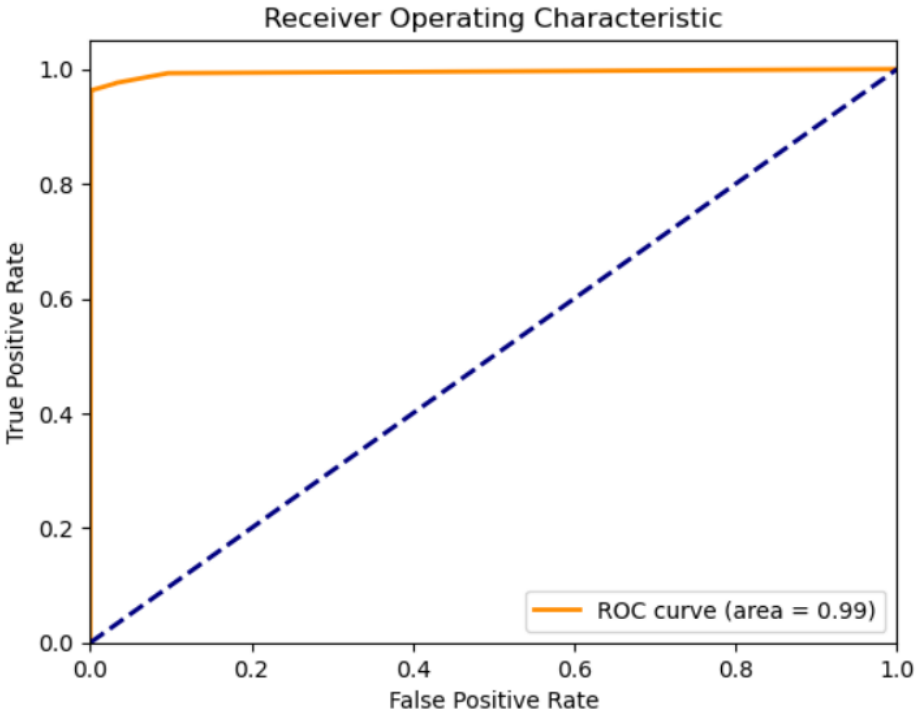
Tablo 4.5 : Birinci öznitelik indirgeme yönteminin sonuçlarına göre sınıflandırmaya ait test sonuçları.

Sınıflayıcı	Doğruluk (Accuracy Score)	Kesinlik (Precision Score)	Duyarlılık (Recall Score)	F1 Score
KNN	0,9723	0,9708	0,9746	0,9806
NB	0,6193	0,7719	0,6193	0,5604
LR	0,9619	0,9816	0,9424	0,9616
DT	0,9813	0,9970	0,9660	0,9812
RF	0,9819	0,9970	0,9671	0,9818

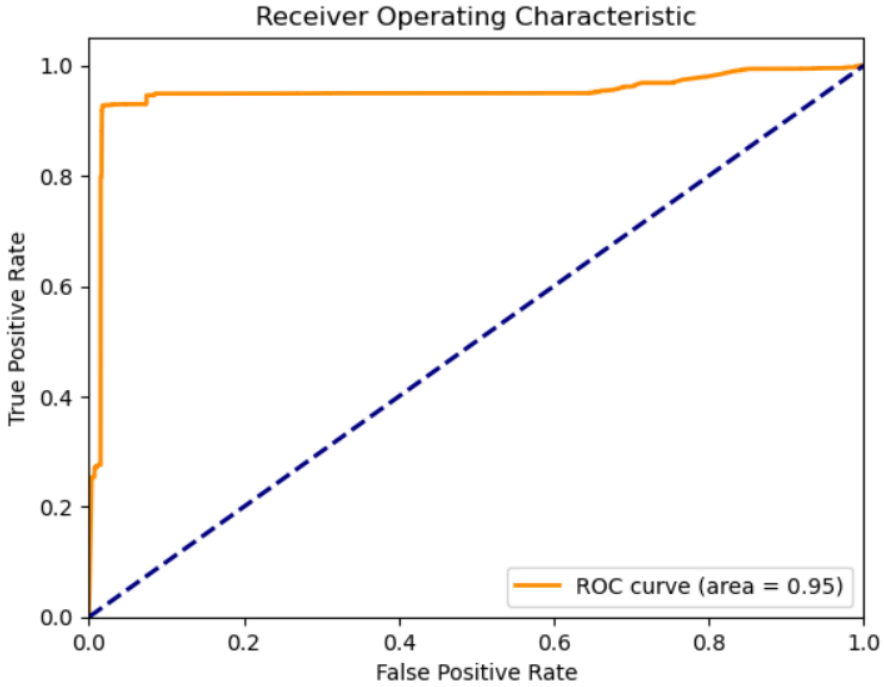
Tablo 4.6 : Birinci öznitelik indirgeme yönteminin sonuçlarına göre sınıflandırmaya ait doğrulama sonuçları.

Sınıflayıcı	Karmaşıklık Matrisi $\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$	Çapraz Doğrulama
KNN	$\begin{bmatrix} 11604 & 359 \\ 311 & 11964 \end{bmatrix}$	0,9784
NB	$\begin{bmatrix} 11878 & 85 \\ 9140 & 3135 \end{bmatrix}$	0,6263
LR	$\begin{bmatrix} 11747 & 216 \\ 706 & 11569 \end{bmatrix}$	0,9643
DT	$\begin{bmatrix} 11928 & 35 \\ 417 & 11858 \end{bmatrix}$	0,9834
RF	$\begin{bmatrix} 11928 & 35 \\ 403 & 11872 \end{bmatrix}$	0,9841

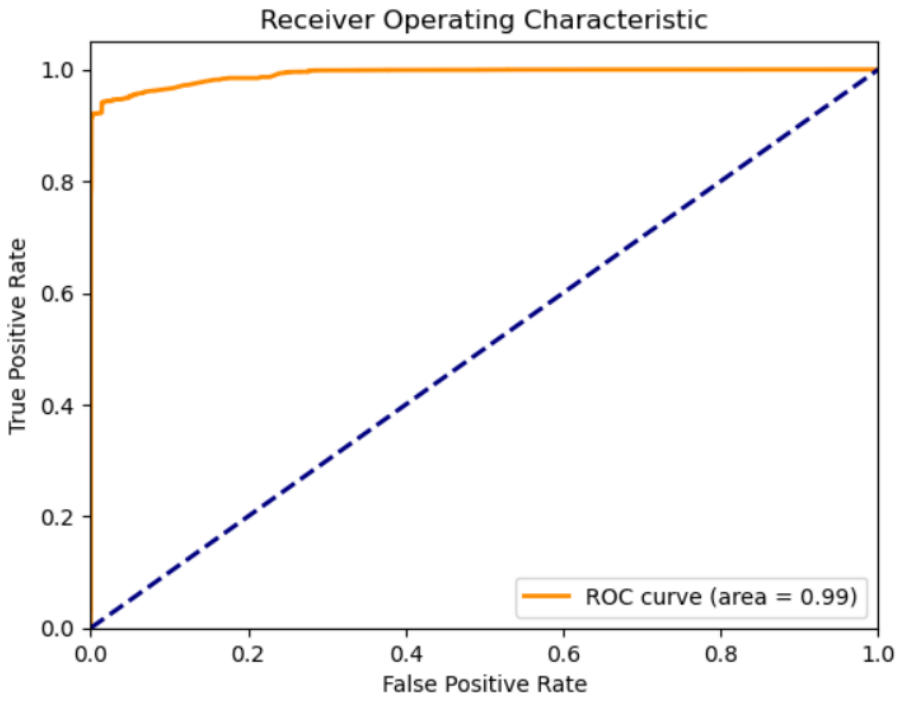
İlk sınıflandırma sonuçlarına göre gösterilen ROC eğrileri Şekil 4.3, Şekil 4.4, Şekil 4.5, Şekil 4.6 ve Şekil 4.7'de gösterilmektedir.



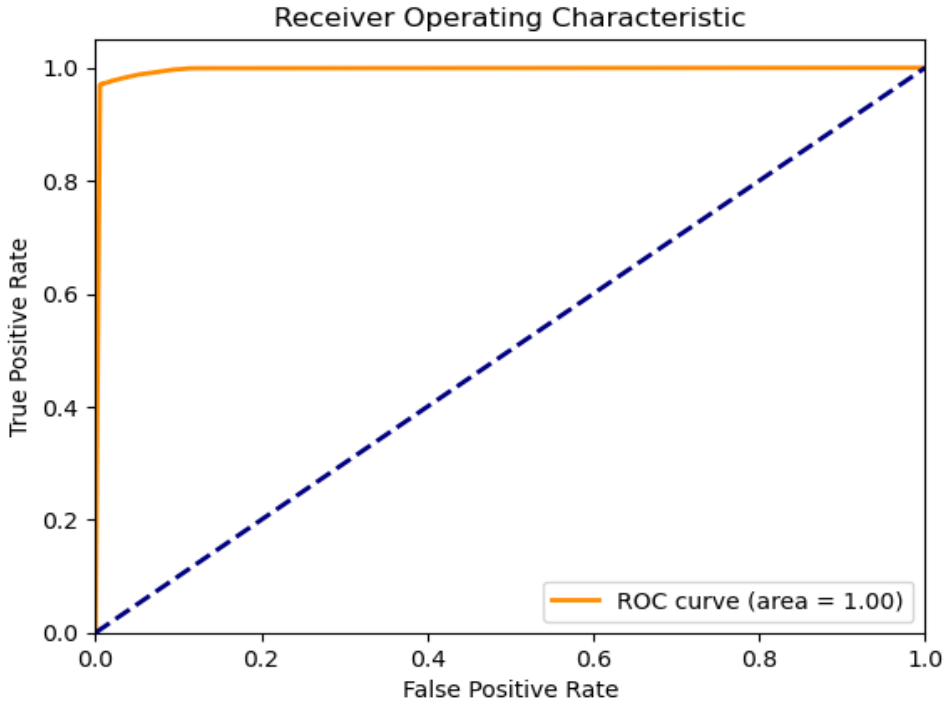
Şekil 4.3 : Birinci deneyde KNN sınıflayıcısına ait ROC eğrisi



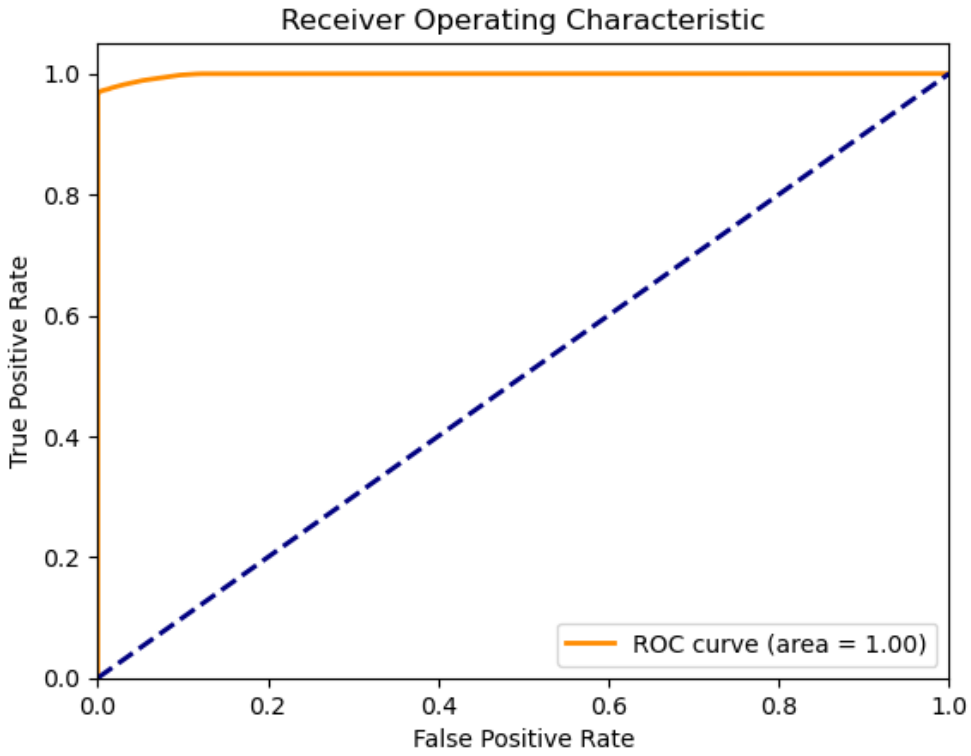
Şekil 4.4 : Birinci deneyde NB sınıflayıcısına ait ROC eğrisi



Şekil 4.5 : Birinci deneyde LR sınıflayıcısına ait ROC eğrisi



Şekil 4.6 : Birinci deneyde DT sınıflayıcısına ait ROC eğrisi



Şekil 4.7 : Birinci deneyde RF sınıflayıcısına ait ROC eğrisi

Bir diğer deneyde ise RFECV + FS yöntemi kullanılarak sınıflandırma işlemleri gerçekleştirilmiştir. Bu süreç, veri kümesinin özniteliklerini seçmek ve sınıflandırma performansını değerlendirmek için iki aşamada gerçekleştirilmiştir.

İlk aşamada, RFECV yöntemi kullanılmıştır. Bu aşamada, Decision Tree sınıflandırıcısı (estimator) kullanılarak Recursive Feature Elimination Cross Validation (RFECV) gerçekleştirilmiştir. RFECV, özniteliklerin sırayla çıkarılması ve çapraz doğrulama ile model performansının değerlendirilmesi işlemidir. Aynı zamanda, veri kümesindeki özniteliklerin önem sıralamasını sağlar. Bu adımdan sonra, RFECV ile belirlenen önemli öznitelikler "important_features" adlı değişkende saklanmıştır.

İkinci aşamada, RFECV ile belirlenen önemli öznitelikler kullanılarak yeni bir veri kümesi olan "X_important" oluşturulmuştur. Bu yeni veri kümesi üzerinde ileri yönelimli seçim (FS) yöntemi uygulanmıştır. Bu yöntemde, RandomForestClassifier kullanılarak SequentialFeatureSelector sınıflandırıcısı tanımlanmış ve n_features_to_select parametresi 20 olarak belirlenmiştir. Veri kümesi "Y_important" olarak tanımlanmış ve sınıflandırıcı "selector" ile öznitelik seçimi gerçekleştirilmiştir. En iyi özniteliklerin indeksleri "selected_features" değişkeninde saklanmıştır.

Sonuç olarak, RFECV + FS yöntemi ile belirlenen en iyi öznitelikler kullanılarak sınıflandırma işlemleri gerçekleştirilmiş ve bu yöntemin öznitelik seçimi aşamasındaki etkisi değerlendirilmiştir. Sınıflandırma işlemlerine ait sonuçlar Tablo 4.7 ve Tablo 4.8' de gösterilmektedir.

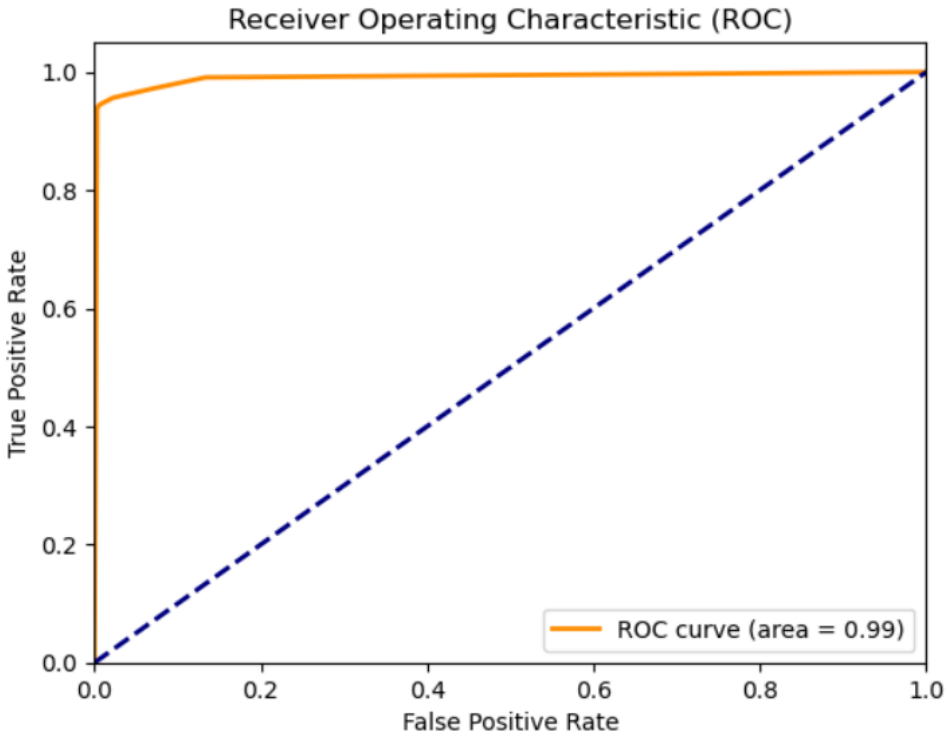
Tablo 4.7 : İkinci öznitelik indirgeme yönteminin sonuçlarına göre sınıflandırmaya ait test sonuçları.

Sınıflayıcı	Doğruluk (Accuracy Score)	Kesinlik (Precision Score)	Duyarlılık (Recall Score)	F1 Score
KNN	0,9668	0,9776	0,9564	0,9669
NB	0,8914	0,9010	0,8914	0,8909
LR	0,9702	0,9930	0,9479	0,9699
DT	0,9812	0,9966	0,9662	0,9812
RF	0,9820	0,9971	0,9672	0,9819

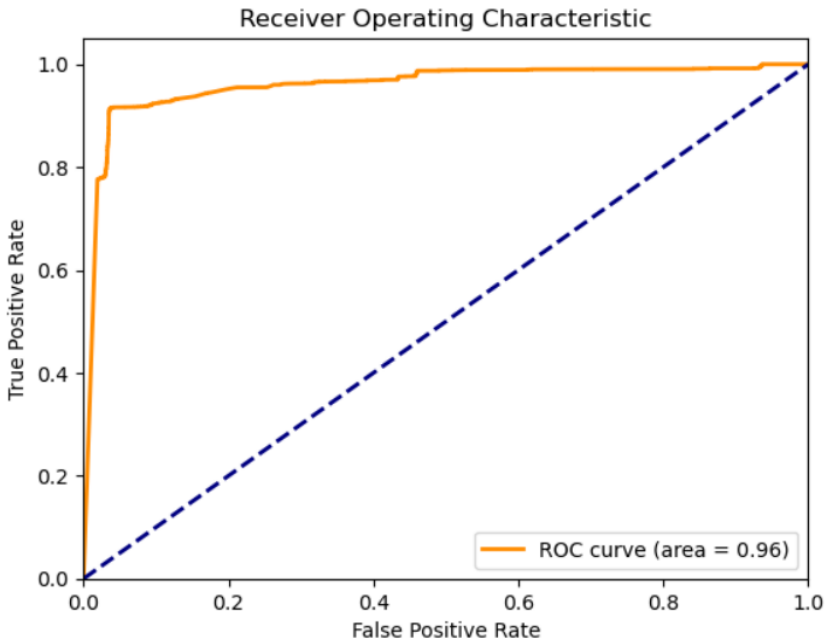
Tablo 4.8 : İkinci öznitelik indirgeme yönteminin sonuçlarına göre sınıflandırmaya ait doğrulama sonuçları.

Sınıflayıcı	Karmaşıklık Matrisi $\begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$	Çapraz Doğrulama
KNN	$\begin{bmatrix} 11941 & 63 \\ 1599 & 48603 \end{bmatrix}$	0,947
NB	$\begin{bmatrix} 11573 & 390 \\ 2240 & 10035 \end{bmatrix}$	0,8949
LR	$\begin{bmatrix} 11881 & 82 \\ 639 & 11636 \end{bmatrix}$	0,9716
DT	$\begin{bmatrix} 11923 & 40 \\ 414 & 11861 \end{bmatrix}$	0,9834
RF	$\begin{bmatrix} 11929 & 34 \\ 402 & 11873 \end{bmatrix}$	0,9842

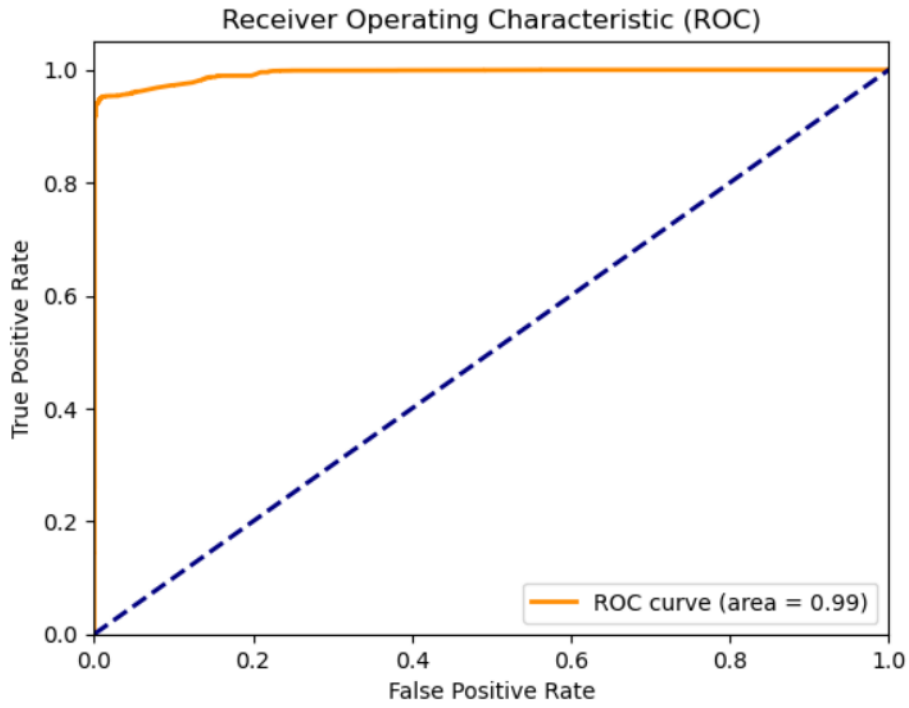
RFECV ve FS hibrit yöntemi kullanılarak 20 özniteliğe indirgenen veri kümesi üzerinden yapılan analizler sonucunda elde edilen ROC eğrileri Şekil 4.8, Şekil 4.9, Şekil 4.10 Şekil 4.11 ve Şekil 4.12' de görülmektedir.



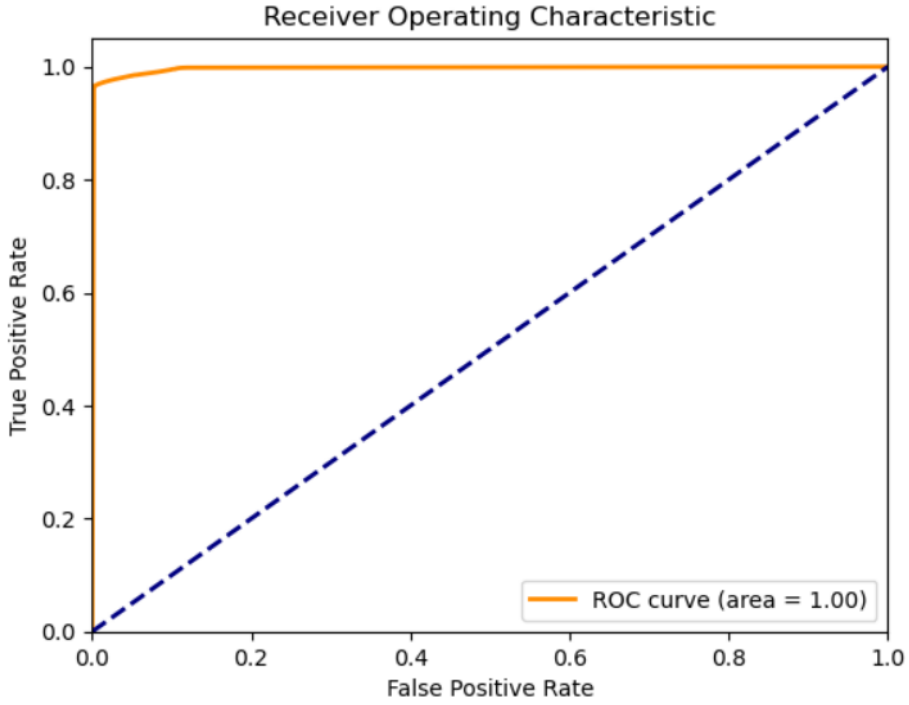
Şekil 4.8 : İkinci deneyde KNN sınıflayıcısına ait ROC eğrisi



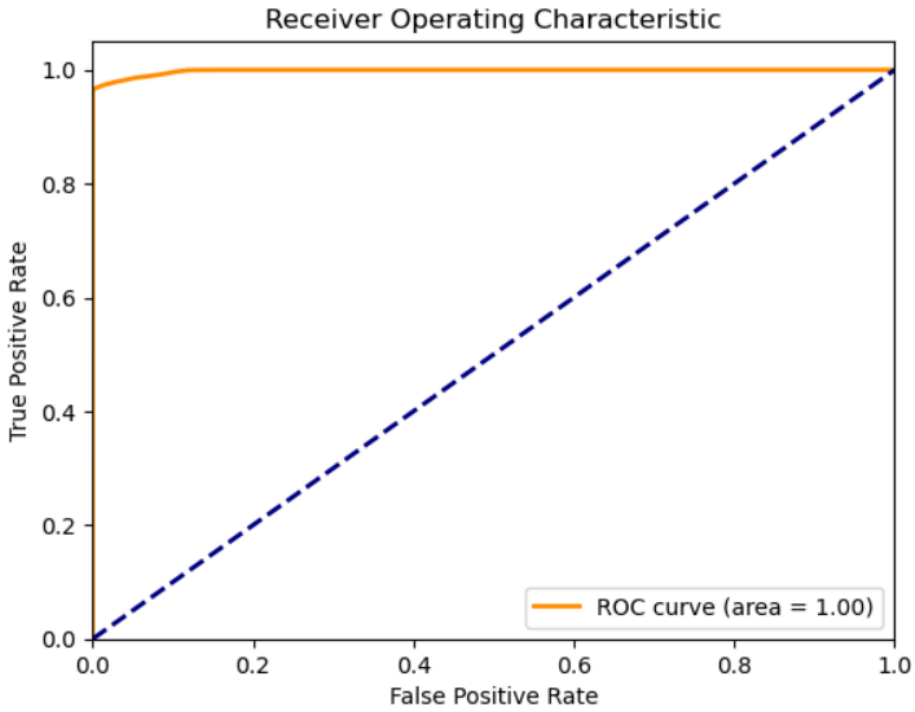
Şekil 4.9 : İkinci deneyde NB sınıflayıcısına ait ROC eğrisi



Şekil 4.10 : İkinci deneyde LR sınıflayıcısna ait ROC eğrisi



Şekil 4.11 : İkinci deneyde DT sınıflayıcısına ait ROC eğrisi



Şekil 4.12 : İkinci deneyde RF sınıflandırıcısına ait ROC eğrisi

4.6. İki Farklı Öznitelik Gruplarıyla Yapılan Deney Sonuçlarının Tartışılması

Tablo 4.3, Tablo 4.5, Tablo 4.6 ve Tablo 4.7 'de sunulan sonuçları incelediğimizde, öznitelik indirgeme yöntemi kullanılmadan yapılan sınıflandırmalarla, PCA+RFECV ve RFECV+FS yöntemleriyle gerçekleştirilen öznitelik indirgeme işlemlerinin sınıflandırma performansına etkileri görülmektedir.

İlk tabloya (öznitelik indirgeme uygulanmadan) odaklanıldığında, Naive Bayes sınıflandırıcısının performansının oldukça tatmin edici olduğu görülmektedir. Ancak PCA+RFECV yöntemi uygulandığında, doğruluk, kesinlik ve F1 skoru metriklerinde belirgin bir düşüş yaşanmıştır. Bu, PCA+RFECV yönteminin, özellikle Naive Bayes için, veri kümesinin özelliklerine veya özellikler arasındaki ilişkilere müdahale ettiğini gösteriyor olabilir. Öte yandan, RFECV+FS yöntemi ile elde edilen sonuçlarda Naive Bayes'in performansı tekrar yükselmiş; fakat başlangıç değerlerine nazaran hala daha düşük kalmıştır. Bu da RFECV+FS yönteminin, PCA+RFECV'ye kıyasla, Naive Bayes için daha uygun bir öznitelik seçimi yöntemi olduğunu işaret etmektedir.

Logistic Regression (LR) sınıflandırıcısının performansını değerlendirdiğimizde, PCA+RFECV öznitelik indirgeme yöntemi uygulandığında, doğruluk, kesinlik,

duyarlılık ve F1 skoru gibi temel metriklerde artışlar kaydedilmiştir. Bu artışlar, PCA+RFECV'nin LR sınıflandırması için veri kümesinin yapısal özelliklerini ve özellikler arasındaki ilişkileri optimize ettiğini düşündürmektedir. Bunun yanında, RFECV+FS öznitelik indirgeme yöntemi uygulandığında da benzer bir artış gözlemlenmiştir. Ancak bu artış, başlangıç değerlerine kıyasla daha belirgindir. Bu durum, RFECV+FS'nin LR sınıflandırıcısı için daha etkili bir öznitelik seçimi yöntemi olabileceğini göstermektedir.Genel bir değerlendirme yapılacak olursa; iki farklı öznitelik indirgeme yönteminin, Logistic Regression sınıflandırmasının performansının olumlu bir şekilde etkilediği söylenebilmektedir. Bu etki, özellikle RFECV+FS yöntemi ile daha da belirginleşmiştir. Bu sonuçlar, öznitelik indirgeme ve seçimi yöntemlerinin, LR sınıflandırıcısının performansı üzerinde ne kadar kritik bir rol oynadığını göstermektedir.

Decision Tree (DT) ve Random Forest (RF) sınıflandırıcıları için, her iki öznitelik indirgeme yönteminin de sınıflandırma sonuçları üzerinde olumlu bir etkisi olduğu gözlemlenmektedir. Özellikle RF'nin doğruluk oranı, PCA+RFECV ve RFECV+FS yöntemleri arasında neredeyse değişmemiştir. Bu, RF algoritmasının öznitelik seçimi yöntemlerine karşı daha dirençli olabileceğini göstermektedir.

K-En Yakın Komşu (KNN) sınıflandırıcı performansı, öznitelik indirgeme yöntemlerinin uygulanması sonucunda genel olarak artış göstermiştir. RFECV+FS yönteminin, KNN'nin kesinlik ve duyarlılık metriklerini iyileştirmede özellikle etkili olduğu görülmektedir. Bu durum, KNN'nin öznitelik seçimi ve boyut indirgeme yöntemlerine oldukça duyarlı olduğunu belirtmektedir. Özellikle, KNN'nin yapısal özellikleri dikkate alındığında, az sayıda öznitelikle daha hızlı ve etkili sonuçlar üretebildiği anlaşılmaktadır. Bu nedenle, KNN sınıflandırıcısında öznitelik indirgeme yöntemlerinin kritik bir rol oynadığı söylenebilmektedir.

BÖLÜM 5. SONUÇLAR

Bu çalışmanın amacı, ağ trafiğindeki anomali veya anomali davranışları saptamak ve bunları raporlamak için bilgisayar sistemleri oluşturmaktır. Anomali tespiti sistemleri, ağ trafiğinin normal davranışlarını öğrenerek, beklenmeyen, anormal davranışları tespit edebilmektedirler. Bu sistemler, ağdaki riskleri azaltmak ve önemli verilerin korunmasına yardımcı olmak için kullanılabilirler. Bu hedefi gerçekleştirmek için daha önce yapılan çalışmaların detaylı incelemesi yapılmıştır ve literatür araştırmalarında ağ anomali tespiti için önemli kısımlar belirlenmiştir. Araştırmalar incelendiğinde son zamanlarda makine öğrenmesi algoritmalarının yüksek doğruluk oranına sahip olduğu görülmüştür. Farklı türde ağ saldırıları içerdiği ve gerçek dünya senaryolarını yansıttığı için KDDCUP99 veri kümesi kullanılmıştır.

Çalışma da sınıflayıcı olarak Decision Tree (DT), Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF) ve K-Nearest Neighbors algoritmaları (KNN) kullanılmış ve değerlendirme metrikleri olarak Çapraz doğrulama ile birlikte ROC eğrileri tercih edilmiştir.

Önişleme sürecinde orjinal veri kümesinin oldukça büyük olmasından kaynaklı işlemesi oldukça zaman alacağından corrected olarak adlandırılan ve orjinal veri kümesinden daha küçük boyutlu olan veri kümesi kullanılmıştır. Bu veri kümesi orjinal veri kümesinin bazı özellikleri açısından filtrelenerek elde edilmiştir. Bu filtreleme işlemi, bazı özellikleri düşük olan ağ trafiği paketlerini çıkartmak ve daha anlamlı verilerle çalışmak için kullanılmıştır. corrected isimli veri kümesinden modele katkısı olmadığı ve performansı olumsuz etkilediği için duration sütunu çıkartılmıştır. Kategorik veriler olan protocol_type, flag ve service öznitelikleri one-hot-encoding yöntemi kullanılarak sayısal değerlere dönüştürülmüştür. Son olarak veri kümesi normalizasyon işlemine tabi tutulup hazır hale getirilmiştir.

Tablo 4.5, Tablo 4.6, Tablo 4.7 ve Tablo 4.8' de sunulan sonuçlar, öznitelik seçimi sürecinde kullanılan iki farklı hibrit yöntemin - PCA + RFECV ve RFECV + FS - sınıflandırma performansına olan etkilerini açıkça göstermektedir. Bu sonuçlar, veri kümesinin boyutunu etkili bir şekilde azaltmanın ve anlamlı özniteliklerin seçilmesinin sınıflandırma işlemlerinin başarısını artırabileceğini göstermektedir.

Öznitelik indirgeme uygulamadan elde edilen sonuçlar, özellikle Random Forest (RF), Decision Tree (DT) ve K-Nearest Neighbors (KNN) sınıflandırıcıları için yüksek doğruluk, kesinlik, duyarlılık ve F1 metrikleri ile dikkat çekicidir. Ancak PCA+RFECV öznitelik indirgeme yöntemi uygulandığında, özellikle Naive Bayes (NB) için metriklerde dramatik bir düşüş gözlemlenmiştir. Bu, PCA'nın, özellikle NB algoritması için, veri kümesinin özellikler arasındaki ilişkisini bozabileceğini düşündürebilir. Ancak, RFECV+FS yöntemi uygulandığında, NB'nin performansındaki düşüşe rağmen, diğer sınıflandırıcıların, özellikle Logistic Regression (LR), performansında bir artış gözlemlenmiştir. Bu, RFECV+FS'nin sınıflandırma performansını optimize etme kapasitesini gösteriyor olabilir.

KNN sınıflandırıcısı, özellikle RFECV+FS öznitelik indirgeme yöntemi uygulandığında en yüksek doğruluk, kesinlik, duyarlılık ve F1 metriklerini sunmuştur. Bu, KNN'nin, daha az öznitelikle daha hızlı ve daha etkili sonuçlar üretebildiğini ve özellikle öznitelik indirgeme yöntemlerine oldukça duyarlı olduğunu göstermektedir.

Genel olarak, bu çalışmanın sonuçları, öznitelik seçiminin ve indirgemenin sınıflandırma modellerinin başarısını belirlemedeki kritik önemini vurgulamaktadır. Sınıflandırma performansını artırmada kullanılan öznitelik indirgeme yöntemlerinin etkinliği, sınıflandırma algoritmasına göre değişkenlik gösterebilir. Bu çalışma, sınıflandırma problemleri üzerinde çalışan akademisyenler ve endüstriyel kurumlar için önemli bir referans niteliğindedir. Gelecekte, öznitelik seçimi yöntemlerinin daha geniş veri kümeleri ve farklı sınıflandırıcılarla nasıl performans gösterdiğinin derinlemesine incelenmesi önerilebilir.

KAYNAKLAR

[1] Thottan, M., Ji, C. (2003). Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, 51(8), 2191-2204

[2] Url-1 <*https://www.datascience.com/blog/python-anomaly-detection*>, erişim tarihi: 24.04.2023.

[3] Denning, D. E. (1987). An intrusion detection model. *IEEE Transactions on Software Engineering*, 13(2), 222-232.

[4] Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. *In Proceedings of the IEEE International Joint Conference on Neural Networks* (Vol. 2, pp. 1702-1707).

[5] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2), 18-28.

[6] Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). Intrusion detection in 21st century: A survey. Journal of Network and Computer Applications, 75, 303-322.

[7] Ahmed, M., Mahmood, A. N., Hu, J., A survey of network anomaly detection techniques. *J Netw. Comput. Appl.*, 60: 19-31, 2016.

[8] Url-2 <*https://www.sans.org/*>, erişim tarihi : 03.02.2023.

[9] Paliwal, S., Gupta, R. (2012). Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm. *International Journal of Computer Applications and Technology*, 60(19), 57-62.

[10] Clarke, N., Li, F., & Rak, M. (2009). SQL Injection Attacks and Defense. *Syngress*.

[11] Haines, J. W., Rossev, L. M., Lippmann, R. P., Cunningham, R. K. (2001). Extending the DARPA off-line intrusion detection evaluations. *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, Anaheim, CA, USA, 35-45.

[12] Hasan, D. (2017). Cost-sensitive access control for detecting Remote to Local (R2L) and User to Root (U2R) Attacks. *International Journal of Computer Applications and Technology*, 43(2), 124-129.

[13] Anley, C. (2002). Advanced SQL Injection In SQL Server Applications. *NGSSoftware Insight Security Research (NISR).*

[14] Saleh, M., & Shamsuddin, A. (2013). Cross Site Scripting Attacks: Types, Methods and Prevention. *International Journal of Engineering and Technology*, 5(2), 1756-1761.

[15] Jakobsson, M., & Myers, S. A. (2007). Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft. *IEEE Security & Privacy*, 5(6), 26-33.

[16] Deshmukh, S. R., & Bharadi, V. A. (2017). ARP Spoofing Attack Detection and Prevention Technique using OpenFlow. *International Journal of Computer Applications*, 161(12), 30-34.

[17] Masek, P., & Celeda, P. (2016). Detection and Prevention of DNS Spoofing Attacks. *Proceedings of the 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 109-114.

[18] Sun, C., & Fang, B. (2018). Detecting IP Spoofing Attacks Using Machine Learning Algorithms. *Proceedings of the International Conference on Information Science and System (ICISS)*, 90-94.

[19] Mirkovic, J., Reiher, P., & Knightly, E. (2002). Simulation-based Analysis of Denial-of-Service Countermeasures. *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 57-68.

[20] Dittrich, D., Kennerly, K., & Longstaff, T. (2000). Internet Denial of Service: Attack and Defense Mechanisms. *Prentice Hall.*

[21] Newsham, T., & Anthony, D. (2003). Attacks on Internet Protocols: A Comprehensive View. IEEE Communications Surveys & Tutorials, 5(4), 64-87.

[22] Stallings, W. (2013). *Network Security Essentials: Applications and Standards*. Pearson Education.

[23] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A. (2009). A detailed analysis of KDDCUP 99 data set. *IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada.*

[24] Moustafa, N., Slay, J. (2015). The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Data Set and the Comparison with the KDD99 Data Set. *Information Security Journal: A Global Perspective*, 24(1-3), 18-31.

[25] Url-3 <*https://www.unb.ca/cic/datasets/ids-2017.html*>, erişim Tarihi: 17.09.2019.

[26] Hirose, Shunsuke, Yamanishi, Kenji, Nakata, Takayuki, & Fujimaki, Ryohei. (2009). Network anomaly detection based on Eigen equation compression. Sayfa 1185-1194. doi: 10.1145/1557019.1557147.

[27] Sheyner O., Haines J., Javitz H., Stolfo S.,(2000) Intrusion Detection in Computer Networks Based on Statistical Analysis of Traffic Parameters. *International Conference on Cyber Conflict*. IEEE.

[28] Last, M., Kandel, A., Bunke, H. (2004). *Data Mining in Time Series Databases*. World Scientific.

[29] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: a survey. *ACM Computing Surveys*, 15(3).

[30] Liu, D., Lung, C.-H., Lambadaṅs, I., & Seddigh, N. (2013). Network traffic anomaly detection using clustering techniques and performance comparison. In *Canadian Conference on Electrical and Computer Engineering* (pp. 1-4).

[31] Zhao, S., Chandrashekar, M., Lee, Y., & Medhi, D. (2015). Real-time network anomaly detection system using machine learning. *In 2015 11th International Conference on the Design of Reliable Communication Networks, DRCN 2015* (pp. 267-270).

[32] Thing, V. L. L. (2017). IEEE 802.11 Network Anomaly Detection and Attack Classification: A Deep Learning Approach. *In 2017 IEEE Wireless Communications and Networking Conference (WCNC*) (pp. 1-6). San Francisco, CA, USA: IEEE. https://doi.org/10.1109/WCNC.2017.7925567.

[33] Kyaw, T., Oo, M. Z., & Khin, C. S., (2020), Machine-Learning Based DDOS Attack Classifier in Software Defined Network, *17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 431-434, https://doi.org/10.1109/ECTI-CON49241.2020.9158230.

[34] Zhang, J., Perdisci, R., Lee, W., Sarfraz, U., & Luo, X., (2011), Detecting stealthy P2P botnets using statistical traffic fingerprints, *DSN*, 121-132, https://doi.org/10.1109/DSN.2011.5958212.

[35] Tan, Z., Jamdagni, A., He, X., Nanda, P., & Liu, R., (2011), Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis, *International Journal of Recent Technology and Engineering (IJRTE)*, 756-765, https://doi.org/10.1145/2490428.2490450.

[36] Limthong, K. (2015). *A wavelet-based anomaly detection for outbound network traffic.* (Doctor of Philosophy thesis). Department of Informatics, School of Multidisciplinary Sciences, The Graduate University for Advanced Studies (SOKENDAI).

[37] Bloedorn, E., Christiansen, A., Hill, W., Skorupka, C., Talbot, L., & Tivel, J., (2002), Data Mining for Network Intrusion Detection: How to Get Started, *InternationalConference on Data Mining*, IEEE.

[38] Lakshman, M., (2003), Detecting Network Intrusions via Sampling: A Game Theoretic Approach, *Proceedings of the IEEE INFOCOM 2003 - The Conference on Computer Communications, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies,*1880.

[39] Boughaci, D., Drias, H., Bendib, A., Bouznit, Y., & Benhamou, B., (2006), Distributed Intrusion Detection Framework based on Autonomous and Mobile Agents, *2006 International Conference on Dependability of Computer Systems*, 248-255, doi: 10.1109/DEPCOS-RELCOMEX.2006.19.

[40] Jain, R., & Abouzakhar, N., (2013), A Comparative Study of Hidden Markov Model and Support Vector Machine in Anomaly Intrusion Detection, *Journal of Internet Technology and Secured Transaction*, 2, 176-184, doi: 10.20533/jitst.2046.3723.2013.0023

[41] Shyu, M.-L., Chen, S.-C., Sarinnapakorn, K., & Chang, L., (2003), A Novel Anomaly Detection Scheme Based on Principal Component Classifier, *in*

conjunction with the Third IEEE International Conference on Data Mining (ICDM'03), IEEE Foundations and New Directions of Data Mining Workshop.

[42] García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E., (2009), Anomaly-based network intrusion detection: Techniques, systems and challenges, *Computers & Security*, 28, 18-28, doi: 10.1016/j.cose.2008.08.003.

[43] G. Poojitha, K. N. Kumar and P. J. Reddy, Intrusion Detection using Artificial Neural Network, *2010 Second International conference on Computing, Communication and Networking Technologies*, Karur, India, 2010, pp. 1-7, doi: 10.1109/ICCCNT.2010.5592568.

EKLER

EK-1 : Veri Kümesindeki Saldırı Verileri

Saldırı Adı	Kayıt Sayısı
Smurf.	164091
normal.	60593
neptune.	58001
snmpgetattack.	7741
mailbomb.	5000
guess_passwd.	4367
snmpguess.	2406
Satan	1633
Warezmaster	1602
Back	1098
Mscan	1053
Apache2	794
Processtable	759
Saint	736
Portsweep	354
Ipsweep	306
Httptunnel	158
Pod	87
Nmap	84
Buffer_overflow	22
Multihop	18
Sendmail	17
Ps	16
Rootkit	13

Xterm	13
Teardrop	12
Xlock	9
Land	9
Xsnoop	4
ftp_write	3
Loadmodule	2
Perl	2
Udpstorm	2
Worm	2
Phf	2
Sqlattack	2
Imap	1