

**YALOVA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**-BİTİRME TEZİ-**

**YAZILIM TANIMLI AĞLARA YAPILAN HİZMET DIŞI BIRAKMA SALDIRILARININ MAKİNE ÖĞRENMESİ ALGORİTMALARIYLA TESPİTİ**

**Mert KAHRAMAN**

**Bitirme Tezi Danışmanı: Dr. Öğr. Üyesi Yunus ÖZEN**

**YALOVA, 2024**

**YALOVA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAZILIM TANIMLI AĞLARA YAPILAN DAĞITIK HİZMET ENGELLEME SALDIRILARININ MAKİNE ÖĞRENMESİ ALGORİTMALARIYLA TESPİTİ**

**Mert KAHRAMAN**

**190101028**

**1. Bitirme Tezi Danışmanı: Dr. Öğr. Üyesi Yunus ÖZEN**

1. **Jüri Üyesi :** **Dr. Öğr. Üyesi Güneş HARMAN**
2. **Jüri Üyesi : Doç. Dr. Adem TUNCER**

**Bitirme Tezinin Dönemi: 2023 – 2024 Bahar Yarıyılı**

**İÇİNDEKİLER**

[KISALTMA LİSTESİ iv](#_Toc166948679)

[ŞEKİL LİSTESİ v](#_Toc166948680)

ÖNSÖZ..........................................................................................................................................vi

[ÖZET vi](#_Toc166948681)i

[ABSTRACT vii](#_Toc166948682)i

1. GİRİŞ...........................................................................................................................................1

[1.1 Tez Çalışmasının Amacı 2](#_Toc166948683)

[1.2 Literatür Taraması 2](#_Toc166948684)

1.2.1 Yazılım tanımlı ağların avantajları................................................................................2

1.2.2 Güvenlik tehditleri ve zorlukları...................................................................................3

1.2.3 Makine öğrenmesi ile güvenlik çözümleri....................................................................3

[2. MATERYAL ve METOD 4](#_Toc166948685)

[2.1 Araştırma Materyali 4](#_Toc166948686)

2.1.1 Simülasyon ve deney ortamları.....................................................................................4

2.1.2 Makine öğrenmesi algoritmaları...................................................................................4

2.1.3 Veri setleri.....................................................................................................................4

[2.2 Araştırma Metodu 4](#_Toc166948686)

2.2.1 Simülasyon kurulumu ve deneyler................................................................................4

2.2.2 Makine öğrenmesi uygulamaları...................................................................................5

2.2.3 Veri analizi....................................................................................................................5

2.2.4 Sonuçlar ve değerlendirmeler.......................................................................................5

[2.3 Kullanılan Araçlar 5](#_Toc166948686)

[2.3.1 Mininet nedir?...............................................................................................................5](#_Toc166948687)

[2.3.2 POX nedir?...................................................................................................................6](#_Toc166948687)

2.4 Veri Seti İçin Gerekli Araçlar.................................................................................................6

[2.4.1 WireShark nedir?..........................................................................................................6](#_Toc166948688)

[2.5 Yapay Zeka Modeli İçin Gerekli Kütüphaneler 7](#_Toc166948691)

[2.5.1 Scikit-learn nedir? 7](#_Toc166948692)

[2.5.2 Matplotlib nedir? 8](#_Toc166948693)

2.5.3 Seaborn nedir?...............................................................................................................8

2.6 En Yakın Komşu....................................................................................................................9

[2.7 Rastgele Orman 10](#_Toc166948694)

2.8 XGBoost 11

2.9 Eğim Arttırma Makinesi 11

[3. VERİ SETİNİN ELDE EDİLMESİ VE DÜZENLENMESİ 12](#_Toc166948695)

[3.1 Verilerin Elde Edilmesi 12](#_Toc166948696)

3.2 Veri setlerinin içeriği...............................................................................................................................12

[3.3 Veri Setlerinin Düzenlenmesi 14](#_Toc166948697)

[4. MODELİN OLUŞTURULMASI VE EĞİTİMİ 15](#_Toc166948698)

[4.1 Makine Öğrenmesi Modellerinin Seçilmesi 15](#_Toc166948699)

[4.2 Modelin Oluşturulması 15](#_Toc166948700)

[4.3 Modelin Eğitilmesi 17](#_Toc166948701)

5. BULGULAR VE TARTIŞMA ..................................................................................................19

[6. GENEL SONUÇLAR ve ÖNERİLER 23](#_Toc166948702)

[KAYNAKLAR ...24](#_Toc166948703)

# 

# KISALTMA LİSTESİ

YTA Yazılım Tanımlı Ağlar

DHE Dağıtık Hizmet Engelleme

EYK En Yakın Komşu

ROC Receiver Operating Characteristic

TCP Transmission Control Protocol

UDP User Datagram Protocol

ICMP Internet Control Message Protocol

ARP Address Resolution Protocol

HTTP Hyper Transfer Text Protocol

# ŞEKİL LİSTESİ

Şekil 2.1 POX .................................................................................................................................6

Şekil 2.2 Wireshark .........................................................................................................................6

Şekil 2.3 Scikit-learn .......................................................................................................................7

Şekil 2.4 Matplotlib .........................................................................................................................8

Şekil 2.5 Seaborn ............................................................................................................................8

Şekil 2.6 En Yakın Komşu .............................................................................................................10

Şekil 2.7 Rastgele Orman …..........................................................................................................11

Şekil 3.1 Wireshark ile toplanan verinin sütunları ve açıklamaları ................................................13

Şekil 3.2 Çevrimiçi kaynaklardan elde edilen veri setinin sütunları ve açıklamaları .....................14

Şekil 4.1 Veri setinin yüklenmesi ..................................................................................................15

Şekil 4.2 Verilerin tam sayıya çevrilmesi ......................................................................................16

Şekil 4.3 Karar değişkenlerin belirlenmesi ....................................................................................16

Şekil 4.4 Karar değişkenlerin yorumlanması ................................................................................16

Şekil 4.5 Verilerin düzenlenmesi ...................................................................................................17

Şekil 4.6 Modelin uygulanması ....................................................................................................17

Şekil 4.7 Eğitim verilerinin ayrılması ............................................................................................18

Şekil 5.1 İki veri setine ait karmaşıklık matrisleri ..........................................................................20

Şekil 5.2 İki veri setine ait ROC (Receiver Operating Characteristic) eğrileri ...............................21

Şekil 5.3 İki veri setine ait başarım sonuçları ................................................................................22

**ÖNSÖZ**

Bu tezde, oluşturulan bir YTA (Yazılım Tanımlı Ağlar) üzerindeki bilgisayarlara siber saldırının bir türü olan DHE (Dağıtık Hizmet Engelleme) uygulanmış olup, buradan toplanan veri kümeleri kullanılarak bir takım makine öğrenmesi modelleri eğitilmiştir. Bu modeller kullanılarak YTA üzerindeki normal trafik ve DHE birbirinden ayırt edilmiştir.

Yazdığım tezimde her zaman bana yardımcı olan, takıldığım yerlerde yol gösteren, büyük bir samimiyetle yaklaşan, bilgi ve deneyiminden yararlandığım Dr. Öğr. Üyesi Yunus ÖZEN’e teşekkürlerimi sunarım.

**Mert KAHRAMAN**

# ÖZET

YTA (Yazılım Tanımlı Ağlar) günümüzde bilgisayar ağları kurmanın en popüler yöntemidir. Geleneksel fiziksel yönteme göre birçok artısı vardır, birçok sorunu çözer. Fakat kendisiyle birlikte birtakım sorunları da beraberinde getirir. Bunların en büyüğü ise DHE (Dağıtık Hizmet Engelleme) olarak bildiğimiz paket saldırılarıdır. Bu tez, Mininet adı verilen bir YTA oluşturma aracını, POX adında bir kontrolcü ile beraber kullanarak oluşturulan bir YTA üzerindeki bilgisayarlara, aynı YTA üzerindeki farklı bilgisayarlar tarafından yapılan DHE saldırılarının ve normal ağ trafiğinin Wireshark adı verilen bir araç ile yakalanmasını, ardından yakalanan trafiğin veri seti olarak kullanılıp, EYK (En Yakın Komşu), Rastgele Orman, XGBoost ve Eğim Arttırma Makinesi makine öğrenmesi modelleri ile DHE’lerin normal ağ trafiğinden ayırt edilmesini konu alıyor.

**Anahtar Kelimeler**: Yazılım Tanımlı Ağlar, Dağıtık Hizmet Engelleme, Mininet, POX, Wireshark, En Yakın Komşu, Rastgele Orman

# ABSTRACT

SDN (Software Defined Networks) is currently the most popular method for building computer networks. It has many advantages over the traditional physical method and solves many problems. However, it also brings along some issues. The biggest of these is packet attacks, known as DDoS (Distributed Denial of Service) attacks. This thesis focuses on capturing DDoS attacks and normal network traffic on an SDN created using a tool called Mininet along with a controller named POX. It then addresses the distinction between DDoS attacks and normal network traffic using machine learning models such as kNN (k-Nearest Neighbors ), Random Forest, XGBoost and Gradient Boosting Machine after capturing the traffic using a tool called Wireshark.

**Keywords**: Software Defined Networks, Distributed Denial of Service, Mininet, POX, Wireshark, Nearest Neighbor, Random Forest

**1. GİRİŞ**

Günümüzde, bilgisayar ağları, işletmelerin ve kurumların temel altyapıları arasında yer almaktadır. Ancak, geleneksel ağ yapısı, karmaşık, esnek olmayan ve yönetimi zor bir yapıya sahiptir. İşte tam da bu noktada, YTA (Yazılım Tanımlı Ağlar) devreye giriyor ve ağ dünyasına yeni bir soluk getiriyor.

Yazılım Tanımlı Ağlar, ağ altyapısını geleneksel donanım odaklı yöntemlerden uzaklaştırarak, ağ kaynaklarının merkezi bir yazılım kontrol paneli aracılığıyla yönetilmesine imkan tanır. Bu yaklaşım, birçok avantajı beraberinde getirir. Öncelikle, ağ kaynaklarının merkezi kontrolü sayesinde, ağ yöneticileri daha esnek bir şekilde ağ trafiğini yönetebilir ve hızlı bir şekilde yeni hizmetler ekleyebilirler. Ayrıca, YTA, otomatikleştirilmiş bir yönetim sağlayarak işletme maliyetlerini düşürebilir ve ağın güvenliğini artırabilir.

Ancak, YTA'nın getirdiği bu yeniliklerin yanında, bazı zorluklar da bulunmaktadır. En önemlisi, DHE (Dağıtık Hizmet Engelleme) saldırıları gibi ağ güvenliği tehditleridir. Bu tür saldırılar, ağ trafiğini yoğun bir şekilde hedef alarak ağ kaynaklarını tüketir ve hizmet kesintilerine neden olabilir.

Neyse ki, bu tür tehditlere karşı mücadele etmek için güçlü bir araç elimizde: Makine öğrenmesi. Makine öğrenmesi algoritmaları, ağ trafiğini analiz ederek normal ve anormal davranışları tanımlayabilir ve hızla müdahale edebilir. Bu sayede, Dağıtık Hizmet Engelleme saldırılarını algılayabilir ve engelleyebiliriz.

Sonuç olarak, Yazılım Tanımlı Ağlar, bilgisayar ağlarının geleceğini şekillendiren önemli bir teknolojidir. Ağ yöneticileri, YTA'nın getirdiği avantajlardan yararlanırken, aynı zamanda makine öğrenmesi gibi güçlü araçları kullanarak ağlarını daha güvenli hale getirebilirler. Bu sayede, işletmeler daha verimli, esnek ve güvenli ağ altyapılarına sahip olabilirler.

## **1.1 Tez Çalışmasının Amacı**

Bu tezde, bir Yazılım Tanımlı Ağ üzerindeki bir bilgisayardaki trafiğin, makine öğrenmesi ile incelenip Dağıtık Hizmet Engelleme saldırılarının normal trafikten ayrılması amaçlanmaktadır.

**1.2 Literatür Taraması**

Bilgisayar ağları, modern işletmelerin ve kurumların vazgeçilmez altyapı bileşenleridir. Geleneksel ağ yapılarının karmaşıklığı ve yönetim zorlukları, yeni yaklaşımların gelişmesine neden olmuştur. Bu bağlamda, YTA (Yazılım Tanımlı Ağlar), ağ yönetiminde devrim niteliğinde yenilikler sunmaktadır. Bununla birlikte, YTA'nın güvenlik zorlukları, özellikle DHE (Dağıtık Hizmet Engelleme) saldırıları, önemli bir endişe kaynağıdır. Makine öğrenmesi algoritmaları, bu güvenlik tehditlerine karşı etkili çözümler sunmaktadır. Bu literatür taraması, YTA'nın temel avantajlarını, karşılaşılan güvenlik tehditlerini ve makine öğrenmesinin bu tehditleri nasıl hafifletebileceğini incelemektedir.

**1.2.1 Yazılım Tanımlı Ağların Avantajları**

Yazılım Tanımlı Ağlar, ağ yönetimini yazılım tabanlı bir kontrol düzlemi aracılığıyla merkezileştirir ve bu sayede ağ trafiğinin dinamik ve esnek bir şekilde yönetilmesini sağlar. Literatürde, YTA’nın sağladığı avantajlar şu başlıklar altında toplanmaktadır:

**Esneklik ve hızlı hizmet sunumu:** YTA, ağ kaynaklarının hızlı bir şekilde konfigüre edilmesine ve yeni hizmetlerin hızla devreye alınmasına olanak tanır. Shen vd. (2021), YTA'nın hizmet sunum hızını artırarak işletmelerin rekabet avantajı elde etmelerini sağladığını vurgulamıştır.

**Yönetim ve kontrol:** YTA, ağ kaynaklarının merkezi bir yazılım paneli üzerinden yönetilmesini mümkün kılar. Bu merkezi yapı, ağ yöneticilerinin ağ trafiğini daha verimli bir şekilde izlemesini ve yönetmesini sağlar (Xu vd., 2020).

**Otomasyon ve maliyet tasarrufu:** YTA, ağ yönetim süreçlerini otomatikleştirerek insan müdahalesini azaltır ve bu sayede operasyonel maliyetleri düşürür (Chen vd., 2021).

**1.2.2 Güvenlik Tehditleri ve Zorluklar**

YTA'nın sağladığı avantajların yanı sıra, bu teknoloji yeni güvenlik zorluklarını da beraberinde getirir. Literatürde, YTA'nın karşılaştığı başlıca güvenlik tehditleri şunlardır:

**Dağıtık hizmet engelleme (DHE) saldırıları:** DHE saldırıları, YTA sistemlerini hedef alarak ağ kaynaklarını tüketir ve hizmet kesintilerine neden olabilir (Ahmed vd., 2022). Bu tür saldırılar, özellikle YTA'nın merkezi kontrol yapısını hedef alarak ağın genel işleyişini bozabilir.

**Kontrol düzlemi güvenliği:** YTA'nın merkezi kontrol düzlemi, saldırganlar için kritik bir hedef oluşturur. Bu düzlemin güvenliğinin sağlanması, ağın genel güvenliği açısından hayati öneme sahiptir (Kim vd., 2021).

**1.2.3 Makine Öğrenmesi ile Güvenlik Çözümleri**

Makine öğrenmesi, YTA'nın karşılaştığı güvenlik tehditlerini hafifletmek için güçlü bir araçtır. Literatürde, makine öğrenmesi tabanlı güvenlik çözümlerinin etkinliği şu şekilde belirtilmektedir:

**Anomali tespiti:** Makine öğrenmesi algoritmaları, ağ trafiğini analiz ederek normal ve anormal davranışları tespit edebilir. Bu sayede, potansiyel DHE saldırıları erken aşamada tespit edilip önlenebilir (Zhang vd., 2021).

**Güvenlik otomasyonu:** Makine öğrenmesi, güvenlik politikalarının otomatik olarak uygulanmasını sağlar ve bu sayede insan hatalarını minimize eder. Li vd. (2023), makine öğrenmesi ile otomatik güvenlik politikalarının ağ güvenliğini artırdığını göstermiştir.

**Gerçek zamanlı müdahale:** Makine öğrenmesi, gerçek zamanlı veri analizi yaparak hızlı ve etkili güvenlik müdahaleleri sağlar. Bu durum, saldırıların etkisini azaltmada kritik bir rol oynar (Wang vd., 2022).

# 2. MATERYAL ve METOD

**2.1 Araştırma Materyali**

Bu çalışma, YTA (Yazılım Tanımlı Ağlar) ve makine öğrenmesi kullanılarak ağ güvenliğinin nasıl artırılabileceğini araştırmaktadır. Araştırma materyali olarak aşağıdaki kaynaklar ve araçlar kullanılmıştır:

**2.1.1 Simülasyon ve deney ortamı**: YTA'nın (Yazılım Tanımlı Ağlar) performansını ve güvenliğini değerlendirmek için bir simülasyon ortamı oluşturulmuştur. Bu ortamda, OpenFlow tabanlı bir YTA mimarisi ve POX kontrolcüsü kullanılmıştır. OpenFlow, ağ cihazlarının veri yollarını bir merkezi kontrol düzleminden dinamik olarak yönetmeye olanak tanıyan, yazılım tanımlı ağların (YTA) temel protokolüdür. Mininet gibi ağ emülasyon araçları, ağ topolojilerinin oluşturulması ve test edilmesi için kullanılmıştır.

**2.1.2 Makine öğrenmesi algoritmaları**: Anomali tespiti ve saldırı önleme için çeşitli makine öğrenmesi algoritmaları (örneğin, XGBoost, Rastgele Orman, En Yakın Komşu algoritması) kullanılmıştır. Python programlama dili ve ilgili kütüphaneler (scikit-learn, pandas, numpy vd.) bu algoritmaların uygulanması için tercih edilmiştir.

**2.1.3 Veri setleri**: Ağ trafiği analizi ve anomali tespiti için hem proje kapsamında oluşturulmuş hem de çevrimiçi ortamdan edinilmiş iki veri seti kullanılmıştır. Bu veri setleri, gerçek ağ trafiği kayıtları ve simülasyon ortamında oluşturulan saldırı senaryolarından elde edilen verilerden oluşmaktadır. Bu iki veri setinin detayları makalenin üçüncü bölümünde bulunmaktadır.

**2.2 Araştırma Metodu**

Bu çalışmada kullanılan metodlar, YTA'nın (Yazılım Tanımlı Ağlar) avantajlarını ve güvenlik zorluklarını incelemeyi, makine öğrenmesi algoritmaları ile ağ güvenliğini artırmayı amaçlamaktadır. Araştırma metodu şu adımlardan oluşmaktadır:

**2.2.1 Simülasyon kurulumu ve deneyler**: Mininet ve POX kullanılarak bir YTA (Yazılım Tanımlı Ağlar) simülasyon ortamı kurulmuş ve çeşitli ağ topolojileri oluşturulmuştur. Bu ortamda, normal ve anormal ağ trafiği senaryoları simüle edilmiştir. DHE (Dağıtık Hizmet Engelleme) saldırıları ve diğer güvenlik tehditleri bu simülasyon ortamında test edilmiştir.

**2.2.2 Makine öğrenmesi uygulamaları**: Anomali tespiti için çeşitli makine öğrenmesi algoritmaları uygulanmıştır. Bu algoritmalar, normal ve anormal ağ trafiğini ayırt etmek için eğitilmiş ve test edilmiştir. Performans değerlendirmesi, doğruluk, kesinlik, geri çağırma ve F1 skoru gibi metrikler kullanılarak yapılmıştır.

**2.2.3 Veri analizi**: Elde edilen veriler, makine öğrenmesi algoritmalarının etkinliğini değerlendirmek için analiz edilmiştir. Bu analiz, YTA'nın (Yazılım Tanımlı Ağlar) güvenlik açıklarını belirlemek ve makine öğrenmesi tabanlı çözümlerin bu açıkları nasıl hafifletebileceğini göstermek amacıyla yapılmıştır.

**2.2.4 Sonuçların değerlendirilmesi**: Simülasyon ve deney sonuçları, makine öğrenmesi algoritmalarının YTA (Yazılım Tanımlı Ağlar) güvenliği üzerindeki etkisi ve bu algoritmaların pratik uygulanabilirliği tartışılmıştır.

Bu metodoloji, YTA'nın (Yazılım Tanımlı Ağlar) ağ yönetiminde sağladığı avantajları ve karşılaştığı güvenlik zorluklarını derinlemesine anlamayı ve makine öğrenmesi kullanarak bu zorluklara etkili çözümler geliştirmeyi hedeflemektedir.

## **2.3 Kullanılan Araçlar**

**2.3.1 Mininet nedir?**

Mininet, YTA (Yazılım Tanımlı Ağlar) ve ağ protokollerini test etmek için kullanılan bir ağ emülasyon platformudur. Kullanıcıların sanal ağ topolojileri oluşturmasına, düğümler, anahtarlar ve bağlantılar eklemesine olanak tanır. Mininet, gerçek Linux kernel ve ağ yığınlarını kullanarak hafif sanal ağ ortamları sağlar, bu sayede ağ uygulamaları gerçek dünya koşullarına yakın şekilde test edilebilir. Özellikle ağ araştırmaları ve eğitim amaçlı yaygın olarak kullanılır. Kurulumu ve kullanımı kolaydır, bu da hem öğrenciler hem de profesyoneller için ideal bir araç olmasını sağlar.

**2.3.2 POX nedir?**

A purple squares on a black background

Description automatically generated

Şekil 2.1 POX

POX (Şekil 2.1), YTA (Yazılım Tanımlı Ağlar) için kullanılan bir açık kaynaklı kontrol platformudur. Python dili ile yazılmıştır ve ağ uygulamalarının hızlı geliştirilmesine olanak tanır. POX, ağ yöneticilerinin ağ trafiğini yönetmesini, yönlendirme protokollerini uygulamasını ve ağ güvenliği politikalarını gerçekleştirmesini sağlar. Eğitim ve araştırma projelerinde sıkça kullanılır, çünkü kullanıcı dostu bir arayüz sunar ve genişletilebilir bir mimariye sahiptir. Mininet ile kullanılabilir, bu sayede sanal ağ ortamlarında test ve deneyler yapmak kolaylaşır. POX, YTA uygulamaları geliştirmek isteyenler için güçlü ve esnek bir araçtır.

**2.4** **Veri Seti İçin Gerekli Araçlar**

### 2.4.1 WireShark nedir?



##### Şekil 2.2 Wireshark

Wireshark (Şekil 2.2), ağ protokol analizörü olarak kullanılan ücretsiz ve açık kaynaklı bir yazılımdır. Ağ trafiğini gerçek zamanlı olarak yakalar ve ayrıntılı analizler yapar. Kullanıcılar, ağ paketlerini inceleyebilir, sorunları teşhis edebilir ve ağ performansını izleyebilir. Wireshark, geniş protokol desteği ve kullanıcı dostu arayüzü ile hem eğitim hem de profesyonel amaçlar için idealdir. Ağ güvenliği uzmanları, ağ yöneticileri ve geliştiriciler tarafından yaygın olarak kullanılır. Windows, macOS ve Linux gibi çeşitli işletim sistemlerinde çalışabilir. Wireshark, ağ iletişimlerini derinlemesine anlamak için güçlü bir araçtır.

## **2.5 Yapay Zeka Modeli İçin Gerekli Kütüphaneler**

### 2.5.1 Scikit-learn nedir?



##### Şekil 2.3 Scikit-learn

Scikit-learn (Şekil 2.3), Python programlama dili için makine öğrenmesi kütüphanesidir. Veri madenciliği ve veri analizi görevleri için basit ve verimli araçlar sunar. Sklearn, sınıflandırma, regresyon, kümeleme, boyut indirgeme ve model seçimi gibi yaygın makine öğrenmesi algoritmalarını içerir. NumPy, SciPy ve matplotlib üzerine inşa edilmiştir ve bu kütüphanelerle uyumlu çalışır. Kullanıcı dostu API'si ve kapsamlı dokümantasyonu sayesinde hem yeni başlayanlar hem de deneyimli veri bilimciler tarafından tercih edilir. Sklearn, veri ön işleme, model eğitimi, değerlendirme ve hiper parametre ayarlama süreçlerini kolaylaştırır, bu da makine öğrenmesi projelerinde verimliliği artırır.

### 2.5.2 Matplotlib nedir?



##### Şekil 2.4 Matplotlib

Matplotlib (Şekil 2.4), Python programlama dili için güçlü ve esnek bir veri görselleştirme kütüphanesidir. Grafikler, çizimler ve görseller oluşturmayı kolaylaştırır. Çizgi grafikleri, çubuk grafikleri, histogramlar, saçılım grafikleri ve daha fazlasını destekler. Matplotlib, NumPy ile uyumlu çalışır ve veri bilimciler, mühendisler ve araştırmacılar tarafından yaygın olarak kullanılır. Kullanıcı dostu API'si ile hızlı ve özelleştirilebilir görseller oluşturmayı sağlar. Matplotlib, yayın kalitesinde grafikler üretmek için geniş stil ve format seçenekleri sunar. Veri analizini daha etkili ve anlamlı hale getirmek için ideal bir araçtır.

**2.5.3 Seaborn nedir?**



Şekil 2.5 Seaborn

Seaborn (Şekil 2.5), Python programlama dili için yüksek seviyeli bir veri görselleştirme kütüphanesidir. Matplotlib üzerine inşa edilmiştir ve estetik olarak hoş, bilgilendirici istatistiksel grafikler oluşturmayı kolaylaştırır. Veri setlerinin yapısını ve ilişkilerini anlamak için özellikle kullanışlıdır. Seaborn, kategorik veriler, dağılımlar ve regresyon modelleri için özelleştirilmiş grafik türleri sunar. Ayrıca, stil ve renk paletleri ile görsellerin görünümünü kolayca özelleştirir. Veri analizi ve görselleştirme projelerinde yaygın olarak tercih edilir. Seaborn, görselleştirme sürecini basitleştirir ve analizlerin daha anlamlı ve görsel olarak çekici olmasını sağlar.

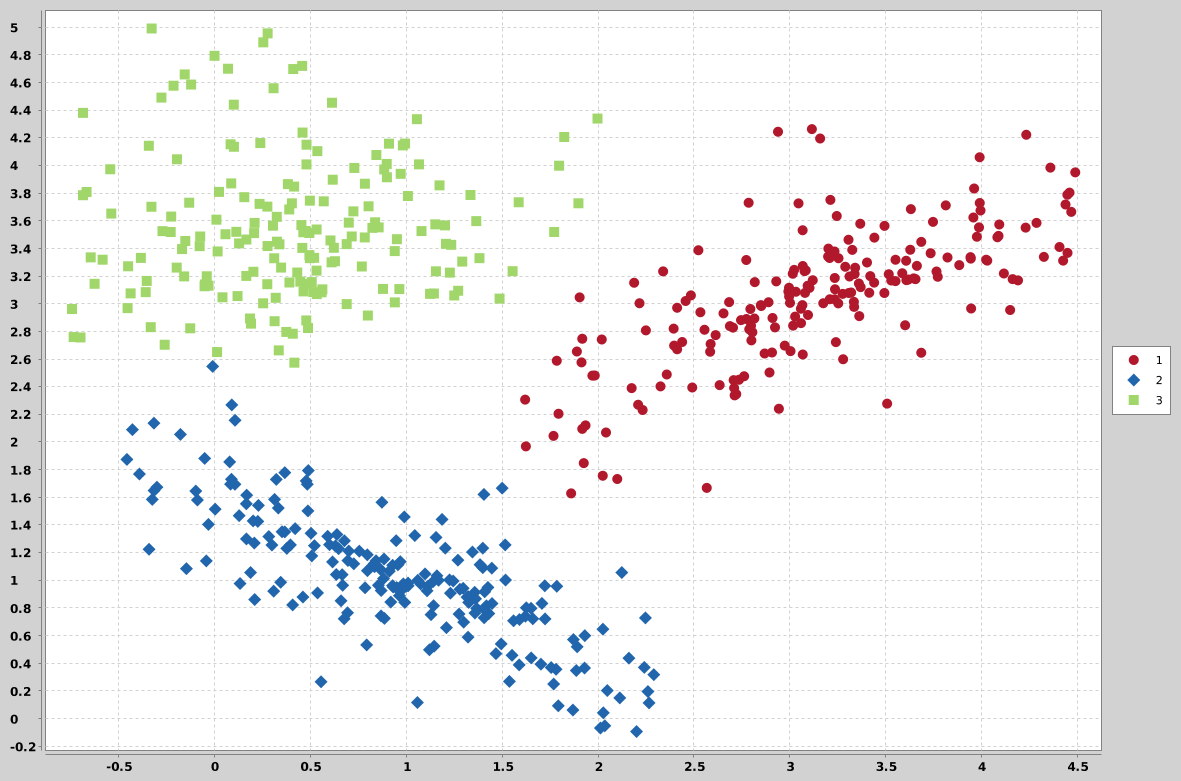
**2.6 En Yakın Komşu**

EYK (En Yakın Komşu) metodu, denetimli öğrenme algoritmaları arasında yer alan, hem sınıflandırma hem de regresyon problemlerinde kullanılan basit ve etkili bir makine öğrenmesi yöntemidir. Temel prensibi, bir veri noktasının sınıfını veya değerini belirlemek için Şekil 2.6’da da göründüğü gibi bu noktanın veri uzayındaki en yakın komşularına bakmaktır.

Öncelikle, veri seti eğitim ve test olarak bölünür. Eğitim verisi, modeli eğitmek için kullanılırken, test verisi modeli değerlendirmek için kullanılır. Veri noktaları arasındaki benzerliği ölçmek için genellikle Öklidyen mesafe kullanılır. Diğer mesafe metrikleri de (Manhattan, Minkowski gibi) kullanılabilir. Kullanılacak komşu sayısı k olarak adlandırılır ve genellikle deneysel olarak belirlenir. Küçük k değerleri modelin aşırı uyum yapmasına, büyük k değerleri ise yetersiz uyuma neden olabilir. Test veri noktası için, eğitim veri setindeki her bir nokta ile mesafe hesaplanır. Hesaplanan mesafelere göre en yakın komşular seçilir. Sınıflandırma problemlerinde, k komşunun sınıfları arasında çoğunluk oylaması yapılır ve en sık görülen sınıf test noktasına atanır. Regresyon problemlerinde, k komşunun ortalama değeri test noktasına atanır.

EYK metodunun bazı avantajları ve dezavantajları vardır. Avantajları arasında basitlik ve kolay uygulanabilirlik yer alır. EYK algoritması, uygulanması kolay ve anlaşılır bir yöntemdir. Karmaşık modelleme gerektirmez. Hem sınıflandırma hem de regresyon problemlerinde kullanılabilir. Eğitim aşamasında sadece veri saklama işlemi yapar, bu da hızlı bir eğitim süreci sağlar. Dezavantajları arasında ise hesaplama maliyeti ve bellek kullanımı bulunur. Büyük veri setlerinde mesafe hesaplamaları zaman alıcı olabilir, bu da yavaş çalışma süresine neden olabilir. Tüm eğitim verilerini bellekte saklamayı gerektirir, bu da büyük veri setlerinde bellek kullanımını artırır. EYK, veri noktalarının ölçeklerinden etkilenir. Bu nedenle, özelliklerin aynı ölçeğe getirilmesi (normalizasyon veya standardizasyon) gereklidir. K değerinin belirlenmesi, veri setine göre değişir ve genellikle deneysel olarak ayarlanmalıdır.

EYK, örüntü tanıma, görüntü işleme, öneri sistemleri ve metin madenciliği gibi birçok alanda kullanılır. Örneğin, kullanıcıların geçmiş beğenileri ve davranışları temel alınarak kişiselleştirilmiş önerilerde bulunmak için öneri sistemlerinde yaygın olarak kullanılır. Sonuç olarak, En Yakın Komşu metodu, basitliği ve esnekliği sayesinde makine öğrenmesi uygulamalarında sıkça tercih edilen bir algoritmadır. Ancak, büyük veri setlerinde hesaplama maliyeti ve bellek kullanımı gibi zorluklarla karşılaşılabileceği unutulmamalıdır. Bu nedenle, EYK algoritması kullanılırken bu dezavantajların dikkate alınması önemlidir.



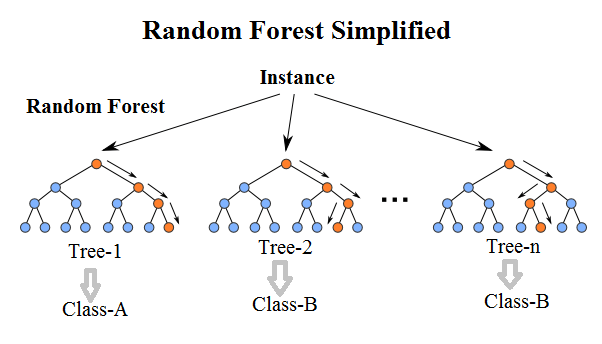
Şekil 2.6 En Yakın Komşu

## **2.7 Rastgele Orman**

Rastgele Orman, makine öğrenmesi alanında hem sınıflandırma hem de regresyon problemleri için kullanılan güçlü bir denetimli öğrenme algoritmasıdır. Temel olarak, Şekil 2.7’deki gibi birden fazla karar ağacının birleşiminden oluşur ve bu ağaçlar eğitim verisinin rastgele alt kümeleri üzerinde eğitilir. Her ağacın dallanma noktalarında özellikler de rastgele seçilir, bu da modelin genelleme yeteneğini artırır ve aşırı uyum riskini azaltır.

Rastgele Orman algoritmasının avantajları arasında yüksek doğruluk, aşırı uyuma karşı direnç, yüksek boyutlu veri setlerinde etkili çalışma ve önemli özelliklerin belirlenmesine yardımcı olma yer alır. Ayrıca, eksik veri ve aykırı değerlere karşı dayanıklıdır. Ancak, modelin açıklanabilirliği azalır ve büyük veri setlerinde bellek ve işlemci kaynaklarını yoğun bir şekilde kullanabilir.

Rastgele Orman, örüntü tanıma, metin sınıflandırma, biyoinformatik ve finans gibi birçok alanda başarıyla uygulanmaktadır. Karmaşık ve yüksek boyutlu veri setlerinde önemli özelliklerin belirlenmesi ve tahmin performansının artırılması için sıkça tercih edilen bir yöntemdir. Sonuç olarak, Rastgele Orman algoritması, esnekliği ve doğruluğu sayesinde makine öğrenmesi projelerinde yaygın olarak kullanılan bir araçtır.



Şekil 2.7 Rastgele Orman

##### 

**2.8 XGBoost**

XGBoost, "Extreme Gradient Boosting" anlamına gelir ve karar ağaçları ile çalışan, denetimli öğrenme algoritmalarına dayalı güçlü bir makine öğrenmesi kütüphanesidir. Gradient boosting yöntemini optimize eden XGBoost, hız, performans ve genel doğruluğu artırmak için tasarlanmıştır. Büyük veri kümeleri üzerinde etkili çalışabilmesi ve paralel işleme yetenekleri sayesinde, yarışmalarda ve endüstriyel uygulamalarda popüler hale gelmiştir. Kayıp fonksiyonlarını minimize ederek modelin hatalarını adım adım azaltır. Overfitting'i önlemek için düzenleme (regularization) teknikleri kullanır ve eksik verilerle başa çıkabilir. Python, R, C++, Java gibi çeşitli programlama dilleriyle uyumludur.

**2.9 Eğim Arttırma Makinesi**

Eğim Arttırma Makinesi (GBM), bir dizi zayıf öğrenici (genellikle karar ağaçları) kullanarak güçlü tahmin modelleri oluşturan bir makine öğrenme tekniğidir. Her bir yeni öğrenici, önceki öğrenicilerin hatalarını düzelterek modeli aşamalı olarak iyileştirir. GBM, regresyon ve sınıflandırma problemlerinde yüksek performans sağlar ve gerçek dünya koşullarına yakın test edilebilme olanağı sunar. Esnek ve özelleştirilebilir yapısı sayesinde çeşitli veri setlerinde etkili sonuçlar verir. Ancak, hesaplama maliyeti yüksektir ve dikkatli hiperparametre ayarları gerektirir.

# 3. [VERİ SETİNİN ELDE EDİLMESİ VE DÜZENLENMESİ](#_Toc138406595)

## **3****.1 Verilerin Elde Edilmesi**

Verilerin elde edilmesinde ilk olarak Ubuntu işletim sisteminde kurulan Mininet üzerinde, şişko ağaç topolojisi ile oluşturulan bilgisayarların birbirlerine Xterm ekranları üzerinden hem sağlıklı hem de DHE (Dağıtık Hizmet Engelleme) trafiklerini oluşturması kullanılmıştır. Bunu sağlayabilmek için Mininet üzerinden “<Bilgisayar ismi> xterm” kodu ile bilgisayarların Xterm ekranı açılıp, daha sonra da hedef bilgisayarın Xterm ekranı üzerinde “iperf -s -p 80” ve “iperf -s -u -p 80” komutlarıyla hem TCP hem de UDP sunucuları açılmıştır. Örnek olması açısından HTTP’ye ait “80” numaralı port kullanılmıştır. Saldırılar için de ICMP (Internet Control Message Protocol), TCP (Transmission Control Protocol) ve UDP (User Datagram Protocol) için sırasıyla “ping (hedef adres)”, “iperf -c <hedef IP adresi> -t <istenen saniye> -b <bant genişliği> -l <paket büyüklüğü> -p 80” ve “iperf -c <hedef IP adresi> -u -t 300 -b <bant genişliği> -l <paket büyüklüğü> -p 80” komutları kullanıldı. Bu trafik sağlanırken POX ile oluşturulmuş L2 Learning Kontrolcüsü kullanılmıştır. Oluşturan bu trafik Wireshark aracı ile yakalanmış ve CSV formatına dönüştürülmüştür. İkinci veri seti ise data.mendeley.com adresinden toplanmıştır [1].

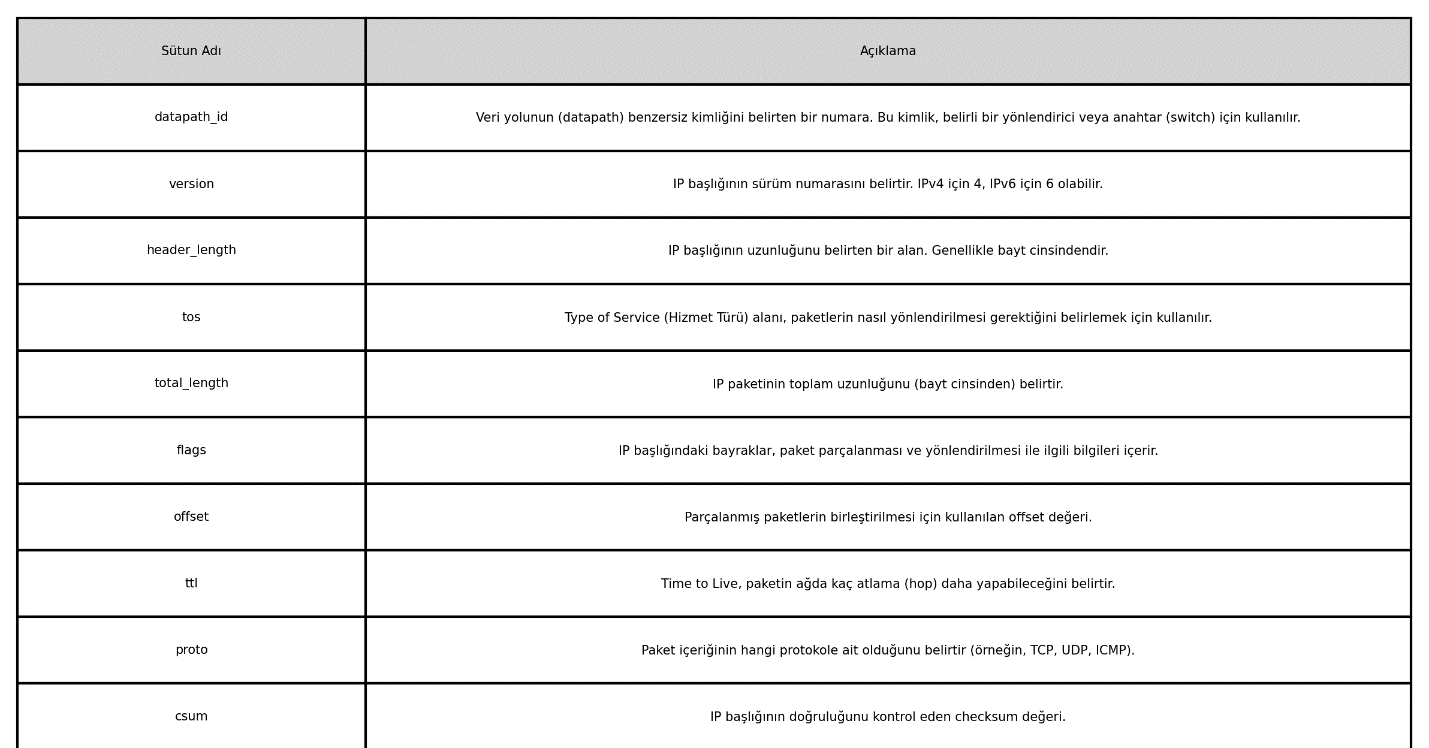
###### 3.2 Veri setlerinin içeriği

İlk veri seti Mininet aracında oluşturulan bir şişko ağaç topolojisinde gerçekleştirilen TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol) paketlerine sahip, hem DHE (Dağıtık Hizmet Engelleme) hem de sağlıklı paketlerden oluşan bir trafiği (Şekil 3.1) içerir. Bu trafik Wireshark aracı ile CSV formatında toplanmıştır. Yaklaşık 55.000 veriden oluşur. Hedef sütunu olarak daha sonra bu veri setine “DDoS” sütunu eklenmiştir. Bu sütunda paketin DHE olup olmadığı gösterilir.

A screenshot of a computer

Description automatically generated Şekil 3.1 Wireshark ile toplanan verinin sütunları ve açıklamaları

İkincisi veri seti (Şekil 3.2) ise çevrimiçi kaynaklardan elde edilmiştir. Tercih edilmesindeki en büyük sebep sütun sayısı olarak Wireshark’a göre çok daha fazla detay bulundurması, dolayısı ile daha fazla bilgi içermesidir. Aynı zamanda veri sayısı da yaklaşık 420.000 kadardır. Hedef sütun olarak “label” sütunu oluşturulmuştur. Bu sütunda paketin DHE (Dağıtık Hizmet Engelleme) olup olmadığı ve hangi çeşitte trafik olduğu birlikte (DHE – UDP, DHE değil – TCP vs.) tutulur.



A list of text on a white background

Description automatically generated

Şekil 3.2 Çevrimiçi kaynaklardan elde edilen veri setinin sütunları ve açıklamaları

## **3****.3 Veri Setlerinin Düzenlenmesi**

Her iki veri setinde de kararların paketlerin kaynağına ve hedefine göre belirlenmemesi için bu sütunlar düzenlemede kaldırıldı. Eğer bu değerler kaldırılmasaydı, DHE (Dağıtık Hizmet Engelleme) paketleri sağlayan kaynakların adresi, farklı bir YTA (Yazılım Tanımlı Ağlar) ortamındaki bir kaynakla aynı olduğu durumda oluşturulan projenin yanlış kararlar verme ihtimali yükselirdi. Ayrıca ikinci veri setinde aynı görevi gören “src\_ip, dst\_ip, src\_host, dst\_host” sütunları da kaldırıldı.

Bununla beraber, ilk veri setinde “No.” sütunu da kaldırıldı. Bunun sebebi ise gerçek hayatta paket numarasının DHE saldırılarıyla ile ilgisinin bulunmaması, fakat simülasyon ortamında makine öğrenmesi algoritmasının bunu önemli görebilme ihtimali bulundurmasıdır.

Son olarak her iki veri setine hedef sütunlar eklendi, bu hedef sütunlar üzerinden etiketlendirmeler yapıldı.

**4. MODELİN OLUŞTURULMASI VE EĞİTİMİ**

## **4.1 Makine Öğrenmesi Modellerinin Seçilmesi**

**Temel seçimler**

İlk olarak her iki veri seti için de en temel modellerden olan EYK (En Yakın Komşu) modeli ile deney yapıldı.

**Bilinçli seçimler**

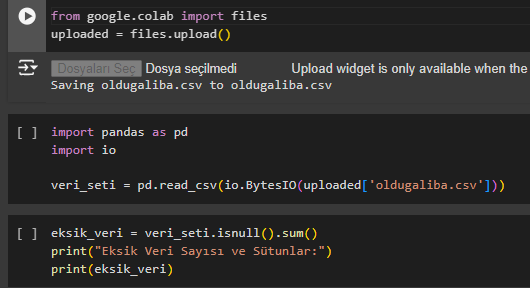
Trafik ve DHE (Dağıtık Hizmet Engelleme) saldırılarının tanımı için en çok işe yarayacak modeller incelendi [8] ve Rastgele Orman modeli üzerinde karar kılındı. İki veri setine de uygulandı.

Daha sonrasında ise, birçok ağaç metodunun birleşmesinden oluştuğu için gayet iyi sonuçlar veren XGBoost metodu tercih edildi. İyi verim alındıktan sonra, yakınlığından dolayı Eğim Arttırma Makinesi modeli de kullanıldı.

## **4.2 Modelin Oluşturulması**

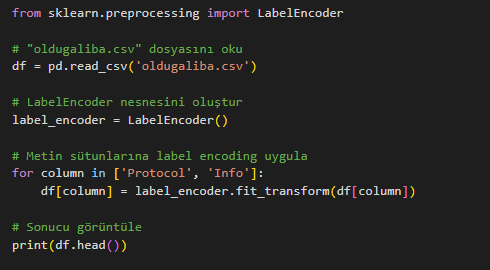
**Elde edilen veri seti üzerine:**

Google Colab üzerinden proje oluşturulmaya başlandı. Gerekli kütüphaneler eklendi. Veriler, “oldugaliba.csv” adı altında Colab ortamına Şekil 4.1’deki kodlarla yüklendi.



Şekil 4.1 Veri setinin yüklenmesi

Veriler üzerinde düzenleme (Şekil 4.3, Şekil 4.4, Şekil 4.5) yapıldı. Makine öğrenmesine el verişli olabilmesi için verilerdeki değerlerin boş olup olmadığı kontrol edildi. Ardından tam sayı formatına Şekil 4.2’deki şekilde getirildi.



Şekil 4.2 Verilerin tam sayıya çevrilmesi

A computer screen with colorful text

Description automatically generated

Şekil 4.3 Karar değişkenlerin belirlenmesi

A computer screen with text

Description automatically generated

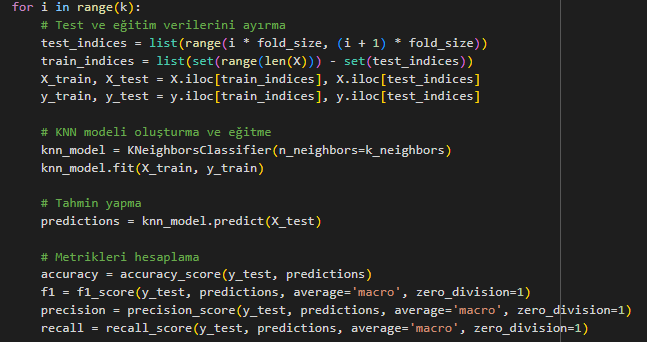
Şekil 4.4 Karar değişkenlerinin yorumlanması

A computer screen with text

Description automatically generated

Şekil 4.5 Verilerin düzenlenmesi

Eğitim ve test verileri belirlendi. Ardından belirlenen modeller uygulandı (Şekil 4.6). Sonuç metrikleri ve grafikler yazıldı.



Şekil 4.6 Modelin uygulanması

**Hazır veriler üzerine:**

Hazır veriler de tam olarak aynı işlemlerden geçti. Fakat ikili sınıflandırma yerine verinin doğasına uygun olacak şekilde 6 çeşit sınıfa ayrıldı. Bu sınıflar şu şekildedir:

0 = ICMP DHE (Internet Control Message Protocol Dağıtık Hizmet Engelleme)

1 = TCP DHE (Transmission Control Protocol DHE)

2 = UDP DHE (User Datagram Protocol DHE)

3 = ICMP NORMAL

4 = TCP NORMAL

5 = UDP NORMAL

## **4.3 Modelin Eğitilmesi**

Eğitim ve test verilerinin belirlenmesi için K Kat Çaprazlama yöntemi tercih edildi. “K” sayısı olarak “10” seçildi (Şekil 4.7). Bu sayının seçilmesindeki en büyük sebep hem yeteri kadar test verisi sunması, hem de yeteri kadar rastgelelik sağlaması oldu.

A computer screen with text and numbers

Description automatically generated

Şekil 4.7 Eğitim verilerinin ayrılması

**5. BULGULAR VE TARTIŞMA**

Yazılım Tanımlı Ağ çalışmaları Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz (12 CPUs), ~2.6GHz işlemcisine, NVIDIA GeForce GTX 1650 Ti ekran kartına ve 16384MB RAM’e sahip bir Windows 11 işletim sistemine sahip sistemde, Ubuntu 22.04 sanal makinesi altında yapılmıştır.

Makine öğrenmesi çalışmaları ise Google Colab’in kendi alt yapısı üzerinde yürütülmüştür.

Sonuçlar ve grafikleri (Şekil 5.1 ve Şekil 5.2) aşağıdaki gibidir:

A blue squares with white text

Description automatically generatedA screenshot of a graph

Description automatically generated

A blue squares with white text

Description automatically generatedA graph of a number of blue squares

Description automatically generated with medium confidence

A blue squares with white text

Description automatically generatedA graph of a number of blue squares

Description automatically generated with medium confidence

A blue squares with white text

Description automatically generatedA graph of a number of blue squares

Description automatically generated with medium confidence

Şekil 5.1 İki veri setine ait karmaşıklık matrisleri (En Yakın Komşu, XGBoost, Eğim Arttırma Makinesi, Rastgele Orman)

Normal şartlar altında bu kadar yüksek değerler aşırı öğrenme denilen bir durumun göstergesi sayılır. Bu durum, algoritmanın çok kesin kararlar vermesi ve bunun sonucunda yeni değerler karşısında karar verememe durumudur. Fakat ağ trafiğindeki değerler çok kesin olduğu için bu yorumu her zaman yapmak pek de doğru olmaz. Bir modelin aşırı öğrenmeye eğilimli olup olmadığını belirlemek için, modelin eğitim setindeki performansı ile doğrulama setindeki performansı arasındaki farka dikkat etmek gerekir. Eğer eğitim setindeki performans doğrulama setindeki performanstan belirgin şekilde daha iyiyse, model muhtemelen aşırı öğrenmeye eğilimlidir.

Ayrıca, modelin karmaşıklığı da bir etkendir. Çok karmaşık modeller genellikle eğitim verilerine daha fazla uyarlanır ve dolayısıyla aşırı öğrenme eğilimindedir. Bu nedenle, modelin karmaşıklığını kontrol etmek ve aşırı uyum riskini azaltmak için uygun düzeyde düzenleme ve doğrulama teknikleri kullanmak önemlidir.

A graph with a line and a point

Description automatically generated with medium confidence A graph of a function

Description automatically generated with medium confidence

Şekil 5.2 İki veri setine ait Rastgele Orman ROC (Receiver Operating Characteristic) eğrileri

Her iki veri setinin (önce elde edilen, sonra hazır alınan) kullanıldığı algoritmaların test etmede kullanılan değerlendirme ölçeklerine göre başarım sonuçları Şekil 5.3’te sunulmaktadır.

A white table with black text

Description automatically generatedA white table with black text

Description automatically generated

A white rectangular box with black text

Description automatically generatedA screenshot of a computer

Description automatically generated

A white rectangular box with black text

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Şekil 5.3 İki veri setine ait başarım sonuçları (En Yakın Komşu, XGBoost, Eğim Arttırma Makinesi, Rastgele Orman)

# 6. GENEL SONUÇLAR ve ÖNERİLER

Bu makalede kullanılan veri setlerinde en iyi sonuç veren modeller dikkate alınmıştır. Çok sayıda model deneyimleri ve bu modeller üzerinden tercih yapılması bu makaleyi öne çıkarmıştır. Bu modeller ilk veri setinde Rastgele Orman, ikincisinde ise Rastgele Orman, XGBoost ve Eğim Arttırma Makinesi olmuştur. Her iki proje için de değerlendirme metriklerinin birbirine bu kadar yakın, hatta aynı çıkması bize projelerin her dört metrik için de gayet sağlıklı çalıştığını göstermektedir. Fakat bu metriklerin bu kadar yüksek çıkması da bir o kadar kuşku uyandırıcıdır. Buna rağmen dikkate alınmalıdır ki, ağ trafiği değişkenlerin çok az olduğu, kalıpların net olduğu bir veri konusudur. Bu sebeple makine öğrenmesi algoritmaların sonuçlarının çok yüksek olması kaçınılmazdır.

Bu şekilde, projelerde DHE (Dağıtık Hizmet Engelleme) saldırıları sağlıklı trafikten neredeyse kesin olarak ayırt edilebilmiştir. Bu sayede birçok kurum ve kuruluş, birey ve her türlü YTA (Yazılım Tanımlı Ağlar) kullanıcısı bu projeleri kullanıp kendi sistemlerini DHE saldırılarından koruyabilir.

Teknolojinin kaçınılmaz ilerleyişiyle birlikte her zaman daha farklı tehditler ve daha farklı metotlar oluşacaktır. Gelecek çalışmalarda bu makaleden temel alınarak ilerisine gidilip, Yazılım Tanımlı Ağlar üzerinde daha farklı tehditler, daha farklı metotlar ile çözümlenebilir. Bu çalışmaya başkalaşım geçirtilip birçok ağ temelli alanda kullanılabilir.

# KAYNAKLAR

Ahmed, I., Hassan, A., & Asif, M. (2022). Mitigating DDoS attacks in SDN using machine learning techniques. Journal of Network and Computer Applications, 184, 102914.

Chen, X., Li, Y., & Zhu, M. (2021). Cost-effective network management using SDN. IEEE Transactions on Network and Service Management, 18(2), 199-213.

Kim, S., Park, J., & Lee, K. (2021). Security challenges in software-defined networking: A survey. IEEE Communications Surveys & Tutorials, 23(1), 123-144.

Li, Y., Wang, J., & Zhang, P. (2023). Automated security policy enforcement in SDN using machine learning. Computer Networks, 202, 108578.

Shen, Y., Liu, H., & Zhou, X. (2021). Enhancing service agility in SDN. Future Generation Computer Systems, 114, 345-357.

Wang, K., Wu, L., & Liu, J. (2022). Real-time DDoS detection and mitigation in SDN with machine learning. Journal of Parallel and Distributed Computing, 154, 97-105.

Xu, D., Sun, Y., & Zhang, T. (2020). Centralized management in SDN: Benefits and challenges. IEEE Network, 34(3), 85-91.

Zhang, L., Chen, W., & Liu, Y. (2021). Anomaly detection in SDN using deep learning. IEEE Transactions on Information Forensics and Security, 16, 1630-1642.

[1] https://data.mendeley.com/datasets/hkjbp67rsc/1