T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

SINIFLANDIRMA VE REGRESYON AĞAÇLARI İLE RASTGELE ORMAN ALGORİTMASI KULLANARAK BOTNET TESPİTİ: VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ ÖRNEĞİ

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Duygu KORKMAZ
DANIŞMAN: Prof. Dr. H. Eray ÇELİK

VAN-2018

T.C.
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
İSTATİSTİK ANABİLİM DALI

SINIFLANDIRMA VE REGRESYON AĞAÇLARI İLE RASTGELE ORMAN ALGORİTMASI KULLANARAK BOTNET TESPİTİ: VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ ÖRNEĞİ

YÜKSEK LİSANS TEZİ

HAZIRLAYAN: Duygu KORKMAZ

VAN-2018

KABUL VE ONAY SAYFASI

İstatistik Anabilim Dalı'nda Prof. Dr. H. Eray ÇELİK danışmanlığında, Duygu KORKMAZ tarafından sunulan "**Sınıflandırma ve Regresyon Ağaçları ile Rastgele Orman Algoritması Kullanarak Botnet Tespiti: Van Yüzüncü Yıl Üniversitesi Örneği** " isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince 18/07/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile başarılı bulunmuş ve Yüksek Lisans Tezi olarak kabul edilmiştir.

Başkan	:Prof. Dr. H. Eray ÇELİK	İmza:
Üye	:Doç. Dr. Hamit MİRTAGİOĞLU	İmza:
Üye	:Dr. Öğr. Üyesi Murat CANAYAZ	İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..15../..08.../2018 tarih ve 2018/89-I sayılı kararı ile onaylanmıştır.

Prof. Dr. Suat ŞENSOY

Enstitü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Duygu KORKMAZ

ÖZET

SINIFLANDIRMA VE REGRESYON AĞAÇLARI İLE RASTGELE ORMAN ALGORİTMASI KULLANARAK BOTNET TESPİTİ: VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ ÖRNEĞİ

KORKMAZ, Duygu
Yüksek Lisans Tezi, İstatistik Anabilim Dalı
Tez Danışmanı: Prof. Dr. H. Eray ÇELİK
Temmuz 2018, 81 sayfa

Bir botnet, kötü amaçlı yazılım (bot) kodunun bulaşmış olduğu, bir veya daha fazla makineden oluşan bir ağdır. Botnet, Botmaster denilen kişiler tarafından yönetilir ve DDos, Spam, Kimlik Hırsızlığı gibi faaliyetler için kullanılmaktadır. Bu çalışmanın amacı, bir Network üzerinde Botnet bulaşmış Network cihazı olup olmadığını, Makine Öğrenmesi Algoritmalarından, Sınıflandırma Ağaçları ve Regresyon Ağacı (CART) ile Rastgele Orman teknikleriyle tespit etmek ve sınıflandırmaktır. Modellerin sınıflandırma performansları bazı performans ölçütleri bakımından ölçülmüş ve kıyaslanmıştır. Ele alınan değişkenler, ekleyip çıkarılarak doğruluk ve bazı performans ölçütleri üzerindeki değişimler Sınıflandırma Ağaçları Yöntemi ve Rastgele Orman Algoritması Yöntemi ile incelenmiştir ve bir ağda Botnet tespiti yapmak için önemli olan değişkenler önerilmiştir.

Anahtar kelimeler: Düğüm, Gini, Hata matrisi, Phyton, Siber güvenlik

ABSTRACT

BOTNET DETECTION BY USING CLASSIFICATION AND REGRESSION TREES WITH RANDOM FOREST ALGORITHMS: EXAMPLE OF VAN YUZUNCU YIL UNIVERSITY

KORKMAZ, Duygu
M.Sc. Thesis, Department of Statistics
Supervisor : Prof. Dr. H. Eray ÇELİK
Temmuz 2018, 81 pages

A botnet is a network of malware code infected by one or more machines. Botnet is managed by Botmaster and is used for activities such as Ddos, Spam, and Identity Theft. The purpose of this study is to identify and classify whether or not there is a network device infected by a botnet on a network using Classification and Regression Trees and Random Forest techniques from Machine Learning Algorithms. Classification Performance of Models are measured and compared in terms of some performance measures. Variations on accuracy and some performance measures were examined on Classification and Regression Trees and Random Forest techniques, by adding and subtracting variables and this study suggests variables that are important for Botnet Detection in a Network.

Keywords: Confusion matrix, Cyber Security, Gini, Node, Phyton

ÖN SÖZ

Bu tez çalışmasında verdiği desteklerden dolayı her türlü ilgi ve yardımlarını esirgemeyen danışmanım Sayın Prof. Dr. H. Eray ÇELİK'e, Yüzüncü Yıl Üniversitesi Bilgisayar Bilimleri Araştırma ve Uygulama Merkezi'nde çalışan Bilgisayar Mühendisi Mesut KAPAR'a, Arş.Gör. Fatih ULUDAĞ'a ve diğer hocalarıma değerli katkılarından dolayı, ayrıca çalışmalarımda bana her zaman destek olan aileme teşekkürlerimi sunarım.

2018

Duygu KORKMAZ

İÇİNDEKİLER

Sayfa

ÇİZELGELER

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLER LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
$\boldsymbol{X}$	Açıklayıcı değişkenlerden oluşan ölçüm vektörü
$\boldsymbol{j}$	Sınıf
$\boldsymbol{C}$	Sınıfların bir kümesi
$\boldsymbol{d(x)}$	Sınıflandırma kuralı fonksiyonu
$\boldsymbol{A_J}$	X' in alt kümesi
$\boldsymbol{N}$	Gözlem sayısı
$\boldsymbol{L}$	Eğitim seti
$\boldsymbol{R^*(d)}$	$d(x)$ için yanlış sınıflandırma oranı
$\boldsymbol{R(d)}$	Yeniden yerine koyma kestirimi
$\boldsymbol{t}$	Düğüm
$\boldsymbol{i(t)}$	Safsızlık ölçüsü
$\boldsymbol{\theta}$	t'nin safsızlık ölçüsünün bir fonksiyonu
$\boldsymbol{\delta}$	Aday bölünme
$\boldsymbol{\varphi}$	Sorular kümesi
$\boldsymbol{T}$	İkili ağaç
$\boldsymbol{\tilde{T}}$	Ulaşılan terminal düğüm
$\boldsymbol{R^*(T)}$	Tüm ağacın genel yanlış sınıflandırma oranı için yeniden yerine koyma tahmini
$\boldsymbol{T'}$	Budanmış ağaç

Kısaltmalar	Açıklama
ABD	United States of America
AID	Automatic Interaction Detection
C&C	Commend and Control
CART	Classification and Regression Trees
COM	Communication
DDOS	Distributed Denial of Service
DNS	Domain Name System
DNS	Domain Name System
E-POSTA	Electronic post
FCBF	Fast Correlation Based Filter
FIN	Fınısh
FNR	False Negative Ratio
FP	False Positive Ratio
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection Systems
IP	Internet Protocol
IRC	Internet Relay Chat
JPEG	Joint Photographic Experts Group
MARS	Multivariate Adaptive Regression Splines
ML	Machine Learning
NPV	Negative Predictive Value
OOB	Out Of Bag
PHP	Hypertext Preprocessor
PPV	Positive Predictive Value
P2P	Peer To Peer
QUEST	Quick, Unbiased, Efficient Statistical Tree
ROC	Receiver Operating Characteristics
SLIQ	Supervised Learning in Quest
SPC	Specivity
SYN	Synchronization

TCP	Transmission Control Protocol
TPR	True Positive Rate
UDP	User Datagram Protocol
WWW	World Wide Web

EKLER DİZİNİ

Sayfa

1. GİRİŞ

Teknolojik gelişmeler yaşam kalitesini yükseltmekle beraber, bilişim alanındaki gelişmeler; yeni tip suçların ortaya çıkmasına sebep olmuştur. Bu yeni tip suçlara siber suç adı verilmektedir (Shinde ve Tittel, 2002).

Siber, bilgisayar ve bilgisayar ağlarını ilgilendiren veya içeren kavram ya da varlıkları tanımlamak için kullanılan bir kelimedir. Siber suçlar temel anlamıyla, bilgisayar sistemleri üzerinden yasadışı bir şekilde bilgi alınması ve bu sistemlere izinsiz erişim sağlanmasıdır (Shinde ve Tittel, 2002).

Sosyal Medya ve Mobil Kullanıcı İstatistikleri'ne göre dünya nüfusunun % 53'ü aktif bir şekilde internet kullanmaktadır (Anonim, 2018a). Her geçen yıl bu oran artmakta ve buna bağlı olarak siber suçların da oranı ve bu suçlara maruz kalan kullanıcılar artmaktadır.

Türkiye, dünya genelinde en çok siber saldırı alan ülkelerden biri konumundadır. Avrupada en fazla siber saldırının yaşandığı ülke Türkiye iken, dünya genelinde ABD ve Brezilya'dan sonra üçüncü sırada yer almaktadır. Bu saldırıların çoğu bilgisayar, akıllı telefon veya diğer ağ cihazlarından botnet yoluyla gerçekleştirilmektedir. Fortinet şirketi tarafından yapılan araştırmanın sonuçlarına göre botnet, exploit kit ve fidye yazılım tehdidi altındaki dünya ülkeleri arasında Türkiye beşinci sırada yer almaktadır (Anonim, 2018b).

Botnet saldırıları, spam, Dağıtık Hizmet Reddi Saldırıları (DDos), kimlik hırsızlığı ve kimlik avı gibi bir çok internet saldırısının temel platformudur (Gu ve ark., 2008).

Bir botnet, kötü amaçlı yazılım (bot) kodunun bulaşmış olduğu, bir veya daha fazla makineden oluşan bir ağdır. Botnetler, botmaster denilen kişiler tarafından yönetilir ve DDos, Spam, Phishing, Kimlik Hırsızlığı gibi faaliyetler için kullanılırlar (Gu ve ark., 2008).

Botnet saldırıları, yasal iletişim kanalları aracılığıyla gerçekleştirilir. Internet Relay Chat (IRC) uygulamaları 2000'lerin başlarına kadar geleneksel botnetler arasında en yaygın iletişim yolu olarak kullanılmaktadır. Ağ içerisindeki makinelere botnet yazılımları bulaştıktan sonra, botnetler bir IRC sunucusuyla bağlantı kurar ve bu kötü

amaçlı yazılım Botmaster tarafından yerleşik IRC komutları ve kontrol (C&C) kanallarını kullanır. Botmasterın amacı, bulaştırılan zararlı yazılımın (botların) sistemde sürekliliğini sağlamaktır. Botlar güncelleme almak için belirli aralıkla bot yöneticisine bağlanır ve botlar merkezi olarak yönetildikleri için, bot yöneticisinin kapatılması durumunda devre dışı kalmaktadırlar.

Ağ trafiğinin izlenebilmesiyle birlikte botnet tespiti konusunda daha fazla araştırma yapılmıştır. 2000'li yıllardan itibaren Peer to Peer (p2p) protokolünün gelişmesi ile, bot yazılımları istemci ve sunucu arasında hareket eder ve merkezi bir nokta olmadığı için çalışması sekteye uğramaz. P2P botnetleri yönetilebilirlik açısından ve sınırlandırmalarının olmasından ötürü HTTP tabanlı C &C protokolüyle uygulanmaya başlanmıştır. HTTP tabanlı botnetlerin, güvenlik duvarı ve benzeri sistemlere takılmaması için botnet paketleri şifrelenerek bu güvenlik duvarlarının geçilmesi sağlanmaktadır (Zhao ve ark., 2013).

Bu çalışmanın amacı, bir ağ üzerinde Botnet bulaşmış network cihazı olup olmadığını, sanal ortamda belli bir zaman diliminde botnet bulaştırılmış ve bulaştırılmamış ağ trafiği akışı birlikte izlenerek toplanan veriler üzerinde, Makine Öğrenmesi Algoritmalarından Sınıflandırma ve Regresyon Ağacı (CART) ile Rastgele Orman teknikleri kullanarak botnetli ve normal akışı tespit etmek, bu iki akışı ayırt eden bir sınıflandırma modeli oluşturmak ve modele ait performans ölçülerini hesaplamaktır.

1.1. Botnet Aktiviteleri

Botnetler, binlerce bottan (büyük ölçekli botnetlere) oluşan büyük bir botnet ya da yüzlerce veya daha az sayıda bottan (küçük ölçekli botnetler) oluşan küçük bir botnete kadar geniş bir yapıya sahip olabilir. Karmaşık yapılarına ve amaçlarına bakılmaksızın, botnetlerin temel özelliği bilgisayar ağlarında kötü niyetli faaliyetlere hizmet etmesi için yaratılmış olmalarıdır (Eslahi ve ark., 2012).

Botnetler sadece bilgisayar ağları için tehlikeli bir tehdit olmaktan öte, diğer tehdit ve saldırı türlerinde de yer almaktadır. Bu saldırıların bazıları, Ddos, Spam yapma, kişisel bilgi hırzılığı, yasadışı barındırma satış veya kira servisleri ve tıklama sahtekarlığıdır (Choi ve Lee, 2012).

Ddos: Bu saldırı, çok sayıda UDP (Bağlantı kurulum işlemlerini,akış kontrolü ve tekrar iletim işlemlerini yapmayarak veri iletim süresini en aza indirmek için kullanılan pakettir) paketi, ICMP (Hata oluşumlarında geribildirim sağlamak veya paket başka bir yoldan gideceği zaman geribildirim sağlamak için kullanılan bir protokoldür) isteği veya TCP (Gelişmiş bilgisayar ağlarında paket anahtarlamalı bilgisayar iletişiminde kayıpsız veri gönderimi sağlayabilmek için kullanılan bir protokoldür) senkronizasyon taşması göndererek gerçekleştirilen hizmet reddi veya DOS saldırısının dağıtılmış şeklidir.

Bot yöneticileri, virüslü bilgisayarlardaki botlara belirli bir komut göndererek amaçlarını gerçekleştirebilirler (Eslahi ve ark., 2012).

Spam yapma: İstenmeyen mesajlara aynı içeriğe sahip olan Spam, E-posta, Messenger, Blog veya haber gruplarındaki yorumlar gibi farklı ortamlara yüksek hacimli olarak gönderilir. Spam etkinliklerinin yaklaşık % 85'i botnet'ler tarafından üretilmektedir. Bu nedenle spamlar, botnetlerin ana platformu olarak görülebilir. Saniyede ortalama üç spam e-postası ile bot göndermek mümündür (Choi ve Lee, 2012; Eslahi ve ark., 2012;).

Kişisel bilgilerin hırsızlığı: Kişisel bilgilerin çalınması her zaman en önemli internet tehditlerinden biri olarak görülür ve Zeus botnet tek başına yaklaşık 3,5 milyon bilgisayara bulaşmış ve kişisel bilgileri çalmaya teşebbüs etmiştir. Anahtar kaydediciler gibi kişisel parolalar ve çevrimiçi bankacılık gibi finansal verilere ulaşmak için botlar yayılır (Bilge ve ark., 2009; Eslahi ve ark., 2012).

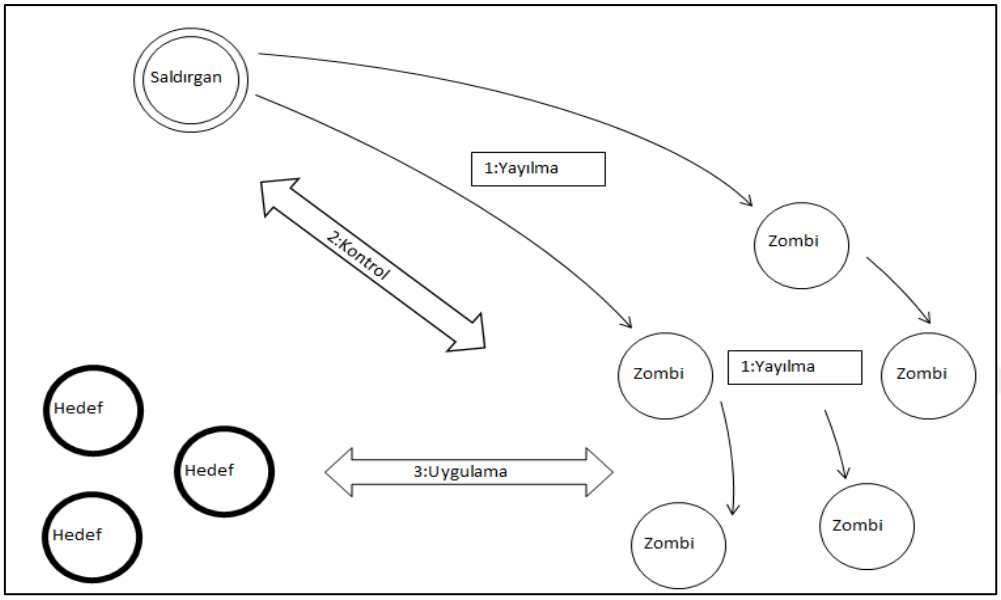
Yasadışı barındırma, satış veya kira servisleri: Internet'e büyük bir depolama alanı ve yüksek bant genişliğine sahip bir bilgisayar veya sunucu, dosya paylaşımı ve yasadışı barındırmalar için bir botmasterın kontrolü ele geçirmesi için hedef olabilir. Botnet programları ve barındırma hizmetleri, kötü amaçlı olarak satılabilir veya kiralanabilir. Bu hizmetlerin amaçlarından biri; müşteriler ile kanun uygulayıcıları arasında daha fazla engel ve boşluk bırakmaktır (Eslahi ve ark., 2012).

Tıklama sahtekârlığı: Botnetler ve diğer internet tehditleri arasındaki temel farklardan biri, botnetin tıklama sahtekârlığıyla para kazanmak için kullanılmasıdır. Botmasterlar, web sitelerine yapılan her ziyaret için veya reklama yapılan her bir tıklama için küçük bir miktar para ödeyen açık web sitelerinde botlarını kullanarak çok para kazanabilirler. Saldırılara ilave olarak, botnetler virüs, truva atı, solucan gibi farklı

türdeki bilgisayar tehditlerini yaymak için kullanılabilmektedir. Yani botnetlerin yalnızca bir tehdit değil, aynı zamanda dağıtım için bir platform olduğu söylenebilir (Juels ve ark.,2007; Eslahi ve ark., 2012).

1.2. Botnet Yaşam Döngüsü

Botnet'ler varlıkları boyunca benzer bir adımı izlerler. Bu adımlar bir yaşam döngüsü olarak karakterize edilebilir. Şekil 1'de bir botnetin genel yaşam döngüsü gösterilmektedir (Hachem ve ark., 2011).



Şekil 1.1. Botnet Yaşam Döngüsü (Hachem ve ark., 2011).

1.2.1. Yayılma ve enjeksiyon

Botnetler, yeni zombi makinelere diğer malwarelere benzer yöntemlerle bulaşmaktadır. Bu yöntemler aşağıdaki başlıklarda anlatılmıştır (Hachem ve ark., 2011).

Kötü amaçlı e-posta yoluyla: Makineleri zombi makinesi haline getirmek için kullanılan sosyal mühendislik yöntemlerinden biri kötü amaçlı e-posta dağıtımı yapmaktır. Bu yöntem sosyal medya üzerinden de uygulanmaktadır. Bu yöntemde, botnet adminler, kullanıcıların, kötü amaçlı yazılımı çalıştırmaları için kullanıcıları

kandırmaktadır. Böylelikle kötü amaçlı yazılımın çalıştığı makine zombi makine haline dönüşmüş olmaktadır (Gómez ve ark., 2013).

Yazılımsal güvenlik açığı ile: Botnetlerin bir diğer bulaşma yolu da, kullanıcının makinesinde bulunan yazılımsal güvenlik açıklığının uzaktan istismar edilmesidir. Bu tür güvenlik açıklarını başarıyla istismar eden saldırgan, bir sistemin kontrolünü tamamen ele geçirebilir. Kontrolü tamamen ele geçirdikten sonra, yeni programlar yükleyebilir, verileri görüntüleyebilir, değiştirelibilir, silebilir veya tam haklara sahip yeni kullanıcı hesapları oluşturabilir (Hachem ve ark., 2011).

Bazı anlık mesajlaşma uygulamalarında yayılan bilgisayar solucanları ile : Uzaktan kontrol edilen makinelerden oluşan geniş bir botnet ağı oluşturmak için kullanılmaktadır. Bu tür solucanlar, kötü niyetli çalıştırılabilir bir programı, bir 'jpeg' benzeri uzantılara sahip dosyalara veya arkadaşlarından geliyor gibi görünen bir mesaja ekleyip bulaştırabilmektedir. Bu yöntemler, botnetlerin yayılması için, yazılımsal güvenlik açıklarından daha fazla kullanılmaktadır (Hachem ve ark., 2011).

Diğer botnetleri kullanarak: Botnet bulaşmış bir makine, başka makinelere de ulaşıp botnet yazılımını bulaştırabilir. Böylelikle geniş bir botnet ağı kurulabilir (Hachem ve ark., 2011).

1.2.2. Komut ve kontrol (C&C)

C&C kanalı, komutları göndermek için, farklı modeller, topolojiler ve çeşitli uygulamaları (Örn, HTTP, P2P, IRC vb.) kullanabilir (Feily ve ark., 2009).

C&C modeli: C&C modellerini kategorize etmek için birçok yaklaşım vardır. Bu yaklaşımları iki farklı model olarak sınıflandırılmaktadır. Bunlar merkezi ve dağıtılmış modellerdir (Ramachandran ve ark., 2006; Hachem ve ark., 2011).

Merkezi C% C modeli: İstemciler arasındaki mesajları iletmek veya yayınlamak için merkezi bir nokta kullanılır. Bu merkezi modelde botmaster, C&C sunucusu olacak bir ana makine seçer. Ana makineye bağlanan diğer makineler botmasterın kontrolü altına girer. Bu ana makineler de IRC ve HTTP gibi servisleri çalıştırır (Ramachandran ve ark., 2006; Hachem ve ark., 2011).

Dağıtılmış C&C modeli: Bu modelde iletişim sistemi merkezi olarak yapılandırılmış bir veya bir kaç sunucuya bağlı değildir. İletişim sistemi, merkezi olarak sunuculara bağlı olmadığından, merkezi modele kıyasla, dağıtılmış modelin keşfedilmesi ve yok edilmesi çok daha zordur (Ramachandran ve ark., 2006; Hachem ve ark., 2011).

İç haberleşme protokolü: Botlar ve botmasterlar kendi haberleşme kanallarını kurmak için mevcut veya daha önce kullanılmış teknikleri, protokolleri kullanmaktadırlar. Bundan dolayı protokolü anlamak, botnetleri kaldırmak ve imha etmek konusunda strateji oluşturmak için büyük öneme sahiptir (Micro, 2006).

İletişim karakteristiklerini çözebilmek için ilk olarak botnette kullanılmış olan olası yazılım araçlarını bilmek gerekir. İkinci olarak da , bu iletişim uygulamalararında kullanılan protokollerin anlaşılması, botnet araştırmacısının tarfiği izlemesine ve çözmesine yardımcı olur (Micro, 2006).

Botnet iç iletişim protokollerine örnek olarak, IRC, HTTP, ICMP,P2P örnek olarak verilebilir (Micro, 2006).

İletişim: İletişimi başlatmak için 2 yol vardır. Bunlar itme ve çekme metodlarıdır.

İtme metodu: Bu yöntemde botmaster botlara komut gönderir ve gönderilen komut, açık bir iletişim kanalında saklanmaktadır. Böylelikle yeni eklenen botlar da bu komutları alabilmektedirler (Jakobsson ve Ramzan, 2008).

Çekme metodu: Bu yöntemde bot, yeni komutları, botmaster tarafından belirlenen belirli periyotlara göre ana makineden kontrol eder ve uygulamaktadır (Jakobsson ve Ramzan, 2008).

1.2.3. Botnet Uygulama Alanları:

Botnetler hem meşru hem de meşru olmayan amaçlarla kullanılmaktadırlar. Botnetlerin meşru kullanımlarından biri IRC yöntemine yardımcı olmaktadır.

Botnetlerin yasadışı veya yıkıcı olarak kullanma yönleri aşağıdaki şekilde kategorize edilmiştir;

-DDOS Saldırıları,

-Spam, kötü amaçlı yazılım ve reklam yayılması,

-Casusluk,

-Kötü amaçlı uygulamaları ve faaliyetleri barındırma (Bailey ve ark., 2009)

1.2.4. Direnç Teknikleri

Yeni teknolojiler geliştikçe, suçlular, kötü amaçlı yazılımların esnekliklerini ve dirençlerini arttırabilecek yeni teknikler aramaktadırlar. Bu teknikler botnet yaşam döngüsünün üç ayrı aşamasında da bulunmaktadır (Jakobsson ve Ramzan, 2008; Hachem ve ark., 2011).

1.3. Botnet Karekteristikleri

Irc: Irc, istemci/sunucu modunu kullanan uygulama katmanı protokolüdür. Sunucu tarafında, birden fazla istemcinin aynı anda birbirine bağlanması ve birbiriyle iletişim kurmasını sağlayan bir sohbet odası kanalıdır. Irc tam özellikli bir protokoldür ve elde edilmesi kolay olduğundan, bilgisayar korsanlarının botnet kurması için ilk tercihtir (Zilong ve ark., 2010).

Http: Http protokolü, bir sunucu üzerinden dijital verileri iletmek için kullanılan uygulama katmanı protokolüdür. Sunucu, bir bilgisayar ağı üzerinden, istemcilerin bağlanabilmesi için yapılandırılmıştır (Del ve Klemets, 2000).

Http tabanlı botnetler, en yeni botnet çeşitlerinden biridir. Http botnetler web trafiğini taklit eder ve mevcut ağ güvenlik sistemlerini aşmak için bu protokolü kullanırlar. Ayrıca, Http protokolü, normal uygulamalar ve web sitelerinde yaygın olarak kullanılır, böylece Http botnetlerin algılanması dikkate değer bir sorun haline gelmiştir (Eslahi ve ark., 2013).

Dns: Çoğu internet kullanıcısı, IP adresi denilen sayısal adresleri hatırlamakta zorlanır. Buna çözüm olarak oluşturulan DNS protokolü, alan adını IP adreslerine çevirerek, göndermektedir. Böylelikle alan adından kolaylıkla erişilmek istenen yere erişilir (Whyte ve ark., 2005).

Botlar, dağıtık bir şekilde aynı anda DNS sorgusu gönderdiklerinde, DNS hizmeti veren sunucunun hizmeti durur ve DNS hizmeti veremez hale gelir (Choi ve ark., 2007).

P2P: Peer to peer (P2P) iki veya daha fazla istemci arasında veri paylaşmak için kullanılan bir ağ protokolüdür. HTTP veya DNS protokolünün aksine sunucuya ihtiyaç duymaz. Bu nedenle dağıtık bir yapıdadır ve P2P ağlarda botnet tespiti de diğer protokollerindekine göre daha zordur (Noh ve ark., 2009).

1.4. Botnet Tespit Yöntemleri

Botnet tespiti için çeşitli yöntemler geliştirilmiştir. Önceleri tespit yöntemlerinden biri olan balküpü yöntemiyle ağ davranışlarını izleyerek botnet algılama yaygın olsa da günümüzde ağ davranışlarını izlemeyerek botnet algılama yöntemleri daha yaygındır. Botmaster'lar son güvenlik güncellemelerine karşın yeni yöntemler geliştirmektedirler. Bu kısımda botnet tespit yöntemleri anlatılacaktır (Barazi ve ark., 2014).

1.4.1. Balküpleri

Balküpleri belirli bir problemi çözmez. Bunun yerine saldırılar konusunda bilgi toplayıp, hedef şaşırma işlemlerinin yapılmasını sağlamaktadır.

Networke yapılan saldırıların farkına varmak, saldırı metodlarını belirlemek ve yeni geliştirilen saldırı çeşitlerinden önceden bilgi sahibi olma amacıyla dizayn edilmiş tuzak sistemlerdir (Spitzner, 2003).

Bu sistemlerin üzerinde çalışan yazılımlar, gerçek sistemleri taklit ederek, saldırganların gerçek sistemlere saldırdığı izlenimi verir. Böylelikle gerçek sistemlerle iletişim kurulmadan önce tuzak sistemlerle iletişim kurmuş olur (Spitzner, 2003).

1.4.2. Saldırı tespit sistemleri

IDS(Saldırı Tespit Sistemleri), kötü amaçlı yazılımlar ve kural ihlallerini tespit edip bildiren yazılım uygulamasıdır. Bu sistemler, imza tabanlı botnet algılaması ve

anormalliğe dayalı botnet algılaması olarak iki gruba ayrılmaktadır (Feily ve ark., 2009).

İmza tabanlı botnet algılama :İmza listelerinde bulunan bayt veya paket dizilerini, ağ trafiğinin izlenmesi sonucunda, gelen paketler içerisinde eşleştirip tespit eden algılama yöntemidir. Belirli bir ağ trafiğinde bir eşleşme olduğunda, sistem yöneticileri uyarılır ve sistem tarafından önceden tanımlanmış bir eylem gerçekleşir (Alauthman, 2016). Örneğin, açık kaynak kodlu olan Snort yazılımı, ağ trafiğini izinsiz girişler için izleyen bir saldırı tespit sistemidir. Çoğu saldırı tespit sistemine benzer şekilde snort da trafiği izleyen ve şüpheli kabul edilen trafiği kaydetmek için bir dizi kural ve imza ile yapılandırılmıştır. Sadece bilinen botnetlerin tespiti için imza tabanlı teknikler kullanıldığından dolayı bu yaklaşım bilinmeyen botlar için faydalı değildir (Binkley ve Singh, 2006; Feily ve ark., 2009).

Anormallik tabanlı botnet algılama: Anormallik tespit terimi, ağ trafiğinde beklenmeyen iletişim şekillerini bulma problemini ifade etmektedir. Bu uygun olmayan şekiller genellikle çeşitli uygulamalarda anormallikler, istisnalar veya uyumsuz gözlemlerden oluşan belirgin veya saptanması zor olan ağ davranışlarıdır (Karim ve ark., 2014). Anormallik tespit sistemlerinin avantajı, sıfır gün saldırıları olarak bilinen yeni saldırıları tespit edebilme kabiliyetidir (Alauthman, 2016). Bununla birlikte, anormal tabanlı tespit sistem teknikleri, yüksek derecede yanlış botnet uyarısı vermektedir (Li ve ark., 2009).

Host tabanlı botnet algılama: Bu algılama yönteminde, ağ trafiği izlenmeden, bilgisayar sisteminin iç yapısı izlenmektedir. Botnet algılama yazılımları botnet olup olmadığını tespit etmeye çalışır. Fakat kötü niyetli olmayan bazı bilgisayar yazılımları da anormal bir trafik oluşturur ve botnet tespit eden yazılımlar bu trafiği de botnet trafiği sanıp yanlış kararlar verebilmektedir (Zeidanloo ve ark., 2010).

Ağ tabanlı botnet algılama: Bu algılama yönteminde, ağ trafiği izlenerek botnet tespiti yapılır. Bu yöntemin birkaç avantajı vardır; Birincisi botnet yaşam döngüsünün herhangi bir aşamasında tespit etmek mümkün olması ikincisi ise bu yöntemin daha ucuz olmasıdır (Saad ve ark., 2011). Bu yöntem aktif ve pasif izleme olarak ikiye ayrılmaktadır;

Aktif izleme: Bu izleme yönteminde, sunucuya veya uygulamalara paketler gönderilerek bu uygulama veya sunucuların bu paketlere olan tepkileri ölçülmektedir (Zeidanloo ve ark., 2010).

Pasif izleme: Bu izleme yönteminde ağ üzerinde kurulu olan cihaz veya yazılımlar kullanılır. Bu cihaz veya yazılımlar ağdaki etkinliği arttırmazlar. Sadece üzerinden geçen ağ paketlerini analiz ederek botnet olup olmadığına karar vermek için kullanılırlar (Zeidanloo ve ark., 2010). Ağ üzerinde çok sayıda paket vardır. Bunların çoğu botnet tespiti için önemli olmadığından, alakasız paketlerin filtrelenmesi gerekmektedir. Trafik azaltma algoritmaları, botnet algılama sisteminin daha iyi çalışmasını sağlamaktadır (Kempanna ve Kannan, 2015).

2. KAYNAK BİLDİRİŞLERİ

Saad ve ark. (2011), çalışmalarında, botnetlerin DDOS saldırıları ile spam oluşturduğunu ve tıklama sahtekarlığı ile gizli bilgilerin çalınmasından hareketle, ağ trafiği davranışlarını kullanarak botnetleri tanımlamak ve saptamak için yeni bir yaklaşım öne önermişlerdir. En yeni ve en zorlu botnet tiplerini temsil eden P2P botları üzerinde yaygın kullanılan 5 makine öğrenme tekniğinin erken saptama yeteneğini incelemişlerdir. Kullandıkları veri üzerinde botnetlerin Commend and Control (C&C) fazındayken ve ataklar başlamadan yalnızca trafik davranışlarını kullanarak etkili bir şekilde tespit edildiğini göstermişlerdir. Ancak kullanılan teknikler çözüm ihtiyaçlarını tek seferde ayrı ayrı gidermede başarısız olmuştur. Çalışma; C&C fazında botları tespit etmeye yoğunlaşmıştır. Ağ trafik davranışları üzerinde; Karar Destek Makineları, Yapay Sinir Ağları, K En Yakın Komşu Sınıflandırıcısı, Gaus Tabanlı Sınıflandırıcı ve Naive Bayes sınıflandırıcısı kullanılmıştır. 10 kat çapraz geçerlikle Weka ve Javada uygulama gerçekleştirilmiş ve 5 yöntemin kıyasında Karar Destek Makineları, Yapay Sinir Ağları ve K En Yakın Komşu Sınıflandırıcısı ön plana çıkmıştır. Yapay Sinir Ağları Yöntemi toplam sınıflandırma hata oranında en aza sahip olduğu belirlenmiştir.

Gu ve ark. (2008), botnet C&C Protokolünden ayrı olarak, Botminer olarak adlandırılan hem C&C hem de P2P'i hedefleyen bir botnet tespit yöntemi sunmuştur. Botminerin çalışma prensibine göre botlar benzer kalıpları kullanıp, benzer zararlar veren yazılımlardır. Gerçek bir ağ üzerinde önerilen bu yaklaşım incelenerek yüksek oranda tespit gücüne sahip olduğu gösterilmiştir. Botminer, Botnet C&C Trafik tespiti için veri madenciliği teknikleri uygulayan en yeni yaklaşımdır. Botsniffein bir gelişmiş halidir ve benzer iletişim trafiği ve benzer zararlı trafiği kümelemeye dayanır. Kullanılan mimaride iki düzlemde kümeleme yapılmıştır; A düzlemi ve C düzlemi. A ve B düzlemleri daha sonra çapraz düzlem korelasyonun da sergilenerek her iki düzleminde bot olarak algıladığı kısımlar gösterilmiştir. A ve C Düzlem kümelemelerinin ayrı ayrı kısıtlandığı durumlar söz konusudur. C düzleminde C&C akışındaki benzerliklerin, tesadüfi paketlerin (gürültü) azalması, A düzleminde tespit

edilemeyen hareketler (Gmail kullanılarak gönderilen mailler) örnek olarak gösterilebilir.

Singh ve ark. (2014), gerçek bir zaman diliminde yüksek boyutlu veri için trafik yönetiminin zor olduğunu, bunun için imza tabanlı tespit yönteminin uygulanabileceğini; öte yandan anormal temelli olan Peer to Peer botnetlerde olduğu gibi atakların daha düşük hızda ve çoklu Makinelardan türeyen botnetlerin tespitinde imza temelli yaklaşımın uygulayamayacağını belirtmişlerdir ve buna istinaden çalışmalarında Hadoop ve Hive gibi açık kaynaklardan gerçek zamanlı bir tespit sistemi inşa etmişlerdir. Uygulamada Peer to Peer Botnet ataklarını tespit etmek için makine öğrenmesi teknikleri kullanılmıştır. Bu makine öğrenmesi tekniklerinden Rastgele Orman ve Karar ağaçları yöntemleri denenmiştir. Gerekli olan özelliklerin seçimi (veri ayıklaması) yapıldıktan sonra eğitim seti % 90, test seti % 10 olarak belirlenip; 10 ağacın kullanıldığı Rastgele Orman algoritması ile % 99.7 doğru sınıflandırma ile başarılı bir performans sergilendiği ROC eğrisi ile gösterilmiştir. 84030 örnekli bu veride yalnış sınıflandırma oranı çok düşük çıkmıştır. Karar ağacı ise bir topluluk yöntemi olan Rastgele Ormandan daha kötü sonuç vermiştir.

Livadas ve ark. (2006), IRC Temelli botnetlere ait C&C trafiğini tanımlamak için makine öğrenmesi tekniklerinden sınıflandırma yöntemlerini kullanmışlardır. Bu görev için 2 senaryo oluşturmuşlardır. Birinci senaryodaki amaç, IRC ve IRC olmayan trafiği ayırt etmek olurken; ikinci senaryodaki amaç, botnet ve normal IRC trafiğini ayırt etmek olmuştur. Birinci senaryo için ; J48, Naive Bayes ve Bayesian Network sınıflandırıcılarının doğru sınıflandırma oranlarına bakılmıştır ve eğitim setinin boyutu için bir sınıflandırma hassasiyeti tanımlanmıştır. İkinci senaryoda kullanılan IRC trafiğinin Botnet olup olmamasınına göre sınıflandırılmasında ise bozukluklar olmuştur ve böylece bunun zorlu bir görev olduğu belirlenmiştir. Ağ akışını , TCP ve IP paket okuyucularına dayanan özellikleri kullanarak karakterize etmişlerdir. Daha sonra veri seti gereksiz özelliklerden ayıklanmıştır. Ağ akışını IRC veya IRC değil diye sınıflandırmak için J48, Naive Bayes ve Bayesian Networks'un model performansları kıyaslanmıştır. Naive Bayes düşük yanlış negatif orana sahipken en yüksek yanlış pozitif oranına sahiptir. Bayesian Network tekniği düşük yanlış pozitif oranı, ama daha yüksek yanlış negatif oranına sahiptir. J48'e ait, FNR ve FPR ise dengede olduğu belirlenmiştir. Sadece Naive Bayes sınıflandırıcısı düşük

FNR'ye ulaşmada başarılı olmuştur. Naive Bayes "doğru sınıflandırma" yaptığından J48 ve Bayesian Network sınıflandırıcılarına tercih edilebilir olduğu belirlenmiştir. İkinci sahne için Botnet ve gerçek IRC trafiğini ayırt etme görevi üstlenilmiştir. Burada ise en başarılı sınıflandırıcı yanlış negatif ve yanlış pozitif oranlarının dengede olduğu Bayesian Network sınıflandırıcısı olarak bulunmuştur. Burada dikkat çekilen nokta ise sınıflandırma problemlerinin çözümünde, veriyi etiketleme işleminin doğru yapılması gerektiğidir.

Moore ve ark. (2005), çalışmalarında ağ trafiğini sınıflandırmak için denetimli Makine öğrenmesi tekniği olan Naive Bayes kestirimcisini kullanmışlardır. Veri seti bir internet sitesinden toplanmıştır. Biyoloji ile alakalı birçok çalışmanın olduğu bu site Genome Kampus olarak isimlendirilmiştir. Bir sınıflandırma modeli oluşturmak için Naive Bayes tekniği kullanılmış ve iki güçlü düzeltme (Kernel yoğunluk tahmin teorisi ve özellik seçimi yapmak için kullanılan Fast Correlation-Based Filter(FCBF)) kombinasyonu ile doğruluk % 65 den % 95 lere çıkabildiği tespit edilmiştir. Diğer geleneksel yöntemler % 50-70 arası başarı gösterirken elde edilen bu başarı yüksek bir başarı olarak nitelendirilmiştir. FCBF işleminden sonra uygulanan Naive Bayesin doğru sınıflandırma oranı % 94.29 olarak belirlenmiştir.

Kalaivani ve ark. (2016), yoğun internet trafik akışında, botnetli trafiğin normal trafikten ayırt edilmesinin zorlu bir görev olduğunu ve mevcut tespit yöntemlerinin bu zorlu görevin üstesinden gelmede yetersiz kaldığını belirtmişlerdir. Bu sorunun, ağ trafiğini, botnetli trafik ve botnetli olmayan trafik olarak ikili sınıflandırma modeli ile çözülebileceğini belirtmişlerdir. Ağ trafiğini ikili (binary) olarak kategorize etmek için güçlü makine öğrenmesi tekniklerden; Destek Vektör Makineları, Naive Bayes, Karar Ağaçları ve Sinir Ağları'nı kullanmışlardır. 500 botnetli ve 500 normal akış toplanmıştır. Sınıflandırma modeli inşasında özellik seçiminin (future selection) hayati rol oynadığını belirtmişlerdir ve botnet tespiti için gerekli/önemli olduğunu düşündükleri özellikleri seçerek analize dahil etmişlerdir. Sonuç olarak, Destek Vektör Makineleri tekniğinin % 99.8 doğru sınıflandırma oranı ile Naive Bayes, Karar Ağaçları ve Sinir Ağları yöntemlerinin önüne geçtiğini belirtmişlerdir. Ağ trafiğinde makine öğrenmesi tekniklerinin botnet tespit etmede oldukça başarılı olduğu sonucuna ulaşmışlardır.

Kumar ve Kaur (2014), ağ trafiğini, pasif izleme yöntemi ile izlemişlerdir. İstatistiksel yöntemler ve Makine öğrenmesi temelli tekniklerin ortak kullanılması ile iyi bir trafik sınıflandırıcısı modeli geliştirilebileceğini belirtmişlerdir ve ağ trafiğini otomatik yakalayan bir sistemle saat bazında yakalanan dosyaları kaydetmişlerdir. Saldırı verileri ile normal kullanıcı verileri toplanıp birleştirilip, ağ trafiğinin düşük seviyeli özelliklerine dayanan otomatik sınıflandırma sistemi geliştirmişlerdir. Makine öğrenmesi tekniklerinden Naive Bayes Sınıflandırıcısı kullanılmıştır ve bu tekniğe ilişkin doğru sınıflandırma oranı ise % 66.111 bulunmuştur.

Zhao ve ark. (2012), zararlı ve zararsız paketler içeren iki akış trafiğini birleştirip 1.672.575 ağ akışına sahip bir veri seti elde ettikten sonra kısa bir süre içinde akışın karakteristiğini inceleyerek gerçek bir zaman diliminde botnet saldırılarını tespit etmeye olanak sağlayan bir yöntem sunmuşlardır. Bayesian Network ve Karar Ağaçları teknikleri kullanılan bu çalışmada, azaltılmış hata budama algoritması kullanılarak bir karar ağacı seçilmiştir. Bu algoritma ile gürültülü verilere karşı tespit doğruluğu artmıştır ve sınıflandırma karmaşasını da azaltarak ağaç boyutunun küçülmesini sağlanmıştır. 10 katlı çapraz geçerlik kullanılarak azaltılmış hata budama algoritması ile % 99 doğruluk elde edilirken, Bayesian Network sınıflandırıcısı % 90 başarı sağlamıştır.

Gu ve ark. (2008), diğer yazılımların aksine botnetlerin bir C&C kanalına sahip olması neticesinde tespitinin zor hale geldiğini belirtmişlerdir. Buna istinaden çalışmalarındaki amaç, önceden bir imza bilgisi veya C&C sunucu adresi olmadan bir bölgesel alanda network temelli anormal tespit yöntemi kullanılarak botnet tespiti sağlamaktır. Bu yaklaşımın hem C&C sunucusunu hem de alıcıya bulaşan botları tanımlamak için kullanılabilir olduğunu vurgulamışlardır. IRC temelli C&C itme tipi olarak değerlendirilirken; HTPP tabanlı C&C çekme tipi olarak değerlendirilmiştir. Çalışmada, önceden C&C bilgisi veya imza gerektirmeyen botların tespiti için çok sayıda bota gerek duymayan, yanlış pozitif ve yanlış negatif oranlarının düşük olduğu hem IRC hem de HTTP tabanlı C&C leri bağımsız bir şekilde tanımlamak için anormal tabanlı tespit algoritmaları önerilmiştir ve son olarak üniversite ağı üzerinden önerdikleri anormal tespit algoritmasına dayalı ve açık kaynaklı Snort (bir saldırı tespit ve saldırı engelleme sistemidir) için birkaç eklenti olarak kullanılan Botsniffer kullanarak elde edilen sonuçları incelemişlerdir. Botsniffer'ın, ağ trafiğini incelemek

için birkaç korelasyon ve benzerlik analizi algoritması kullanan bir teknik olmasıyla birlikte aynı botnet içindeki botların faaliyetlerinde çok güçlü benzerlikler ortaya çıkaracağı varsayımına dayandığını vurgulamışlardır. Sonuç olarak yüksek doğruluk ve düşük yanlış pozitif oranı elde edilmiştir.

Barthakur ve ark. (2013), Makine öğrenmesi tekniklerinden sınıflandırma algoritmasını P2P botnetlerinin tespiti için C&C trafik akışı üzerinde kullanmışlardır. Kullanılan üç yöntem; Karar ağacı, Bayesnet ve Doğrusal Destek Vektör Makineleridir. Sınıflandırma sonucu elde edilen doğruluk oranlarından Bayesnet % 99.68 ile ön plana çıkarken eğri altında kalan alan % 99.7 olarak hesaplanmıştır. Böylece bu çalışmadaki en iyi sınıflandırıcı olduğunu göstermişlerdir.

Kirubavathi ve Anitha (2018), botnet tehdidinin teknolojik gelişmelerle yayılması sonucunda cep telefonlarına bulaşan zararlı yazılımlara yönelik tespit yöntemleri mevcutken, android botnet uygulamalarındaki botnetin tespit edilememesi problemini ele almışlardır. Çalışmalarında Google Android market gibi açık kaynaklı siteler kullanarak veri toplamışlardır. Sonrasında toplanan veriler sınıflandırma tekniklerinden; Naive Bayes, Destek Vektör Makineleri ve REPTree teknikleri ile analiz edilmiştir. En yüksek doğru sınıflandırma oranı, Destek Vektör Makineleri Yönteminde gözlenmiştir. Çalışmada makine öğrenme teknikleri kullanılarak, android uygulamaları iyi huylu veya botnet olarak sınıflandırmaya dayalı model oluşturmanın yüksek doğrulukla mümkün olduğunu belirtmişlerdir.

Hoang ve Nguyen (2018), çalışmalarında T1, T2 ve T3 olmak üzere üç eğitim veri seti ve aynı veri kümesinden TEST adı verilen bir test veri seti seçmişlerdir. T1 seti benign (iyi huylu) adını alan 10,000 kayıt ve kötü niyetli olarak adlandırılan 10.000 kayıt içerirken, T2 seti benign'den 5000 kayıt ve kötü niyetli olarak adlandırılan 15.000 kayıt içermektedir, son olarak T3 kümesi 15.000 iyi 5000 kötü kayıt içermektedir. Test seti ise 10000 iyi 10000 kötü kayıt içermektedir. Yöntem olarak makine öğrenmesi tekniklerinden; K En Yakın Komşuluk, C4.5, Rastgele Orman ve Naive Bayes kullanılmıştır. Sonuç olarak üç veri setinde de en iyi sınıflandırma performansına sahip olan yöntemin Rastgele Orman yöntemi olduğunu belirtmişlerdir.

3. MATERYAL ve YÖNTEM

3.1. Materyal

Botnetli ve normal trafik akışını elde etmek için sanal ortam kurulmuştur. sanal ortamın kurulması için Microsoft Windows 10 üzerinde kurulu gelen Hyper-v sanallaştırma ortamı kullanılmıştır. Network üzerinden paket toplamak için ve paket üzerinden filtrelemeler yapabilmek için wireshark adlı network sniffing aracı kullanılmıştır. Veri hazırlama aşamasında literatürde önemli olduğu düşünülen özellikler PHP yazılım dili yardımı ile MYSQL veri tabanına aktarılarak veriler burada düzenlenip (sayısal hale getirilip) Excell formatına çevrilmiştir. İstatistiksel analizler Phyton'da gerçekleştirilmiştir.

3.2. Yöntem

3.2.1. Makine öğrenmesi

Örüntü tanıma (pattern recognition) alanı, bilgisayar algoritmalarının kullanımı yoluyla verinin altında yatan desenin otomatik olarak keşfedilmesi ve bu desene dayanarak verileri farklı kategorilere sınıflandırılması gibi eylemlerde bulunmak için kullanılmasıyla ilgilidir. Makine öğrenmesi algoritmaları ise bu desenleri keşfetmek için kullanılan algoritmalardır (Bishop, 2012).

Makine öğrenmesi, verinin altında yatan şeklin tesadüfi olmadığı ve gözlemlenen veriyi açıklayan bir yapının olduğu varsayımına dayanır (Kelleher ve ark., 2015).

Makine öğrenmesi, örnek veri veya geçmiş deneyimleri kullanarak bilgisayar programlamaya verilen isimdir. Makine öğrenmesinde temel görev bir örneklemden çıkarım yapmak olduğundan, matematiksel modellerin oluşturmak için istatistik teorisini kullanır (Alpaydın, 2014).

Makine öğrenim sistemleri, bilgisayarlara açık bir şekilde programlanmadan öğrenebilme olanağı sağlar. Algoritmalar, daha sonra yeni verilere maruz kaldıkça güncellenen ve değişen modeller oluşturur. Makine öğreniminin tarihi 1950'lerde başlamıştır, ancak son gelişmeler, olasılıksal araçların 1990'larda geliştirilmesi ile büyük veri setlerine erişim ve son yıllarda grafik işlemede kullanılan güçlü çipler arasındaki yakınsamanın bir sonucudur (Anonim, 2018c).

Makine öğrenimi, özünde, bilgiyi eyleme dönüştürülebilen algoritmalarla ilgilidir. Bu gerçek, makine öğrenimini günümüzün büyük verilerine dayanan çağa uygun hale getirmektedir. Makine öğrenimi olmadan, kitlesel bilgi akışına ayak uydurmak neredeyse imkansız olurdu (Lantz, 2013).

Makine öğrenmesi algoritmaları, örneklerden genelleştirerek önemli görevlerin nasıl gerçekleştirileceğini anlayabilen algoritmalardır. Bu algoritmalar genellikle manuel programlamanın olmadığı durumlarda uygulanabilir ve uygun maliyetlidir. Daha fazla veri ortaya çıktıkça, daha iddialı problemler bu algoritmalarla ele alınabilir. Makine öğrenmesi bilgisayar bilimi ve diğer alanlarda yaygın olarak kullanılmaktadır (Domingos, 2012).

Makine öğrenmesi bir veri tabanı problemi olmanın yanında yapay zekanın da bir parçasıdır. Bir sistemin zeki olması için değişen koşullar altında öğrenme kabiliyetine sahip olmalısı gerekir. Sistem bu öğrenmeyi kendi başına yapabilirse sistem tasarımcısının olası bir problem karşısında efor sarfetmesine gerek de kalmaz (Alpaydın, 2014).

Makine öğrenmesi uygulamaları ;

Uyarlanabilen web siteleri,

Bilgi işlem,

Biyoenformatik,

Makine arayüzleri,

DNA dizilerinin sınıflandırılması,

Kredi kartı sahtekarlığının tespiti,

Oyun,

İnternet Sahtekarlığı tespiti,

Borsa,

Yazılım Mühendisliği,

Konuşma ve el yazısı tanımlama,
Hesaplamalı reklam,
Hesaplamalı finans,
Öneri sistemleri,
Desen tanıma,
Sürücüsüz araç kontrolü,
Optik karakter tanımlama,
Tıbbi tanı gibi alanlarda kullanılmaktadır (Mohri ve ark., 2012).

Makine öğrenmesi Algoritmaları ise;
Sınıflandırma,
Regresyon,
Sıralama,
Kümeleme,
Boyut azaltma olmak üzere sıralanabilir (Mitchell,1997).

Makine öğrenmesi yaşam döngüsü :Makine öğrenimi, farklı öğrenme görevleri ile ilgili çeşitli alt alanlara ayrılmıştır (Joachims, 2002).

Denetimli öğrenme: Bir tahmin edici model veri kümesindeki diğer özelliklerin değerini kullanarak bir özelliğin değerini tahmin eder. Öğrenme algoritması, hedef özellik değeri ve diğer özelliklerin değerleri arasındaki ilişkiyi keşfetmeye ve modellemeye çalışır. Bir örneğin hangi kategoriye ait olduğunu kestirirken kullanılan denetimli makine öğrenmesi görevi sınıflandırma olarak adlandırılır. Sınıflandırma örneği olarak,bir e postanın spam olup olmadığı, bir bireyin kanser olup olmadığı, bir futbol takımının maçı kazanıp kazanmayacağı verilebilir (Lantz, 2013).

K-en yakın komşuluk, Naive Bayes, Karar ağacı, Karar Destek Makineleri, Yapay Sinir Ağları ve Topluluk Yöntemleri bu öğrenme türüne örnek verilebilir (Rechenthin, 2014).

Denetimsiz öğrenme : Veri kümesi bir hedef özellik ya da bilinen bir sonuca sahip değildir. Sınıf değerleri veri kümesinde önceden bilinmediği için bu öğrenme yönteminde hedeflenen, birbirine benzeyen özelliklerden kümeler oluşturmaktır (Rechenthin, 2014).

Kümeleme ve Boyut İndirgeme Denetimsiz Öğrenmeye örnek verilebilir (Hastie ve ark., 2009).

Yarı denetimli öğrenme: Yarı denetimli öğrenme denetlenen ve denetlenmeyen öğrenme açısından yarı yarıyadır. Etiketlenmemiş verilere ilaveten, algoritma etiketlenmiş verilere de sahiptir (Chapelle ve ark., 2009**).**

Sınıflandırma, Regresyon ve Sıralama problemlerini içerebilir. Modern Makine Öğrenme teknikleri ile bu yöntemin denetimli öğrenmeden daha iyi bir performans sergilemesi umulmaktadır (Mohri ve ark., 2012).

Transdüktif çıkarsama: Öğrenme algoritması, yarı denetimli öğrenme algoritmasına benzer şekilde etiketsiz ve etiketli eğitim verisi içermektedir (Joachims, 1999).

Ancak burada amaç, hedeflenen test örneklerinin etiketlerini kestirmektir. Yarı Denetimli Öğrenme Algoritması gibi diğer öğrenme algoritmalarından daha iyi sonuç vermesi beklense de henüz bu konuda soru işaretleri mevcuttur (Mohri ve ark., 2012).

Çevrimiçi öğrenme: Diğer öğrenme algoritmalarından farklı olarak eğitim ve test aşamaları bu öğrenme türünde karıştırılmıştır. Etiketsiz bir nokta aldıktan sonra buna ilişkin bir tahmin yapılmakta ve gerçek etiket ile bu tahmin kıyaslayıp bir hata hesaplanmaktadır. Diğerlerinin aksine herhangi bir dağılım varsayımı yoktur ve amaç kümülatif hatayı minimum yapmaktır (Mohri ve ark., 2012).

Takviyeli öğrenme: Çevrimiçi öğrenmeye benzer şekilde test ve eğitim aşamaları burada da karıştırılmıştır. Bilgi toplamak için öğrenici çevreyle etkileşim halindedir ve bazı durumlarda çevreyi etkileyerek gösterdiği her bir girişim için ödül alır. Burada amaç, bu ödül miktarının maksimum olmasıdır (Mohri ve ark., 2012).

Aktif öğrenme: Burada amaç denetimli öğrenmeden daha az etiketlenmiş örneklerle Denetimli Öğrenmedeki gibi bir performans elde etmektir. Etiketli veri toplamanın masraflı olduğu, biyoloji gibi uygulama alanlarında kullanılmaktadır (Mohri ve ark.,2012).

Daha az eğitim örneği ile daha iyi performans gösterecektir. Bu neden algoritmaların öğrenilmesi için arzu edilen bir özelliktir (Settles, 2012).

Makine öğrenmesi adımları :Tahmin edici modeller için en yaygın kullanılan süreçlerden biri Çapraz Endüstri Standart Süreç Modelidir (CRISP). Bu model 6 adımdan oluşmaktadır (Kelleher ve ark., 2015);

- Problemi Tanımlamak,
- Veriyi Anlamak,
- Veriyi Hazırlamak
- Modellemek,
- Modeli Değerlendirmek,
- Uygulamak.

Problemin tanımlanması: Bu altı adımlık sürecin en önemli basamağıdır. Araştırma konusunun tanımlama basamağı durumun değerlendirilmesini, araştırmanın amacını, bu altı adımlık sürecin amaçlarını ve projenin planlama sürecinin belirlenmesini kapsamaktadır (Kantardzic, 2011).

Veriyi anlama: İlk adım veri toplamaktır. Bir veri kümesi oluşturmak, veri niteliklerini tanımlamak, verileri keşfetmek ve verinin altında yatan yapıyı belirlemek olarak adımlar devam etmektedir (Larose, 2006).

Veriyi hazırlamak: Ham veriden verinin son haline kadar yapılması gereken tüm işlem ve düzeltmeleri içeren basamaktır. Veriyi hazırlama, veri dönüşümü ve modelleme yapabilmek için veri ayıklama özelliklerini içinde barındıran bir basamaktır (Kantardzic, 2011) .

Modelleme : Bu basamak sürecin en önemli basamağıdır. Çünkü bilgi çekmek için ileri çözümleme yöntemleri kullanılır. Modelleme yönteminin seçimi, modelin gelişimi ve tahmin işlemlerinden oluşur. Her probleme özgü farklı yöntemler olduğundan veri tipine uygun olmayıp belli tanımlamalar gerektirebilir. Bu yüzden gerekirse veri hazırlama basamağına geri dönülür (Larose, 2006).

Bu çalışmada Makine öğrenmesi algoritmalarından kullanılan yöntemler; aşağıda sırası ile verilmiştir.

3.2.1.1. Sınıflandırma ve regresyon ağaçları (CART)

Bir sınıflandırma çalışmasında amaç, probleme bağlı olarak doğru bir sınıflandırıcı üretmek veya problemin yapısına uygun kestirimci bir yapı ortaya çıkarmaktır. Eğer araştırmacı; kestirimci bir model bulma çabasındaysa, veri setindeki değişkenlerden hangisinin veya hangilerinin olayı basit bir şekilde karakterize ettiğini ve diğer sınıflardan ayırdığını anlamaya çalışmalıdır. Problemin türüne göre hem veriyi anlamak hem de kestirimci bir model elde etmek amaçlanabilir ve bazen bu amaçlardan biri diğerine göre daha önemli hale gelebilir (Bock 2002, Breiman, 2017).

Bu amaca uygun bir yöntem olan Sınıflandırma ve Regresyon ağaçları Morgan ve Sonquist'in AID (Automatic Interaction Detection) adlı karar ağacı algoritmasının devamı niteliğinde olup Breiman tarafından 1984 yılında önerilmiştir (Breiman, 2017).

Sınıflandırma ve Regresyon Ağaçları (CART) yöntemi, çoklu bağlantı, sapkın gözlem, kayıp verinin etkilerine büyük direncinden ve modelde hangi değişkenin içerilmesi gerektiğinin belirlenmesinden önce kestirimciler/tahminciler arasında yüksek seviyeli interaksiyonları tanımlamaya yeteneğinden dolayı parametrik yaklaşımlara tercih edilebilir (De'ath, 2000).

Sınıflandırma ve Regresyon Ağaçları Metodolojisi 3 kısımdan oluşmaktadır: maksimum ağacın oluşturulması, uygun ağaç genişliğinin seçimi, oluşturulan ağaçtan hareketle yeni verilerin sınıflandırılması (Timofeev, 2004).

Bölünmeler olarak sınıflandırıcılar: Formulasyon olarak, ölçümleri $x_1, x_2, \dots,$ olarak alalım. Burada x_1 ve x_2 açıklayıcı değişkenler olarak ölçülsün. Gözlemlere karşılık gelen bir ölçüm vektörü $(x_1, x_2, \dots,)$ tanımlanır ve tüm mümkün ölçüm vektörlerini içeren ölçüm uzayı X olarak adlandırılır.

Varsayalım ki, olaylar ya da nesneler j sınıflarına düşsünler. Sınıfların sayısı $1,2,\dots,j$ ve C de sınıfların bir kümesi $C = \{1, \dots, j\}$ olsun.

Sınıf kestiriminin bir sistematik yolu, X deki her bir x ölçüm vektörüne bir sınıf atama kuralıdır (Loh, 2011).

Tanım: Bir sınıflandırıcı ya da sınıflandırma kuralı, X üzerinde tanımlı bir $d(x)$ fonksiyonudur ve böylece her x için $d(x)$, $1,2,\dots,j$ sayılarından (sınıflarından) birine eşit olur.

Bir sınıflandırıcıya bakmanın diğer bir yolu $d(x) = j$ üzerinde X 'in alt kümesi olarak A_J tanımlamaktır;

$A_J = \{x; d(x) = j\}. A_1, \dots, A_J$ ayrıklardır ve $X = \bigcup_j A_J$ dir. Böylece her $x \epsilon A_J$ için kestirilen sınıf "j" dir (Loh, 2011; Breiman, 2017) (Bkz. Şekil 4.1.).

Veri setinin sınıflandırıcı inşasında kullanımı: Sınıflandırıcılar geçmiş tecrübelere dayanarak oluşurlar. Sistematik bir sınıflandırıcı inşasında, eski tecrübeler bir eğitim seti ile özetlenir. Bu, geçmişte gözlenen N gözlemle onların gerçek sınıfını içeren ölçüm verisinden oluşur (Morgan,2014).

Tanım : Bir eğitim seti, N durumlu $(x_1, j_1), \dots, (x_N, j_N)$ verisinden oluşur. Burada $x_n \epsilon X$, $j_n \epsilon \{1, \dots j\}$, $n = 1, \dots, N$'dir (Morgan, 2014). Eğitim setine L denirse;

$$L = \{(x_1, j_1), \dots, (x_N, j_N)\} \tag{4.1}$$

Doğruluğun kestirimi: C 'deki değerleri alan X üzerinde tanımlı verilen bir sınıflandırıcı, yani verilen bir fonksiyon $d(x)$ için, $R^*(d)$ onun gerçek yanlış sınıflandırıcısı oranı olarak tanımlanır. "Peki gerçek nedir ve nasıl kestirilir ?" sorusuna cevap için (yani $R^*(d)$ 'yi tahmin etmenin bir yolu olarak) sınıflandırıcıyı, doğru sınıflandırıldığı bilinen bir durumda test etmek gerekir. Örneğin, 1972-1975 yılları arasında toplanan bir verinin sınıflandırıcısını, 1976-1977 yılında toplanan veri seti üzerinde test ederek, sınıflandırıcının doğruluğu kestirilir. Yani, $R^*(d)$, eski veriler üzerinde oluşturulan $d(x)$ kullanıldığında, 1976-1977 yılına ait veriler üzerinde yanlış sınıflandırma oranı olarak tahmin edilir (Loh, 2011; Breiman, 2017).

Bunu anlamak için bir olasılıksal modele ihtiyaç duyulur. Tüm çiftlerin bir seti olarak (x, j) ve $X * C$ şeklinde bir uzay tanımlansın, burada $x \epsilon X$, j bir sınıf etiketi ve $j \epsilon C$ ' dir.

$P(A, j)$; $X * C$, $A \subset X$ ve $j \in C$ üzerinde bir olasılık olsun. $P(A, j)$ 'nin yorumu, ilgili popülasyondan rastgele çekilen bir durum, $P(A, j)$ olasılığına sahiptir şeklindedir. Yani ölçüm vektörü olan x, A'dadır ve sınıfı j'dir.

Eğitim seti olan L, N durum $(x_1, j_1), \dots, (x_N, j_N)$ içeren, bağımsız bir şekilde $P(A, j)$ dağılımından tesadüfi çekilsin. Burada L (eğitim seti) kullanarak $d(x)$ (kestirim modeli) inşa edilir. Daha sonra $R^*(d)$ (yanlış sınıflandırma oranı), aynı dağılımdan

çekilen yeni bir örneği sınıflandıracak olan d'nin (modelin) yanlış sınıflandırma oranı olarak tanımlanır (Safavian ve Landgrebe, 1991 ; Loh, 2011; Breiman, 2017).

Tanım: $x\epsilon\, X,\ \ Y\epsilon C$ olmak üzere $P(A,j)$ olasılık dağılımından çekilmiş yeni bir örnek (x,Y) olsun;

$P(x\,\epsilon\, A\,, Y=j) = P(A,j)$ ve $(X,Y),\ L$'den bağımsız ise;

$$R^*(d) = P(d(x) \neq Y) \tag{4.2}$$

olarak tanımlanır.

$P(d(x) \neq Y)$ olasılığının değerlendirilmesinde, L kümesi sabit düşünülür. Daha kesin bir notasyonla $P(d(x) \neq Y|L)$,verilen öğrenim örneği L'de, yeni örneğin yanlış sınıflandırma olasılığıdır (Loh, 2011).

Gerçek problemlerde, L 'deki veri sınıflandırılmış durumların ek bir geniş örneğini almada az bir olasılığa sahiptir. Bu yüzden L, hem $d(x)$ oluşturmada hem de $R^*(d)$ 'yi tahminde kullanılmalıdır. $R^*(d)$ 'nin bu tahminini iç tahmin olarak isimlendirilir (Breiman, 2017).

İç tahminlerden, ilki en az doğru ve en yaygın olan yeniden yerine koyma kestirimidir. d inşa edildikten sonra, L'deki gözlemler, sınıflandırıcının içinden geçer. Yanlış sınıflandırılan durumların oranı, yerine koyma kestirimidir. Bunun eşitlik formu için, Yeniden Yerine Koyma Kestirimi, $R(d)$ olsun (Safavian ve Landgrebe,1991).

$$R(d) = \frac{1}{N}\sum_{n=1}^{N} X(d(x_n) \neq j_n) \tag{4.3}$$

Yeniden yerine koyma kestiriminin problemi, bağımsız bir örneklem yerine d' yi inşa etmek için aynı veriyi kullanmaktır. Tüm sınıflandırma prosedürleri, ister direkt ister dolaylı olarak $R(d)$'yi minimize etmeye çabalar. $R^*(d)$'nin bir kestirimi olarak $R(d)$'nin sonraki değerinin kullanılması, d'nin bir optimistik resmini yüzeysel olarak verebilir (Loh, 2011; Breiman, 2017).

İkinci yöntem, test örneği kestirimidir. Burada, L'deki durumlar ; L_1 ve L_2, olmak üzere ikiye bölünsün. Sadece L_1'deki durumlar kullanılarak d inşa edilir. Daha

sonra L_2 'deki durumlar kullanılarak $R^*(d)$ 'nin tahmini yapılır. Eğer N_2 , L_2 deki durumların sayısı ise, test örneği kestirimi (Safavian ve Landgrebe, 1991);

$$\mathrm{R^{ts}(d)} = \frac{1}{\mathrm{N_2}} \sum_{(\mathrm{x_n},\mathrm{j_n})\in \mathrm{L_2}}^{\mathrm{N}} \mathrm{X(d(x_n) \neq j_n)} \tag{4.4}$$

Bu yöntemde, L_2 , L_1 'den bağımsız ve aynı dağılımdan çekilmiş olarak düşünülebilir. Sık sık L_2, L'deki durumların $1/3$'ü olarak alınır, ancak bu bölünmenin herhangi bir teorik doğrulaması var mıdır bilmiyoruz. Bu bölünme etkili örnek boyutunu azaltacağı için yani verinin sadece $2/3$'ü ile d oluşturulacağı ve $1/3$'ü ile $R^*(d)$ tahmini yapılacağı için bir dezavantaja sahiptir. Örnek büyüklüğü yeterince genişse bu büyük bir problem olmayacaktır (Safavian ve Landgrebe, 1991; Kohavi, 1995; Breiman, 2017).

Bir diğer yöntem, daha küçük örnek boyutu için $v-$kat çapraz geçerliktir. Burada L'deki durumlar tesadüfi bir şekilde olabildiğince eşit v alt kümeye bölünür. Bu alt kümeler $L_1, \dots L_v$ 'dir. Prosedür,bir sınıflandırıcı oluşturmak için herhangi bir öğrenme kümesine uygulanabilir. Her v için, $v = 1, \dots, v$ öğrenim örneği $L - L_V$ olarak prosedür uygulanır.Yani L'deki durumlar L_V de yoktur ve $d^v(x)$ sınıflandırıcı sonucu olarak adlandırılsın. d^v 'nin oluşumunda, L_V deki durumlar kullanılmayacağından,

$$\mathrm{R^{ts}(d^v)} = \frac{1}{\mathrm{N_v}} \sum_{(\mathrm{x_n},\mathrm{j_n})\in \mathrm{L_v}}^{\mathrm{N}} \mathrm{X(d^v(x_n) \neq j_n)} \tag{4.5}$$

olur.

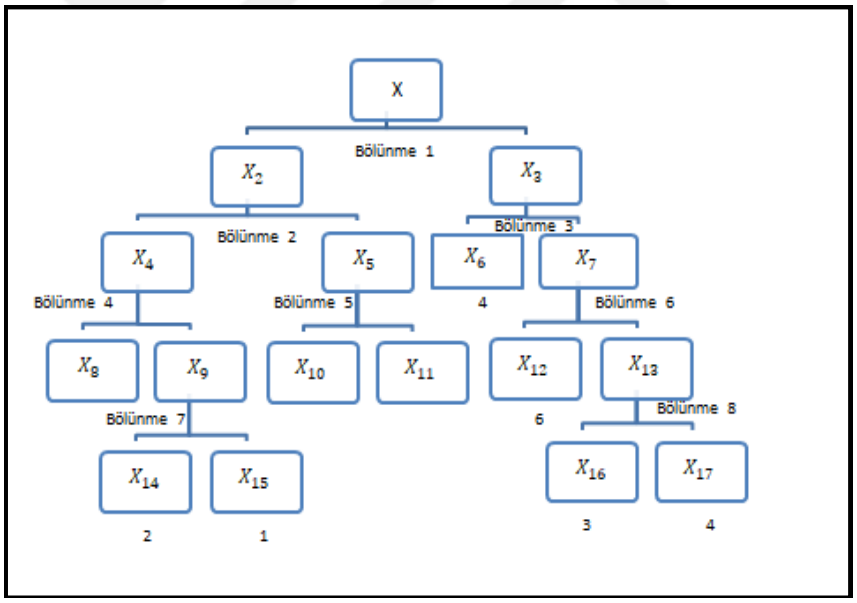
Burada $N_v = N/v$, L_V 'deki durumların sayısıdır. Aynı prosedürü uygulayarak tüm L'yi kullanıp sınıflandırıcı d inşa edilsin; geniş v için, her bir v sınıflandırıcısı neredeyse L kadar geniş $N(1-1/v)$ boyutlu bir öğrenim kümesi kullanarak oluşturulur. Çapraz geçerliğin temel varsayımı, prosedürün istikrarlı olmasıdır. Yani, sınıflandırıcı d^v, $v = 1, \dots, v$ herbiri neredeyse tüm L kullanılarak inşa edilir ve yanlış sınıflandırma oranı $R^*(d^{(v)})$ neredeyse $R^*(d)$ ye eşittir. Bu bulgusallığa dayanarak, $R^{cv}(d)$;

$$\mathrm{R^{cv}(d)} = \frac{1}{\mathrm{v}} \sum_{\mathrm{v=1}}^{\mathrm{V}} \mathrm{R^{ts}(d^v)} \tag{4.6}$$

N-kat çapraz geçerlik "birini dışarda bırak" kestirimidir. Her n için, $n = 1, \ldots, N$ n'inci durum bir tarafta tutulur ve sınıflandırıcı diğer $N - 1$ durum kullanılarak inşa edilir. Sonra n'inci durum bir tek test örneği olarak kullanılır ve $R^*(d)$, bir önceki eşitlikle kestirilir (Loh, 2011).

Çapraz geçerlik tutucudur. L'deki her durum d inşası için kullanılır ve her durum test örneğinde sadece bir kez kullanılır. Ağaç yapılı sınıflandırıcılarda *10*-kat çapraz geçerlik kullanılır ve sonuçlanan kestirimciler, simülasyon verisinde tatmin edici bir şekilde $R^*(d)$' ye yakındır (Loh,2011; Breiman, 2017).

Ağaç yapılı sınıflandırıcılar: Ağaç yapılı sınıflandırıcılar ya da daha doğru bir şekilde ikili ağaç yapılı sınıflandırıcıların bir X kümesini, kendisinden başlayarak tekrarlı bir şekilde X'in alt kümelere bölünmesi ile inşa edilir (Loh, 2011; De'ath ve Fabricius, 2000). Bu süreç altı sınıflı bir ağaç için Şekil 3.1.'de gösterilmiştir.



Şekil 3.1. Sınıflandırma ağacının yapısı.

Şekilde X_2 ve X_3 ayrıktır.$X = X_2 \cup X_3$ olup benzer şekilde X_4 ve X_5 de ayrıktır ve $X_2 = X_4 \cup X_5$ ve $X_3 = X_6 \cup X_7$ olur.

Bilinmeyen alt kümeler $X_6, X_8, X_{10}, X_{11}, X_{12}, X_{14}, X_{15}, X_{16}$ ve X_{17} terminal alt kümeler olarak adlandırılır. Şekilde belirtilen terminal düğümler dikdörtgen ile, terminal olmayanlar ise dairelerle gösterilmiştir (Loh, 2011; Breiman, 2017).

Terminal alt kümeler X'in bir bölünmesini temsil eder. Her bir terminal alt kümesi bir sınıf etiketi ile gösterilir. Aynı sınıf etiketine sahip iki ya da daha fazla terminal alt kümeler olabilir. Sınıflandırıcıya karşılık gelen bölünme aynı sınıfa karşılık gelen tüm terminal alt kümelerin birleştirilmesiyle elde edilir. Yani, $A_1 = X_{15}$,$A_2 = X_{11} \cup X_{14}$, , $A_3 = X_{10} \cup X_{16}$, , $A_4 = X_6 \cup X_{17}$, $A_5 = X_8$, $A_6 = X_{12}$ 'dir.Bölünmeler $X = (X_1, X_2 \ldots)$ noktasının koordinatlarının koşulları ile oluşturulur. Örneğin, X'in X_2 ve X_3'e bölünmesi;

$$\mathrm{X}_2 = \{\mathrm{x};\ \mathrm{x}_4 \leq 7\}, \mathrm{X}_3 = \{\mathrm{x};\ \mathrm{x}_4 > 7\} \tag{4.7}$$

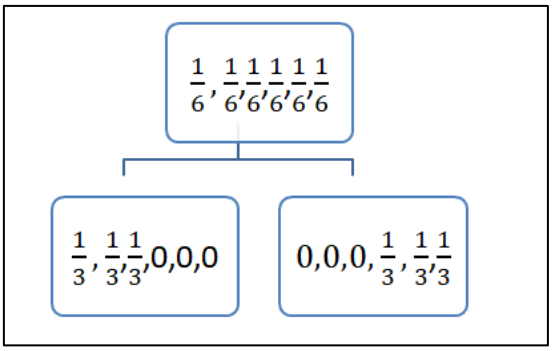
şeklinde olabilir (Loh 2011; Breiman, 2017).

Ağaç sınıflandırıcısı x ölçüm vektörü için bir sınıfı şöyle tahmin eder: İlk bölünmenin tanımından x'in X_2'ye ya da X_3'e gidip gitmeyeceği belirlenir. Örneğin (Bkz. 4.7) $X_4 \leq 7$ ise x, X_2 kümesine, eğer $X_4 > 7$ ise X_3 kümesine gider. x terminal kümeye ulaştığında, x'in sınıfı gittiği terminal alt sınıfın kümesi olarak tahmin edilir.

Teorik olarak X'in bir alt kümesine t düğümü, X'in kendisine t_1 kök düğümü, terminal alt kümelere terminal düğümler, terminal olmayan alt kümelere de terminal olmayan düğümler denir.

Ağaç sınıflandırıcı inşası: Ağaç inşasında ilk problem L datasının X'in ikili bölünmelerini sürdürerek daha küçük parçalara bölmek için nasıl kullanılacağıdır. Temel fikir bir alt kümenin her bir bölünmesini aşağıdaki alt kümeleri (child), yukardaki (parent) alt kümelere göre daha saf olacak şekilde seçebilmektir (Guttman,1984).

Örneğin altı sınıflı bir gemi probleminde,herhangi bir düğümde $p_1, p_2, \ldots, p_6$ ile herhangi bir düğümde sınıf $1,2,\ldots 6$' dan birine eşit olma oranlarını gösterelim. t_1 kök düğüm için; $p_1, p_2 \ldots p_6 = \left(\frac{1}{6}, \frac{1}{6} \ldots \frac{1}{6}\right)$ 'dır. t_1'in iyi bir bölünmesi ile sınıf $1,2,3$ 'e sahip gemilerin sol düğüme, sınıf $4,5,6$ ' ya sahip gemilerin sağ düğüme ayrılması olabilir.



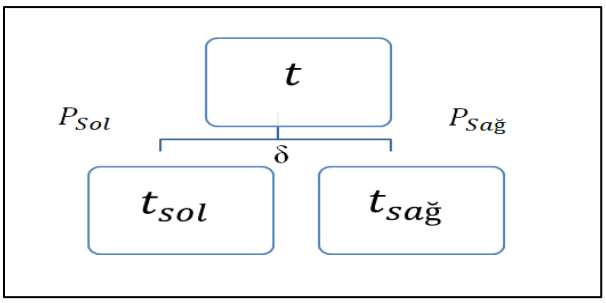
Şekil 3.2. Belirli bir sınıfa ait olma oranları.

t_1'in iyi bir bölünmesi bulunduğunda, arama t_2 ve t_3'ün iyi bir bölünmesini bulmak ile devam eder. Düğümlerin daha saf düğümler oluşturması için bölünmesi fikri şu şekilde uygulanır (Breiman, 2017); $p(j\backslash t)$, $j = 1,2 \ldots,6$ düğüm oranları, $x_n \in t,\ j$ sınıfına dahil olma oranı $p\left(\frac{1}{t}\right) + \cdots p\left(\frac{6}{t}\right) = 1$ olacak şekilde tanımlanır.

t'nin safsızlık ölçüsü $i(t)$, $\mathrm{p}\left(\frac{1}{\mathrm{t}}\right) + \cdots \mathrm{p}\left(\frac{6}{\mathrm{t}}\right)$'nin negatif olmayan bir θ fonksiyonu olarak tanımlanır.Bu θ fonksiyonu:

- $\theta\left(\frac{1}{6},\frac{1}{6},\ldots,\frac{1}{6}\right) = max,$
- $\theta(1,0,\ldots,0) = 0,$
- $\theta(0,0,\ldots,1) = 0$ şartlarını sağlar.

Yani, düğüm safsızlığı tüm sınıflar bir düğümde eşit şekilde yer aldığında maksimum,bir düğüm sadece bir sınıf içerdiğinde minimumdur (Alpaydın, 2009).
Herhangi bir t düğümü için,varsayalım ki bu düğümü,durumların $p_{sağ}$ oranında $t_{sağ}$'a p_{sol} oranında t_{sol}'a ayıran bir δ aday bölünmesi olsun.



Şekil 3.3. Aday Bölünme.

Bu durumda bu bölünmelerin iyiliği safsızlıktaki azalma olarak tanımlanır vr şöyle verilir;

$$\Delta \mathrm{i}(\delta, \mathrm{t}) = \mathrm{i}(\mathrm{t}) - \mathrm{p}_{\mathrm{sol}}\mathrm{i}(t_{sol}) - \mathrm{p}_{\mathrm{sağ}}\mathrm{i}(t_{sağ}) \tag{4.8}$$

Son olarak, her düğümde δ ikili bölünmelerin bir aday seti S tanımlanır. Genellikle, ϑ sorular kümesinde her sorunun $x \in A$?, $(A \subset X)$ şeklindeki sorularla üretilerek bölünmelerin S setini kavramak kolaydır. Daha sonra δ'ye ilişkin bölünme, t'deki tüm x_n'leri yanıt "evet" ise t_{sol}'a yanıt "hayır" ise $t_{sağ}$'a gönderir.

6 sınıflı bir sınıflandırma probleminde düğüm safsızlığı ;

$$\mathrm{i}(\mathrm{t}) = -\sum_1^6 \mathrm{p}(\mathrm{j}|\mathrm{t}) \log \mathrm{p}\,(\mathrm{j}|\mathrm{t}) \tag{4.9}$$

şeklinde olabilir.

Her bir düğümdeki ikili bölünmelerin bir aday kümesi tanımlanır ve ağaç şu şekilde oluşturulur. t_1 kök düğümünde safsızlıktaki en yüksek azalmayı sağlayan δ^* bölünmesi bulunur;

$$\Delta \mathrm{i}(\delta^*, \mathrm{t}_1) = \max_{\delta \in \mathrm{S}} \Delta \mathrm{i}(\delta, \mathrm{t}_1) \tag{4.10}$$

Burada S kümesi olası tüm bölünmeler kümesidir. Daha sonra t_1, δ^* bölünmesi kullanılarak t_2 ve t_3 düğümlerine bölünür ve aynı prosedür t_2 ve t_3 için en iyi $\delta \in S$ ile tekrarlanır (Loh, 2011).

Ağaç oluşturmasını durdurmak için, sezgisel bir kural oluşturulmuştur. Eğer bir t düğümü için safsızlıkta önemli derecede bir azalma olmuyorsa bu düğüm terminal düğüm olarak belirlenir. Terminal düğümün sınıfı,çokluk kuralı ile belirlenir (Chipman ve ark. 1998; Breiman, 2017).

Büyüyen ilk ağaç metodolojisi: Ağaç bir sınıflandırıcı olarak şu şekilde kullanılır; eğer bir gözlemin sınıfı belirli değilse, gözlem ağaca dahil edilir ve bu gözlemin ulaştığı terminal düğümün sınıfı bu gözlemin sınıfı olarak belirlenir.

Ağaç oluşturulurken ihtiyaç duyulan dört temel eleman şunlardır:

- $x \in A$? şeklindeki ikili sorular kümesi ,

- Durdurma kuralı,
- Herhangi bir t düğümündeki herhangi bir bölünmeyi değerlendirecek $\theta(\delta, t)$ bölünme kriteri iyiliği,
- Her bir terminal düğüme sınıf atama kuralı (Breiman, 2017).

Standart sorular kümesi: Eğer veri seti standart bir yapıya sahipse, φ sorular kümesi de standartlaştırılabilir. Ölçüm vektörlerinin $X = (X_1, X_2, \dots X_M)$ formunda olduğunu varsayalım.Burada M sabit bir boyuta sahip olurken $X_1, X_2, \dots X_M$ değişkenleri kategorik ve sürekli değişkenler olabilir. Bu durumda φ sorular kümesi şu şekilde tanımlanabilir.

- Her bir bölünme sadece bir değişkene dayanır.
- Her bir x_m sürekli değişken için sorular $\{x_m \leq c\ ?\}$ şeklindedir. Burada $c \in (-\infty, \infty)$'dir.
- Eğer $x_m, \{b_1, b_2, \dots b_L\}$ değerlerini alan kategorik bir değişken ise φ kümesi $x_m \in S\ ?$ şeklindeki sorulardan oluşur. Burada S, $\{b_1, b_2, \dots b_L\}$'nin tüm alt kümelerini belirtir (Loh, 2011; Breiman, 2017)

Bölünme ve bölünme durma kuralı: Bölünmenin iyiliği kriteri safsızlık fonksiyonundan elde edilir.

Tanım: $p_j \geq 0$ ve $j = 1,2, \dots j$ için $\sum_j p_j = 1$ olsun.$(p_1, p_2, \dots p_j)$ sayılarından oluşan tüm j-tuple kümesi üzerinde tanımlanan θ fonksiyonu,

- θ , $\left(\frac{1}{j}, \frac{1}{j}, \dots, \frac{1}{j}\right)$ noktasında maksimum değerini alır.
- θ, minimum değerini $(1,0, \dots ,0)$,$(0,1, \dots ,0)\dots$ $(0,0, \dots ,1)$ noktalarında alır.
- $\theta, p_1, p_2, \dots p_j$ 'nin simetrik bir fonksiyonudur.

şartlarını sağlarsa bir safsızlık fonksiyonu olur (Alpaydın,2009)

Tanım:Verilen bir θ safsızlık fonksiyonu için $i(t)$ safsızlık ölçüsü herhangi bir t düğümü için, $i(t) = \theta\left(p\left(\frac{1}{t}\right), p\left(\frac{2}{t}\right), \dots p\left(\frac{j}{t}\right)\right)$ ile tanımlanır.

Herhangi bir t düğümü için, varsayalım ki bu düğümü,durumların $p_{sağ}$ oranında $t_{sağ}$'a p_L oranında t_{sol}'a ayıran bir δ aday bölünmesi için, safsızlıktaki azalmayı;

$\Delta i(\delta, t) = i(t) - p_{sol} i(t_{sol}) - p_{sağ} i(t_{sağ})$ şeklinde tanımlamıştık.

O zaman $\theta(\delta, t)$'nin bölünme iyiliği $\Delta i(\delta, t)$ olsun.Birkaç bölünme yaptığımızı ve mevcut terminal düğümlerin bir setine ulaştığımızı varsayalım. Kullanılan bölme seti, ikili ağaç olarak adlandırdığımız T'yi tanımlar. $\tilde{T}$'yi ; $I(t) = i(t)p(t)$ ulaştığımız terminal düğümlerin seti olarak adlandıralım ve ağaç safsızlığını;

$I(T) = \sum_{t\in\tilde{T}} I(T) = \sum_{t\in\tilde{T}} i(t)p(t)$ ile tanımlayalım. Kolayca görülebileceği gibi $\Delta i(\delta, t)$'yi maksimum yapan bölünme tüm ağaç safsızlığını minimum yapan $I(T)$ değerine eşittir. Bir $t \in \tilde{T}$ düğümü ve t' yi $t_{sağ}$ ve t_{sol} olarak bölen bir δ bölünmesi alalım.Yeni ağaç olan T' için safsızlık;

$$\mathrm{I(T')} = \sum_{\tilde{T}-\{t\}} \mathrm{I(t)} + \mathrm{I(t_{sol})} + \mathrm{I(t_{sağ})} \tag{4.11}$$

Safsızlıktaki azalma ise;

$$\mathrm{I(T)} - \mathrm{I(T')} = \mathrm{I(t)} - \mathrm{I(t_{sol})} - \mathrm{I(t_{sağ})} \tag{4.12}$$

ile tanımlanır.

Bu sadece t düğümü ve onu bölen δ' ye bağlıdır. Böylece t üzerindeki bölünmelerle safsızlıktaki azalışın maksimizasyonu;

$$\Delta\mathrm{I}(\delta, \mathrm{t}) = \mathrm{I(t)} - \mathrm{I(t_{sol})} - \mathrm{I(t_{sağ})} \tag{4.13}$$

t_{sol} ve $t_{sağ}$'a giden t düğüm popülasyonun sırasıyla p_{sol} ve $p_{sağ}$ oranları ise

$$\mathrm{p_{sol}} = \frac{\mathrm{p(t_{sol})}}{\mathrm{p(t)}} \tag{4.14}$$

$$\mathrm{p_{sağ}} = \frac{\mathrm{p(t_{sağ})}}{\mathrm{p(t)}} \tag{4.15}$$

$$\mathrm{p_{sol}} + \mathrm{p_{sağ}} = 1 \tag{4.16}$$

$$\Delta\mathrm{I}(\delta, \mathrm{t}) = \left[\mathrm{i(t)} - \mathrm{p_{sol}i(t_{sol})} - \mathrm{p_{sağ}i(t_{sağ})}\right]\mathrm{p(t)} = \Delta\mathrm{i}(\delta, \mathrm{t})\mathrm{p(t)} \tag{4.17}$$

olur.

$\Delta I(\delta, t)$ ve $\Delta i(\delta, t)$ birbirinden $p(t)$ faktörü ile ayrıldığından, aynı bölünme olan δ^* iki ifadeyi de maksimum yapan bölünmedir. Böylece,bölme seçme

prosedürü,tüm ağaç safsızlığını tekrarlı bir şekide minimize etme çabası olarak düşünülebilir. Bölünmeyi durdurmak için bir $\beta > 0$ eşik değeri belirlenir ve eğer;

$$\max_{\delta \in S} \Delta I(\delta, t) < \beta \tag{4.18}$$

ise bu t düğümü bir terminal düğüm olarak belirlenir (Safavian ve Landgrebe, 1991; Loh, 2011).

Sınıf ataması ve yeniden yerine koyma kestirimi: Bir T ağacının inşa edildiğini ve $\widetilde{T}$ terminal düğümleri olduğunu varsayalım.

Tanım: Bir sınıf atama kuralı, her $t \in \widetilde{T}$ terminal düğümüne bir sınıf atar. $t \in \widetilde{T}$ ye atanılan düğüm $j(t)$ tarafından atanır. Sınıf atama kuralı $j(t)$ için, $\sum_{j \neq j(t)} p(j|t)$, t düğümüne düşen bir durumun yanlış sınıflandırma olasılığının yeniden yerine koyma kestirimidir. Burada $j^*(t)$'yi sınıf atama kuralı olarak alınır, kural yeniden yerine koyma kestirimini minimize etmektir. (Loh, 2011; Breiman, 2017). Şöyle ki;

Tanım: Sınıf atama kuralı $j^*(t)$ şöyle verilir ; Eğer $p(j|t) = max_i\, p(i|t)$ ise o zaman $j^*(t) = j$ 'dir. Eğer maksimum iki ya da daha fazla farklı sınıflarda edinilirse, $j^*(t)$ maksimum sınıflardan herhangi birine keyfi olarak atanır. (Timofev, 2004; Loh 2011; Breiman, 2017). Bu kuralı kullanarak,

Tanım: t düğümüne düşen bir durumun yanlış sınıflandırma olasılığının yeniden yerine koyma kestirimi (Loh, 2011);

$$r(t) = 1 - \max_j p(j|t) \tag{4.19}$$

$$R(t) = r(t)p(t) \tag{4.20}$$

Daha sonra T ağacı sınıflandırıcısının tüm ağacın genel yanlış sınıflandırma oranı için yeniden yerine koyma tahmini (Breiman, 2017) $R^*(T)$;

$$R^*(T) = \sum_{t \in \tilde{T}} R(t) \tag{4.21}$$

Ağaç boyutunun seçimi: Burada 2 temel konu, doğru boyutlu (right size) bir ağacın *"T"* eldesi ve beklenen yanlış sınıflandırma maaliyeti olan $R^*(T)$ 'nin yanlış sınıflandırma kestiriminin doğru eldesidir.

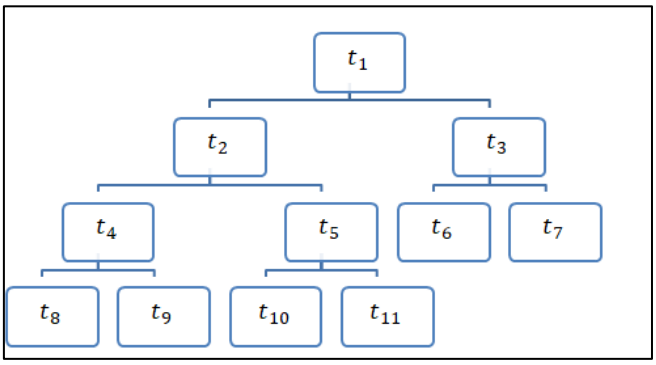
Adımsal ağaç yapısı, verinin mümkün bölünmelerinin her adımda geniş bir setini optimize eder. Eğer sadece yeniden yerine koyma hatası kullanılırsa, genel sonuçlar çok fazla bölünmeye sebep olur ve ağaç, verinin sağladığından çok daha geniş bir yapıda olur ayrıca yeniden yerine koyma kestirimi $R(T)$ aşağı doğru yanlı olur (aşırı uyum sorunu).

Örneğin, eğer bölünme, her terminal düğüm sadece verideki bir durumu içerirken yürütülüyorsa, o zaman her düğüm içerdiği durumla sınıflandırılır ve yeniden yerine koyma hatası kestirimi "0"yanlış sınıflandırma oranını verir. (Loh 2011; Breiman, 2017).

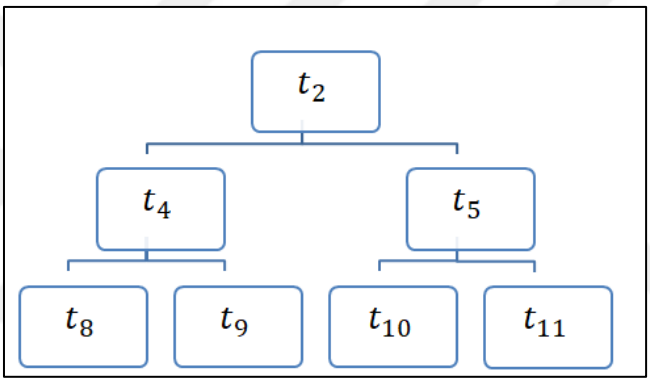
Genelde ne kadar çok bölünme olursa o kadar az yeniden yerine koyma kestirimi $R(T)$ elde edilir. Buradaki yapı adımsal doğrusal regresyondakine benzer. Çok büyük bir ağaç, doğru büyüklükteki ağaçtan daha yüksek bir yanlış sınıflandırma oranına sahip olacaktır. Doğru büyüklükte ağacın inşası için (Breiman, 2017);

- Durdurma yerine ağacı budamak. Bir ağacı olabildiğince geniş olarak büyütmek ve yukarı doğru,sonunda kök düğüme ulaşıncay kadar doğru bir şekilde budamak.
- Budanmış alt ağaçlar arasından doğru boyuttaki ağacı seçmek için $R^*(T)$ nin daha doğru bir kestirimini kullanmak (Timofev, 2004; Loh 2011; Breiman, 2017).

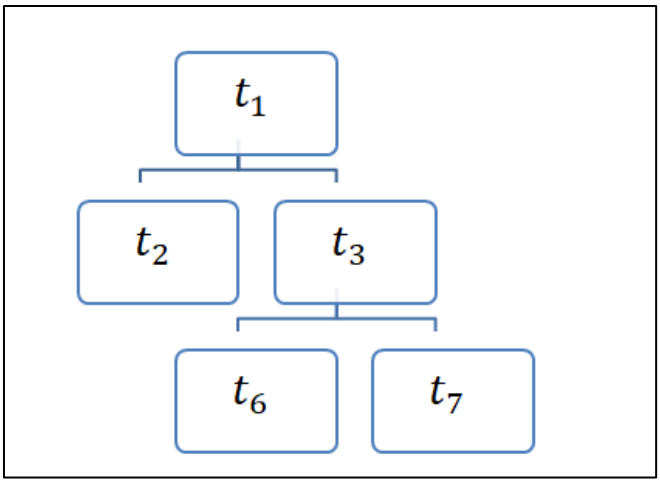
Budama :



Şekil 3.4. T ağacı.



Şekil 3.5. T_{t2} yaprakları.



Şekil 3.6. $T - T_{t2}$ ağacı.

Tanım: Bir T ağacının t kök düğümlü T_t dalı, t düğümünden ve tüm alt düğümlerinden oluşur.

Tanım: T ağacından T_t dalını budamak t düğümünün tüm alt düğümlerini T ağacından silmektir. .Bu şekilde budanan ağaç $T - T_t$ ile gösterilir.

Tanım: Eğer T', T ağacının budanması ile elde edilmişse, T' budanmış alt ağaç olarak isimlendirilir ve $T' < T$ ile gösterilir.

30-40 düğüme sahip orta büyüklükte bir ağacın bile oldukça fazla alt ağacı bulunmaktadır ve hatta t_1 düğümünden bile budamanın oldukça fazla seçeneği vardır. Bu durumda seçici bir budama işlemi gereklidir. Yani mantıklı sayıda alt ağaç seçilmelidir. Kabaca konuşulursa, seçilen her bir alt ağaç büyüklük olarak en iyi alt ağaç olmalıdır. En iyi kelimesi bir alt ağacın ne kadar iyi olduğunun ölçülmesini gerektirmektedir. $R(T)$, $R^*(T)$ 'nin iyi bir tahmini olmasa bile aynı büyüklükteki farklı ağaçların karşılaştırılması için kullanılabilir. Bu ağaçlar arasında $R(T)$' si en düşük olan ağaç seçilir (Safavian ve Landgrebe 1991; Breiman, 2017).

Bölme kuralları**:** Bölme kuralları, $\emptyset(\delta, t)$ bölme iyiliği fonksiyonunun belirlenmesi ile oluşur. İki sınıf problemi için bölme kriteri ise;
$\emptyset(p_1, p_2) = 1 - max(p_1, p_2) = min(p_1, p_2) = min(p_1, 1 - p_1)$ olmak üzere, Gini Kriteri;

$$\emptyset(p_1, \dots, p_j) = -\sum_j p_j \log p_j \tag{4.22}$$

olarak belirlenir.
Gini İndeksi;

$$i(t) = \sum_{j \neq i} p(j|t)\, p(i|t) \tag{4.23}$$

Ayrıca,

$$i(t) = \sum_j (p(j|t))^2 - \sum_j p^2(j|t) = 1 - \sum_j p^2(j|t) \tag{4.24}$$

şeklinde de yazılabilir.İkili sınıflandırma problemlerinde bu indeks;

$$i(t) = 2p(1|t)p(2|t) \quad (4.25)$$

şeklindedir (Alpaydın, 2014).

3.2.1.2. Rastgele orman

Boosting (Freund ve Schapire, 1996) ve Bagging (Breiman, 1996) ağaçların sınıflandırılmasında toplu öğrenme için çok iyi iki yöntem olarak bilinmektedir.

Boosting'de, ardışık gelen ağaçlar bir öncekine bağımlıdır. Bir önceki öncüller tarafından yanlış tahmin edilmiş noktalar için ekstra ağırlık verilir. Sonrasında ağırlıklı oy tahmin için alınır (Liaw ve Wiener, 2002).

Bagging'de eğitim verisi kullanılarak her bir ağaç inşa edilir. Ardışık gelen ağaçlar bir öncekinden bağımsızdır ve en çok oy alan ağaç tahmin için kullanılır. 1996'da Breiman tarafından ilk amaçlanan Bagging tekniği kullanılarak her bir ağacın birbirinden bağımsız olarak eğitim verileri ile oluşturulmasını sağlamıştır (Liaw ve Wiener, 2002).

Rastgele Orman yöntemi oluşturulmak istenen ağaç sayısınca Sınıflandırma Ağacının veya amaca uygun olarak Regresyon Ağacının topluluklarından oluşmaktadır. Bu yüzden topluluk yöntemlerinden en yaygın olarak kullanılan algoritmalardan biri de Rastgele Ormandır. Yöntemin altında yatan temel fikir, çok sayıda tahminci ağaçlar arasından rastgele seçilen bir alt kümesi yardımıyla topluluklar oluşturmaktır (Breiman, 2001).

Rastgele Orman Yöntemi hem kategorik hem sürekli hem de her ikisinin yer aldığı veri setlerinde; aynı zamanda büyük veya küçük boyutlu veri setlerinde rahatlıkla kullanılabilmektedir. Yöntemin dez avantajı olarak, Sınıflandırma Ağacı Yönteminin aksine çıktı olarak bir ağaç vermemesidir (Akman, 2011).

Bu şekilde rastgele tahminci seçmenin avantajı, topluluktaki ağaçlar arasında daha az korelasyon elde edildiği için oluşan modelin doğruluğu daha yüksektir (Suchetana ve ark., 2017).

Bu yöntemde de Sınıflandırma ve Regresyon ağaçlarında olduğu gibi bölünme kriteri olarak daha önce (4.26) eşitliği ile vermiş olduğumuz Gini indeksi kullanılmaktadır. Gini indeksinin azalması istenen bir durumdur çünkü saflığın arttığına

işarettir ve bu indeksin nihai olarak sıfıra eşit olması demek maksimum saflık demektir (Watts ve ark., 2011).

Bu yöntemle geliştirilen ağaçlarda budama yapılmaz (Breiman, 2006).

Değerlendirme basamağı: Bu basamakta uygun modeller kurulduktan sonra uygulama sonuçları göz önüne alınarak araştırma probleminin amaçlarını gerçekleştirip gerçekleştirmediği değerlendirilir. Sonuçların değerlendirilmesi bu altı adımlık sürecin gözden geçirilmesi ve sonraki adımların ne olacağı konusunu içermektedir (Larose,2006).

Hangi sınıflandırıcının birbirine göre daha iyi bir sınıflandırma performansı sergilediğini bulabilmek için çeşitli performans ölçüm yöntemleri geliştirilmiştir. (Kartal ve Balaban, 2015).

Gerçek sonuç değerleri ile model aracılığıyla tahmin edilen değerlere ait çizelge (Fawcett, 2006) aşağıda verilmiştir. Bu çizelgeye göre ikili (binary) sınıflandırma sonuçlarının belirlenmesi ve değerlendirilmesi için hesaplanan bazı model performans değerlendirme ölçüleri de çizelgeden sonra verilmiştir (Kartal ve Balaban, 2015).

		GERÇEK		
		POZİTİF	NEGATİF	TOPLAM
TAHMİN	POZİTİF	Doğru Pozitif (dp)	Yanlış Pozitif (yp)	TPoz
	NEGATİF	Yanlış Negtaif (yn)	Doğru negatif (dn)	TNeg
	TOPLAM	Poz	Neg	M

Çizelge 3.1. Hata Matrisi.

Sınıflandırıcıların doğru sınıflandırma oranı ve yanlış tahmin oranı;

$$\text{Doğruluk} = \frac{d_p + d_n}{\text{m}} \tag{4.26}$$

$$\text{Hata Oranı} = 1 - \text{Doğruluk} \tag{4.27}$$

Sınıflandırıcının, pozitif sınıflara ait etiketlerini kestirmedeki etkililiğine verilen isim duyarlılıktır (Flach, 2004; Kartal ve Balaban, 2015);

$$\text{Duyarlılık} = \frac{d_p}{d_p + y_n} \tag{4.28}$$

Sınıflandırıcının negatif sınıflara ait etiketlerini kesirmedeki etkililiğine verilen isim belirleyiciliktir (Flach, 2004; Kartal ve Balaban, 2015);

$$\text{Belirleyicilik} = \frac{d_n}{d_n + y_p} \tag{4.29}$$

Gerçekte negatif olup, yanlışlıkla pozitif olarak sınıflandırılan örneklerin, tüm negatif etiketli örneklere oranına yanlış pozitif oranı denilmektedir. Gerçekte pozitif olup, yanlışlıkla negatif olarak sınıflandırılmış örneklerin, tüm pozitif etiketli örneklere oranına ise yanlış negatif oranı denilmektedir (Glas ve ark.,2003; Kartal ve Balaban, 2015);

$$\text{Yanlış Pozitif Oranı} = \frac{\mathrm{y_p}}{\mathrm{y_p} + \mathrm{d_n}} \tag{4.30}$$

$$\text{Yanlış Negatif Oranı} = \frac{\mathrm{y_n}}{\mathrm{y_n} + \mathrm{d_p}} \tag{4.31}$$

Doğru sınıflandırılmış olan pozitif örneklerin toplam pozitif kestirilen örneklere oranı pozitif öngörü değeri olarak adlandırılırken, doğru sınıflandırılmış negatif sınıf etiketine sahip örneklerin toplam negatif kestirilen örneklere oranına negatif öngörü değeri olarak adlandırılır (Glas ve ark.,2003; Kartal ve Balaban, 2015);

$$\text{Pozitif Öngörü Değeri} = \frac{d_p}{d_p + y_p} \tag{4.32}$$

$$\text{Negatif Öngörü Değeri} = \frac{d_n}{d_n + n} \tag{4.33}$$

F-ölçüsü, pozitif öngörü değeri ve duyarlılık ölçülerinin harmonik ortalamasıdır ve her iki ölçünün birlikte incelenmesine olanak tanır (Glas ve ark.,2003; Kartal ve Balaban, 2015) ;

$$F = \frac{2 * \text{Pozitif Öngörü Değeri} * \text{Duyarlılık}}{\text{Pozitif Öngörü Değeri} + \text{Duyarlılık}} \tag{4.34}$$

Pozitif olabilirlik oranı, tahmin sonucunun gerçekte pozitif sınıf etiketinin varlığında pozitif çıkma olasılığının, negatif sınıf etiketi varlığında pozitif çıkma olasılığına oranıdır. Negatif olabilirlik oranı, tahmin sonucunun gerçekte pozitif sınıf etiketinin varlığında negatif çıkma olasılığının, negatif sınıf etiketi varlığında negatif çıkma olasılığına oranına negatif olabilirlik oranıdır (Flach, 2004; Glas ve ark., 2003 ; Kartal ve Balaban, 2015) ;

$$\text{Pozitif Olabilirlik Oranı} = \frac{\text{Duyarlık}}{1-\text{Belirleyicilik}} \tag{4.35}$$

$$\text{Negatif Olabilirlik Oranı} = \frac{1-\text{Duyarlık}}{\text{Belirleyicilik}} \tag{4.36}$$

4. BULGULAR

Botnet trafiği için gerçek network kullanmak networke zarar verebileceğinden, sanal ortamda bir network kurarak (demo ortamında) paket toplaması yapılmıştır. Bu paketlere ek olarak internet üzerinden hazır botnet paketleri de indirilip, diğer botnet paketlerine eklenmiştir. Sanal ortamın kurulması için Microsoft Windows 10 üzerinde kurulu gelen Hyper-v sanallaştırma ortamı kullanılmıştır. Bu sanal ortamda toplamda dört adet bilgisayar bulunmaktadır. Bu bilgisayarlardan üçü botnetli akışa sahipken biri Van Yüzüncü Yıl Üniversitesine ait normal ağ akışına sahiptir.

Network üzerindeki anormal aktiviteleri incelemek için network trafiğinin sniff (süzülmesi) edilmesi gerekmektedir. Network üzerinden paket toplamak için ve paket üzerinden filtrelemeler yapabilmek için Wireshark adlı network sniffing aracı kullanılmıştır. Bunun için ağ katmanı yani layer 2' de çalışılmıştır Sanal ortamda bulunan bilgisayarlara botnet malwareler bulaştırılıp, bu dört trafik arasındaki trafik 40 bin veri akışı toplanana kadar izlenmiştir. Normal trafik verisinin akışı için ise botnetsiz olan üniversiteye ait normal trafik akışı izlenmiştir.

Toplamda 40 bin botnet trafik paketi, 40 bin normal trafik paketi toplanıp makine öğrenmesi algoritmalarıyla sınıflandırılması gerçekleştirilmiştir. Ancak 1 gözlem botnet sınıfına, 1 gözlem de normal akış sınıfına ait olmak üzere, toplam iki gözlem isimleri dışında değişken içermediğinden analizden çıkarılmıştır ve analiz 79998 gözlem üzerinden yapılmıştır. Veri hazırlama aşamasında literatürde önemli olduğu düşünülen özellikler PHP yazılım dili yardımı ile MYSQL veri tabanına aktarılarak veriler burada düzenlenip (sayısal hale getirilip) Excell formatına çevrilmiştir. Uygulama Phyton'da gerçekleştirilmiştir. Uygulamaya ait değişkenler ve elde edilen bulgular bu bölümde verilmiştir.

Çalışmada Kullanılan Nitelikler: Özellik seçimi, sınıflandırıcı oluşturmada çok önemli bir rol oynamaktadır. Varolan veri kümesi, başlangıç zamanı, kaynak ve hedef bağlantı noktası belirtme sırasında kullanılan süre, ağ trafiğinde yer alan paket için kullanılan kaynak ve hedef IP adresi,varlıklar arasındaki etkileşimi belirleyen protokol, toplam bayt gibi temel özellikleri içerir. Bu özellikler, normal trafikten botnet trafiğini ayırt etmek için yeterli değildir. Bu nedenle, yüksek sınıflandırma doğruluğuna ulaşmak

için botnet ve normal trafiğin özelliklerini açıklayan özellikler seçilir (Kalaivani ve Vijaya, 2016).

Dolayısıyla, ortalama bayt oranı, ortalama paket hızı, paket boyutu gibi botnet trafiğini belirlemede önemli olan özellikler seçilmiştir (Kalaivani ve Vijaya, 2016).

Bu kısımda, veri setinde yer alan özellikler ve onların sayısallaştırma işlemlerine değinilmiş ve analize ilişkin bulgular sunulmuştur.

Ele alınan özellikler; Devam Süresi, Protokol Tipi, Kaynak ve Hedef IP, Kaynak ve Hedef Port, Syn Bayrak Durumu, Reset Bayrak Durumu, Ack Bayrak Durumu, Toplam Paket, Reset Bağlantı Sayısı, Ortalama Bayt, Ortalama Paket Oranı , Paket Boyutu ve Botnet'tir.

Devam süresi: Akışın süresi, akışı tamamlamak için alınan toplam süreyi gösterir. Bu süre, ortalama paket oranını ve ortalama bayt oranını hesaplamak için kullanılır (Kalaivani ve Vijaya, 2016).

Protokol: Protokol, iletişim kurduklarında telekomünikasyon bağlantı kullanımındaki noktaları sonlandıran özel kurallar kümesidir. Protokoller, iletişim kurucu varlıklar arasındaki etkileşimleri belirler.

Kullanılan farklı protokol türleri vardır (Kalaivani ve Vijaya,2016). Bu çalışmada yer alan protokoller; TCP, UDP, İCMPV6, İGMP ,DATA, İPV6,.HOPOPTSİ ve İCMP'dir.

Kaynak IP : İnternet akış alanlarının temel özelliklerinden biridir. Kaynak IP adresi kullanıcı bilgisayarının IP adresidir (Karasaridis ve ark.,2007). Kaynak IP adresi, daha fazla işlem için ondalık formata PHP yazılım dilinde IP2LONG (IP adreslerini sayısal değerlere dönüştürüp veritabanına kayıt işlemlerini yapan fonksiyondur) fonsiyonu kullanılarak dönüştürülmektedir.

Hedef IP: Hedef IP, bir mesajın gönderildiği IP adresidir. IP adresleri, bir ağ üzerinden veri paketlerini iletmek için kullanılır (Karasaridis ve ark., 2008; Kalaivani ve Vijaya, 2016). Hedef IP adresi, daha fazla işlem için ondalık formata için ondalık formata Php yazılım dilinde IP2LONG fonsiyonu kullanılarak dönüştürüldü.

Kaynak ve hedef port : Port numarası, verdiğimiz hizmeti veya uygulamayı tanımlamamıza, isteğimizin gönderilmesine ve yapılmasına izin veren önceden belirlenmiş numaradır. Port numaraları. saldırıları amaçlayan uzak sistemlere ilişkin bilgi edinmek için kullanılabilirler. 80, 53, 25 numaralı bağlantı noktası, farklı botnet

saldırıları türlerine sahip kötü amaçlı akış olarak işaretlenir; bunlar sırasıyla, HTTP tabanlı botnet, spam botnet ve DNS sunucu tabanlı botnettir (Karasaridis ve ark., 2008; Kalaivani ve Vijaya, 2016).

Bayraklar : Ağ akışını temsil eden SYN, RST, CON, ACK, FIN olmak üzere farklı bayrak türleri vardır (Kalaivani,Vijaya, 2016). Bunların herbirinin farklı bir durumu temsil eder (Kalaivani,Vijaya, 2016).

Toplam paket: Toplam paket özelliği, belirli bir akış sırasında aktarılan paketlerin sayısı olarak anlamına gelir. Belirli bir süre veya akışta iletilen paketlerin sayısı kaydedilir (Karasaridis ve ark., 2008).

Reset bağlantı sayısı : Bir akışın reset bağlantısı sayısı, sunucunun bağlantı yapmayı reddetmesidir. Reset bağlantısının sayısı, aynı IP adresinden tekrarlanan RST bağlantısını analiz ederek hesaplanır. Çok fazla RST bağlantısının alınması, alıcının enfekte olduğu anlamına gelmektedir (Kalaivani ve Vijaya, 2016).

Ortalama bayt : Ortalama bayt oranı, toplam bayt ve süre gibi var olan özellikler aracılığı ile hesaplanır.Ortalama bayt,belirli bir sürede akış içinde aktarılan ortalama baytları ifade eder (Karasaridis ve ark., 2008 ; Kalaivani ve Vijaya, 2016).

$$\text{Ortalama Bayt} = \frac{\text{Toplam bayt}}{\text{Devam Süresi}}$$

Burada toplam bayt: İstemcinin istek dahilinde gönderdiği toplam bayt sayısı anlamına gelir (Kalaivani ve Vijaya, 2016).

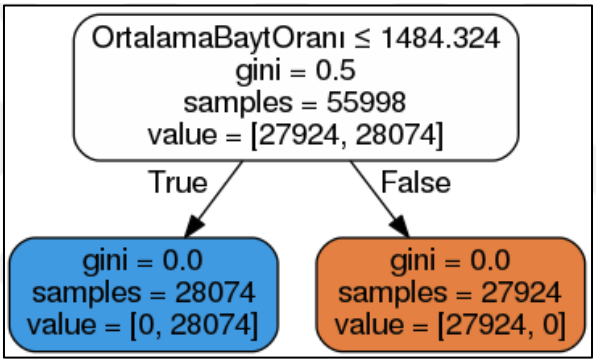
Ortalama paket oranı : Belirli bir zaman aralığındaki akışta aktarılan ortalama paket sayısıdır (Kalaivani ve Vijaya, 2016; Chen ve ark., 2017).

$$\text{Ortalama Paket Oranı} = \frac{\text{Toplam paket}}{\text{Devam Süresi}}$$

Paket boyutu: Belirli bir zaman aralığındaki akışta aktarılan paketin bayt cinsinden boyutudur (Kalaivani ve Vijaya, 2016; Chen ve ark., 2017).

4.1. Sınıflandırma Ağacı Yöntemine İlişkin Bulgular

Öncelikle Botnet Sınıflandırma problemimizde ele aldığımız açıklayıcı değişkenler; Devam Süresi, Protokol Tipi, Kaynak IP, Hedef IP, Kaynak Port, Hedef Port, Syn Bayrak Durumu, Reset Bayrak Durumu, Ack Bayrak Durumu, Toplam Paket, Reset Bağlantı Sayısı, Ortalama Bayt, Ortalama Paket Oranı ve Paket Boyutudur. Yanıt değişkenimiz ise botnettir. Sınıflandırma Ağacı oluşturmak için analize tüm veri setinden başlanarak performans sonuçları incelenmiş, ağacın önemli bulduğu ve ağaç inşasında tek bir bölünmeyle yanlılığa sebep olan değişkenler tek tek çıkarılarak doğruluktaki azalma ele alınmıştır. Bu kapsamda Phyton Programlama Dilinden elde edilen bulgular;



Şekil 4.1. Tüm değişkenler analize dahil edildiğinde oluşan sınıflandırma ağacı.

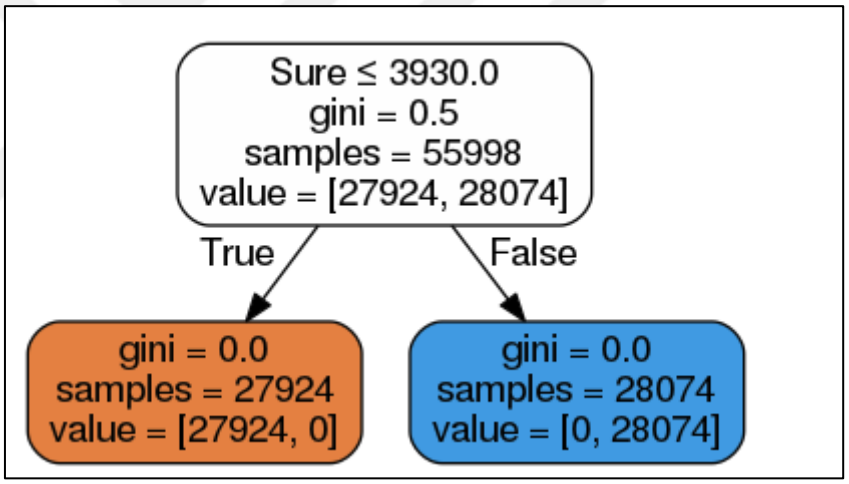
Ağacın ilk bölünmesi Ortalama Bayt Oranı ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5 tir. Ortalama Bayt Oranı 1484.324 değerinden küçükse Botnet olmayan örnek sayısı 27924 olurken, Botnet olan örnek sayısı 28074 dür.

Çizelge 4.1. Tüm değişkenler analize dahil edildiğinde sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
TAHMİN	Botnet Değil	12075	0	12075
	Botnet	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.2. Tüm değişkenler analize dahil edildiğinde sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 100	1	1	0	0	1	1	1



Şekil 4.2. Ortalama bayt oranı değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

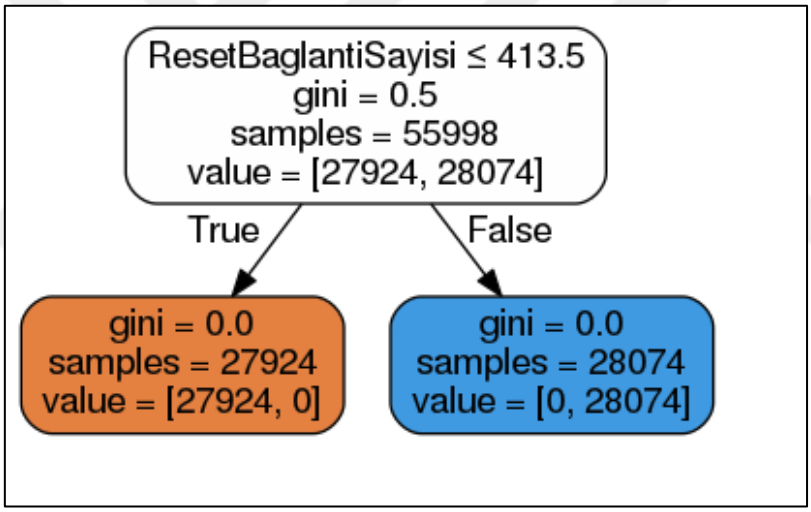
Ortalama Bayt Oranı analizden çıkarıldığında ağacın ilk bölünmesi “Süre” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5’tir. Süre değişkeni 3930 değerinden küçük olduğunda Botnet olmayan örnek sayısı 27924 olurken, Botnet olan örnek sayısı 28074 dür.

Çizelge 4.3. Ortalama bayt oranı değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		POZİTİF	NEGATİF	TOPLAM
Tahmin	POZİTİF	12075	0	12075
	NEGATİF	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.4. Ortalama bayt oranı değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 100	1	1	0	0	1	1	1



Şekil 4.3. Süre değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

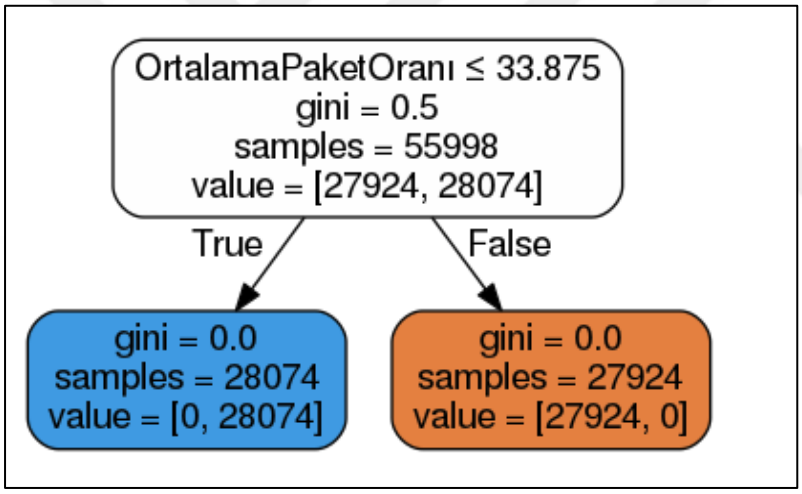
Süre değişkeni analizden çıkarıldığında ağacın ilk bölünmesi “Reset Bağlantı Sayısı” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5’tir. Ortalama Bayt Oranı 413.5 değerinden küçük olduğunda Botnet olmayan örnek sayısı 27924 olurken, Botnet olan örnek sayısı 28074’dür.

Çizelge 4.5. Süre değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.6. Süre değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 100	1	1	0	0	1	1	1



Şekil 4.4. Reset bağlantı sayısı değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

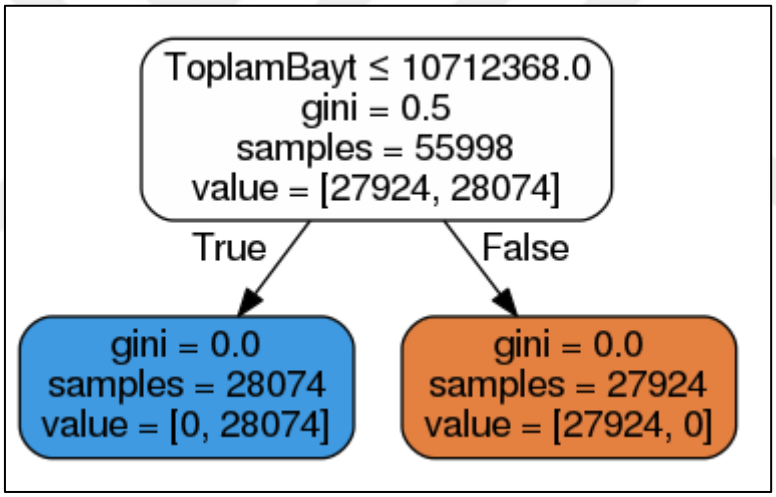
Reset Bağlantı Sayısı analizden çıkarıldığında ağacın ilk bölünmesi “Ortalama Paket Oranı” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5’tir. Ortalama Bayt Oranı 33.875 değerinden küçük olduğunda botnet olmayan örnek sayısı 27.924 olurken, botnet olan örnek sayısı 28.074 dür.

Çizelge 4.7. Reset bağlantı sayısı değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.8. Reset bağlantı sayısı değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 100	1	1	0	0	1	1	1



Şekil 4.5. Ortalama paket değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

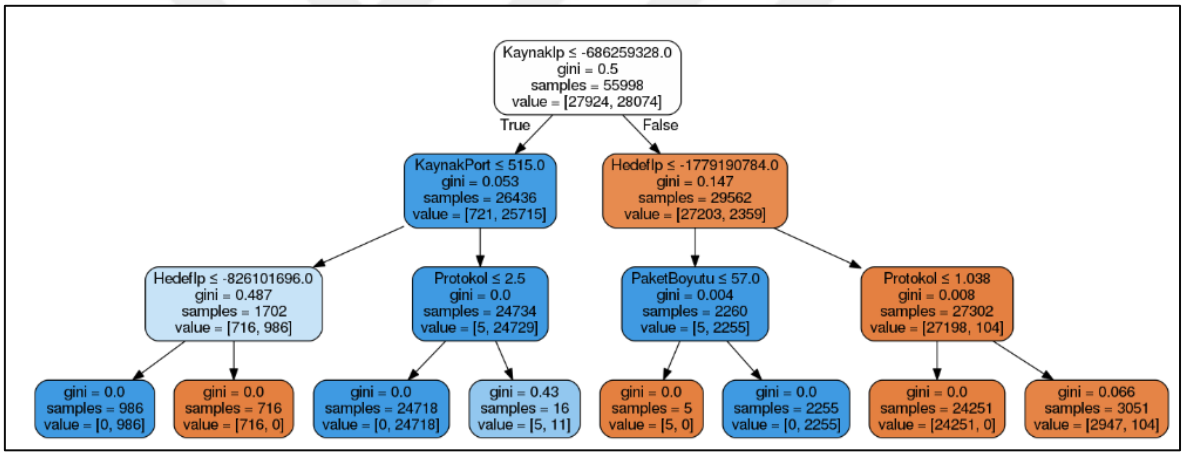
Ortalama Paket Oranı analizden çıkarıldığında ağacın ilk bölünmesi “Toplam Bayt” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5’tir. Ortalama Bayt Oranı 10712368 değerinden küçük olduğunda botnet olmayan örnek sayısı 27924 olurken, botnet olan örnek sayısı 28074 dür.

Çizelge 4.9. Ortalama paket değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.10. Ortalama paket değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 100	1	1	0	0	1	1	1



Şekil 4.6. Toplam bayt değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Toplam Bayt değişkeni analizden çıkarıldığında ağacın ilk bölünmesi “Kaynak IP” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5 tir. Kaynak IP değişkeninin değeri -686259328 değerinden küçük olduğunda Kaynak Port düğümüne ulaşılır. Kaynak Port düğümünde 25715 gözlem botnet olurken, 721 gözlem botnetli değildir. Kaynak Port değeri 515.0 dan küçükse Hedef Ip düğümüne, büyükse Protokol Düğümüne ulaşılır. Hedef Ip değeri -826101696’dan küçükse botnetli gözlem sayısı 716 olurken botnetsiz gözlem sayısı 986 dır. Diğer yandan, Kaynak IP değeri -686259328 değerinden küçük olmadığında, Hedef IP düğümüne ulaşılır. Bu düğümde 27203 botnet olmayan gözlem varken, 2359 botnetli

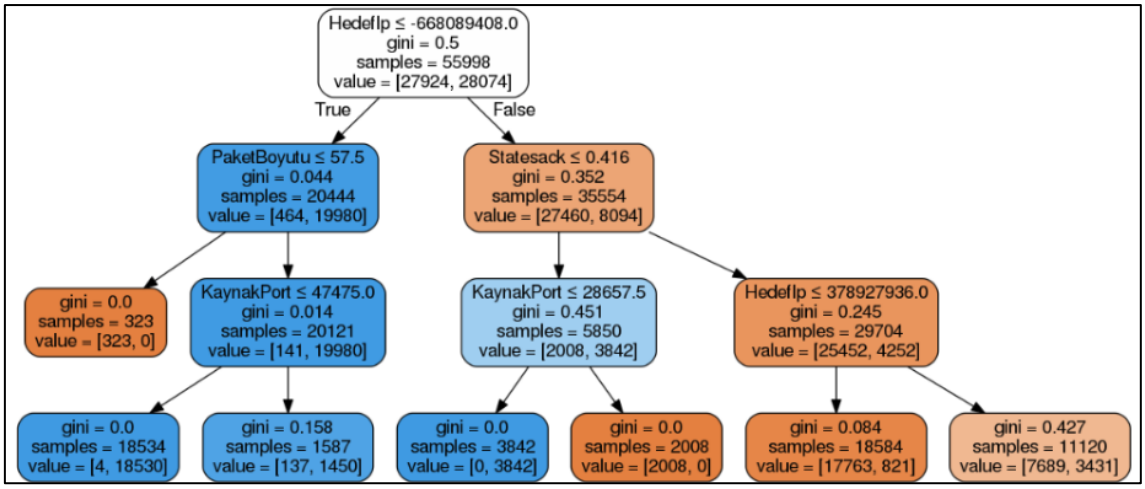
gözlem vardır. Hedef Ip değeri -1779190784 değerinden küçükse “Paket Boyutu” düğümüne ulaşılır. Bu düğümde 2260 gözlemden 5 tanesi botnet olmayan gözlemken, 2255 gözlem botnetli gözlemlerdir. Hedef IP değeri -1779190784 değerinden küçük değilse “Protokol” düğümüne ulaşılır. 27198 gözlem botnetli olmazken 104 gözlem botnetlidir.

Çizelge 4.11. Toplam bayt değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.12. Toplam bayt değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 100	1	1	0	0	1	1	1



Şekil 4.7. Kaynak IP değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Kaynak Ip değişkeni analizden çıkarıldığında ağacın ilk bölünmesi “Hedef IP” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5 tir. Hedef IP değişkeninin değeri -668089408 değerinden küçük olduğunda “Paket

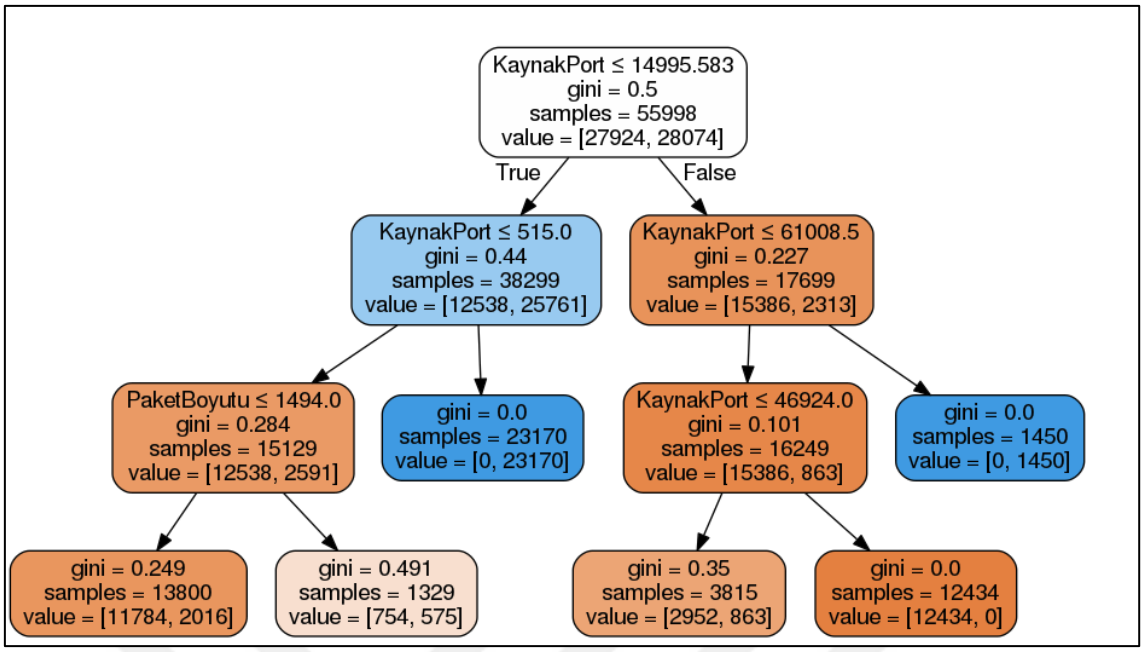
Boyutu" düğümüne ulaşılır. Paket boyutu düğümünde 20444 gözlemden 19980 gözlem botnet olurken, 464 gözlem botnetli değildir. Paket boyutu değeri 57.5 den küçükse 323 gözlem botnetli olmadan terminal düğüm olmuştur,. Paket boyutu değeri 57.5 den büyükse "Kaynak Port" düğümüne ulaşılır. Kaynak port düğümünde 20121 gözlem vardır bunların 141'i botnetli gözlem olmazken, 19980 gözlem botnetlidir. Kaynak port değeri 47475 değerinden küçükse botnetli gözlem sayısı 4 olurken botnetsiz gözlem sayısı 18530 dur. Diğer yandan, hedef IP değeri -668089408 değerinden küçük olmadığında, "Statesack" düğümüne ulaşılır. Bu düğümde 27203 botnet olmayan gözlem varken, 2359 botnetli gözlem vardır. Hedef IP değeri -1779190784 değerinden küçükse "Paket Boyutu" düğümüne ulaşılır. Bu düğümde 35554 gözlemden 27460 tanesi botnet olmayan gözlemken, 8094 gözlem botnetli gözlemlerdir. Statesack değeri 0.416 değerinden küçükse "Kaynak Port", değilse "Hedef IP" düğümüne ulaşılır ve Kaynak port düğümünde 2008 botnet olmayan gözlem varken, 3842 botnetli gözlem vardır. Hedef IP düğümünde 25452 botnet olmayan gözlem varken, 4252 botnetli gözlem vardır.

Çizelge 4.13. Kaynak IP değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12813	62	12075
	Botnet	1827	11925	11925
	TOPLAMI	14640	11987	24000

Çizelge 4.14. Kaynak IP değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 92.129	0.875	0.993	0.007	0.125	0.995	0.846	0.93



Şekil 4.8. Hedef IP değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

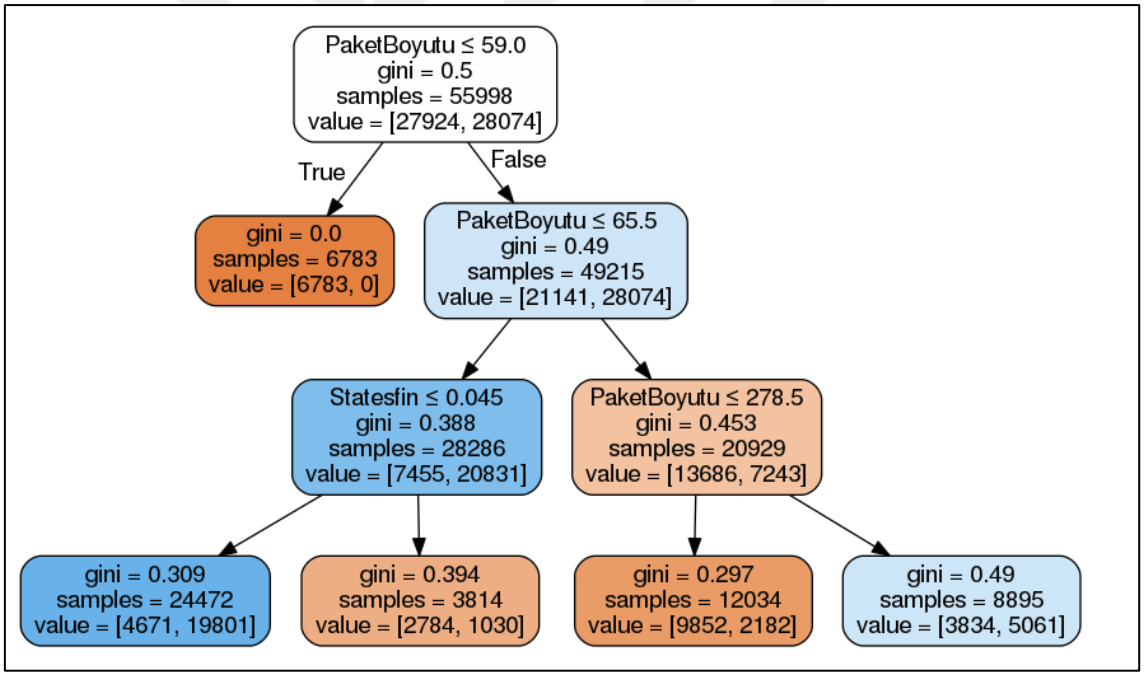
Hedef IP değişkeni analizden çıkarıldığında ağacın ilk bölünmesi “Kaynak Port” ile başlamıştır. Eğitim seti 55.998 gözlem içermektedir. Gini indeksinin kesim noktası 0.5’tir. Kaynak IP değişkeninin değeri -686259328 değerinden küçük olduğunda “Kaynak Port” düğümüne ulaşılır. Kaynak port düğümünde 25715 gözlem botnet olurken, 721 gözlem botnetli değildir. Kaynak port değeri 515.0 dan küçükse “Hedef IP” düğümüne, büyükse “Protokol” düğümüne ulaşılır. Hedef IP değeri -826101696’ dan küçükse botnetli gözlem sayısı 716 olurken botnetsiz gözlem sayısı 986 dır. Diğer yandan, Kaynak IP değeri -686259328 değerinden küçük olmadığında, “Hedef IP” düğümüne ulaşılır. Bu düğümde 27203 botnet olmayan gözlem varken, 2359 botnetli gözlem vardır. “Hedef IP” değeri -1779190784 değerinden küçükse “Paket Boyutu” düğümüne ulaşılır. Bu düğümde 2260 gözlemden 5 tanesi botnet olmayan gözlemken, 2255 gözlem botnetli gözlemlerdir. Hedef IP değeri -1779190784 değerinden küçük değilse “Protokol” düğümüne ulaşılır. 27198 gözlem botnetli olmazken 104 gözlem botnetlidir.

Çizelge 4.15. Hedef IP değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12813	62	12075
	Botnet	1827	11925	11925
	TOPLAMI	14640	11987	24000

Çizelge 4.16. Hedef IP değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 93.916	0.892	1	0	0.108	1	0.877	0.94



Şekil 4.9. Kaynak port değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Kaynak port değişkeni analizden çıkarıldığında ağacın ilk bölünmesi "Paket Boyutu" ile başlamıştır. Paket boyutu 59 değerinden küçükse 6783 gözlem botnetsiz olarak terminal düğüm olmuştur. Paket boyutu 59 değerinden büyükse 21141 botnetsiz gözlem varken, 28074 botnetli gözlem vardır. Paket boyutu 65.5 değerinden küçükse "Statisfin" düğümüne ulaşılır, değilse "Paket Boyutu" düğümüne ulaşılır ve "Statesfin"

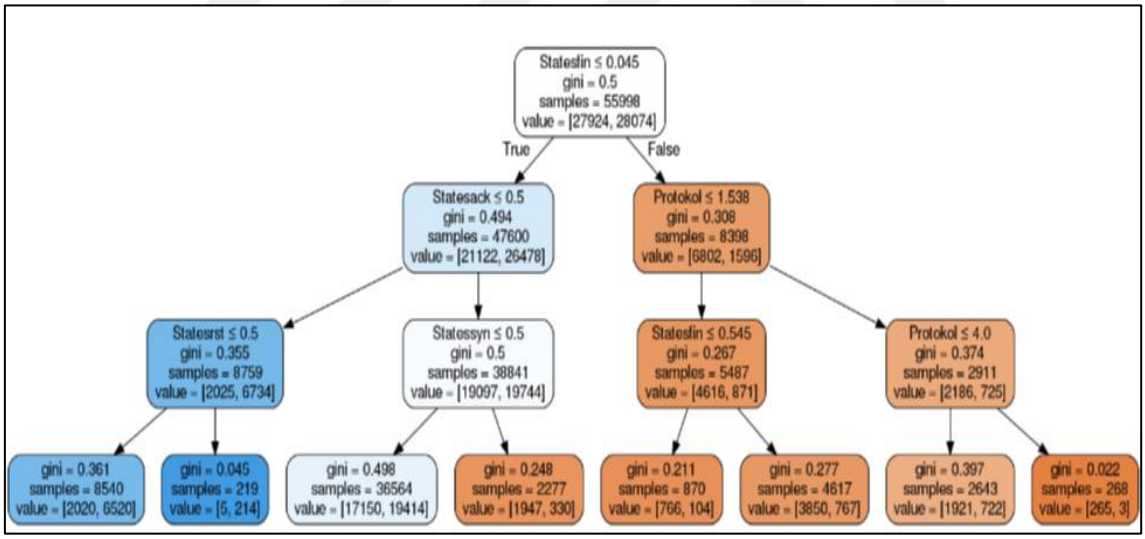
düğümünde 7455 gözlem botnetsiz olurken 20831 gözlem botnetlidir. Paket boyutu ise 278.5 değerinden küçükse 13686 botnetsiz, 7243 botnetli gözlem vardır.

Çizelge 4.17. Kaynak port değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12813	62	12075
	Botnet	1827	11925	11925
	TOPLAMI	14640	11987	24000

Çizelge 4.18. Kaynak port değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 78.383	0.854	0.735	0.265	0.146	0.687	0.881	0.78



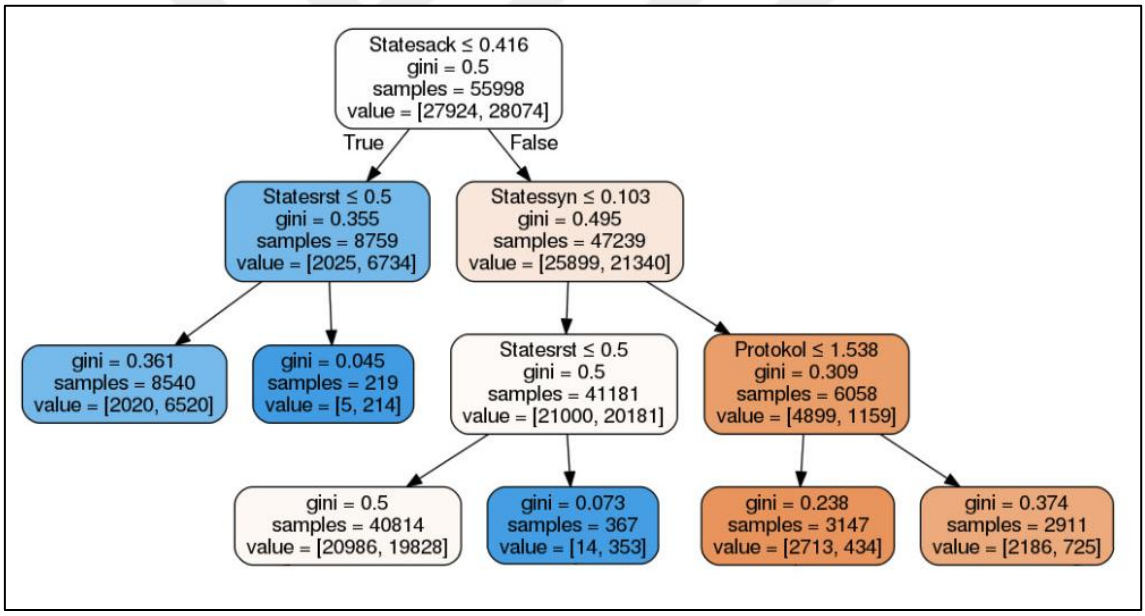
Şekil 4.10. Paket boyutu değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Çizelge 4.19. Paket boyutu değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	3866	8209	12075
	Botnet	857	11068	11925
	TOPLAMI	4723	19277	24000

Çizelge 4.20. Paket boyutu değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 62.224	0.818	0.574	0.426	0.182	0.242	0.928	0.58



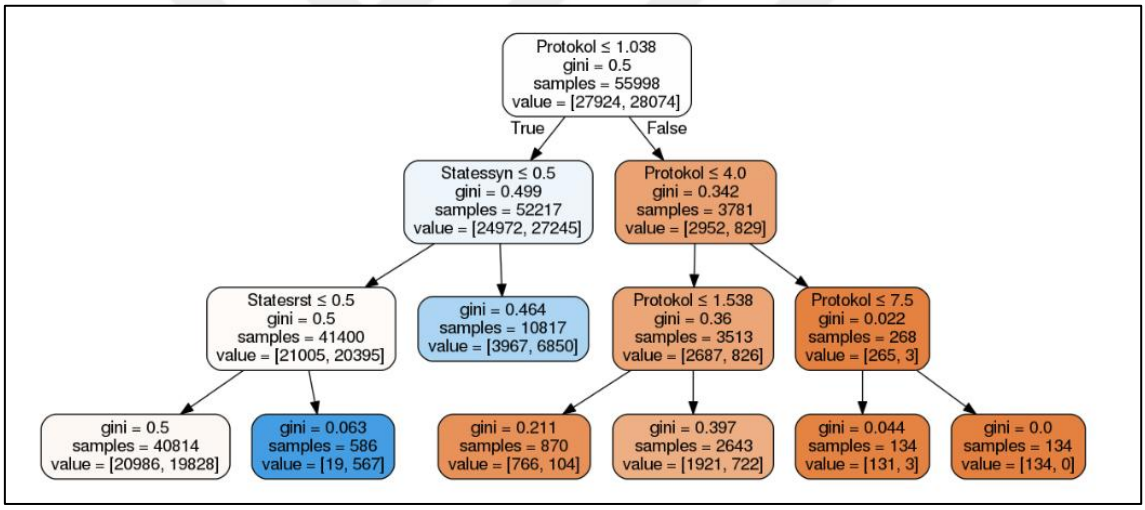
Şekil 4.11. Statesfin değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Çizelge 4.21. Statesfin değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	11231	844	12075
	Botnet	8924	3001	11925
	TOPLAMI	20155	3845	24000

Çizelge 4.22. Statesfin değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 59.3	0.557	0.00026	0.999	0.443	0.930	0.251	0.54



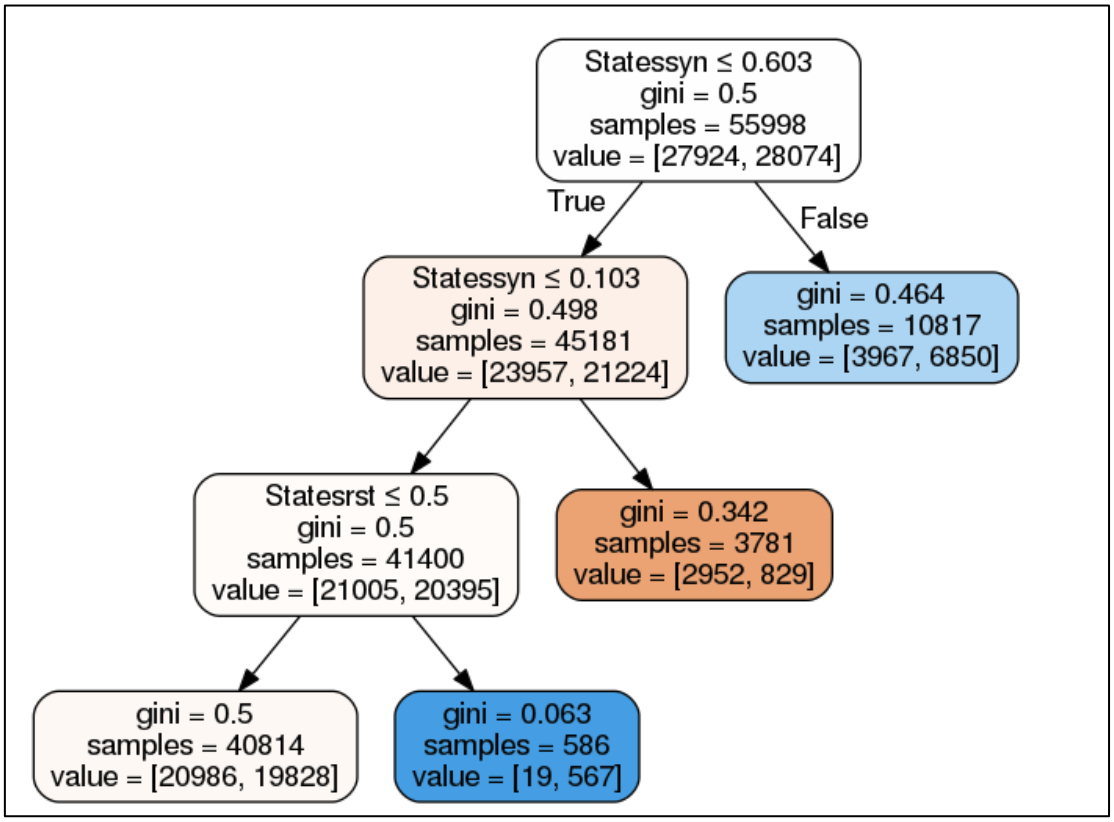
Şekil 4.12. Statesack değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Çizelge 4.23. Statesack değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	10367	1708	12075
	Botnet	8788	3137	11925
	TOPLAMI	19155	4845	24000

Çizelge 4.24. Statesack değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
56.26	0.541	0.647	0.353	0.459	0.858	0.263	0.52



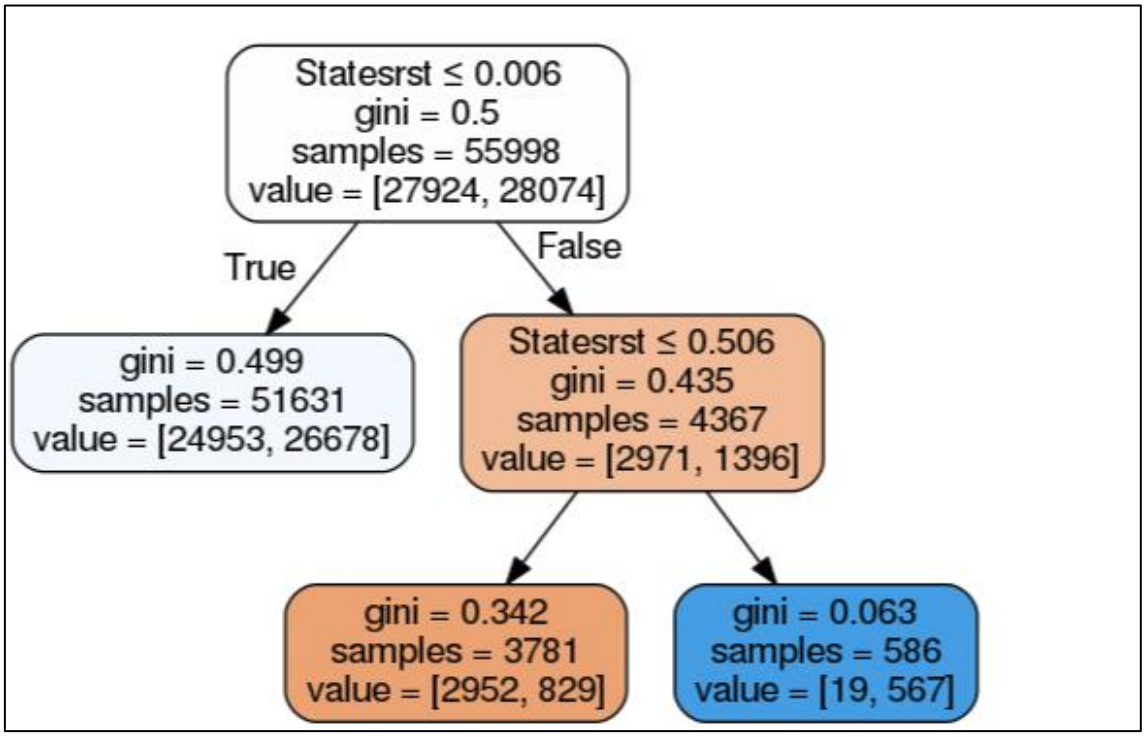
Şekil 4.13. Protokol değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Çizelge 4.25. Protokol değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	10367	1708	12075
	Botnet	8788	3137	11925
	TOPLAMI	19155	4845	24000

Çizelge 4.26. Protokol değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 56.26	0.0006	0.517	0.483	0.999	0.106	0.96	0.52



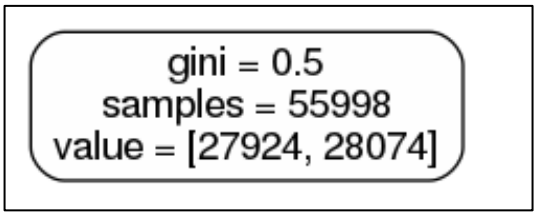
Şekil 4.14. Statesyn değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Çizelge 4.27. Statesyn değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	1280	10795	12075
	Botnet	366	11559	11925
	TOPLAMI	1646	22354	24000

Çizelge 4.28. Statessyn değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 53.495	0	0.49	0.51	1	0	1	0.43



Şekil 4.15. Staterst değişkeni analizden çıkarıldığında oluşan sınıflandırma ağacı.

Çizelge 4.29. Staterst değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	0	12075	12075
	Botnet	0	11925	11925
	TOPLAMI	0	24000	24000

Çizelge 4.30. Staterst değişkeni analizden çıkarıldığında sınıflandırma ağacına ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV	F
% 49.68	0	0.49	0.51	1	0	0	0.33

4.2. Rastgele Orman Yöntemine İlişkin Bulgular

Rastgele Orman Analizinde, tıpkı Sınıflandırma ve Regresyon Ağaçlarında yapılan analizdeki gibi önce bütün değişkenler analize dahil edilmiştir, sonrasında sırasıyla; Ortalama Bayt Oranı, Süre, Reset Bağlantı Sayısı, Ortalama Paket Oranı, Toplam Bayt, Kaynak Ip, Hedef Ip, Kaynak Port, Paket Boyutu, Statisfin, Statesack, Protokol, Statissyn ve Statesrst değişkenleri analizden çıkarılarak oluşturulan modellerin performansları bazı özellikler bakımından degerlendirilmiştir.

Ortalama Bayt Oranı, Süre, Reset Bağlantı Sayısı, Ortalama Paket Oranı değişkenleri teker teker çıkarıldığında Hata Matrisi hep aynı sonucu vererek Doğru Sınıflandırma Oranı % 100 olarak bulunmuştur.

Çizelge 4.31. Tüm değişkenler analize dahil edildiğinde ve ortalama bayt oranı, süre, reset bağlantı sayısı, ortalama paket oranı değişkenleri teker teker çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	0	11925	11925
	TOPLAMI	12075	11925	24000

Çizelge 4.32. Tüm değişkenler analize dahil edildiğinde ve ortalama bayt oranı, süre, reset bağlantı sayısı, ortalama paket oranı değişkenleri teker teker çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 100	1	1	0	0	1	1

Çizelge 4.33. Toplam bayt değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	34	11891	11925
	TOPLAMI	12109	11891	24000

Çizelge 4.34. Toplam bayt değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 99.858	0.997	1	0	0.003	1	0.997

Çizelge 4.35. Kaynak IP değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12075	0	12075
	Botnet	35	11890	11925
	TOPLAMI	12110	11890	24000

Çizelge 4.36. Kaynak IP değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 99.854	0.997	1	0	0.003	1	0.997

Çizelge 4.37. Hedef IP değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	12036	39	12075
	Botnet	806	11119	11925
	TOPLAMI	12842	11158	24000

Çizelge 4.38. Hedef IP değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 96.47	0.937	0.996	0.004	0.063	0.996	0.932

Çizelge 4.39. Kaynak port değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	9355	2720	12075
	Botnet	606	11319	11925
	TOPLAMI	9961	14099	24000

Çizelge 4.40. Kaynak port değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 86.141	0.93	0.806	0.194	0.07	0.774	0.949

Çizelge 4.41. Paket boyutu değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	3866	8209	12075
	Botnet	857	11068	11925
	TOPLAMI	4723	19277	24000

Çizelge 4.42. Paket boyutu değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 62.224	0.818	0.57	0.43	0.182	0.320	0.928

Çizelge 4.43. Statesfin değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	11231	844	12075
	Botnet	8924	3001	11925
	TOPLAMI	20155	3845	24000

Çizelge 4.44. Statesfin değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 59.3	0.557	0.00026	0.999	0.443	0.930	0.251

Çizelge 4.45. Statesack değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	10367	1708	12075
	Botnet	8788	3137	11925
	TOPLAMI	19155	4845	24000

Çizelge 4.46. Statesack değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 56.26	0.541	0.647	0.353	0.459	0.858	0.263

Çizelge 4.47. Protokol değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	10367	1708	12075
	Botnet	8788	3137	11925
	TOPLAMI	19155	4845	24000

Çizelge 4.48. Protokol değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 56.26	0.541	0.647	0.353	0.459	0.858	0.263

Çizelge 4.49. Statessyn değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	1280	10795	12075
	Botnet	366	11559	11925
	TOPLAMI	1646	22354	24000

Çizelge 4.50. Statessyn değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 53.495	0.0006	0.517	0.483	0.999	0.106	0.96

Çizelge 4.51. Staterst değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hata matrisi.

		GERÇEK		
		Botnet Değil	Botnet	TOPLAM
Tahmin	Botnet Değil	0	12075	12075
	Botnet	0	11925	11925
	TOPLAMI	0	24000	24000

Çizelge 4.52. Statesrt değişkeni analizden çıkarıldığında rastgele orman yöntemine ilişkin hesaplanan bazı model performans ölçütleri.

Doğruluk	TPR	SPC	FPR	FNR	PPV	NPV
% 49.68	0	0.49	0.51	1	0	0

5. TARTIŞMA ve SONUÇ

Siber güvenlik alanında önemli bir yere sahip olan botnet tespiti ve sınıflandırılması için, botnet akışı ile normal akışı bibirinden ayırt edeceği düşünülen değişkenler ele alınarak, Sınıflandırma ve Regreyon Ağaçları ile Rastgele Orman Yöntemleri ile analiz edilmiştir.

Sınıflandırma Ağacında önemli bulunan değişkenler ağacın ilk bölünmesine sebep olan değişkenler olduğu için bir sonraki ağacı oluşturmada bu değişken analizden çıkarılmıştır. Böylece ağacın bir tek değişkene göre bölünmesine bağlı olarak ortaya çıkan yanlılık (% 100 doğruluk), değişkenleri tek tek analizden çıkarılarak azaltılmaya çalışılmıştır. Her bir değişken çıkarıldıktan sonra modelde oluşan değişimler bazı model performans ölçütleri ile hesaplanmıştır. Buradaki amaç, ağ akışının botnetli mi yoksa normal mi olduğuna karar verirken önemli olan özelliklerin belirlenmesi ve seçimi ve performansa katkısını ölçmektir.

Yapılan iki analiz sonucu da Ortalama Bayt Oranı, Süre, Reset Bağlantı Sayısı, Ortalama Paket Oranı değişkenleri Botnetli bir trafiği belirlemede % 100 başarı gösterdiği için önemli bulunmuştur, bu değişkenlerden her biri modelden çıkarıldığında sınıflandırma modelinin doğruluğunda bir azalma olmamıştır. Dolayısıyla bu çalışmada kullanılan ağ akışını botnetli mi, normal mi olup olmadığını ayırt etmek için Ortalama Bayt Oranı, Süre, Reset Bağlantı Sayısı, Ortalama Paket Oranı değişkenlerinden birini baz almak yeterli olacaktır.

Ağacın sadece bir düğüm oluşturmasına sebep olan değişken, yanıt değişkeni ile yüksek korelasyonlu olduğundan diğer değişkenleriniz ne olursa olsun, ağacınız her zaman bu değişkene ait bir düğümle sonuç verecektir. Bu çalışmada bu sorunun çözümü için değişkenleri tek tek analizden çıkarmak uygun bulunmuştur. Ayrıca her bir değişken çıkarıldığında sınıflandırmaya ait doğrulukta meydana gelen azalma, o değişkenin modele ne kadar katkısı olduğuna dair bilgiler vermektedir. Aşağıda, her iki yöntem, önemli görülen her bir değişken çıkarıldıktan sonra hesaplanan bazı model performans ölçülerinin sonuçları bakımından kıyaslanarak aktarılmıştır.

Sınıflandırma Ağacı Yöntemi sonucunda elde edilen bulgular ışığında; tüm değişkenlerden sırasıyla Ortalama Bayt Oranı, Süre, Reset Bağlantı Sayısı, Ortalama Paket Oranı, Toplam Bayt, değişkenleri analizden çıkarılsa da analizin doğru sınıflandırma oranı % 100 olarak kalmaya devam ederken, yanlış pozitif ve yanlış negatif oranı sıfır olarak elde edilmiştir.

Rastgele Orman Yönteminde ise Ortalama Bayt Oranı, Süre, Reset Bağlantı Sayısı, Ortalama Paket Oranı değişkenleri teker teker çıkarıldığında hata Matrisi hep aynı sonucu vererek doğru sınıflandırma oranı % 100, yanlış pozitif ve yanlış negatif oranı sıfır olmuştur.

Sınıflandırma ve Regresyon Ağaçlarında Toplam Bayt değişkeni analizden çıkarıldığında doğruluktan pay düşmezken, Rastgele Ormanda bu oran 99.858 olmuştur. Kaynak IP değişkeni analizden çıkarıldığında Sınıflandırma Ağacının doğru sınıflandırma oranı % 92.129 olurken yanlış pozitif oranı 0.007 olurken yanlış negatif oranı 0.125 dir. Kaynak IP değişkeni analizden çıkarıldığında Rastgele Orman Yönteminin doğru Sınıflandırma Oranı % 99.854 olurken yanlış pozitif oranı 0 ve yanlış negatif oranı 0.003'tür. Bu değişkene ilişkin Rastgele Orman Yönteminin daha iyi bir performans sergilediğini görülmüştür.

Hedef IP değişkeni analizden çıkarıldığı zaman, Sınıflandırma Ağacı Yöntemi sonucu doğruluk, % 93.916 olurken yanlış pozitif oranının 0, yanlış negatif oranın da 0.108'dir. Hedef IP değişkeni analizden çıkarıldığı zaman, Rastgele Orman Yönteminin % 96.47 Doğruluk, 0.004 yanlış pozitif oranı ve 0.063 yanlış negatif oranı ile daha üstün bir performans sergilediğini söylemek mümkündür.

Kaynak Port değişkeni analizden çıkarıldığında, Sınıflandırma Ağacı Yöntemi sonucu doğruluk, % 92.129 olurken yanlış pozitif oranının 0.007, yanlış negatif oranın da 0.125 olmuştur. Kaynak Port değişkeni analizden çıkarıldığında, Rastgele Orman Yönteminin % 86.141 Doğruluk, 0.007 yanlış pozitif oranı ve 0.774 yanlış negatif oranı ile Sınıflandırma Ağacı Yöntemine göre daha zayıf bir performans sergilediğini söylenebilir.

Paket Boyutu değişkeni analizden çıkarıldığında, Sınıflandırma Ağacı Yöntemi sonucu Doğruluk, % 78.383 e düşerken, yanlış pozitif oranı 0.265 e, yanlış negatif oranın da 0.146 ya yükselmiştir. Paket Boyutu değişkeni analizden çıkarıldığı zaman, Rastgele Orman Yönteminin % 62.224 doğruluk, 0.43 yanlış pozitif oranı ve 0.182

yanlış negatif oranı ile Sınıflandırma Ağacı Yöntemine göre daha zayıf bir performans sergilemiştir.

Statisfin değişkeni analizden çıkarıldığında Sınıflandırma Ağacında doğruluk hızla % 59.3e kadar düşerken yanlış pozitif oranı 0.443 ve Yanlış Negatif Oranı da 0.930 a yükselmiştir. Aynı değişkene ait Rastgele Orman Yönteminin sonucuna bakıldığında ise doğruluğun % 59.3 olduğunu Yanlış Pozitif Oranının 0.999 ve Yanlış Negatif Oranının da 0.443 olduğunu, dolayısıyla her ne kadar aynı Sınıflandırma Doğruluğuna sahip olsalar bile diğer ölçümler bakımından her iki yönteme ait sonuçların farklı olabileceği görülmüştür.

Statesack değişkeni analizden çıkarıldığında Sınıflandırma Ağacında doğruluk hızla % 56.26 olurken yanlış pozitif oranı 0.459 ve yanlış negatif oranı da 0.858 olmuştur. Aynı değişkene ait Rastgele Orman Yönteminin sonucuna bakıldığında ise Doğruluğun % 59.26 olduğu yanlış pozitif Oranı 0.353 ve yanlış negatif oranının da 0.459 olduğu tespit edilmiştir. Yine buradan anlaşılacağı üzere her ne kadar aynı doğruluğa sahip olurlarsa olsunlar her iki yöntemi performans bakımından kıyaslamada kullanılan diğer ölçümlerin, daha detaylı fikir sahibi olmamız için önemi ön plana çıkmaktadır.

Protokol değişkeni analizden çıkarıldığında Sınıflandırma Ağacında doğruluk hızla % 56.26 olurken yanlış pozitif oranı 0.483 ve yanlış negatif oranı da 0.999 olarak olmuştur. Aynı değişkene ait Rastgele Orman Yönteminin sonucuna bakıldığında ise doğruluğun % 56.26 olduğunu yanlış pozitif oranının 0.353 ve yanlış negatif oranının da 0.459 olduğu tespit edilmiştr.

Statissyn değişkeni analizden çıkarıldığında Sınıflandırma Ağacında doğruluk hızla % 53.495 olurken yanlış pozitif oranı 0.51 ve yanlış negatif oranı da 1 olarak hesaplanılmıştır. Aynı değişkene ait Rastgele Orman Yönteminin sonucuna bakıldığında ise Doğruluğun % 53.495 olduğunu yanlış pozitif oranı 0.483 iken yanlış negatif oranı 0.999 olarak elde edilmiştir.

Son olarak Statesrst değişkeni analizden çıkarıldığında Sınıflandırma Ağacında doğruluk % 49.68 olurken yanlış pozitif oranı 0.51 ve yanlış negatif oranı da 1 olarak hesaplanılmıştır. Aynı değişkene ait Rastgele Orman Yönteminin sonucuna bakıldığında ise doğruluğun % 49.68 olduğunu yanlış pozitif oranı 0.51 iken yanlış negatif oranı

1'dir. Yani her iki yöntemde tüm performans ölçütlerinde ilk kez aynı sonuçları verdiği belirlenmiştir.

Kalaivani ve Vijaya, (2016) çalışmalarında botnet tespiti için bu çalışmadakine benzer açıklayıcı değişkenler kullanmışlardır ve Makine Öğrenmesi Tekniklerinden, Bayes sınıflandırıcı, NN sınıflandırıcı, Karar ağacı ve SVM sınıflandırıcı kullanarak 10 kat çapraz geçerlik kullanarak SVM sınıflandırıcı algoritmasının % 99.8 doğruluk yakaladığını belirtmişlerdir.

Buczak ve Guven (2016), çalışmalarında, Kaynak IP adresi, Hedef IP adresi, IP Protokolü, Kaynak Bağlantı Noktası, Hedef Bağlantı Noktası ve IP Hizmet Türü açıklayıcı değişkenlerini kullanarak, ağ trafiğine ait akışın Makine Öğrenmesi ve Veri Madenciliği teknikleri ile sınıflandırılabileceğini vurgulamışlardır.

Ryu ve Yang (2018), çalışmalarında Gaussian Naive Bayes, Sinir Ağları ve Karar Ağacı algoritmalarının yanı sıra topluluk yöntemlerinden Voting, Adaboosting, ve Bagging'i kullanarak botnet tespiti ve sınıflandırması yapmışlardır. Karar ağaçları, Sinir Ağları algoritmasından daha hızlı ve daha yüksek doğruluk sağlarken; Gausian Naive Bayes en hızlı çalışan algoritma olmasına rağmen, doğruluk veri kümesine göre değiştiğini vurgulamışlardır. Ortak beklentiden farklı olarak, doğruluğu artırmak üzere botnet tespiti için makine öğrenme algoritmaları üzerinde topluluk metotlarının kullanılmasının ise tercih edilmemesi gerektiğini çünkü çok daha fazla zaman harcayarak daha kesin sonuç vermediğini belirtmişlerdir.

Mathur ve ark. (2018), çalışmalarında, Akış Başlangıç Zamanı, Akış Bitiş Zamanı, Akış Süresi, Kaynak IP Adresi, Hedef IP Adresi, Kaynak Portu, Hedef Portu, Protokol , Flg Bayrakları, Giriş Paketleri ve Giriş Baytları açıklayıcı değişkenlerini kullanarak botnet sınıflandırması problemini çözmek için Lojistik Regresyon, Random SubSpace, Randomizable Filtered Classifier, Multiclass Classifier ve Random Committee yöntemlerini kullanmışlardır ve Lojistik Regresyon tekniğini yüksek doğruluk ve hıza sahip olduğu için önermişlerdir.

Feizollah ve ark. (2013), çalışmalarında Makine Öğrenmesi Tekniklerinden Naive Bayes, K-nearest Neighbor, Decision Tree, Multi-Layer Perceptron ve Support Vector Machine (SVM) kullaranak botnetli ağ akışını tespit etmeye çalışmışlardır. 10 kat çapraz geçerlik tekniği kullanıldığında % 99.94 doğru pozitif oranı ile K-nearest Neighbor Tekniği diğer tekniklerin önüne geçtiğini vurgularken, % 30 test seti ile

oluşturulan sınıflandırma modelinde % 97.33 doğru pozitif oranı ile Multi-Layer Perceptron ön plana çıktığını belirtmişlerdir.

Buradan hareketle botnet tespitinde Makine Öğrenmesi Tekniklerinin başarılı bir şekilde sınıflandırma yaptığı ve botnet tespitinde kullanılan açıklayıcı değişkenlerin, çalışmalarda aşağı yukarı benzer değişkenler olduğu söylenebilir.

Çalışmada, Yüzüncü Yıl Üniversitesinin ağ akışından ve sanal ortamdan topladığımız normal ve botnetli akışı ayırt etmede belirleyici olarak düşünülen değişkenlere Sınıflandırma Tekniklerinden iki yöntem olan Sınıflandırma ve Regresyon Ağaçları ile Rastgele Orman uygulanmıştır

. Sınıflandırma Ağaçları ve Regresyon Ağaçları ile toplanan tüm akış analiz edildiğinde akışın normal olup olmadığını belirlemede değişkenler, analizden çıkarma sırasına göre önemli bulunmuştur. Hem Sınıflandırma Ağaçları ve Regresyon Ağaçları hem de Rastgele Orman Yöntemi, bu akışta başarılı bir sınıflandırıcı performansı sergilemişlerdir. Bu çalışmada yakalanan bu başarı ile kullanılan yöntemlerin, ağ akışı incelemelerinde ve diğer zararlı yazılımları tespit etmede etkin olduğunu söylenebilir.

Çalışmanın, Siber Güvenlik Alanına dikkat çekmesi, Sınıflandırma ve Regresyon Ağaçları'nın yorumunun kolay olduğunun gösterilmesi, ağ akışında botnet tespitinde kullanılan açıklayıcı değişkenler için bir fikir oluşturması, sanal bilgisayar kurulumu gibi konulara değinmesi bakımından önemli olduğu düşünülmektedir. Diğer Makine Öğrenmesi Tekniklerinden Sınıflandırma Algoritmaları, ağ akışı incelemelerinde kullanılabilir ve yöntemlerin gösterdiği performanslar ayrı ayrı ele alınarak bu minvaldeki çalışmaların kapsamı genişletilebilir.

KAYNAKLAR

Akman, M., Genç, Y., Ankarali, H. 2011. Random forests yöntemi ve sağlık alanında bir uygulama. ***Turkiye Klinikleri Journal of Biostatistics,*** **3** (1): 36-48.

Alauthman, M., 2016. ***An Efficient Approach to Online bot Detection Based on a Reinforcement Learning Technique*** (PhD Thesis). Northumbria University, New Castle, UK.

Alpaydin, E., 2014. ***Introduction to Machine Learning.*** MIT press, 3rt edition, 640

Anonim, 2018a. Sosyal Medya ve Mobil Kullanıcı İstatistikleri. ***https://dijilopedi.com/2018-internet-kullanimi-ve-sosyal-medya-istatistikleri/*** Erişim tarihi: 01.03.2018

Anonim, 2018b. Avrupadaki en fazla siber saldırı türkiyede ***http://www.sigortacigazetesi.com.tr/avrupadaki-en-fazla-siber-saldiri-turkiyede/*** Erişim tarihi: 01.03.2018

Anonim, 2018c. An executives guide to machine learning ***https://www.mckinsey.com/industries/high-tech/our-insights/anexecutives guide-to-machine-learning*** Erişim tarihi: 08.04.2018

Bailey, M., Cooke, E., Jahanian, F., Xu, Y., Karir, M., 2009. A survey of botnet technology and defenses. ***CATCH'09 Cybersecurity Applications and Technology .*** 3-4 March 2009, Washington, DC, USA. 299-304.

Barazi, J., Jakalan, A., XiaoWei, W., 2014. Botnet detection techniques. ***The International Journal of Computer Science and Communication Security(IJCSCS)***, **14:** 61-64.

Barthakur, P., Dahal, M., Ghose, M. K., 2013. An efficient machine learning based classification scheme for detecting distributed command & control traffic of P2P botnets. ***International Journal of Modern Education and Computer Science,*** **5** (10): 9-18.

Bilge, L., Strufe, T., Balzarotti, D., Kirda, E., 2009. All your contacts are belong to us:automated identity theft attacks on social networks. ***In Proceedings of the 18th International Conference on World Wide Web***. 20-24 April 2009, Spain, Madrid. 551-560.

Binkley, J. R., Singh, S., 2006. An Algorithm for Anomaly-based Botnet Detection***. SRUTI,*** **6**: 7.

Bock, H. H. 2002. Data mining tasks and methods: Classification: the goal of classification. ***In Handbook of Data Mining and Knowledge Discovery,*** **6**:254-258.

Bradley, A. P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. ***Pattern Recognition,*** **30** (7): 1145-1159.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J., 2017. ***Classification and Regression Trees.*** Taylor Francis, Berkeley, California. 368

Breiman, L., 1996. Bagging predictors. ***Machine Learning***, **24** (2): 123-140.

Breiman, L., 2001. Random forests. ***Machine Learning,*** **45** (1): s.5-32.

Buczak, A. L., Guven, E., 2016. A survey of data mining and machine learningmethods for cyber security intrusion detection. ***IEEE Communications Surveys & Tutorials,*** **18** (2): 1153-1176.

Chapelle, O., Scholkopf, B., Zien, A. 2009. Semi-supervised learning. ***IEEE Transactions on Neural Networks,*** **20** (3): 542-542.

Chen, R., Niu, W., Zhang, X., Zhuo, Z., Lv, F., 2017. An effective conversation-based botnet detection method. ***Mathematical Problems in Engineering***,**42**: 1-9.

Chipman, H. A., George, E. I., McCulloch, R. E., 1998. Bayesian CART modelsearch. ***Journal of the American Statistical Association, 93*** (443): 935-948.

Choi, H., Lee, H., Lee, H., Kim, H., 2007. Botnet detection by monitoring group activities in DNS traffic**.** ***In Computer and Information Technology***. 16-19 Oct. 2007, Aizu-Wakamatsu, Fukushima, Japan. 715-720.

Choi, H., Lee, H., 2012. Identifying botnets by capturing group activities in DNS traffic**.** ***Computer Networks,*** **56** (1): 20-33.

Del Val, D., Klemets, A. E., 2000. Method and apparatus for communication media commands and media data using the HTTP protocol*.* ***Patent and Trademark Office.***

Domingos, P., 2012. A few useful things to know about machine learning***. Communications of the ACM,*** **55** (10): 78-87.

Eslahi, M., Hashim, H., Tahir, N. M, 2013. An efficient false alarm reduction a pproach in HTTP-based botnet detection. ***In Computers & Informatics (ISCI), 2013 IEEE Symposium.*** 7-9 April 2013, Langkawi, Malaysia. 201-205

Eslahi, M., Salleh, R., Anuar, N. B., 2012. Bots and botnets: An overview of characteristics, detection and challenges. ***In Control System, Computing and Engineering (ICCSCE), IEEE International Conference.*** 23-25 Nov. 2012, Penang, Malaysia. 349-354.

Fawcett, T., (2006). An introduction to ROC analysis. ***Pattern Recognition Letters,*** **27** (8): 861- 874.

Feily, M., Shahrestani, A., Ramadass, S. 2009. A survey of botnet and botnet detection. ***2009 Third International Conference on Emerging SecurityInformation, Systems and Technologies.*** 18-23 June 2009, Athens, Glyfada, Greece.20-22.

Feizollah, A., Anuar, N. B., Salleh, R., Amalina, F., Shamshirband, S., 2013. A study of machine learning classifiers for anomaly-based mobile botnet detection. ***Malaysian Journal of Computer Science,*** **26** (4): 251-265.

Flach, P., (2004). The many faces of ROC analysis in machine learning. ***Icml Tutorial***.

Freund, Y., Schapire, R. E., 1996. Experiments with a new boosting algorithm. ***Icml,*** **96:** 148-156.

Gislason, P. O., Benediktsson, J. A., Sveinsson, J. R., 2006. Random forests for land cover classification. ***Pattern Recognition Letters***, **27** (4): 294-300.

Glas, A. S., Lijmer, J. G., Prins, M. H., Bonsel, G. J., Bossuyt, P. M., 2003. The diagnostic odds ratio: a single indicator of test performance. ***Journal of clinical Epidemiology, 56*** (11): 1129-1135.

Gu, G., Zhang, J., & Lee, W., 2008. BotSniffer: Detecting botnet command and control channels in network traffic. ***17th USENIX Security Symposium.*** 8 Feb 2008, San Diego, CA.15.

Gu, G., Perdisci, R., Zhang, J., & Lee, W. ; 2008, July. BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection. ***In Usenıx Security Symposium,*** **5** (2): 139-154.

Hachem, N., Mustapha, Y. B., Granadillo, G. G., & Debar, H. ; 2011. Botnets: lifecycle and taxonomy. ***In Network and Information Systems Security (SAR-SSI), Conference.*** 18-21 May 2011, La Rochelle, France. 200-208.

Hastie, T., Tibshirani, R., Friedman, J., 2009. Unsupervised learning. ***In The Elements of Statistical Learning.*** Springer, New York. 100.

Hoang, X. D., Nguyen, Q. C., 2018. Botnet detection based on machine learning techniques using DNS query data. ***Future Internet,*** **10** (5): 43.

Horning, N., 2010. Random Forests: An algorithm for image classification and generation of continuous fields data sets. ***In Proceedings of the International Conference on Geoinformatics for Spatial Infrastructure Development in Earth and Allied Sciences.*** Osaka, Japan.911.

Jakobsson, M., & Ramzan, Z. 2008. ***Crimeware: Understanding New Attacks and Defenses***. Addison-Wesley Professional; 1 edition. 608.

Joachims, T., 1999, June. Transductive inference for text classification using support vector machines. ***ICML,*** **99**: 200-209.

Joachims, T., 2002. ***Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms***. Norwell: Kluwer Academic Publishers, UK.186.

Juels, A., Stamm, S., Jakobsson, M., 2007. Combating Click Fraud via Premium
Clicks. ***In Usenıx Security Symposium***. 6-10 August 2007, Boston. 17-26.

Kaelbling, L. P., Littman, M. L., Moore, A. W., 1996. Reinforcement learning: A survey. ***Journal of Artificial Intelligence Research,*** **4:** 237-285.

Kalaivani, P., Vijaya, M. 2016, Mining based detection of botnet traffic in network
flow. ***International Journal of computer Science and information Technology &Security***, **6**: 535-540

Kantardzic, M., 2011. ***Data Mining: Concepts, Models, Methods, and Algorithms***. Wiley-IEEE Press; 2 edition. 552.

Karasaridis, A., Rexroad, B., Hoeflin, D. A., 2007. Wide-Scale Botnet Detection and Characterization. ***HotBots,*** **7**: 7.

Karim, A., Salleh, R. B., Shiraz, M., Shah, S. A. A., Awan, I., Anuar, N. B., 2014. Botnet detection techniques: review, future trends and issues. ***Journal of Zhejiang University Scıence C.,*** **15** (11): 943-983.

Kartal, E., BALABAN, M. E., 2015. ***Sınıflandırmaya Dayalı Makine Öğrenmesi Teknikleri ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama*** (doktora tezi). İstanbul Üniversitesi.

Kelleher, J. D., Mac Namee, B., D'Arcy, A., 2015. ***Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples and Case Studies***. MIT Press, 1.edition. 624.

Kempanna, M., & Kannan, R. J. ; 2015. A novel traffic reduction technique and anfıs based botnet detection. ***In International Conference on Circuits, Systems, Signal and Telecommunications.*** January 2015.31.

Kirubavathi, G., Anitha, R., 2018. Structural analysis and detection of android botnets using machine learning techniques. ***International Journal of Information Security,*** **17** (2): 153-167.

Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. ***Ijcai,*** **14** (2): 1137-1145.

Kumar, R., Kaur, T., 2014. Machine learning based traffic classification using low level features and statistical analysis. ***International Journal of Computer Applications,*** **108**: 12.

Larose, D. T., 2006. ***Data Mining Methods & Models***. John Wiley & Sons.385

Li, C., Jiang, W., Zou, X. ; 2009. Botnet: Survey and case study***. In Innovative Computing, Information and Control (Icicic), 2009 Fourth International Conference on,*** 7-9 Dec. 2009, Kaohsiung, Taiwan. 1184-1187.

Lantz, B., 2013. ***Machine Learning with R***. Packt Publishing Ltd.396.

Liaw, A., Wiener, M., 2002. Classification and regression by random Forest. ***Renews,*** **2** (3): 18-22.

Livadas, C., Walsh, R., Lapsley, D., Strayer, W. T., 2006 Using machine learning technliques to identify botnet traffic. ***In Local Computer Networks, Proceedings 2006 31st IEEE Conference on***, 967-974.

Loh, W. Y. 2011. Classification and regression trees. ***Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,*** **1** (1): 14-23.

Mathur, L., Raheja, M., Ahlawat, P., 2018. Botnet Detection via mining of network traffic flow. ***Procedia Computer Science,*** **132**: 1668-1677.

Micro, T. 2006. Taxonomy of botnet threats. ***Whitepaper***.

Mitchell, T. M. 1997. Machine learning. ***Burr Ridge, IL: McGraw Hill,*** **45** (37): 870-877.

Mohri, M., Rostamizadeh, A., Talwalkar, A., 2012. ***Foundations of Machine Learning.*** MIT Press, 1th edition.432.

Moore, A. W., Zuev, D., 2005. Internet traffic classification using bayesian analysis techniques. ***In Acm Sıgmetrıcs Performance Evaluation Review***, **33** (1): 50-60.

Morgan, J.,2014. ***Classification and Regression Tree Analysis*** (master thesis). Boston: Boston University.

Murthy, S. K., 1998. Automatic construction of decision trees from data: A multi-disciplinary survey. ***Data Mining and Knowledge Discovery,*** **2** (4): 345-389.

Noh, S. K., Oh, J. H., Lee, J. S., Noh, B. N., Jeong, H. C, 2009 Detecting P2P botnets using a multi-phased flow model. ***ICDS'09. Third International Conference on***. 1-7 Feb. 2009, Cancun, Mexico. 247-253.

Ramachandran, A., Dagon, D., Feamster, N., 2006. Can DNS-based blacklists keep up with bots?. ***CIES.***

Rechenthin, M. D., 2014. ***Machine-Learning Classification Techniques For The Analysis And Prediction Of High-Frequency Stock Direction*** (Phd thesis). The University of Iowa.

Rodríguez-Gómez, R. A., Maciá-Fernández, G., García-Teodoro, P., 2013. Survey and taxonomy of botnet research through life-cycle. ***ACM Computing Surveys (CSUR),*** **45** (4): 45.

Ryu, S. ve Yang, B., 2018. A comparative study of machine learning algorithms and their ensembles for botnet detection. ***Journal of Computer and Communications,*** **6** (05): 119.

Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Hakimian, P., 2011. Detecting P2P botnets through network behavior analysis and machine learning. ***In Privacy, Security and Trust (PST), Ninth Annual International Conference on.*** 19-21 July 2011, Montreal, QC, Canada .174-180.

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

Settles, B., 2012. Active learning. ***Synthesis Lectures on Artificial Intelligence and Machine Learning,* 6** (1): 1-114.

Singh, K., Guntuku, S. C., Thakur, A., Hota, C., 2014. Big data analytics framework for peer-to-peer botnet detection using random forests. ***Information Sciences,* 278**: 488-497.

Shinder, D. L., Tittel, E. 2002. ***Scene of the Cybercrime: Computer Forensics Handbook***. Syngress Publishing.

Spitzner, L., 2003. Honeypots: Catching the insider threat. ***In Computer Security Applications Conference, Proceedings. 19th Annual,*** 8-12 Dec. 2003, Las Vegas,Usa. 170-179.

Spitzner, L., 2003. ***Honeypots: Tracking Hackers***. Boston: Addison-Wesley.480.

Suchetana, B., Rajagopalan, B., Silverstein, J., 2017. Assessment of wastewater treatment facility compliance with decreasing ammonia discharge limits using a regression tree model. ***Science of the Total Environment,* 598**: 249-257.

Timofeev, R., 2004. ***Classification and Regression Trees (CART) Theory and Applications*** (master thesis). Humboldt University, Berlin.

Watts, J. D., Powell, S. L., Lawrence, R. L., Hilker, T., 2011. Improved classification of conservation tillage adoption using high temporal and synthetic satellite imagery. ***Remote Sensing of Environment,* 115** (1): 66-75.

Whyte, D., Kranakis, E., Van Oorschot, P. C., 2005. DNS-based Detection of Scanning Worms in an Enterprise Network. ***In NDSS***. February 3-4, 2005, San Diego,USA. 17.

Zeidanloo, H. R., Shooshtari, M. J. Z., Amoli, P. V., Safari, M., Zamani, M., 2010, A taxonomy of Botnet detection techniques.***Technology (ICCSIT) 2010 3rd IEEE International*** *Conference on.*Dec 23-25 2010, Bangladesh. 158-162.

Zilong, W., Jinsong, W., Wenyi, H., Chengyi, X., 2010. The detection of IRC botnet based on abnormal behavior. ***In Multimedia and Information Technology (MMIT), 2010 Second International Conference on.*** 24-25 April 2010 Kaifeng, China .146-149.

Zhao, D., Traore, I., Ghorbani, A., Sayed, B., Saad, S., Lu, W., 2012. Peer to peer botnet detection based on flow intervals. ***In IFIP International Information Security Conference*.** 4-6 June, Greece. 87-102.

Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., Garant, D., 2013. Botnet detection based on traffic behavior analysis and flow intervals. ***Computers & Security,* 39:** 2-16.

EKLER

EK 1. Algoritma Kodları

Sınıflandırma ağaçları kodları

```
import pandas as pd
import numpy as np
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn import tree
from sklearn import datasets, metrics, tree, cross_validation
from sklearn.metrics import confusion_matrix
from sklearn import metrics

np.random.seed(0)

data=pd.read_csv("/home/duygu/Desktop/data123.csv")
data=data.fillna(data.mean())
degiskenler=["Sure","OrtalamaBaytOrani","Protokol","KaynakIp","HedefIp","KaynakP
ort","HedefPort","Statesyn","Staterst","Statefin","Stateack","ToplamPaket","ResetBağl
antiSayısi","OrtalamaBayt","OrtalamaPaketOrani","PaketBoyutu"]
y=data["botnet"]
X=data[degiskenler]

print X.shape,y.shape

print data["Age"].head()

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size = 0.3, random_state =
100)
print X_train.shape, X_test.shape
```

```
clf_gini = DecisionTreeClassifier(criterion = "gini", random_state = 100,max_depth=3,
min_samples_leaf=5)
clf_gini.fit(X_train, y_train)
y_pred = clf_gini.predict(X_test)
print "Accuracy is ", accuracy_score(y_test,y_pred)*100
cm = confusion_matrix(y_test, y_pred)
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred, pos_label=2)
print "*****"
print metrics.auc(fpr, tpr)
print(cm)
metrics.confusion_matrix(y_test, clf_gini.predict(X_test))
print metrics.classification_report(y_test, clf_gini.predict(X_test))
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(clf_gini, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = degiskenler)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

Rastgele orman tekniğinin kodları

```
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
import numpy as np
from sklearn.cross_validation import train_test_split
from sklearn.metrics import accuracy_score
from sklearn import tree
```

```
from sklearn import datasets, metrics, tree, cross_validation
from sklearn.metrics import confusion_matrix
from sklearn import metrics

np.random.seed(0)

data=pd.read_csv("/home/duygu/Desktop/data123.csv")
data=data.fillna(data.mean())
degiskenler=["Sure","OrtalamaBaytOrani","Protokol","KaynakIp","HedefIp","KaynakP
ort","HedefPort","Statesyn","Staterst","Statefin","Stateack","ToplamPaket","ResetBağl
antiSayısi","OrtalamaBayt","OrtalamaPaketOrani","PaketBoyutu"]
y=data["botnet"]
X=data[degiskenler]

print X.shape,y.shape

print data["Age"].head()

X_train, X_test, y_train, y_test = train_test_split( X, y, test_size = 0.3, random_state =
100)
print X_train.shape, X_test.shape

random_forest = RandomForestClassifier(n_estimators=30, max_depth=10,
random_state=1)
random_forest.fit(X_train, y_train)

y_predict = random_forest.predict(X_test)
print "Accuracy is ", accuracy_score(y_test,y_predict)*100
pd.DataFrame(
   confusion_matrix(y_test, y_predict),
)
```

```
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_predict, pos_label=2)
print "*****"
print metrics.auc(fpr, tpr)
```

ÖZ GEÇMİŞ

1991 yılında Seyhan'da doğdu. İlk, orta ve lise öğrenimini Adana'da tamamladı. 2014 yılında Fırat Üniversitesi İstatistik Bölümünden bölüm birinciliği derecesi ile mezun oldu. 2016 yılında Van Yüzüncü Yıl Üniversitesi İstatistik Anabilim Dalı'nda yüksek lisans yapmaya başladı. 2017 yılında Erasmus Programı kapsamında Polonyada staj eğitimi gördü.

YÜZÜNCÜ YIL ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ
LİSANSÜSTÜ TEZ ORİJİNALLİK RAPORU

Tarih: 14/08/2018

Tez Başlığı / Konusu:SINIFLANDIRMA ve REGRESYON AĞAÇLARI ile RASTGELE ORMAN ALGORİTMASI KULLANARAK BOTNET TESPİTİ VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ ÖRNEĞİ+

Yukarıda başlığı/konusu belirlenen tez çalışmamın Kapak sayfası, Giriş, Ana bölümler ve Sonuç bölümlerinden oluşan toplam 28 sayfalık kısmına ilişkin, 14/08/2018 tarihinde şahsım/tez danışmanım tarafından TURNITIN intihal tespit programından aşağıda belirtilen filtreleme uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 1 (bir) dir.
Uygulanan filtreler aşağıda verilmiştir:
- Kabul ve onay sayfası hariç,
- Teşekkür hariç,
- İçindekiler hariç,
- Simge ve kısaltmalar hariç,
- Gereç ve yöntemler hariç,
- Kaynakça hariç,
- Alıntılar hariç,
-Tezden çıkan yayınlar hariç,
- 7 kelimeden daha az örtüşme içeren metin kısımları hariç (Limit match size to 7 words)

Yüzüncü Yıl Üniversitesi Lisansüstü Tez Orijinallik Raporu Alınması ve Kullanılmasına İlişkin Yönergeyi inceledim ve bu yönergede belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini bilgilerinize arz ederim.

14.08.2018

Adı Soyadı: Duygu KORKMAZ

Öğrenci No: 169102051

Anabilim Dalı: İstatistik

Programı: İstatistik

Statüsü:Y.Lisans ■Doktora□

DANIŞMAN ONAYI
UYGUNDUR

Dr. H. Ergelik

ENSTİTÜ ONAYI
UYGUNDUR