Journal of the Faculty of Engineering and Architecture of Gazi University 38:3 (2023) 1737-1746

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Detection and classification of darknet traffic using machine learning methods

Mesut Uğurlu[1]*, İbrahim Alper Doğru[2], Recep Sinan Arslan[3]

[1]Gazi University, Graduate School of Natural and Applied Sciences, Department of Information Security Engineering, 06570, Ankara, Türkiye
[2]Gazi University, Faculty of Technology, Department of Computer Engineering, 06570, Ankara, Türkiye
[3]Kayseri University, Faculty of Engineering, Department of Computer Engineering, 38039, Kayseri, Türkiye

Highlights:

- Detection and classification of darknet traffic
- Model performance enhancement with feature selection
- Increasing the model success rate with data balancing

Keywords:

- Darknet
- Cyber security
- Encrypted network traffic
- Machine learning
- Classification

Article Info:

Research Article
Received: 13.11.2021
Accepted: 18.08.2022

DOI:

10.17341/gazimmfd.1023147

Correspondence:

Author: Mesut Uğurlu
e-mail: mesut.ugurlu@gazi.edu.tr
phone: +90 542 341 0629

Graphical/Tabular Abstract

In this study, a machine learning-based model has been developed for the detection and classification of the darknet or dark web that cybercriminals and attackers use to hide their identity information and provide encrypted communication. The statistical information of packets was analyzed using machine learning approach without deciphering encrypted network traffic. Feature selection was made to increase the performance of the model. In addition to this process, data balancing was performed in order to increase the detection and classification rate of features with low numbers during the training phase. The created model is given in Figure A.

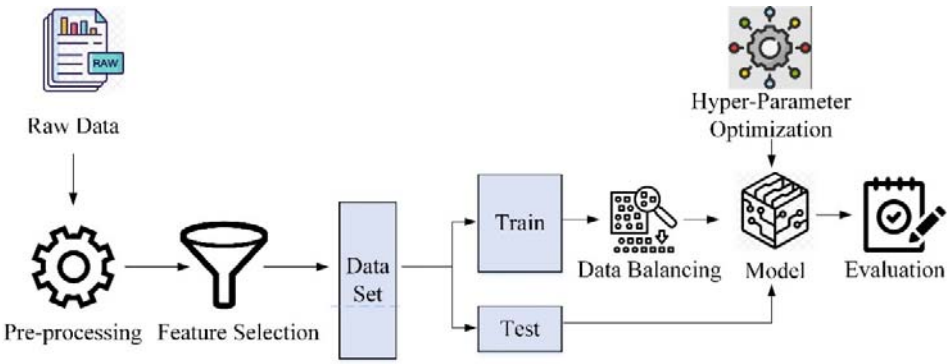


Figure A. Model architecture

Purpose:
The aim of the study is to detect and classify dark network traffic using the statistical properties of network packets.

Theory and Methods:
In the study, firstly, the data was pre-processed, and the data was converted into a suitable format. Afterwards, feature selection was made to reduce the processing load. In order to increase the detection and classification success of the classes with a small number of data, data balancing was performed. The hyperparameters of ten different algorithms were selected by Grid-Search.

Results:
XGBoost algorithm gave the most successful results in binary classification with an accuracy rate of 98.4%, while the Decision Tree algorithm gave the most successful results in multi-classification with an accuracy of 93.2%.

Conclusion:
Due to the increase in cyber-crime rates and the sale of captured information and documents through the darknet, analyzing the traffic to this network is very important. With this study, without deep packet analysis, only the statistical data of the packet was analyzed and the high accuracy rate of traffic to this network was detected and classified.

Journal of the Faculty of Engineering and Architecture of Gazi University 38:3 (2023) 1737-1746

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Karanlık ağ trafiğinin makine öğrenmesi yöntemleri kullanılarak tespiti ve sınıflandırılması

Mesut Uğurlu[1]*, İbrahim Alper Doğru[2], Recep Sinan Arslan[3]

[1]Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgi Güvenliği Mühendisliği Bölümü, 06570 Maltepe Ankara, Türkiye
[2]Gazi Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 06570 Maltepe Ankara, Türkiye
[3]Kayseri Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 38039, Kayseri, Türkiye

ÖNEÇIKANLAR

- Karanlık ağ trafiğinin tespiti ve sınıflandırılması
- Öznitelik seçimi ile model performans artırımı
- Veri dengeleme ile model başarı oranı artırımı

Makale Bilgileri

Araştırma Makalesi
Geliş: 13.11.2021
Kabul: 18.08.2022

DOI:
10.17341/gazimmfd.1023147

Anahtar Kelimeler:
Karanlık Ağ,
siber güvenlik,
şifreli ağ trafiği,
makine öğrenme,
sınıflandırma

ÖZ

Dijitalleşme ile suç dünyası da dijital bir hale gelmiştir ve internet üzerinden işlenen suçların sayısı her geçen gün artmaktadır. Siber suçlular ve saldırganlar kimliklerini gizlemek ve şifreli iletişim sağlamak için Karanlık Ağ adı verilen ve internet üzerinde bulunan gizli ağları kullanmaktadırlar. Karanlık Ağlar normal internet altyapısından farklı ve özel erişim yöntemlerine sahiptirler. Bu ağlara yapılan tüm erişimler şüphelidir ve incelenmesi gerekmektedir. Karanlık Ağ, şifreli iletişim sağladığı için günümüz güvenlik araçları ile tespit edilmesi ve sınıflandırılması zordur. Bu çalışmada şifreli ağ trafiği deşifreleme işlemi yapılmadan sadece paketlerin istatistiki bilgileri makine öğrenmesi yaklaşımı kullanılarak analiz edilmiştir. Veri seti olarak açık kaynak olan CICDarknet2020 veri seti kullanılmıştır. Paket analizi için K En Yakın Komşu, Lojistik Regresyon, Rassal Orman, SVM, Karar Ağacı, Gaussian Naive Bayes, Doğrusal Ayrımcı Analiz, Gradyan Artırma, Ekstra Ağaç ve XGBoost algoritmalarını kapsayan detaylı bir deneysel çalışma gerçekleştirilmiştir. Yapılan deneysel çalışmalarda Karar Ağacı algoritmasının %93,32 doğruluk oranı ile en yüksek sınıflandırma başarısına sahip olduğu görülmüştür.

Detection and classification of darknet traffic using machine learning methods

HIGHLIGHTS

- Detection and classification of darknet traffic
- Model performance enhancement with feature selection
- Increasing the model success rate with data balancing

Article Info

Research Article
Received: 13.11.2021
Accepted: 18.08.2022

DOI:
10.17341/gazimmfd.1023147

Keywords:
Darknet,
cyber security,
encrypted network traffic,
machine learning,
classification

ABSTRACT

With digitalization, the world of crime has also become digital and the number of crimes committed over the internet is increasing day by day. Cybercriminals and attackers use secret networks on the Internet, called the Dark Web, to hide their identities and provide encrypted communication. Darknets have different and special access methods than normal internet infrastructure. All access to these networks is suspect and needs to be investigated. Because the Darknet provides encrypted communication, it is difficult to detect and classify with today's security tools. In this study, only the statistical information of packets was analyzed using machine learning approach without deciphering encrypted network traffic. CICDarknet2020 dataset was used and a detailed experimental study including K Nearest Neighbor, Logistic Regression, Random Forest, SVM, Decision Tree, Gaussian Naive Bayes, Linear Discriminatory Analysis, Gradient Boosting, Extra Tree and XGBoost algorithms was carried out for packet analysis. In experimental studies, it has been observed that the Decision Tree algorithm has the highest classification success with an accuracy rate of 93.32%.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *mesut.ugurlu@gazi.edu.tr, iadogru@gazi.edu.tr, recepsinanarslan@kayseri.edu.tr / Tel: +90 542 341 0629

1. Giriş (Introduction)

İnternetin günlük hayatın vazgeçilmez bir parçası olması ile tüm insanlık interneti farklı amaçlar için kullanmaktadır. İnternetin gelişimi suç dünyasını da değiştirdi ve yeni suçlar ortaya çıktı. Bununla birlikte siber suç ve siber suçlu kavramlarını da literatüre girdi. Siber suç, suçun işleyişinde bir bilgisayar ve ağ içeren bir suç çeşididir [1]. Siber suçlar ulusal güvenliği, kişileri ve şirketleri de tehdit etmektedir [2]. Siber suçlar siber saldırı, siber aldatma/hırsızlık, siber porno ve siber şiddet olarak farklı kategorilerde sınıflandırılabilir [3]. Siber suçlular bilgileri ele geçirmek, ticari ve askeri bilgileri öğrenmek veya kötü amaçlı araştırmaları yapabilmek için bilgisayar ve internet teknolojisini kullanırlar [4]. Siber suçların tespiti ve siber suçluların yakalanması için birçok çalışma yapılmakta ve sistem oluşturulmaktadır. Saldırganlar ve suçlular ise tespit edilmemek ve yakalanmamak için önlemler almaktadır. Bu önlemler biri de gizli ve anonim haberleşmeyi sağlayan Karanlık Ağ (Darknet) altyapısının kullanılmasıdır.

Karanlık Ağ terimi genellikle internet üzerindeki gizli ağları ifade etmek için kullanılır [5]. Karanlık Ağ açık ve bilinen internet altyapısından ayrı olan bir iletişim sistemidir. Genellikle anonim ve şifreli iletişim anlamına gelmektedir. İnternet altyapısı üzerinden erişilebilir olmasına rağmen genel internet ağından ayrıdır [6]. Terörist gruplar, aşırılık yanlısı gruplar ve siber suçlular, Karanlık Ağı suç faaliyetleri yürütmek, ideolojilerini tanıtmak veya uyuşturucu, silah, kredi kartı verileri, sahte belgeler, cinayet, narkotik ve uygunsuz pornografi gibi hizmet veya mal satmak için kullanmaktadır [7].

Birçok farklı Karanlık Ağ olmasına rağmen en çok kullanılan ağ Soğan Yönlendirme (The Onion Router, TOR) ağıdır. TOR, ilk olarak ABD Deniz Araştırma Laboratuvarı tarafından yurtdışında konuşlanmış ajanlarla iletişimi korumak için geliştirilen, ancak daha sonra başkalarıyla anonim olarak iletişim kurmak isteyen herkese açık hale getirilen bir araçtır [5]. Tor istemcisi uygulama sunucusuyla iletişim kurmak istediğinde, özel dizin sunucularından tüm yönlendirici bilgilerini indirir ve giriş, orta ve çıkış soğan yönlendiricisi olarak üç yönlendirici seçer [8]. Her yönlendirici, geçiş halindeki tüm verileri şifreleme ve şifresini çözme yeteneğine sahiptir [9]. Bu şekilde, kullanıcının trafiğini kapsüller ve bir saldırganın üçünün de şifresini çözemeyeceği şekilde onları ara düğümler üzerinden yönlendirir [10]. TOR Karanlık Ağı, iki uç noktanın birbirlerinin kimliğini bilmeden haberleşmeyi sağlayan bir özelliğe sahiptir [7]. TOR ağında birçok Soğan Yönlendiricisi (Onion Router, OR) bulunmaktadır ve bu yönlendiriciler arasında Taşıma Katmanı Güvenliği (Transport Layer Security, TLS) bağlantı bulunmaktadır. Her bir yönlendirici uzun vadeli kimlik anahtarı ve kısa vadeli soğan (onion) adı verilen anahtara sahiptir. Kimlik anahtarı TLS sertifikasını imzalamak için, soğan anahtarı ise bir devre kurmak veya gelen istekleri deşifreleme etmek amacıyla kullanılır [11].

Siber suçlular bilgi alışverişi yapmak ve çalınan kredi kartı bilgileri ve kimlik hırsızlığı veri madenciliği için kullanılan toplu veriler gibi verileri aktarmak ve depolamak için karanlık ağları kullanır [12]. Karanlık Ağ internet üzerinden erişilebilir olmasına rağmen kullanılmayan internet adresine sahiptir. Bu adresler kullanılmadığı için bu adreslere yapılan tüm erişimler şüpheli olarak görülmektedir ve analiz edilmelidir [13]. Saldırganlar kötücül yazılımlar aracılığı kurban sistemlere kullanıcıların bilgisi olmadan sızma, ele geçirme, bilgi sızdırma ve verileri yok etme gibi hedeflerine yönelik işlemler yapmaktadırlar [14, 15]. Karanlık Ağ trafiğini analiz etmek, kötücül yazılımların dağılmadan önce izlenmesine ve dağıldıktan sonra tespit edilerek önlem alınabilmesine yardımcı olur. Bu çalışmada Karanlık Ağ uygulamalarını derin paket analizi yapmadan sadece paketlerin meta verilerini kullanılarak tespit edilmesi ve sınıflandırılması amaçlanmıştır. Çalışma kapsamında Lashkari vd. [16] tarafından üretilen ve açık kaynak olan CICDarknet2020 veri seti kullanılmıştır. Bu veri seti içerisinde iki katmanlı bir yaklaşım mevcuttur ve ilk katmanda trafiğin normal mi yoksa Karanlık Ağa mı ait olduğu belirlenmektedir. İkinci katmanda ise Karanlık Ağ trafiği Ses Akışı (Audio-Stream), Tarayıcı (Browsing), Sohbet (Chat), E-Posta (Email), Eşler Arası (P2P), Transfer, Video Akışı (Video-Stream) ve IP Üzerinden Ses (Voice over IP, VoIP) sınıflarına ayrılmaktadır. Veri setini oluşturmak için ISCXTor2016 ve ISCXVPN2016 veri setleri birleştirilmiştir. VPN ve Non-VPN sınıfları ISCXVPN2016 veri setinden, Tor ve Non-Tor sınıfları ise ISXCTor2016 veri seti içerisinde bulunan verilerden elde edilmiştir.

Bu makalenin ana katkı noktaları aşağıdaki gibi olmuştur:

- Öznitelik seçimi ile önemsiz ve gereksiz öznitelikler belirlenerek çıkarılmış ve model performans artışı sağlanmıştır.
- Veri seti içerisinde bulunan sınıflardaki dengesizliği gidermek için Rastgele Yeniden Örneklendirme (Random Over Sampling, ROS) tekniği kullanılmıştır. ROS işlemi sonrası modelin başarı oranı artmıştır.
- Makine öğrenme algoritmalarının başarısını doğrudan etkileyen hiper parametrelerin seçimi için ızgara arama yöntemi kullanılmıştır. Her bir sınıflandırıcı için parametre arama ve seçme işlemi ayrı ayrı gerçekleştirilerek performans artışı sağlanmıştır.
- Literatürde aynı veri seti ile gerçekleştirilen çalışmalar ile karşılaştırıldığında daha yüksek doğruluk ile sınıflandırma işlemi gerçekleştirilmiştir.

Bu makalenin ikinci kısmında Karanlık Ağ trafiğinin sınıflandırılması için literatürde yapılan çalışmalara yer verilmiştir. Çalışmada kullanılan veri seti ve detayları üçüncü bölümde verilmiştir. Dördüncü bölümde önerilen model mimarisi ve performans ölçüt metrikleri sunulmuştur. Önerilen modelin mevcut problemin çözümü için elde ettiği sonuçlar beşinci bölümde anlatılmıştır. Son bölümde ise önerilen modelin literatürde yapılan diğer çalışmalar ile karşılaştırılması ve gelecek çalışmalar için yapılabileceklere yer verilmiştir.

2. Literatür Taraması (Related Works)

Karanlık Ağ trafiğini tespit edilmesi ve sınıflandırılması siber güvenliğin sağlanabilmesi için kritik bir işlemdir. Literatürde bu alanda birçok çalışma yapılmıştır. Yapılan çalışmalar ve detaylarına bu bölümde yer verilmiştir.

John Barker vd. [17] TOR trafiğini analiz ederek normal şifreli trafikten farklı olduğunu gösteren bir çalışma yapmışlardır. Çalışma kapsamında normal Güvenli Hiper Metin Transfer Protokolü (Hyper-Text Transfer Protocol/Secure, HTTPS), TOR ağı üzerinde HTTP ve TOR ağı üzerinde HTTPS trafikler toplanarak veri seti oluşturulmuştur. Çalışma kapsamında Rassal Orman (Random Forest), J48 ve Adaboost algoritmaları kullanılmıştır. Adaboost algoritması HTTPS ve TOR ağı üzerinde HTTP trafiğini yüksek başarı oranı ile tespit etmiştir. Çalışma sonucunda TOR trafiğin normal şifreli trafikten farklı olduğu ve ayırt edilebilir olduğu gösterilmiştir.

Khalid Shahbar vd. [18] şifreli TOR trafiğini deşifreleme yapmadan sınıflandırmak için makine öğrenmesi algoritmaları kullanmışlardır. Sınıflandırma işlemi için akış ve devre olmak üzere iki farklı yaklaşım kullanılmıştır. Akış yaklaşımında şifrelenmiş trafiğin türünü tahmin etmek için kullanıcı ile yönlendirici arasındaki TCP iletişimi analiz edilmiştir. Devre yaklaşımında ise devreyi oluşturan ve çalıştıran kullanıcıya ait istatistikler incelenmiştir. Çalışma kapsamında 4 farklı makine öğrenme algoritması kullanılmıştır. Çevrim yönteminde

tarama, akış ve Bittorent sınıfları C4.5 algoritması ile %100'e yakın oran ile sınıflandırılmıştır. Akış temelli sınıflandırma yönteminde ise BayesNet %100 doğruluk oranı ile diğer algoritmalardan daha başarılı sonuç vermiştir.

Alaeddin Akmubayed vd. [19] makine öğrenmesi yaklaşımı kullanarak TOR trafiğini tespit etmişlerdir. Veri seti yazarlar tarafından oluşturulmuş ve normal trafik için Alexa tarafından yayınlanan ve en bilinen 100 sayfa üzerinden toplanmıştır. Yakalanan paketler içerisinde 40 adet öznitelik çıkarılmış ve etiketleme işlemi yapılmıştır. 4 farklı makine öğrenme algoritması kullanılmış ve karşılaştırılmıştır. TOR trafiği %99,64 doğruluk oranı ve %0,01 yanlış pozitif oranı ile tespit edilmiştir.

Siti Hajar Aminah Ali vd. [20] yaptıkları çalışmada karanlık ağ üzerindeki paket trafiğini izleyerek Dağıtık Hizmet Engelleme (Distrubuted Denial of Service Attack, DDoS) saldılarını tespit etmeyi amaçlamışlardır. Tüm protokoller ve port numaraları üzerinden yapılan DDoS saldılarını tespiti için Resource Allocating Network with Locality Sensitive Hashing (RAN-LSH) sınıflandırma yöntemi kullanmışlarıdır. Veri seti oluşturmak için Japon National Institute of Information and Communications Technology (NICT) üzerinden karanlık ağ verileri toplanmıştır. Veri seti içerisinde paket sayısı, protokol sayısı ve kaynak port adresi gibi paketin istatistiksel ve meta verilerinden oluşan 17 adet öznitelik bulunmaktadır. Çalışmada RBFN, RAN ve RAN-LSH modelleri kullanılarak karşılaştırma yapılmıştır. RAN-LSH modelinin diğer yöntemlere göre daha başarılı sonuç verdiği belirtilmiştir.

Elike Hodo vd. [21] makine öğrenmesi temelli bir yaklaşım ile TOR olmayan trafiğin sınıflandırılması çalışmasını yapmışlardır. Çalışma kapsamında Yapay Sinir Ağı (YSA) ve Support Vector Machine (SVM) algoritmaları kullanılmıştır. ISCXTor2016 açık kaynak veri seti kullanılmış ve çalışma kapsamında sadece ikili TOR ve TOR olmayan sınıflandırma yapılmıştır. YSA algoritması %99,8 doğruluk oranı ile TOR olmayan trafiği tespit ve SVM'e göre daha başarılı bir sonuç vermiştir. Çalışmada öznitelik seçimi yapılarak veri kümesi %65 azaltılmış ve hesaplama maliyeti düşürülmüştür.

Arash Habibi Lashkari vd. [22] zaman temelli öznitelikler kullanarak TOR trafiğini sınıflandırmışlardır. Veri seti ISCXFlowMeter aracı kullanılarak yazarlar tarafından oluşturulmuş ve 8 adet sınıf ve 23 adet öznitelik içermektedir. Bu çalışmada akış temelli yöntemlerden farklı olarak zaman temelli yöntemde kullanılmıştır. Zaman temelli yaklaşım kapsamında 10, 15, 30, 60 ve 120 saniye veri setleri bulunmaktadır. Çalışmada iki farklı senaryo bulunmaktadır. İlk senaryoda ikili sınıflandırma yapılarak trafiğin TOR olup olmadığı belirlenmiştir. İkinci senaryoda trafik 8 farklı kategoride sınıflandırılmıştır. İlk senaryo için 3 farklı makine öğrenme algoritması kullanılmıştır. İlk senaryoda zaman arttıkça başarı oranı artmış ve C4.5 algoritmasının en başarılı sonucu verdiği belirtilmiştir. Rassal Orman algoritması ikinci senaryoda %84,3 ağırlıklı doğruluk oranı ile en başarılı sonucu vermiştir. İkinci senaryoda ilk senaryodan farklı olarak zaman arttıkça başarı oranının düştüğü gözlemlenmiştir.

Alfredo Cuzzocrea vd. [23] TOR trafiğini tespit etmek için makine öğrenmesi algoritması yaklaşımı kullanmışlardır. Çalışmalarında ISCXTor2016 veri setini kullanmışlar ve ikili sınıflandırma yöntemi ile trafiğin TOR ve TOR olmayan olarak tespit etmişlerdir. Altı farklı makine öğrenme algoritması kullanılmış olup jRip algoritması ile %99,8 kesinlik değeri ile en başarılı sonucu vermiştir.

Yuzong Hu vd. [24] karanlık ağ trafiğini sınıflandırılması için bir çalışma gerçekleştirmişlerdir. Veri seti 4 farklı karanlık ağdan toplanmış ve 8 adet sınıf içermektedir. Veri seti oluşturulurken CICFlowMeter aracı kullanılmış ve zaman bazlı 26 öznitelik seçilmiştir. Kaynak IP, hedef IP, kaynak port ve hedef port öznitelikleri kullanılmamıştır. Çalışmada ilk olarak trafiğin normal mi yoksa karanlık ağ mı olduğu, daha sonra hangi karanlık ağa dahil olduğu ve en son hangi sınıfa ait olduğu sınıflandırılmıştır. Bu yönteme hiyerarşik sınıflandırma adı verilmiştir. Çalışmada 6 farklı makine öğrenme algoritması ve 2 derin öğrenme algoritması kullanılmıştır. XGBoost TOR karanlık ağında %85,38 f1-ölçüt değeri ve I2P karanlık ağında %91,28 f1-ölçüt değeri ile en yüksek başarıyı elde etmiştir. Gradient Boosting Decision Tree (GBDT) algoritması Freenet ağında, LightGBM ise ZeroNet ağında en yüksek başarıyı elde etmiştir.

Aditya Gurunarayanan vd. [25] açık kaynak veri seti olan ISCXTor2016 veri setini kullanarak TOR ve TOR olmayan trafiğin sınıflandırılması için bir çalışma yapmışlardır. Veri seti içerisinde 8 adet sınıf bulunmakta olup sınıflar arası veri dengesizliğini gidermek için aşağı ve yukarı örnekleme yöntemi kullanılmıştır. Çalışma kapsamında altı adet makine öğrenme algoritması kullanılmış ve ızgara yöntemi ile hiper parametreler belirlenmiştir. Rassal Orman algoritması hem aşağı örnekleme hem de yukarı örnekleme yöntemlerinde en başarılı sonucu vermiştir.

3. Metodoloji Ve Yöntemler (Methodology And Methods)

Karanlık Ağ trafiği, veri seti içerisinde bulunan istatistiksel öznitelikler kullanılarak ve makine öğrenmesi yaklaşımı ile analiz edilerek sınıflandırılmıştır. Çalışmada K-En Yakın Komşu (K-Nearest Neighbor, KNN), Lojistik Regresyon (Logistic Regression), Rassal Orman, SVM, Karar Ağacı (Decision Tree), Gaussian Naive Bayes, Doğrusal Ayrımcı Analiz (Linear Discriminant Analysis), Gradyan Artırma (Gradient Boosting), Ekstra Ağaçlar (Extra Trees) ve XGBoost makine öğrenme algoritmaları kullanılmış ve karşılaştırılmıştır. Bu bölümde önerilen model mimarisi ve performans ölçümünde kullanılan metriklere yer verilmiştir.

3.1. Model Mimarisi (Model Architecture)

Makine öğrenme algoritmaları verilerin matematiksel olarak uygulanabilir bir formatta olmasına ihtiyaç duymaktadır [26]. Bu kapsamda veri seti analiz edildiğinde içerisinde boş (NaN) ve sonsuz (infinity) değerler bulunduğu görülmüştür. Önerilen modelin başarılı çalışabilmesi ve performans iyileştirilebilmesi için veri seti içerisinde bulunan bu öznitelikler ön işlemden geçirilmiştir. Boş ve sonsuz değerlerin model başarısını düşürmesini engellemek için bu değerler yerine 0 değeri atanmıştır.

Veri seti içerisinde 82 adet öznitelik ve 2 adet sınıf etiketi bulunmaktadır. Önerilen model, sınıflandırma işlemini paket içeriğine, kaynak ve hedef bilgilerine bakmadan sadece paketin istatistiksel verilerini analiz ederek gerçekleştirmektedir. Bu amaçla veri seti içerisinde bulunan akış numarası, kaynak adresi, hedef adresi, kaynak port numarası, hedef port numarası, protokol bilgisi ve zaman damgası bilgilerini içeren 7 adet öznitelik silinmiştir. Sınıf etiketleri içerisinde bulunan sözel değerleri etiket kodlama yöntemi ile sayısal değerlere çevrilmiştir. Bu adımdan sonra veri kalitesini artırmak için normalizasyon işlemi yapılmıştır. Veri normalizasyonu, verilerin her bir özelliğe eşit katkı sağlayacak şekilde ölçeklendiği veya dönüştürüldüğü ön işleme yaklaşımlarından biridir [27]. Bu sayede veriler ortak bir uzaya indirgenir veya dönüştürülür. Veri seti içerisinde bulunan veriler 0 ile 1 arasında bir uzaya indirgenerek veri normalizasyonu yapılmıştır.

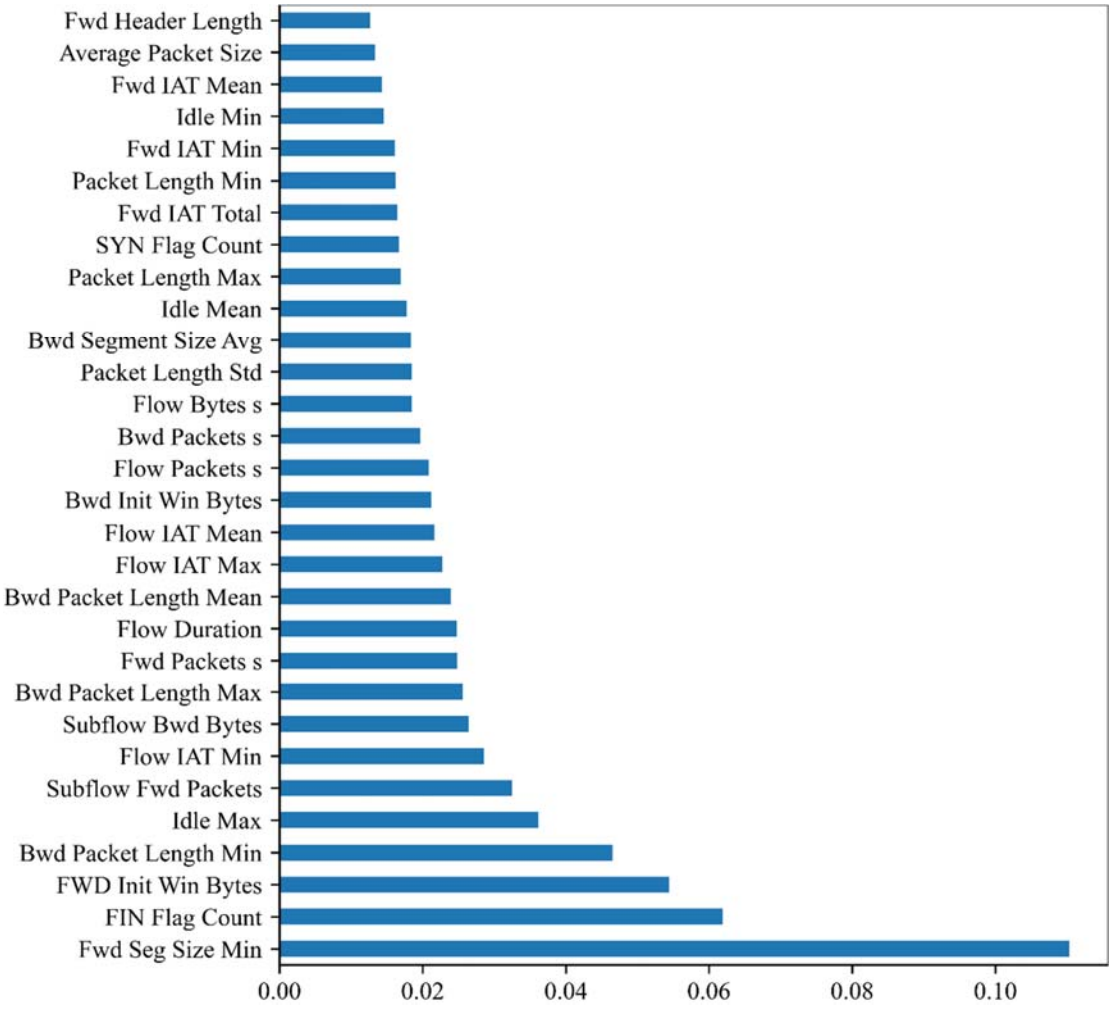
Öznitelik seçimi, alakasız ve gereksiz verileri kaldırarak hesaplama süresini azaltabilen, öğrenme doğruluğunu artırabilen ve öğrenme modeli veya verilerinin daha iyi anlaşılmasını kolaylaştırabilen bir yöntemdir [28]. Bu yöntem ile verilerin boyutunu azaltılabilir, ilgisiz

ve gereksiz öznitelikleri kaldırılabilir ve eğitim süresini kısaltılarak ve öğrenme performansını iyileştirilebilir [29]. Bu amaçlar veri seti içerisindeki öznitelikler sonuca etkisine bakılarak ağırlıklandırılmış ve öznitelik seçimi gerçekleştirilmiştir. Öznitelik seçimi işleminde Extra Tree algoritması kullanılmış ve tüm öznitelik 0-1 arasında ve toplam değeri 1 olacak şekilde ağırlıklandırılmıştır. 14 adet özelliğin ağırlık değeri 0 olduğu görülmüştür ve 76 adet öznitelik arasından 30 adet öznitelik seçilmiştir. Seçilen öznitelikler ve ağırlık derecesini Şekil 1'de gösterilmiştir. Şekil 1'de öznitelik isimleri dikey tarafta ve ağırlık dereceleri yatayda bulunmaktadır.

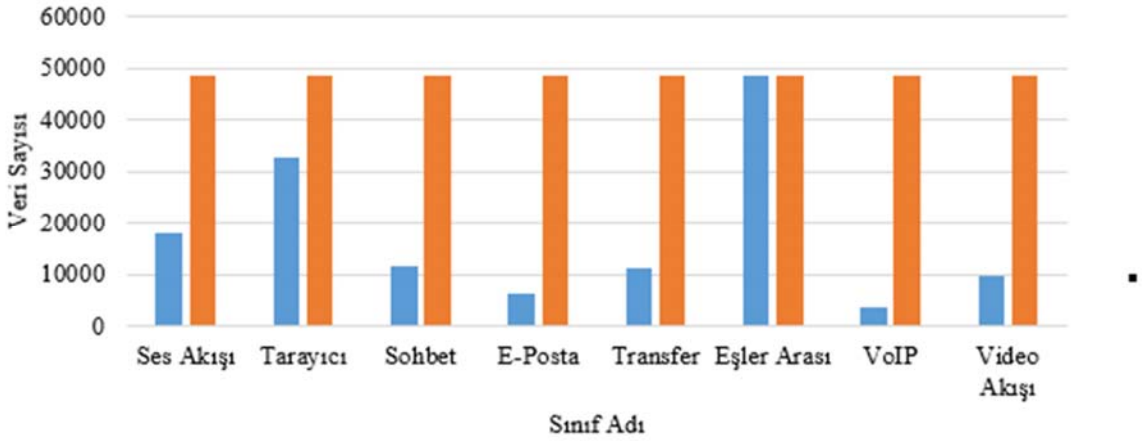
Mevcut problemin model tarafından öğrenilebilmesi için eğitim kümesi ve model performansının test edilebilmesi için test kümesi gerekmektedir. Bu amaçla mevcut veri seti rastgele ve karışık olarak %30 ve %70'lik iki kümeye ayrılmıştır. Büyük küme verileri model eğitiminde, küçük küme verileri model performans testinde kullanılmıştır. Eğitim kümesine ait veriler test aşamasında kullanılmamıştır. Modern makine öğrenimi teknikleri, azınlık sınıfını görmezden gelip çoğunluk sınıfı için hata oranını en aza indirmeye odaklandığı için dengesiz verilerle başa çıkmakta zorlanmaktadır [30]. Sınıf dengesizliği problemleri, verilerdeki küçük sınıfların varlığından dolayı ortaya çıkar ve sınıflandırmada, sınıflandırıcılar çoğunluk sınıfına yönelik önyargılı davranış gösterir [31]. Günümüzde veri sınıflandırmada en çok yaşanan sorunlardan bir tanesi olan dengesiz veri dağılımı nedeniyle nadir olan verilerin sınıflandırılması zor olmaktadır [32]. Çalışmada kullanılan veri seti içerisinde bulunan sınıfların sayısında dengesizlikler bulunmaktadır. Bu problemin çözülebilmesi için ROS algoritması ile veri dengeleme gerçekleştirilmiştir. ROS, az sayıda olan verilerin benzerlerini oluşturarak bu sınıfa ait verilerin tespit oranını yükseltmekte ve modelin genel başarısını artırmaktadır. ROS ile veri dengeleme işlemi yapılmadan önce yapıldıktan sonra oluşan veri sayıları Şekil 2'de gösterilmiştir.

Makine öğrenme algoritmalarının hepsi konfigüre edilmesi gereken hiper-parametrelere sahiptir ve parametrelerin doğru belirlenmesi model başarısını doğrudan etkilemektedir [33]. Hiper-parametrelerin belirlenmesi işlemi zor ve işlem yükü gerektirmektedir [34]. Bu işlemi otomatik araçlar ile yapmak daha başarılı sonuçlar alınmasını ve insan çabasının azaltılmasını sağlamaktadır [35]. Parametre uzayının büyüklüğüne göre farklı yöntemler kullanılmaktadır [36]. Yapılan çalışmada hiper-parametre seçimleri otomatik bir yöntem olan ızgara arama (grid search) kullanılarak yapılmıştır. Bu yöntemde belirlenmesi istenen parametreler ve denenmesi istenen değerler birlikte verilir [37]. Yapılan çalışmada kullanılan algoritmalar için hiper-parametreler ızgara arama yöntemi kullanılarak belirlenmiştir. Kullanılan algoritmalar ve seçilen parametrelere ait değerler Tablo 1'de verilmiştir.

Hiper-parametreler belirlendikten model eğitim aşamasına geçilmiştir. Eğitim kümesi kullanılarak yapılan eğitim sonrasında test küme verileri kullanılarak model test edilmiştir. Model oluşturma aşamaları Şekil 3'te gösterilmiştir.



Şekil 1. Seçilen öznitelikler ve ağırlıkları (Selected features and weights)



Şekil 2. Veri dengeleme (Data balancing)

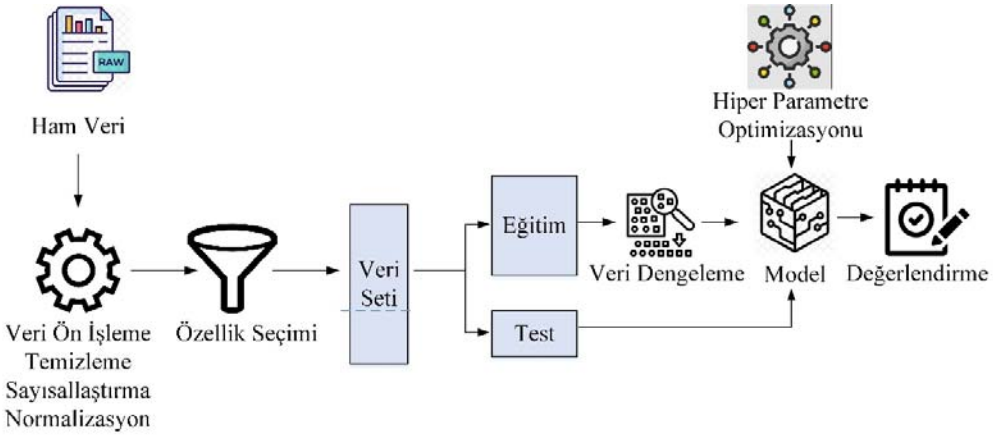
Tablo 1. Izgara araması ile seçilen parametreler (Parameters selected by grid search)

No	Algoritma	Izgara Arama Algoritması Arama Aralığı	Parametreler için seçilen değerler
1	KNN	n_neighbors: (1,20,1) leaf_size: (20,30,1) p: (1,5,1) weights: ('uniform', 'distance'), metric: ('minkowski', 'chebyshev')	'n_neighbors': 1 'leaf_size': 20 'p': 1 'weights': 'uniform' 'metric': 'minkowski'
2	Lojistik Regresyon	C:np.logspace(-3,3,7) penalty:["l1","l2"]	C': 1000.0, 'penalty': 'l2'
3	Rassal Orman	max_depth: [10, 20, None] max_features: ['auto', 'sqrt'] min_samples_leaf: [1, 2, 4] n_estimators: [200, 800, 2000]	max_depth=20 max_features=auto min_samples_leaf=4 n_estimators=200
4	SVM	C: [0.1, 1, 10, 100, 1000] gamma: [1, 0.1, 0.01, 0.001, 0.0001] kernel: ['rbf','linear']	C: 1000, gamma: 0.1, kernel: 'rbf'
5	Karar Ağacı	'criterion':['gini','entropy']	criterion='gini'
6	Gaussian Naive Bayes	'var_smoothing': np.logspace(0,-9, num=100)	'var_smoothing': 0.012
7	Doğrusal Ayrımcı Analiz	solver : ['svd', 'lsqr', 'eigen']	'solver': 'svd'
8	Gradyan Artırma	"learning_rate": [0.01, 0.025, 0.2] "min_samples_split": np.linspace(0.1, 0.3, 3) "min_samples_leaf": np.linspace(0.1, 0.3, 3) "max_depth":[3,8] "max_features":["log2","sqrt"] "subsample":[0.5, 1.0] "n_estimators":[10,100,1000]	'learning_rate': 0.2 'min_samples_split': 0.1 'min_samples_leaf': 0.1 'max_depth': 8 'max_features': 'log2' 'criterion': 'friedman_mse' 'subsample': 1.0 'n_estimators': 10
9	Ekstra Ağaç	'n_estimators': [10,50,100] max_features': ['auto','sqrt','log2']	'max_features': 'log2' 'n_estimators': 100
10	XGBoost	'min_child_weight': [1, 5] 'gamma': [0.5, 2, 5] 'subsample': [0.6, 1.0] 'colsample_bytree': [0.6, 1.0] 'max_depth': [3, 5]	min_child_weight=1 gamma=0.5 subsample=0.8 colsample_bytree=0.6 max_depth=5

3.2. Veri Seti (Dataset)

Karanlık Ağ trafiğinin makine öğrenmesi ile tespit edilebilmesi ve sınıflandırılabilmesi için eğitim ve test verilerine ihtiyaç duyulmaktadır. Yapılan çalışmada Lashkari vd. [16] tarafından üretilen ve açık kaynak olan CICDarknet2020 veri seti kullanılmıştır. Veri seti daha önce oluşturulan ISCXTor2016 ve ISCXVPN2016 veri setlerinin birleştirilmesi ile oluşturulmuştur. Veri seti içerisinde TOR ile şifrelenmiş, TOR olmadan şifrelenmiş, VPN ile şifrelenmiş ve VPN olmadan şifrelenmiş veriler bulunmaktadır.

Veri seti içerisinde iki farklı sınıf etiketi ve iki farklı senaryo bulunmaktadır. İlk senaryoda trafiğin TOR, TOR olmayan, Sanal Özel Ağ (Virtual Private Network, VPN) ve VPN olmayan olarak 4



Şekil 3. Önerilen model mimarisi (Model architecture)

farklı kategoride sınıflandırılması yapılmıştır. İkinci senaryoda ise trafik bu 4 ana kategori altında Ses, Tarayıcı, Sohbet, E-Posta, Eşler Arası, Transfer, Video ve VoIP alt kategorilerine sınıflandırılmıştır. Veri seti içerisinde bulunan sınıflar ve bu sınıflar toplanırken kullanılan uygulamalar Tablo 2'de gösterilmiştir.

Tablo 2. Veri seti sınıf ve kullanılan uygulamalar
(Data set class and applications)

Sınıf Adı	Kullanılan Uygulama
Ses Akışı	Vimeo ve Youtube
Tarayıcı	Firefox ve Chrome
Sohbet	ICQ, AIM, Skype, Facebook, Hangouts
E-Posta	SMTPS, POP3S, IMAPS
Eşler Arası	uTorrent, BitTorrent
Transfer	Skype, Filezilla FTP over SSH, FTP over SSL
Video Akışı	Vimeo ve Youtube
VoIP	Facebook, Skype ve Hangouts sesli arama

Veri seti içerisinde 82 adet öznitelik ve 2 adet sınıf kategorisi bulunmaktadır. Öznitelikler arasında trafiğin kaynak ve hedef bilgileri ile paket sayısı, paket boyutları, paketlerin ağ üzerinde gidiş / geliş zamanları, aktif ve boşta geçen süre gibi istatistiksel özniteliklerden oluşmaktadır. Veri seti dağılımları incelendiğinde veri seti içerisinde 1392 adet TOR, 93356 adet Non-TOR, 22919 adet VPN ve 23863 adet Non-VPN sınıfına ait veri bulunmaktadır. Alt kategoriler incelendiğinde en çok örnek 48300 adet ile Non-TOR P2P sınıfına ve en az örnek 13 adet ile TOR E-Posta sınıfına aittir.

3.3. Performans Ölçümü (Performance Measurement)

Makine öğrenmesi algoritmaları ve modellerinin performanslarının ölçülebilmesi ve değerlendirilebilmesi için birçok metrik bulunmaktadır. Bu metriklere bakılarak önerilen model veya sistemin mevcut problemin çözümünü sağlayıp sağlamadığı anlaşılmaktadır. Doğruluk bu metriklerden en çok kullanılandır [38]. Fakat doğruluk, veri seti dengesiz olduğu durumlarda yanlış veya hatalı sonuç verebilmektedir. Bu durumlarda doğruluk ile kesinlik, hassasiyet ve f1-ölçüt değerleri de kullanılmaktadır [38]. Bu metriklerin hesaplanabilmesi için modelin yaptığı doğru ve yanlış sınıflandırma değerleri ve karmaşıklık matrisi kullanılmaktadır. Karmaşıklık matrisi, bir sınıflandırma sistemi tarafından yapılan gerçek ve tahmin değerleri içeren bir kavramdır [39]. Karmaşıklık matrisinin iki boyutu vardır ve bir boyutu gerçek değerleri içerirken diğer boyutu tahmin edilen değerleri içermektedir. Tablo 3'de karmaşıklık matris tablosu verilmiştir. Model, pozitif sınıfı doğru tahmin ettiğinde bu sonuca Doğru Pozitif (DP) denir. Benzer şekilde negatif sınıfı doğru tahmin ettiğinde ise bu sonuca Doğru Negatif (DN) denir. Eğer model negatif olan veriyi pozitif olarak tahmin ederse buna Yanlış Pozitif (YP) denir. Pozitif olan veri model tarafından negatif olarak belirlenirse buna da Yanlış Negatif (YN) denir. DP ve DN model başarısını ifade ederken, YP ve YN modelin başarısız olduğu durumları temsil etmektedir.

Tablo 3. Karmaşıklık matrisi (Confusion matrix)

		Gerçek Değer	
		Pozitif	Negatif
Model Tahmin Değeri	Pozitif	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	Negatif	Yanlış Negatif (YN)	Doğru Negatif (DN)

Doğruluk, doğru olarak tahmin edilen verilerin sayısının veri seti içerisinde bulunan tüm verilerin sayısına oranı ile hesaplanmaktadır [39]. Doğruluk metrik değerinin hesaplama formülü Eş. 1'de verilmiştir.

$$Doğruluk = \frac{DP+DN}{Tüm\ Veri\ Seti} \quad (1)$$

Kesinlik, model tarafından pozitif olarak tahmin edilen değerlerin gerçekte ne kadarının pozitif olduğunu gösteren bir ölçüttür ve formülü Eş. 2'de verilmiştir. Hassasiyet ise gerçekte pozitif olan verilerin model tarafından ne kadarının tahmin edilebildiği ifade eden bir metriktir. Eş. 3'te hassasiyet formülü gösterilmiştir. F1-ölçüt ise kesinlik ve hassasiyet değerlerinin tek bir sonuç ile ifade edilebilmesi için harmonik ortalamasının alındığı bir metriktir ve formülü Eş. 4'te gösterilmiştir.

$$Kesinlik = \frac{DP}{DP+YP} \quad (2)$$

$$Hassasiyet = \frac{DP}{DP+YN} \quad (3)$$

$$F-ölçütü = \frac{2*Hassasiyet*Kesinlik}{Hassasiyet+Kesinlik} \quad (4)$$

4. Deneysel Sonuçlar (Experimentals Results)

Kullanılan veri seti TOR ve VPN veri setlerini birleştirilmesinden oluşturulan bir veri setidir. Veri seti içerisinde iki adet sınıf etiketi ve buna bağlı olarak iki adet senaryo bulunmaktadır. Bu bölümde her bir senaryo için çalışmada kullanılan makine öğrenmesi algoritmalarının bulduğu sonuçlar verilmiş ve karşılaştırmalar yapılmıştır.

4.1. TOR ve VPN Etiketlerine Göre Sınıflandırma (Classification by TOR and VPN Labels)

Bu senaryoda veri seti içerisinde bulunan ilk sınıf etiketi kullanılmıştır ve trafiğin TOR, Non-TOR, VPN ve Non-VPN olarak sınıflandırılması yapılmıştır. Trafik 4 farklı sınıfa ayrılarak çoklu sınıflandırma işlemi gerçekleştirilmiştir. Kullanılan 10 farklı algoritmadan XGBoost algoritması en başarılı sonucu vermiştir. XGBoost algoritması Non-TOR trafiğini %99,81, TOR trafiğini %94,63, Non-VPN trafiğini %94,44 ve VPN trafiğini %94,58 doğruluk oranı ile sınıflandırmıştır. XGBoost algoritmasına ait karmaşıklık matrisi Tablo 4'te verilmiştir.

Tablo 4. XGBoost karmaşıklık matrisi (XGBoost confusion matrix)

		Non-TOR	Non-VPN	TOR	VPN
		Tahmin Edilen Değer			
Gerçek Değer	Non-TOR	28045	6	0	14
	Non-VPN	13	6873	17	202
	TOR	1	32	389	3
	VPN	24	358	3	6479

Karmaşıklık matrisi analiz edildiğinde modelin VPN trafiğini TOR ile değil kendi içinde VPN veya Non-VPN olarak yanlış sınıflandırdığı görülmektedir. Bu durum diğer dokuz algoritmada da aynı şekilde olduğu tespit edilmiştir. Tablo 5'te 10 algoritmanın ağırlıklı doğruluk, kesinlik, hassasiyet ve f1-ölçüt değerleri verilmiştir.

4.2. Alt Sınıf Etiketlerine Göre TOR ve VPN Trafiğinin Sınıflandırılması (Classification of TOR and VPN Traffic According to Subclass Labels)

Bu senaryoda TOR, Non-TOR, VPN ve Non-VPN trafiğe ait 27 adet sınıf için çoklu sınıflandırma işlemi gerçekleştirilmiştir. Senaryo 2'de veri ön işleme, normalizasyon ve öznitelik seçimi yapıldıktan sonra veri dengeleme işlemi yapılmadan 10 farklı algoritma çalıştırılmıştır. Sonuçlar Tablo 6'da gösterilmiş ve XGBoost algoritmasının en yüksek başarı oranına sahip olduğu görülmüştür. Veri dengeleme işlemi yapılmadan yapılan testlerde XGBoost, Rassal Orman, Karar Ağacı, Gradyan Artırma ve Ekstra Ağaç algoritmalarının diğer 5 algoritmadan daha iyi sonuçlar vermiştir. En iyi beş algoritma seçilerek ROS yöntemi ile veri dengeleme işlemi yapılmıştır ve sonuçlar Tablo 7'de verilmiştir.

Veri dengeleme işlemi yapıldıktan sonra Gradyan Artırma algoritmasını başarı oranı düştüğü, XGBoost algoritmasının başarı oranı değişmediği ve diğer 3 algoritmanın ise başarı oranı arttığı gözlemlenmiştir. Karar Ağacı ve Ekstra Ağaç algoritmaları en yüksek başarı oranı ile sınıflandırma işlemini gerçekleştirmiştir. 5 farklı

Tablo 5. Senaryo 1 performans ölçüt değerleri (Scenario 1 performance results)

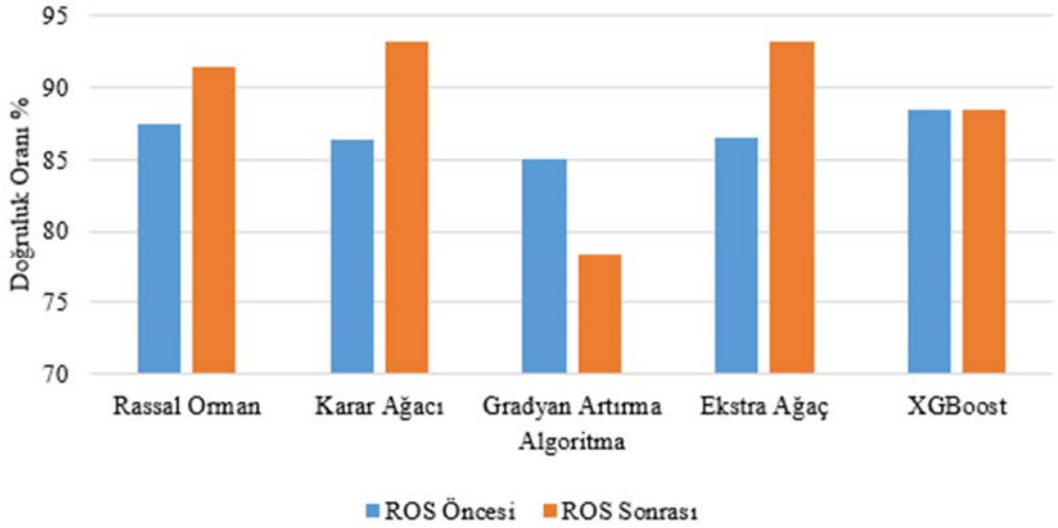
Algoritma	Doğruluk	Kesinlik	Hassasiyet	F1-Ölçüt
KNN	0,967	0,967	0,967	0,967
Lojistik Regresyon	0,832	0,828	0,832	0,829
Rassal Orman	0,982	0,982	0,982	0,982
SVM	0,925	0,923	0,925	0,924
Karar Ağacı	0,978	0,978	0,978	0,978
Gaussian Naive Bayes	0,691	0,699	0,691	0,662
Doğrusal Ayrımcı Analiz	0,792	0,78	0,792	0,783
Gradyan Artırma	0,974	0,974	0,974	0,974
Ekstra Ağaç	0,979	0,979	0,979	0,979
XGBoost	0,984	0,984	0,984	0,984

Tablo 6. Senaryo 2 veri dengeleme olmadan performans ölçüt değerleri (Scenario 2 performance results without data balancing)

Algoritma	Doğruluk	Kesinlik	Hassasiyet	F1-Ölçüt
KNN	0,824	0,823	0,824	0,817
Lojistik Regresyon	0,632	0,606	0,632	0,592
Rassal Orman	0,875	0,873	0,875	0,874
SVM	0,76	0,764	0,76	0,74
Karar Ağacı	0,864	0,864	0,866	0,865
Gaussian Naive Bayes	0,45	0,523	0,45	0,422
Doğrusal Ayrımcı Analiz	0,597	0,561	0,597	0,561
Gradyan Artırma	0,851	0,849	0,851	0,845
Ekstra Ağaç	0,865	0,863	0,865	0,864
XGBoost	0,884	0,883	0,884	0,883

Tablo 7. Senaryo 2 veri dengeleme sonrası performans ölçüt değerleri (Scenario 2 performance results after data balancing)

Algoritma	Doğruluk	Kesinlik	Hassasiyet	F1-Ölçüt
Rassal Orman	0,914	0,926	0,914	0,915
Karar Ağacı	0,932	0,934	0,932	0,932
Gradyan Artırma	0,784	0,794	0,784	0,782
Ekstra Ağaç	0,932	0,934	0,932	0,932
XGBoost	0,884	0,894	0,884	0,884



Şekil 4. Veri dengeleme öncesi ve sonrası karşılaştırma (Comparison before and after data balancing)

algoritma için veri dengeleme işlemi yapılmadan önce ve yapıldıktan doğruluk oranı değişimi Şekil 4'te gösterilmiştir.
Çalışma Intel Core i7 işlemciye, 2 GHz işlemci hızına, 8 GB belleğe ve 4 GB ekran kartına sahip bir dizüstü bilgisayarda gerçekleştirilmiştir. En başarılı sonuç bulan Karar Ağacı ve Ekstra Ağaç algoritmaları zaman bazlı karşılaştırılmıştır. Karar Ağacı algoritması öğrenme ve test işlemini 5,884 saniyede, Ekstra Ağaç algoritması test ve öğrenme işlemini ise 49,695 saniyede gerçekleştirmiştir. Karar Ağacı, Ekstra Ağaç algoritmasına göre yaklaşık 10 kat daha hızlı çalıştığı gözlemlenmiştir. Yapılan tüm testler sonucunda hem başarı oranı hem de çalışma hızı açısından değerlendirildiğinde Karar Ağacı algoritması en başarı sonucu vermiştir ve bu model için tercih edilmiştir.

5. Sonuçlar ve Tartışmalar (Results and Discussions)

Karanlık Ağ tarafından sağlanan şifreleme ve anonimlik sayesinde siber suçlular ve saldırganlar kendilerini gizlemek için bu ağı yoğun olarak kullanmaktadır. Normal tarayıcı ve internet üzerinden erişilemeyen bu ağ üzerinde yapılan her hareket şüpheli olarak tanımlanmaktadır. Siber güvenlik uzmanları ve ağ uzmanları için bu ağlara yapılan erişimlerin tespiti ve engellenmesi siber güvenlik için büyük önem arz etmektedir. Kurumsal bilgi güvenliğinin sağlanabilmesi için görünürlük oldukça önemlidir [40]. Fakat karanlık ağ trafiğini şifreli olması nedeni ile mevcut güvenlik çözümleri ile tespit edilememektedir.

Yapılan bu çalışmada şifreli karanlık ağ trafiği deşifreleme işlemi yapılmadan ağ paketlerinin istatistiksel öznitelikleri kullanılarak analiz edilmiş ve sınıflandırılmıştır. Çalışmada gerçek trafik üzerinden oluşturulan ve açık kaynak olan CICDarknet2020 veri seti tercih edilmiştir. Veri seti içerisinde boş ve sonsuz değerler bulunduğu için veri önişleme uygulanarak veri seti düzenlenmiştir. Veri seti içerisinde bulunan 82 adet öznitelik içerisinde ağırlıklandırma işlemi yapılarak 30 adet öznitelik seçilmiştir. Öznitelik seçimi ile verilerin boyutunu azaltılmış, ilgisiz ve gereksiz öznitelikleri kaldırılmış ve eğitim süresi kısaltılarak öğrenme performansı artırılmıştır. Veri seti içerisinde bulunan verilerin sınıf dağılımında çok büyük oranda dengesizlik bulunmaktadır. Bu durum modelin performansını etkilediği için ROS tekniği ile veri dengeleme yapılmıştır. ROS işlemi sonrası modelin başarı performansında artış görülmüştür. Makine öğrenme algoritmalarında önemli adımlarından bir tanesi hiper-parametrelerin belirlenmesidir. En doğru hiper-parametrelerin seçilebilmesi için ızgara yöntemi ile parametre seçimi işlemi gerçekleştirilmiştir. Yapılan bu işlemler ile model performansı ve hızı artırılmıştır.

Analiz ve sınıflandırma işlemi için 10 farklı makine öğrenme algoritmaları kullanılmıştır. Trafiğin TOR, Non-TOR, VPN ve Non-VPN olarak çoklu sınıflandırıldığı ilk senaryoda XGBoost algoritması en başarılı sonucu vermiştir. XGBoost algoritması Non-TOR trafiğini %99,81, TOR trafiğini %94,63, Non-VPN trafiğini %94,44 ve VPN trafiğini %94,58 doğruluk oranı ile sınıflandırmıştır.

İkinci senaryoda trafik ses akışı, tarayıcı, sohbet, e-posta, transfer, eşler arası, video akışı ve VoIP olarak 8 farklı kategoride sınıflandırılmıştır. Bu senaryoda veri seti dengeleme işlemi yapılmadan önce sınıflandırma işlemi yapılmış ve XGBoost, Rassal Orman, Karar Ağacı, Gradyan Artırma ve Ekstra Ağaç algoritmaları diğer 5 algoritmadan daha başarılı sonuç verdiği gözlemlenmiştir. Bu 5 algoritma seçilerek ROS yöntemi ile veri dengeleme yapılmış ve tekrar sınıflandırma işlemi gerçekleştirilmiştir. Karar Ağacı ve Ekstra Ağaç algoritmaları %93,32 doğruluk oranı ile diğer 3 algoritmaya göre daha başarılı sonuç verdiği gözlemlenmiştir. Veri dengeleme işleminin model başarı oranını artırdığı görülmüştür. Bu iki algoritma oluşturulacak modelin maliyet ve işlem gücü açısından karşılaştırılmış ve Karar Ağacı algoritmasının Ekstra Ağaç algoritmasına göre yaklaşık 10 kat daha hızlı çalıştığı görülmüştür. Bu nedenle en başarılı algoritma olarak Karar Ağacı seçilmiştir. Önerilen modelin karanlık ağı içerisindeki trafiği yüksek başarı oranı ile tespit ettiği görülmüştür. Bu model ile Arash Habibi Lashkari [22] tarafından yapılan çalışmadan %10 ve Yuzong Hu [24] tarafından yapılan çalışmadan yaklaşık %9 daha başarılı bir sonuç elde edilmiştir.

Siber suç oranlarının artması ve ele geçirilen bilgi ve belgelerin Karanlık Ağ üzerinden satılması nedeni ile bu ağa giden trafiğin analiz edilmesi kişi, kurum ve ulusal güvenlik için oldukça önemlidir. Yapılan çalışma ile derin paket analizi yapılmadan sadece paketin istatistiki verileri analiz edilerek bu ağa giden trafik yüksek doğruluk oranı tespit edilmiş ve sınıflandırılmıştır. Mevcut güvenlik çözümlerinin bu ağa giden trafiği deşifreleme yapmadan tespit

edemeyeceği için önerilen model ile bütünleşik çalışması siber güvenlik seviyesini artıracaktır. Model üzerinden gelen alarmlar güvenlik duvarı, saldırı engelleme sistemi ve web güvenlik duvarı gibi siber güvenlik ürünleri tarafından algılanarak bu ağa giden trafiğin engellenmesi sağlanabilir. Gelecek çalışmalarda sadece trafik sınıfı değil kullanılan uygulamalarında tespit edilmesi hedeflenmektedir. Bu amaç için yeni bir veri seti oluşturulabilir. Karanlık ağa giden trafiğin kaynak uygulamasının tespiti de siber güvenlik ihlal olaylarının analiz edilmesi için oldukça faydalı olacaktır.

Kaynaklar (References)

1. Moore R., Cyber crime: Investigating High-Technology Computer Crime, Anderson Publishing, Mississippi, 2005.
2. Okutan A., Çebi Y., A Framework for Cyber Crime Investigation, Procedia Computer Science, 158, 287-294, 2019.
3. Holt T.J., Bossler A.M., Seigfried-Spellar K.C., Cybercrime and Digital Forensics, Routledge, New York, 2018.
4. Sağıroğlu Ş., Alkan M., Siber Güvenlik ve Savunma, Grafiker Yayınları, Ankara, 2018.
5. Meland P.H., Bayoumy Y.F.F., Sindre G., The Ransomware-as-a-Service economy within the darknet, Computers & Security, 92 (101762), 1-9, 2020.
6. Bancroft A., The Darknet and Smarter Crime, Palgrave Macmillan, Cham, 2020.
7. Rathod D., Darknet Forensics, International Journal of Emerging Trends & Technology in Computer Science, 6 (4), 77-79, 2017.
8. Ling Z., Luo J., Yu W., Fu X., Jia W., Zhao W., Protocol-level attacks against Tor, Computer Networks, 57, (4), 869-886, 2013.
9. Yang Q., Gasti P., Balagani K., Li Y., Zhou G., USB side-channel attack on Tor, Computer Networks, 141, 57-66, 2018.
10. Owenson G., Cortes S., Lewman A., The darknet's smaller than we thought: The life cycle of Tor Hidden Services, Digital Investigation, 27, 17-22, 2018.
11. Dingledine R., Mathewson N., Syverson P., Tor: The Second-Generation Onion Router, 13, 1-17, 2004.
12. Mansfield-Devine S., Darknets, Computer Fraud & Security, 12, 4-6, 2009.
13. Bou-Harb E., Debbabi M., Assi C., Cyber Scanning: A Comprehensive Survey, IEEE Communications Surveys & Tutorials, 16 (3), 1496-1519, 2014.
14. Canbek G., Sağıroğlu Ş., Malware and Spyware: A Comprehensive Review, Journal of the Faculty of Engineering and Architecture of Gazi University, 22 (1), 121-136, 2007.
15. Utku A., Doğru İ.A., Permission based detection system for android malware, Journal of the Faculty of Engineering and Architecture of Gazi University, 32 (4), 1015-1024, 2017.
16. Lashkari A.H., Kaur G., Rahali A., DIDarknet: A Contemporary Approach to Detect and Characterize the Darknet Traffic using Deep Image Learning, 10th International Conference on Communication and Network Security, Tokyo, 1-13, November, 2020.
17. Barker J., Hannay P., Szewczyk P., Using traffic analysis to identify The Second Generation Onion Router, IFIP Ninth International Conference on Embedded and Ubiquitous Computing, Melbourne, 72-78, 2011.
18. Shahbar K., Zincir-Heywood A.N., Benchmarking Two Techniques for Tor Classification, IEEE Symposium on Computational Intelligence in Cyber Security, Orlando-USA, 1-8, 9-12 December, 2014.
19. Almubayed A., Hadi A., Atoum J., A Model for DetectingTor Encrypted Traffic using Supervised Machine Learning, Computer Network and Information Security, 7, 10-23, 2015.
20. Ali S.H.A., Ozawa S., Ban T., Nakazato J., Shimamura J., A neural network model for detecting DDoS attacks using darknet traffic features, International Joint Conference on Neural Networks, Vancouver-Canada, 2979-2985, 24-29 July, 2016.
21. Hodo E., Bellekens X., Iorkyase E., Hamilton A., Tachtatzis C., Atkinson R., Machine Learning Approach for Detection of nonTor Traffic, International Conference on Availability, Reliability and Security, Regio Callabria-Italy, 29 Agust – 1 September, 2017.
22. Lashkari A.H., Draper-Gil G., Mamun M.S.I., Ghorbani A.A., Characterization of Tor Traffic Using Time Based Features, International Conference on Information System Security and Privacy, Porto-Portugal, 19-21 February, 2017.
23. Cuzzocrea A., Martinelli F., Mercaldo F., Vercelli G., Tor Traffic Analysis and Detection, IEEE International Conference on Big Data , Boston-USA, 11-14 December, 2017.
24. Hu Y., Zou F., Li L., Yi P., Traffic Classification of User Behaviors in Tor, I2P, ZeroNet, Freenet, 19th International Conference on Trust, Security and Privacy in Computing and Communications, Guangzhou-China, 29-31 December, 2020.
25. Gurunarayanan A., Agrawal A., Bhatia A., Vishwakarma D.K., Improving the performance of Machine LearningAlgorithms for TOR detection, International Conference on Information Networking, Jeju Island-Korea, 13-16 January, 2021.
26. Huang J., Li Y., Xie M., An empirical analysis of data preprocessing for machine learning-based software cost estimation, Information and Software Technology, 67, 108-127, 2015.
27. Singh D., Singh B., Investigating the impact of data normalization on classification performance, Applied Soft Computing, 97, (B), 1-23, 2020.
28. Cai J., Luo J., Wang S., Yang S., Feature selection in machine learning: A new perspective, Neurocomputing, 300, 70-79, 2018.
29. Sheikhpour R., Sarram M.A., Gharaghani S., Chahooki M.A.Z., A Survey on semi-supervised feature selection methods, Pattern Recognition, 64, 141-158, 2017.
30. Thabtah F., Hammoud S., Kamalov F., Gonsalves A., Data imbalance in classification: Experimental evaluation, Information Sciences, 513, 429-441, 2020.
31. Ali H., Najib M.B., Salleh M., Saedudin R., Hussain K., Imbalance class problems in data mining: A review, Indonesian Journal of Electrical Engineering and Computer Science, 14, (3), 1552-1563, 2019.
32. Rustogi R., Prasad A., Swift Imbalance Data Classification using SMOTE and Extreme Learning Machine, International Conference on Computational Intelligence in Data Science, Chennai, 6-7 September, 2019.
33. Li S.A.Y., On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice, Neurocomputing, 415, 295–316, 2020.
34. Tran N., Schneider J., Weber I., Qin A.K., Hyper-parameter optimization in classification: To-do or not-to-do, Pattern Recognition, 103, 2020.
35. Hutter F., Kotthoff L., Vanschoren J., Automated Machine, Springer, Cham, 2019.
36. Gülcü A., Kuş Z., Konvolüsyonel Sinir Ağlarında Hiper-Parametre Optimizasyonu Yöntemlerinin İncelenmesi, Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji, 7, (2), 503-522, 2019.
37. Tanyıldızı E., Demirtaş F., Hiper Parametre Optimizasyonu Hyper Parameter Optimization, 1st International Informatics and Software Engineering Conference, Ankara-Turkey, 1-5, 6-7 November, 2019.
38. Uddin M.F., Addressing Accuracy Paradox Using Enhanched Weighted Performance Metric in Machine Learning, Sixth HCT Information Technology Trends, Ras Al Khaimah-United Arab Emirates, 319-324, 20-21 November 2019.
39. Deng X., Liu Q., Deng Y., Mahadevan S., An improved method to construct basic probability assignment based on the confusion matrix for classification problem, Information Sciences, 340, 250-261, 2016.
40. Vural Y., Sağıroğlu Ş., A review on enterprise information security and standards, Journal of the Faculty of Engineering and Architecture of Gazi University, 23, (2), 507-522, 2008.