Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Makine Öğrenmesi Yöntemleriyle Anormal Ağ Trafiğinin Tespit Edilmesi

Serhat ÖZEKES [a,*], Elif Nur KARAKOÇ [a]

[a] *Bilgisayar Mühendisliği Bölümü, Mühendislik ve Doğa Bilimleri Fakültesi, Üsküdar Üniversitesi, İstanbul, TÜRKİYE*

** Sorumlu yazarın e-posta adresi: serhat.ozekes@uskudar.edu.tr*

ÖZET

Bilgisayar ağlarının ve geliştirilen uygulamaların büyümesi ile saldırıların oluşturacağı hasarın belirgin olarak artması beklenmektedir. Saldırı Tespit Sistemleri (STS) sürekli büyüyen ağ saldırıları karşısında önemli savunma araçlarındandır. Saldırı Tespit Sistemlerinin makine öğrenmesi algoritmaları ile eğitilmesi ve eğitim sonrası gerçek zamanlı olarak saldırıları oluştuğu anda tespit ederek, gerekli tedbirlerin alınmasını sağlaması amaçlanmaktadır. Bu çalışmada da karar ağacı ve rastgele orman yöntemleri kullanılarak bilgisayar ağlarında akan normal ve anormal paketlerin sınıflandırılması amaçlanmaktadır. Sınıflandırma yöntemleri, karar vermek için ağ trafiğinin kaydedildiği PCAP dosyasından CICFlowMeter kullanılarak çıkarılan 78 adet değişkeni kullanmaktadır. Sonuçlar incelendiğinde, önerilen yöntemin bir milyonun üzerindeki kaydı %100'e yakın bir başarıyla sınıflandırdığı ve anormal trafiğin tespitinde etkin olduğu görülmektedir.

Anahtar Kelimeler: *Saldırı Tespit Sistemleri, Karar Ağacı, Rastgele Orman*

Detection of Abnormal Network Traffic by Machine Learning Methods

ABSTRACT

With the growth of computer networks and developed applications, it is expected that the damage caused by the network attacks will increase significantly. Intrusion Detection Systems (IDS) is one of the most important defense tools in avoiding growing network attacks. Intrusion Detection Systems are trained with the machine learning algorithms and after the training, it is aimed to detect the attacks in real time and to take the necessary measures. In this study, it is aimed to classify normal and abnormal packages flowing in computer networks using decision tree and random forest methods. The classification methods use 78 variables which are extracted from the PCAP file where the network traffic is recorded. When the results are examined, it is seen that the proposed method classifies more than one million records with close to 100% success and is effective in detecting abnormal traffic.

Keywords: *Intrusion Detection System, Decision Tree, Random Forest*

Geliş: 17/12/2018, Düzeltme: 28/01/2019, Kabul: 30/01/2019

I. GİRİŞ

Güvenlikle ilgili tehditlerin sayısının ve türlerinin hızla artmasıyla birlikte, güvenlik teknolojilerinde de hızlı bir gelişim yaşanmaktadır. Güvenlik duvarları, anti virus ve saldırı tespit sistemleri gibi yazılımsal veya donanımsal araçlar geliştirilmiştir. Ağ sistemi, saldırganlardan ya da hackerlerden önemli verileri ve sistemleri korumak için bu güvenlik yazılımlarından bir ya da birkaçını kullanmaktadır. Tek başına bir güvenlik duvarı sistemine güvenmek, kurumsal ağlara ya da kişisel ağlara yönelik saldırıları engellemek için yeterli değildir. Bu nedenle, güvenlik duvarının açıklarını kapatmak için aynı zamanda saldırı tespit sistemi de kullanılır [1]. Saldırı tespit sistemleri (STS), tüm tedbirlere karşın bilgisayar sistemlerine yapılan saldırılar gerçekleşirken ya da gerçekleştikten sonra tespit etmek, İnternet veya yerel ağdan gelebilecek, ağdaki sistemlere zarar verebilecek, çeşitli paket ve verilerden oluşan bu saldırıları fark etmek üzere tasarlanmış sistemlerdir. Günümüzde, pek çok araştırmacı, daha etkin saldırı tespit sistemi gerçekleştirilmesi amacıyla çalışma yapmaktadır. Bu amaçla literatürde farklı makine öğrenme teknikleri ile gerçekleştirilmiş çeşitli saldırı tespit sistemleri bulunmaktadır.

Chowdhury ve Ferens tarafından yapılan çalışmada Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) veri kümesindeki 47 özellik birleştirilerek kullanılmış ve anormal sınıflandırması için Karar Vektör Makineleri (KVM) eğitilmiştir [2]. Bu çalışmada KVM %88.03 doğruluk oranıyla sınıflandırma yapmıştır. Karslıgel vd. tarafından önerilen anormallik tespit sistemi NSL-KDD veri kümesi üzerinde, yarı-eğitmenli k-ortalama kümeleme algoritması kullanılarak geliştirilmiş ve %80.119 doğruluk oranı sunmuştur [3]. Ağ saldırı tespiti için derin öğrenme yaklaşımının KDD Cup ’99 ve NSL-KDD veri kümeleri üzerinde uygulandığı Shone vd.’nin çalışmasında, anormal durumlar % 98’ya varan doğrulukla tespit edilmiştir [4]. Javaid vd. [5] derin öğrenme tabanlı çalışmalarında NSL-KDD veri kümesini kullanmışlar ve %75.76’lık bir performans elde etmişlerdir.

STS alanıyla ilgili araştırmalar yıllar boyunca, daha iyi STS sistemleri önermek için geliştirildi. Ancak, birçok araştırmacı test etmek ve değerlendirmek için kapsamlı ve geçerli veri kümeleri bulmakta zorlandığı için yapılan çalışmalar benzer veri kümeleri üzerinde tekrarlanmıştır. Kötü amaçlı yazılım evrimi ve saldırı stratejilerindeki sürekli değişikliklerden dolayı, bu referans veri kümelerini kullanan çalışmalar günümüz problemlerini çözmekte yetersiz kalabilmektedir. Önerilen bu çalışmada, toplumla paylaşılan en güncel veri kümelerinden biri olan CICIDS2017 veri seti kullanılarak güncel ağ saldırılarının yüksek doğrulukla tespit edilmesi amaçlanmıştır. Bu amaçla rastgele orman ve karar ağacı makine öğrenmesi yöntemlerini kullanan yazılım, Python scikit-learn kütüphanesi kullanılarak yazılmıştır.

II. MALZEME ve YÖNTEM

A. VERİ SETİ

Bu çalışmada kullanılan veri kümesi Kanada Siber Güvenlik Enstitüsü tarafından oluşturulmuş olan CICIDS2017 veri kümesidir [6]. Bu veri kümesi oluşturulurken daha önce literatürde bulunan veri kümelerinde bulunan eksiklikler dikkat alınarak oluşturulmuştur. Ayrıca gerçek dünya kriterlerini karşılayan normal ve saldırı ağ trafiği içeren veri seti oluşturulması amaçlanmıştır. Normal ve saldırı trafiği PCAP dosya formatında kayıt edilmiştir. CICFlowMeter kullanılarak PCAP dosyasından 78

adet trafik özelliği çıkarılarak veri kümesi oluşturulmuştur. Saldırı çizelgesine dayanarak üretilen ağ trafiği etiketlenmiştir. Trafik özellikleri ve açıklamaları Tablo 1'de sunulmuştur.

Önerilen çalışma kapsamında CICIDS2017 veri kümesinde bulunan Hizmet Engelleme (Dos), Dağıtılmış Hizmet Reddi (DDoS) ve Port Tarama (PortScan) saldırıları anormal olarak etiketlendirilmiştir. Hizmet Engelleme saldırıları genel olarak TCP/IP protokol yapısındaki açıklardan faydalanılarak bir sunucuya birden çok bağlantı isteği göndererek kullanıcıların hizmet almasını engellemeye yönelik yapılır [7]. Dağıtılmış Hizmet Reddi saldırılarında ise daha önceden ele geçirilmiş olan bilgisayarlar kullanılarak hedef sunucuya aşırı miktarda ağ trafiği gönderilmektedir. Port Tarama saldırıları da bir sunucunun ya da herhangi bir makinanın geçerli IP adreslerini, aktif portlarını veya işletim sistemini öğrenmek için geliştirilmiştir. Normal trafik olarak da HTTP, HTTPS, FTP, SSH ve e-posta protokol trafiği etiketlenmiştir.

Veri üzerinde seçme, birleştirme ve ortalama alınarak eksik değerleri tamamlama gibi ön-işleme aşamaları gerçekleştirilmiştir. Sonuç olarak veri kümesi 629074 adet normal ve 413483 adet normal olmayan olmak üzere toplam 1042557 adet kayıt içermektedir.

Tablo 1. *Sınıflandırmada kullanılan trafik özellikleri ve açıklamaları*

Özellik Adı	Açıklama	Özellik Adı	Açıklama
Destination Port	Hedef port	**Packet Length Mean**	Bir akışın ortalama uzunluğu
Flow Duration	Mikro-saniyedeki akış süresi	**Packet Length Std**	Bir akışın standart sapması
Total Fwd Packets	İleri yönde toplam paket sayısı	**Packet Length Variance**	Bir akışın uzunluk varyansı
Total Backward Packets	Geri yönde toplam paket sayısı	**FIN Flag Count**	FIN içeren paket sayısı
Total Length of Fwd Packets	İleri yönde paketlerin toplam uzunluğu	**SYN Flag Count**	SYN içeren paket sayısı
Total Length of Bwd Packets	Geri yönde paketlerin toplam uzunluğu	**RST Flag Count**	RST içeren paket sayısı
Fwd Packet Length Max	İleri yönde paketlerin maksimum uzunluğu	**PSH Flag Count**	PUSH içeren paket sayısı
Fwd Packet Length Min	İleri yönde paketlerin minimum uzunluğu	**ACK Flag Count**	ACK içeren paket sayısı
Fwd Packet Length Mean	İleri yönde paketlerin ortalama uzunluğu	**URG Flag Count**	URG içeren paket sayısı
Fwd Packet Length Std	İleri yönde paket uzunluklarının standart sapması	**CWE Flag Count**	CWE içeren paket sayısı
Bwd Packet Length Max	Geri yönde paketlerin maksimum uzunluğu	**ECE Flag Count**	ECE içeren paket sayısı
Bwd Packet Length Min	Geri yönde paketlerin minimum uzunluğu	**Down/Up Ratio**	İndirme ve yükleme oranı
Bwd Packet Length Mean	Geri yönde paketlerin ortalama uzunluğu	**Average Packet Size**	Ortalama paket boyutu

Tablo 1. (devam) *Sınıflandırmada kullanılan trafik özellikleri ve açıklamaları*

Bwd Packet Length Std	Geri yönde paket uzunluklarının standart sapması	**Avg Fwd Segment Size**	İleri yönde gözlenen ortalama boyut
Flow Bytes/s	Saniyede akan byte sayısı	**Avg Bwd Segment Size**	Geri yönde gözlenen ortalama boyut
Flow Packets/s	Saniyede akan paket sayısı	**Fwd Header Length_1**	İleri yönde başlıklar için kullanılan toplam byte
Flow IAT Mean	Paketlerin ortalama varış zamanı	**Fwd Avg Bytes/Bulk**	İleri yönde byte/kütle oranının ortalama sayısı
Flow IAT Std	Paketlerin varış zamanlarının standart sapması	**Fwd Avg Packets/Bulk**	İleri yönde paket/kütle oranının ortalama sayısı
Flow IAT Max	Paketlerin maximum varış zamanı	**Fwd Avg Bulk Rate**	İleri yönde kütle oranının ortalama sayısı
Flow IAT Min	Paketlerin Minimum varış zamanı	**Bwd Avg Bytes/Bulk**	Geri yönde byte/kütle oranının ortalama sayısı
Fwd IAT Total	İleri yönde gönderilen iki paket arasındaki toplam zaman	**Bwd Avg Packets/Bulk**	Geri yönde paket/kütle oranının ortalama sayısı
Fwd IAT Mean	İleri yönde gönderilen iki paket arasındaki ortalama zaman	**Bwd Avg Bulk Rate**	Geri yönde kütle oranının ortalama sayısı
Fwd IAT Std	İleri yönde gönderilen iki paket arasındaki zamanın standart sapması	**Subflow Fwd Packets**	İleri yönde bir alt akıştaki ortalama paket sayısı
Fwd IAT Max	İleri yönde gönderilen iki paket arasındaki maksimum zaman	**Subflow Fwd Bytes**	İleri yönde bir alt akıştaki ortalama byte sayısı
Fwd IAT Min	İleri yönde gönderilen iki paket arasındaki minimum zaman	**Subflow Bwd Packets**	Geri yönde bir alt akıştaki ortalama paket sayısı
Bwd IAT Total	Geri yönde gönderilen iki paket arasındaki toplam zaman	**Subflow Bwd Bytes**	Geri yönde bir alt akıştaki ortalama byte sayısı
Bwd IAT Mean	Geri yönde gönderilen iki paket arasındaki ortalama zaman	**Init_Win_bytes_forward**	İleri yönde ilk pencere içinde gönderilen bayt sayısı
Bwd IAT Std	Geri yönde gönderilen iki paket arasındaki zamanın standart sapması	**Init_Win_bytes_backward**	Geri yönde ilk pencere içinde gönderilen bayt sayısı
Bwd IAT Max	Geri yönde gönderilen iki paket arasındaki maksimum zaman	**act_data_pkt_fwd**	İleri yönde en az 1 bayt TCP veri yüküne sahip paket sayısı
Bwd IAT Min	Geri yönde gönderilen iki paket arasındaki minimum zaman	**min_seg_size_forward**	İleri yönde gözlenen minimum segment boyutu
Fwd PSH Flags	İleri yönde hareket eden paketlerde PSH bayrağının aktif olma sayısı (UDP için 0)	**Active Mean**	Boşta kalmadan önce bir akışın aktif olduğu ortalama zaman
Bwd PSH Flags	Geri yönde hareket eden paketlerde PSH bayrağının aktif olma sayısı (UDP için 0)	**Active Std**	Boşta kalmadan önce bir akışın aktif olduğu zamanın standart sapması
Fwd URG Flags	İleri yönde hareket eden paketlerde URG bayrağının aktif olma sayısı (UDP için 0)	**Active Max**	Boşta kalmadan önce bir akışın aktif olduğu maksimum zaman
Bwd URG Flags	Geri yönde hareket eden paketlerde URG bayrağının aktif olma sayısı (UDP için 0)	**Active Min**	Boşta kalmadan önce bir akışın aktif olduğu minimum zaman
Fwd Header Length	İleri yöndeki başlıklar için kullanılan toplam byte	**Idle Mean**	Aktif hale gelmeden önce bir akışın boşta olduğu ortalama zaman

Tablo 1. (devam) *Sınıflandırmada kullanılan trafik özellikleri ve açıklamaları*

Bwd Header Length	Geri yöndeki başlıklar için kullanılan toplam byte	**Idle Std**	Aktif hale gelmeden önce bir akışın boşta olduğu zamanın standart sapması
Fwd Packets/s	Saniyedeki ileri yön paket sayısı	**Idle Max**	Aktif hale gelmeden önce bir akışın boşta olduğu maksimum zaman
Bwd Packets/s	Saniyedeki geri yön paket sayısı	**Idle Min**	Aktif hale gelmeden önce bir akışın boşta olduğu minimum zaman
Min Packet Length	Bir akışın minimum uzunluğu	**Label**	saldırı etiketi
Max Packet Length	Bir akışın maksimum uzunluğu		

B. KARAR AĞACI SINIFLANDIRMA YÖNTEMİ

Karar ağacı, isminden de anlaşılacağı gibi, bir ağaç biçimindedir ve bir tahmin tekniğidir [7, 8]. Karar ağacı karar düğümlerden, dallardan ve yapraklardan oluşur. Karar düğümü gerçekleştirilecek testi belirler. Bu testin sonucu, veri kaybı olmadan dallanmaya neden olur. Her düğümde, test ve dallanma ardışık olarak gerçekleştirilir ve bu dallanma üst seviyeye bağlıdır. Ağacın her dalı, sınıflandırmayı tamamlamak için adaydır. Sınıflandırma gerçekleştirilemezse, bir karar düğümü oluşturulur. Ancak, dalın sonunda belirli bir sınıf oluşmuşsa, bir yaprak vardır [9]. Bu yaprak, verilerden belirlenecek sınıflardan biridir. Karar ağacının işleyişi kök düğümlerini oluşturmaya başlar ve ardışık düğümleri yukarıdan aşağıya doğru yaprağa ulaşana kadar takip eder.

Karar ağacı sınıflandırması, veri setinden seçilen eğitim setini kullanarak bir karar ağacının oluşturulmasına dayanır. Ayrıca karar ağacının kalitesi ağacın büyüklüğüne ve sınıflandırma doğruluğuna bağlıdır [10]. Bu aşamada karar ağacındaki düğümlerin belirlenmesi çok önemlidir. Ağacı inşa etmek için hangi veri kümesinin hangi alanların kullanılacağı belirlenmelidir [11].

Bir ağacın nasıl dallanacağını belirlemek için iki ölçüm kullanılabilir. Gini ölçümü, bir daldaki dağılıma göre rastgele bir etiket seçildiğinde, rastgele örneğin hatalı bir şekilde sınıflandırılmış olma olasılığıdır. Entropi, bir bilgi ölçüsüdür. Her dallanmada bilgi kazancı ve belirsizliğin ne kadar azaltıldığı ölçülür.

Bu çalışmada, karar ağacı sınıflandırıcısı Python dili ve scikit-learn kütüphanesi kullanılarak kodlanmıştır [12]. Kodlama yapılırken dallanma kriteri olarak “criterion :gini” ve “splitter:best” kullanılmıştır. Ağacın maksimum derinliği için “max_depth : None” seçilerek tüm yapraklar saf olana veya tüm yapraklar “min_samples_split=2” örnekten daha az olana kadar düğümler genişletilmiştir. Bir yaprak düğümde olması gereken minimum örnek sayısı “min_samples_leaf: 1” ve en iyi dallanmayı belirlerken tüm girdilerin dikkate alınması için “max_features:None” olarak seçilmiştir. Ağacın en iyi şekilde büyümesi için “max_leaf_nodes:None” seçilerek herhangi bir sınırlama getirilmemiştir.

C. RASTGELE ORMAN SINIFLANDIRMA YÖNTEMİ

Rastgele orman sınıflandırıcısı, rastgele seçilen eğitim örnekleri ve değişkenleri alt kümesini kullanarak çoklu karar ağaçları üreten bir topluluk sınıflandırıcısıdır [13]. Bu sınıflandırıcı, sınıflandırmalarının doğruluğu nedeniyle veri bilimi topluluğunda popüler hale gelmiştir. Adından anlaşılacağı üzere, rastgele bir orman oluşturur ve kurduğu orman, çoğu zaman "bagging" yöntemiyle eğitilen Karar Ağaçları topluluğudur [14, 15]. "Bagging" yönteminin genel fikri, öğrenme modellerinin bir kombinasyonunun genel sonucu arttırmasıdır. Rastgele orman çok sayıda karar ağacı kurarak, daha doğru ve kararlı bir tahmin elde etmek için bu ağaçları birleştirir.

Rastgele Orman, ağaçları büyütürken, modele ek rastgelelik katar. Bir düğümü dallandırırken en önemli özelliği aramak yerine, rastgele bir özellik alt kümesi arasında en iyi özelliği arar. Bu durum, genellikle daha iyi bir modelle sonuçlanan geniş bir çeşitliliğe sebep olur.

Rastgele Orman algoritmasında iki aşama bulunur. Öncelikle rasgele orman oluşturulur, ardından ilk aşamada oluşturulan rasgele orman sınıflandırıcısından tahmin yapılır. Ormanın oluşturulduğu ilk süreç kabaca şu şekilde özetlenebilir [16]:

1. $k << m$ olmak üzere toplam "m" özellikten "k" özellik rastgele seçilir
2. "k" özellik arasında en iyi ayırma noktasını kullanarak "d" düğüm hesaplanır
3. En iyi bölünmeyi kullanarak düğümü alt düğümlere ayrılır
4. "l" düğüm sayısına ulaşılana kadar 1-3 adımları tekrarlanır
5. "n" adet ağaç oluşturmak için "n" defa 1-4 adımları yinelenerek orman oluşturulur

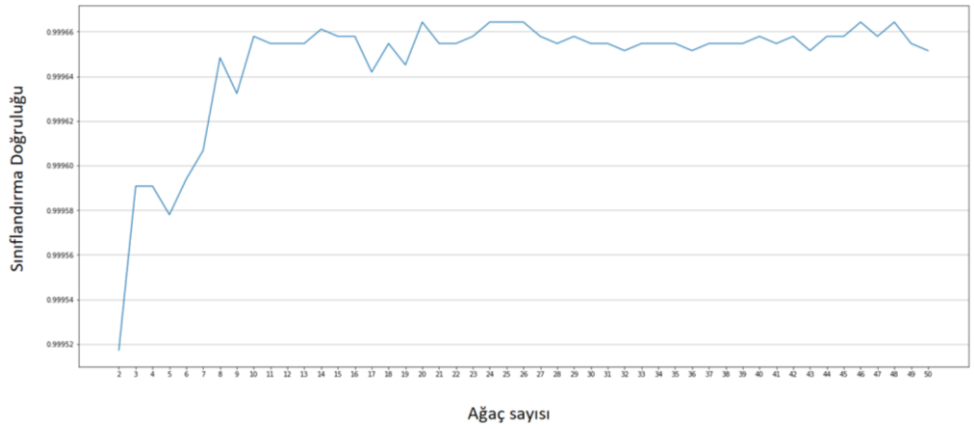
Bir sonraki aşamada, rastgele orman sınıflandırıcısı kullanılarak tahmin yapılması kabaca şu şekilde özetlenebilir [16]:

1. Özellikler alınır ve rastgele oluşturulmuş her karar ağacının kuralları kullanılarak sonuçlar tahmin edilir
2. Her tahmin edilen sonuç için oylar hesaplanır
3. Rastgele orman algoritmasından nihai tahmin olarak en yüksek oy alan tahmin kullanılır

Bu çalışmada, rastgele orman sınıflandırıcısı Python dili ve scikit-learn kütüphanesi kullanılarak kodlanmıştır [12]. Kodlama yapılırken dallanma kriteri olarak "criterion :gini" kullanılmıştır. Ağacın maksimum derinliği için "max_depth : None" seçilerek tüm yapraklar saf olana veya tüm yapraklar "min_samples_split=2" örnekten daha az olana kadar düğümler genişletilmiştir. Bir yaprak düğümde olması gereken minimum örnek sayısı "min_samples_leaf: 1" ve en iyi dallanmayı belirlerken tüm girdilerin dikkate alınması için "max_features:None" olarak seçilmiştir. Ağacın en iyi şekilde büyümesi için "max_leaf_nodes:None" seçilerek herhangi bir sınırlama getirilmemiştir. N_jobs:1 ve verbose:0 seçilerek uydurma ve tahmin etme aşamalarında işlem sayısı ve ayrıntı seviyesi belirlenmiştir. warm_start:False seçilerek önceki aramanın çözümünü tekrar kullanmayıp tamamen yeni bir orman oluşturulması amaçlanmıştır.

III. BULGULAR VE TARTIŞMA

Bu çalışmada karar ağacı ve rastgele orman yöntemleri kullanılarak normal ve anormal ağ trafiği sınıflandırılmıştır. Veri kümesinde 629074 adet normal ve 413483 adet normal olmayan olmak üzere toplam 1042557 adet kayıt bulunmaktadır. Bu verinin %70'i eğitim verisi, %30'u da test verisi olarak rastgele seçilmiş ve kullanılmıştır. Rastgele orman yönteminde, ormanı oluşturacak olan en uygun ağaç sayısının belirlenmesi için farklı büyüklüklerde ormanlar oluşturulmuş ve sınıflandırma doğrulukları incelenmiştir. Şekil 1'de görüldüğü üzere en yüksek sınıflandırma doğruluğunu verecek en küçük orman 20 ağaçla oluşturulabildiğinden, n_estimators parametresi 20 olarak belirlenmiştir.



Şekil 1. *Ormandaki ağaç sayısının belirlenmesi amacıyla incelenen ağaç sayısı-sınıflandırma doğruluğu grafiği*

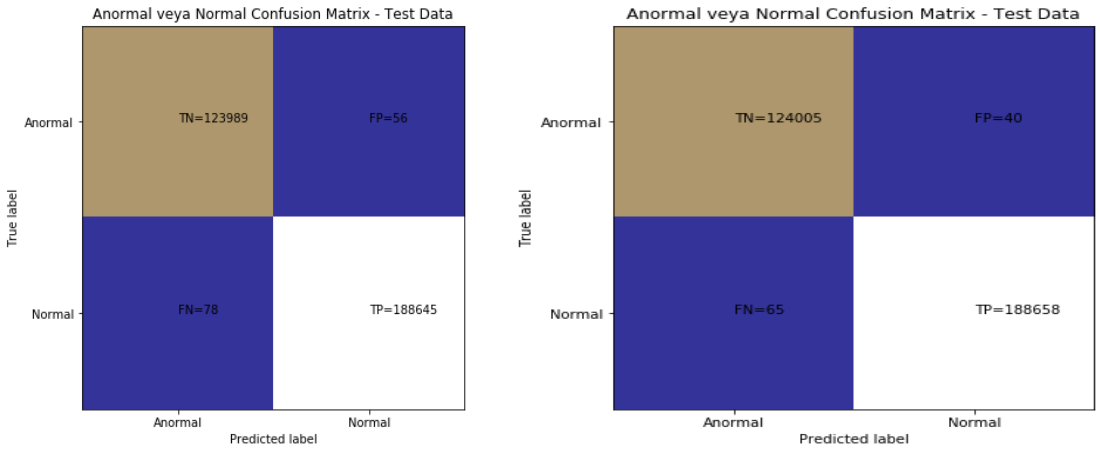
Eğitim sonrası sınıflandırma yöntemlerinin performansları, test verisi üzerinde doğruluk (accuracy), hassasiyet (precision), geri çekilme (recall) ve F1 değeri gibi performans ölçümleri kullanılarak değerlendirilmiştir. Kullanılan bu ölçümler bilgi çıkarımı alanında en çok kullanılan yöntemlerdir [17]. Performans değerlendirme sonuçları tablo 2'de görülmektedir.

Tablo 2. *Değerlendirme sonuçları*

Yöntem		**Hassasiyet**	**Geri Çekilme**	**F1**	**Doğruluk**
Karar Ağacı	Normal	0.99970	0.99959	0.99964	
	Anormal	0.99937	0.99955	0.99946	0.99957
Rastgele Orman	Normal	0.99979	0.99966	0.99972	
	Anormal	0.99948	0.99968	0.99976	0.99966

Tablo 2 incelendiğinde test verisi üzerinde doğru tahmin edilen gözlem sayısının toplam gözlem sayısına oranı olan doğruluk değerlerinin karar ağacı ve rastgele orman yöntemleri için sırasıyla 0.99957 ve 0.99966 olduğu görülmektedir. Test verisinden elde edilen bu değerlerin detayları şekil

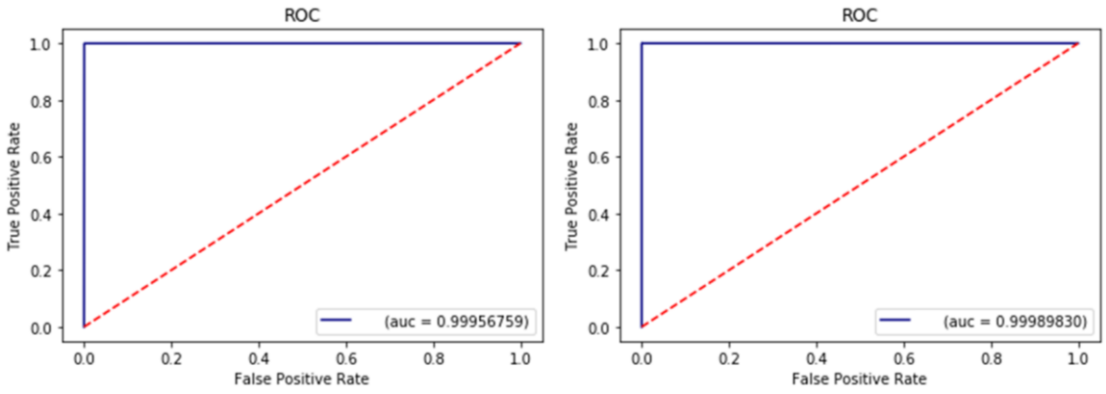
2’de sunulmuştur. Her iki yöntem ve her iki sınıf için de hassasiyet oranlarının 0.999’ın üzerinde olması, doğru tahmin edilen pozitif gözlem sayısının toplam tahmin edilen pozitif gözlem sayısına oranının 1’e çok yakın olduğunu göstermektedir. Diğer adı duyarlılık (sensitivity) olan ve doğru tahmin edilen pozitif gözlem sayısının gerçek sınıftaki tüm gözlem sayısına oranını veren geri çekilme değerlerinin de her iki sınıflandırıcı için 1’e çok yakın olduğu tespit edilmiştir. Hem yanlış pozitif hem de yanlış negatifleri dikkate alan ve hassasiyet ile geri çekilme değerlerinin ağırlıklı ortalaması alınarak hesaplanan F1 değeri de yine 1’e oldukça yakındır.



(a) *(b)*

***Şekil 2. (a)** Karar ağacı ve **(b)** Rastgele orman sınıflandırma yöntemlerinin tahmin sonuçları*

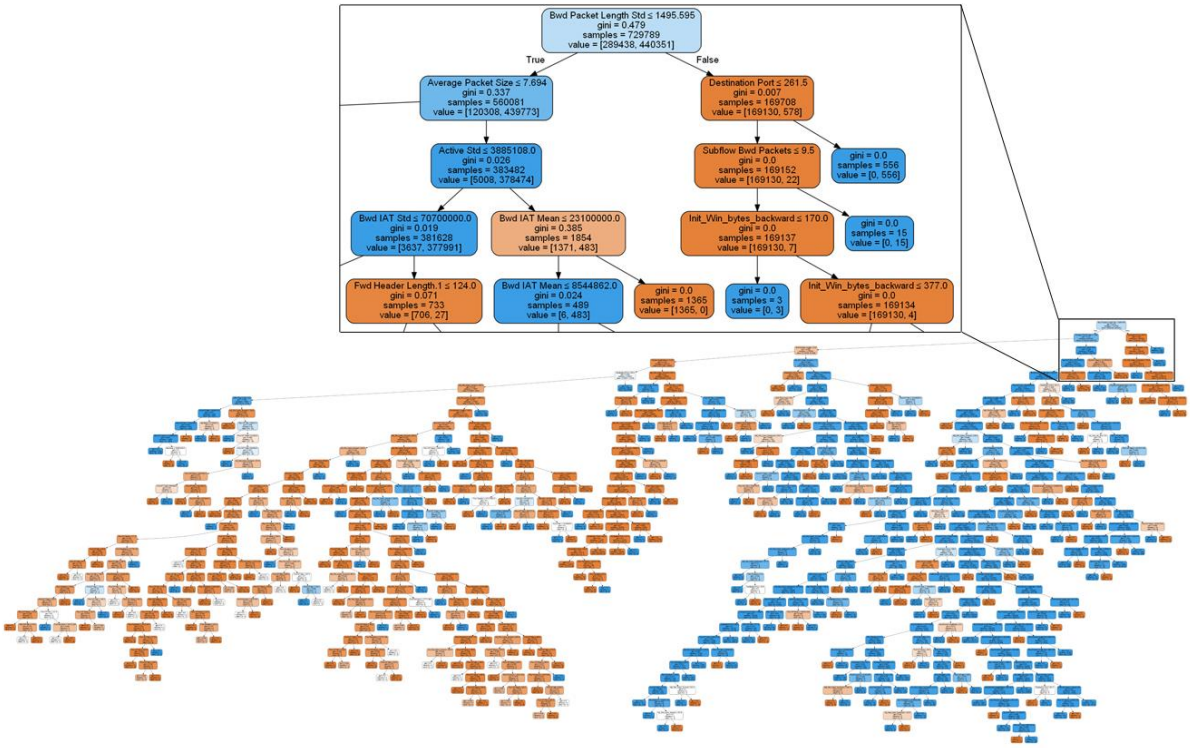
Karar ağacı ve rastgele orman sınıflandırma yöntemlerinin performansları, bir diğer sınıflandırma performans ölçümü olan ROC eğrisi ve eğri altında kalan alan değerleri ile de değerlendirilmiştir. Şekil 3’de verilen grafikler incelendiğinde, her iki sınıflandırıcının da ROC eğrilerinin ideale yakın olduğu ve eğri altında kalan alan değerlerinin de 1’e çok yakın olduğu görülmektedir.



(a) ***(b)***

***Şekil 3. (a)** Karar ağacı ve **(b)** Rastgele orman sınıflandırma yöntemleri için ROC eğrileri ve eğri altında kalan alan değerleri*

Normal ve anormal ağ trafiğini sınıflandırmak amacıyla kurulan karar ağacı şekil 4'de görülmektedir. Ağaç incelendiğinde ağacın üst katmanlarında geri yönde paket uzunluklarının standart sapması, ortalama paket boyutu, hedef port, geri yöndeki başlıklar için kullanılan toplam byte, boşta kalmadan önce bir akışın aktif olduğu zamanın standart sapması, geri yönde bir alt akıştaki ortalama paket sayısı ve bir akışın ortalama uzunluğu gibi değişkenler olduğu tespit edilmiştir. Karar ağacı yapısına göre, belirtilen değişkenlerin normal ağ trafiğini anormal trafikten ayırt etmede kullanılabilecek en önemli ağ trafiği özellikleri olduğu söylenebilir.



Şekil 4. *Normal ve anormal ağ trafiğini sınıflandırmak amacıyla kurulan karar ağacı*

IV. SONUÇ

Saldırı tespit sistemleri, güvenlik yöneticilerinin sızma, saldırı ve kötü amaçlı yazılım gibi kötü niyetli davranışları önceden haber vermelerini amaçlayarak ağ savunma sürecinde hayati bir rol oynar. STS'ye sahip olmak, kritik ağları sürekli olarak artan müdahaleci faaliyetlere karşı korumak için zorunlu bir savunma hattıdır. Bu nedenle, STS alanıyla ilgili araştırma yıllar boyunca, daha iyi STS sistemleri önermek için geliştirilmiştir.

Bu çalışma kapsamında, günümüzde en çok karşılaşılan siber saldırı yöntemlerinden biri olan zararlı ağ trafiğinin tespit edilmesi için makine öğrenmesi algoritmalarından karar ağacı ve rastgele orman yöntemleri kullanılmıştır. Veri kümesi olarak da bu alandaki en güncel ve en kapsamlı veri kümelerinden biri olan CICIDS2017 verisi kullanılmıştır. Bir milyonun üzerindeki kayıt kullanılarak eğitilen ve test edilen sınıflandırıcıların performansları doğruluk, hassasiyet, geri çekilme, F1 değeri, ROC eğrileri ve eğri altında kalan alan gibi çeşitli ölçümlerle incelenmiştir. Bu ölçümler neticesinde çok az bir farkla da olsa rastgele orman yönteminin sınıflandırma performansının, karar ağacı yöntemine oranla daha iyi olduğu söylenebilir.

Önerilen bu çalışmayla, ağ saldırılarının tespit edilmesi ve gerekli tedbirlerin alınması mümkündür. Önemli olan bir diğer konu da saldırı çeşitlerinin belirlenmesidir. Farklı saldırı çeşitleri arasında DoS, PortScan, DDoS, FTP Patator, SSH Patator, Bot, Web Attack, SQL injection, Web Attack XSS, Web Attack BruteForce, Heartbleed ve Infiltration saldırı yöntemleri bulunmaktadır. Sunulan çalışmanın ilerleyen aşamalarında, zararlı ağ trafiğinin tiplerine göre sınıflandırılması amaçlanmaktadır.

V. KAYNAKLAR

[1] Ç. Kaya ve O. Yildiz, "Makine Öğrenmesi Teknikleriyle Saldırı Tespiti: Karşılaştırmalı Analiz", *Marmara University Journal of Science*, c. 26, s.3, ss. 89-104, 2014.

[2] M.N. Chowdhury ve K. Ferens, "Network Intrusion Detection Using Machine Learning", Int'l Conf. Security and Management, Las Vegas, ABD, 2016, pp. 30-35.

[3] M.E. KarsligEl, A.G. Yavuz, M.A. Güvensan, K. Hanifi ve H. Bank, "Network intrusion detection using machine learning anomaly detection algorithms", *25th Signal Processing and Communications Applications Conference*, Antalya, Türkiye, 2017, ss. 1-4.

[4] N. Shone, N. Tran Nguyen, P. Vu Dinh ve Q. Shi, A" Deep Learning Approach to Network Intrusion Detection", *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, 2018.

[5] A. Javaid, Q. Niyaz, W. Sun ve M. Alam, "A deep learning approach for network intrusion detection system", *9th EAI International Conference on Bio-inspired Information and Communications Technologies*, Brüksel, Belçika, 2016, pp. 21–26.

[6] I. Sharafaldin, A.H. Lashkari ve A.A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", *4th International Conference on Information Systems Security and Privacy*, Portekiz, 2018, pp. 108-116.

[7] T. Tuncer ve Y. Tatar, "Karar Ağacı Kullanarak Saldırı Tespit Sistemlerinin Performans Değerlendirmesi", *4. İletişim Teknolojileri Ulusal Sempozyumu*, Adana, Türkiye, 2009, ss. 41-48.

[8] S. Chaudhuri, "Data Mining and Database Systems : Where is the Intersection?", *IEEE Bulletin of the Technical Committee on Data Engineering*, vol. 21, no. 1, pp. 4-8, 1998.

[9] A. Berson, S. Smith ve K. Thearling, *Building Data Mining Applications for CRM*, McGraw-Hill Professional Publishing, 2000, pp. 15-20.

[10] J. Han ve M. Kamber, Data Mining Concepts and Techniques, 2nd Edition, *The Morgan Kaufmann Series in Data Management Systems*, 2006, pp. 1-97.

[11] R. Agrawal, T. Imielinski ve A. Swami "Database Mining:A Performance Perspective", *IEEE Transactions on Knowledge and Data Engineering* vol. 5, no. 6, pp. 914-925, 1993.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot ve É. Duchesnay, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, no. 2, pp. 2825-2830, 2011.

[13] M Belgiu, L Draguţ, “Random forest in remote sensing: A review of applications and future directions”, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, no. 2, pp. 24-31, 2016.

[14] A Demirhan, “Kolektif Öğrenmeye Dayalı Çok Değişkenli Desen Analizinin Klinik Karar Destek Sistemlerinde Uygulanması”, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, c. 6, s. 4, ss. 953-961, 2018.

[15] KJ Archer, RV Kimes, “Empirical characterization of random forest variable importance measures”, *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 2249-2260, 2008.

[16] Anonim, (15 Aralık 2018). [Online]. Erişim: https://syncedreview.com/2017/10/24/how-random-forest-algorithm-works-in-machine-learning.

[17] J Makhoul, F Kubala, R Schwartz, R Weischedel, Performance Measures For Information Extraction, Proceedings of the DARPA Broadcast News Workshop, Washington, ABD, 1999.