Uluslararası Sürdürülebilir Mühendislik ve Teknoloji Dergisi
International Journal of Sustainable Engineering and Technology
Sayı: 1, Cilt: 7, (2023), Sayfa: 31-40

AĞ ALTYAPILARINDA YAPAY ZEKA TABANLI AĞ TRAFİK YÖNETİM MEKANİZMALARININ İNCELENMESİ

Ayşe Nur TEMURÇİN[*1], Mevlüt ERSOY[1]

[1]Süleyman Demirel Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Isparta

Makale Bilgisi

Geliş tarihi: 22.05.2023
Kabul Tarihi: 24.06.2023
Yayın tarihi: 29.06.2023

ÖZET

Bu çalışmanın amacı yazılım tanımlı ağlar üzerinde meydana gelen güvenlik tehditlerinin makine öğrenme algoritmaları ile tespit edilmesi ve kullanılan algoritmalar ile en az hatayla saldırı sınıflandırması yapılmasıdır. Bu çalışmada mininet üzerinde kurulan bir yazılım tanımlı ağ üzerinde kurulan veri seti ve internet ortamından alınan veri seti kullanılmıştır. Hping3 programı kullanılarak DDoS saldırısı gerçekleştirilmiştir. Ağ trafiğinde saldırı paketlerini sınıflandırmak için tehdit var (0) ve tehdit yok (1) şeklinde ikili bir sınıflandırma yapılmıştır. Bu sınıflandırma yapılırken yapay sinir ağı, rastgele orman algoritması ve yükseltme algoritmalarından faydalanılmıştır. Kullanılan algoritmalar ile en az hata ile saldırı sınıflandırması yapılması sağlanmıştır. Elde edilen sonuçlara gerekli istatistikler incelenerek sonuç alınmıştır. Elde edilen sonuçlara göre rastgele orman algoritmasının performans değerleri diğer algoritmalar ile kıyaslandığında daha iyi sonuçlar verdiği görülmüştür. Bu netice ile rastgele orman algoritması ile saldırı sınıflandırması için daha verimli sonuçlar alınmıştır.

Anahtar Kelimeler; Yazılım tanımlı ağlar, mininet, makine öğrenme algoritması, hping3

INVESTIGATION OF AI-BASED NETWORK TRAFFIC MANAGEMENT MECHANISMS IN NETWORK INFRASTRUCTURE

Article Info

Received: 22.05.2023
Accepted: 24.06.2023
Published: 29.06.2023

ABSTRACT

The aim of this study is to detect security threats on software-defined networks with machine learning algorithms and to classify attacks with the least error with the algorithms used. In this study, the data set established on a software defined network established on mininet and the data set obtained from the internet environment were used. A DDoS attack was carried out using the hping3 program. In order to classify attack packets in network traffic, a binary classification of threat (0) and no threat (1) was made. While making this classification, artificial neural network, random forest algorithm and amplification algorithms were used. With the algorithms used, attack classification was achieved with the least error. The results were obtained by examining the necessary statistics. According to the results obtained, the accuracy score of the random forest algorithm gives higher results when compared to other algorithms. With this result, more efficient results were obtained for attack classification with the random forest algorithm.

Keywords; Software defined networks, mininet, machine learning algorithm, hping3

1. Giriş

Yazılım tanımlı ağlar üzerinde gerçekleştirilen veri aktarımının oluşturduğu ağ trafiğinin güvenli bir şekilde yönetilmesi büyük önem taşımaktadır. Gelişmiş ağ uygulamalarının artmasının ve aynı alanda bile çeşitli türde cihazların varlığının yanı sıra, binlerce uç nokta cihazı farklı ağ trafiği modellerini paylaşabilir ve değiş tokuş edebilir(Latah ve Toker,2016). Bu, hali hazırda var

olan ağ altyapısının tüm bu ihtiyaçları karşılayamayacağını ve yeni bir ağ altyapısı yaklaşımının(Latah ve Toker, 2016) sunulması gerektiğini göstermektedir. Bu yeni yaklaşıma Yazılım Tanımlı Ağlar (SDN) adı verilmektedir. Yazılım tanımlı ağlar, ağ iletişimi için devam eden yenilikçi ağ altyapısı sağlama ve ağı daha büyük bulut altyapısının programlanabilir ve takılabilir bir bileşeni olarak etkinleştirme potansiyeline sahip olmuştur(Kobayashi vd., 2013). Ağ trafiğinde yoğun verinin aktarılması ağ trafik yönetimini giderek zorlaştırmaktadır. Bu nedenle ağ yöneticilerinin trafiği devamlı olarak yönetmeleri ve değişiklik yapmalarına neden olmaktadır. Ağ trafiğinin anlık olarak kontrol edilmesi için merkezi kontrol sistemleri ve yazılım tanımlı ağlarla birçok kurum ve kuruluş cihaz altyapılarında güncelleme yapmaya yönelmiştir. Bununla birlikte ağ trafiğinin yönetimini kolay hale getirmek amaçlanmıştır. Ağ trafiğinin yönetilmesi için yapay zeka tabanlı araçlar geliştirilmiştir. Artan teknolojik gelişmeler yapay zeka tabanlı derin makine öğrenmesi yöntemlerini kullanmaya teşvik etmiştir.

Son zamanlarda yapılan çalışmalar trafik mühendisliği, hizmet zincirleme, ağ işlevlerinin sanallaştırılması ve bulut boşaltma gibi bir dizi yönetim görevi için ağ yapılandırmasını kullanmak üzere SDN tabanlı mekanizmaları(Heorhiadi ve Reiter, 2016) kullanmıştır. Trafik mühendisliği, iletilen verilerin davranışını dinamik olarak analiz ederek ve düzenleyerek bir veri ağının performansını optimize etmek için önemli bir mekanizmadır(Akyildiz vd., 2014). Aynı zamanda bulut bilişimde gerçekleşen hızlı büyüme ve büyük ölçekli veri merkezlerinin talebi doğrultusunda uygun bir ağ yönetimi, daha iyi bir sistem performansı için kaynak kullanımını iyileştirebilen yeni ağ oluşturma mimarileri ve daha verimli trafik mühendisliği uygulamaları ortaya çıkmaktadır. Bu mimarilerden bir tanesi PANE(Participatory Networking, Katılımcı Ağ) denetleyicisidir. PANE, kurumsal WAN'lar, veri merkezleri, kampüs veya kurumsal ağlar ve ev ağları dahil olmak üzere tek bir yönetim alanındaki ağlar için tasarlanmıştır(Ferguson vd., 2013). PANE'nin tasarımı, belirli bir sürücü veya hiper yöneticinin kullanımı gibi uç ana bilgisayarların ağ yığınlarında yapılan değişikliklere dayanmaz, bu da onu kullanıcıya ait veya yönetilen cihazlara sahip ağlar için uygun hale getirir(Ferguson vd., 2013).

Ferguson ve diğerleri (2013) tarafından yapılan çalışmada PANE denetleyicisi ile etkileşime girebilen ağ trafiğini yöneten bir uygulama yapmışlardır. PANE prototipini, SDN' leri taklit etmek için mininet platformunda gerçek ağlarla değerlendirmişlerdir ve çalışma sonucunda mevcut OpenFlow özellikli ağlarda PANE API'sini uygulamanın son derece pratik olduğunu göstermişlerdir.

Akbaş ve diğerleri (2016) yaptıkları incelemelerde ağ trafik yönetimi, ağ güvenliği ve ağ trafiğini daha iyi duruma getirecek çözümler üretmek için SDN'deki karmaşık verinin yapay zeka uygulamaları kullanılarak işlenmesinin bilişsel ağların geliştirilebilmesine olanak sağlayacağını söylemişlerdir. Farklı ve karmaşık yapıdaki ağların daha aktif bir şekilde yönetilmesinde yapay zeka tekniklerinin kullanımı artmıştır. Yazılım tanımlı ağlar ile ağın veri trafiği yönetiminin güvenli ve verimli bir şekilde yapılması sağlanmaktadır. Makine öğrenme algoritmaları ile ağ güvenliğini tehdit eden zararlı yazılımlar ve uygulamalar engellenerek ağ trafiğinin güvenli bir şekilde ilerlemesi sağlanmaktadır.

Mihai-Gabriel ve Victor-Valerin (2014) yaptıkları çalışmada SDN'lerde DDoS saldırılarını azaltmak için bir metot önermişlerdir. Önerilen metotta DDoS saldırısı riski her ana bilgisayarda hesaplanmakta ve ağ trafiğinden sorumlu sanal bilgisayarlara iletilmektedir. Gözlemlenen trafiğin riski önceden tanımlanmış bir değeri geçtiğinde, bu akışların kontrolöre iletilmesinin sebep olduğu yükü azaltmak için SDN'in riski önceden görüp harekete geçmesine izin veren kontrolleri yapmak üzere kontrolörün uyarıldığını görmüşlerdir.

Yazılım tanımlı ağlar, ağ trafik kontrolünü esas olarak yapay zeka teknolojileriyle uygulamaktadır. SDN'de konuşlandırılan denetleyici ağ trafiğini gözlemlemekte, ağ trafiğindeki değişikliği tahmin etmekte ve veri akışı temelinde yönlendirme kararları verebilmektedir (Guo vd., 2018). Böyle bir sistem ancak makine öğrenme algoritmaları ile yapılabilmektedir. Akıllı ağ trafiği yönetimi ve optimizasyonu için çeşitli yollar geliştirilmektedir.

Guo ve diğerleri (2018) yaptıkları çalışmada ağ yolu optimizasyonu için gelişmiş öğrenmenin genel algoritmalarını kullanmışlardır. Tüm bu yapay zeka algoritmalarının uygulanmasında Spark-MLlib gibi iyi kurulmuş bir makine öğrenme platformu kullanmışlardır.

Hadi Hadi (2022), yaptığı çalışmada güvenilir sistemler elde etmek için derin öğrenme algoritmalarıyla eğitilen saldırıları belirlemek için akıllı ağ saldırı tespit sistemi (NIDS) kullanılmasını önermiştir. NIDS adı verilen bu sistem, ağı çeşitli saldırılardan korumak için tasarlanmıştır. Elinde bulunan veri setindeki 12 öznitelik üzerinde eğitilen NIDS yaklaşımında DNN(Deep Neural Network, Derin Sinir Ağı), CNN(Convolutional Neural Network, Evrişimli Sinir Ağı), RNN(Recurrent Neural Network, Tekrarlayan Sinir Ağı), GRU(Gated Recurrent Neural Network, Geçitli Tekrarlayan Sinir Ağı) ve LSTM(Long-Short Term Memory, Uzun-Kısa Süreli Bellek) algoritmalarını kullanmıştır. Elde edilen sonuçlarda CNN algoritmasının en yüksek doğruluğa ulaştığı görülmüştür. NIDS yaklaşımının başarısı neticesinde gelecekte derin öğrenmenin SDN güvenliği için etkin bir şekilde kullanılabileceğini göstermiştir.

Özalp (2023), çalışmasında saldırı tespitine yönelik birden fazla yol izlemiştir. İlk aşamasında öznitelik seçiminin önemi ve saldırıların tespitine etkisini araştırmıştır. Kullandığı veri setinden elde edilen öznitelikleri Random Forest, Navie Bayes, J48 ve MLP(Çok Katmanlı Algılayıcılar) algoritmalarıyla sınıflandırmıştır. Bu aşama sonucunda öznitelik seçim yöntemi ve seçilen öznitelik sayısının saldırı tespit sistemleri için önemli bir parametre olduğunu görmüştür. İkinci aşamada ise saldırı tespiti için hibrit bir model üzerine çalışma yapmıştır. Son aşamasında ise derin öğrenme ve makine öğrenmesi algoritmaları kullanarak tasarlanan hibrit bir saldırı tespit sistemi önermiştir. Önerilen hibrit modelde özellikle SQL Injection, Brute Force ve DDoS saldırılarında doğruluk noktasında yüksek performans göstermiştir.

Niyaz, Sun, Javaid (2016), yaptıkları çalışmada bir SDN ortamında saldırı vektörlerinin tespiti için derin öğrenme tabanlı bir DDoS tespit sistemi kurmuşlardır. Kurdukları sistemde, trafik sınıflandırıcı modülü için Yığılmış Otomatik Kodlayıcı(SAE) sınıflandırma modelini kullanmışlardır. Sinir ağı ile saldırı algılama modelleri geliştirmişler ve doğruluk açısından daha iyi performans elde ettiğini görmüşlerdir.

Bu çalışmada SDN ağ yönetim fonksiyonları kullanılarak ağın daha güvenli bir şekilde veri aktarımını sağlayacak bir altyapı oluşturulmuştur. Bu kapsamda veri trafiğinin yönetimi için ağ güvenliğini tehdit eden ağ trafik akışlarının makine öğrenme algoritmaları ile sınıflandırılması sağlanmıştır. Bu sınıflandırmalar sonucunda SDN yönetim mekanizmalarının tetiklenmesini sağlayan bir çalışma gerçekleştirilmiştir. Bu çalışma sayesinde yapay zeka tabanlı ağ trafik yönetimi altyapılarının takibi, denetimi ve yönetimi açısından yapay zekayla kontrol edilmesinin avantaj ve dezavantajları ortaya çıkarılmıştır. Özellikle makine öğrenmesi algoritmaları incelenerek ağ trafik yönetiminin anlık karar verme mekanizmalarının çalışma prensipleri ortaya çıkarılmıştır.

2. SDN Ortamında Makine Öğrenme Algoritmaları ile Güvenlik Yönetimi

Bu çalışmada, ağ simülasyonu amacıyla mininet ortamında yazılım tanımlı bir ağ oluşturulmuştur. Mininet sanal bir bilgisayar üzerinde sanal bir ağ oluşturmamızı sağlamıştır. Mininet üzerinde sanal anahtar ve ağ bilgisayarı koşturulmasına ve Linux ağ isim uzayında işlemleri sanal bir şekilde gerçekleştirmemizi sağlamıştır. Çekirdek isim uzayının kullanılması sanal ağlarda her host için bir isim uzayı oluşturmamıza izin vermektedir. Mininet üzerinde open vswitch ve kontrolcü varsayılan olarak çalıştırılmaktadır. Mininet, OpenFlow tabanlı uygulamaları geliştirmek ve test etmek için basit ve ucuz bir ağ aracıdır. Aynı zamanda fiziksel ağları yapılandırmadan test amaçlı karmaşık ve farklı ağ topolojileri oluşturulmasına izin vermektedir (Kaur vd., 2014). Mininet ile yazılım tanımlı ağlar ile çalışırken Mininet'te geliştirilecek olan kodda bir değişiklik yapılmasına gerek olmadan gerçek ağda da çalıştırılabilmekte ve hızlı bir prototip ağ oluşumuna izin vermektedir. Yazılım tanımlı ağlar için hızlı ve gerçek bir emülasyon sağlayan araçları içermektedir.

Simülasyonlar DELL 5810, 12 çekirdekli Intel Xeon E5-1620 v3, 3.50 GHz ve 16 GB RAM'e sahip bilgisayar üzerinde koşturulmuştur. Mininet üzerinde her ağ üzerinde 10 host olan bir ağaç topolojisi oluşturulmuştur. Ağaç topolojisi uygulamalarda genellikle tercih edildiği için kullanılmıştır. Bu topoloji genellikle veri merkezlerinde tercih edilen bir topolojidir. Test ortamında OpenFlow anahtarlar tercih edilmiştir.

2.1.Izgara Temelli Arama Algoritması(Grid Search Algorithm)

Izgara Arama Algoritması, bir algoritmanın en ideal parametre ayarlarını keşfetmek için, bu algoritmanın ızgara (grid) içerisindeki tüm parametre ayarlarını test eden geleneksel bir tekniktir (Camilleri vd., 2014). Izgara Arama, hiper-parametre yapılandırma

alanını keşfetmek için kullanılan yaygın yöntemlerdendir. Hiper-parametre optimizasyonu ise bir makine öğrenmesi algoritması için belirlenen başarı metriğine göre en uygun hiper-parametre kombinasyonunu bulma işlemidir. Izgara Arama yöntemi bu optimizasyon için uygun bir yöntemdir. Bunun sonucunda parametrelerin değer aralıkları küçük bir arama uzayı oluşturuyorsa başarılı sonuçlar elde etmek için ızgara arama yöntemi kullanılmaktadır. Ancak, parametre sayısının artması ve arama uzayının büyük ölçekli olması durumunda büyük bir hesaplama bütçesi oluşmasına neden olmaktadır. Büyük hesaplama maliyetleri istenmediğinden dolayı, bir algoritmanın en ideal parametre ayarlarını bulmak için genellikle meta-optimizasyon teknikleri kullanılmaktadır(Gencal ve Oral, 2021).

2.2. Trafik üretimi

Bu çalışmada trafik üretimi için hping komutu kullanılmıştır. Hping3 komutu paket analizlerinde ve paket akışı için kullanılan TCP/IP araçları arasındadır. Yoğun bir trafik oluşturma işlemlerinde birçok iş parçacığı üzerinden veri akışı sağlanabilmektedir. Bu çalışmada, mininet ortamında DDoS saldırısı oluşturabilmek için farklı parametrelere sahip hping3 programı kullanılmıştır. Bu programa verilen parametreler ile çeşitli saldırılar gerçekleştirilmiştir. Hping3 hangi cihazların DDoS saldırılarından etkilendiğini bulmak için bir araç olarak kullanılmaktadır (Ahda vd., 2023). Tipik ağ trafiğini simüle etmek için kullanılmaktadır (Ye vd., 2018).

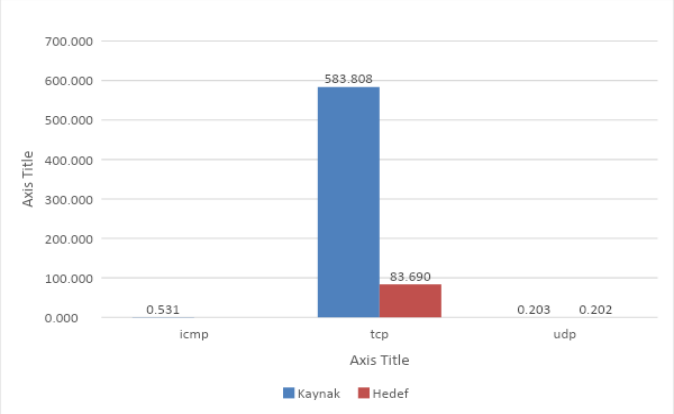
2.3. Ağın eğitimi için veri kümesi

Kurulan ağ üzerinde makine öğrenme algoritmalarının eğitimi için internet üzerinden alınan bir veri seti ve ona ek olarak mininet üzerinde elde edilen veriler eklenerek eğitim seti oluşturulmuştur. Test verisi olarak hping3 ile üretilen farklı parametrelere sahip veriler kullanılmıştır. Bu veriler ICMP, TCP ve UDP protokollerine sahip verilerden oluşmaktadır. Tablo 1'de veriler aktarılırken ortaya çıkan bayt sayıları verilmiştir.

Tablo 1. Veri aktarımında çıkan bayt sayıları

Satır Etiketleri	Toplam Kaynak Veri Boyutu(bayt)	Toplam Hedef Veri Boyutu(bayt)
Icmp	667254	0
Tcp	612166701	87755181
Udp	213231	211433

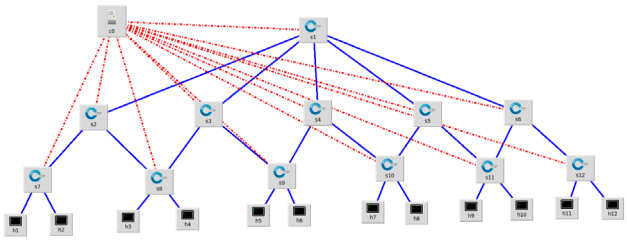
Şekil 1'de görüldüğü üzere veri aktarımı sırasında kaynaktan hedefe gönderilen paket sayıları gösterilmektedir. Bu şekle göre paketlerin genellikle TCP tabanlı olduğu görülmektedir.



Şekil 1. Veri iletiminde kaynaktan hedefe ve hedeften kaynağa gönderilen paket sayıları

3. Araştırma Bulguları

Bu araştırmada Şekil 2'de görünmekte olan ağ tasarımında güvenlik tehditlerini tespit etmek için makine öğrenme algoritmaları kullanılmıştır.



Şekil 2. Mininet ortamında oluşturulan ağ tasarımı

Makine öğrenme algoritmalarının başarısını tespit edebilmek için ağ üzerinden elde edilen veriler ve internet üzerinde yer alan veriler birleştirilerek geniş bir veri seti oluşturulmuştur. Makine öğrenme algoritmaları Python programlama dilinde geliştirilmiş kütüphaneler kullanılarak tespit edilmiştir.

Tablo 2. Kullanılan Yapay sinir ağı parametreleri ve değer aralıkları

Algoritma	Parametreler	Değer Aralıkları
Yapay Sinir Ağı	Öğrenme Oranı	[0,01, 0,1]+0,02
	Momentum Katsayısı	[0,5, 0,9]+0,1
	Döngü(epoch) Sayısı	{10, 25, 50, 100}
	Alt Örneklerin(batch) Boyutu	{10, 20, 40, 60, 80, 100}

Algoritmaların parametrelerini deneme yanılma ile belirlemek yerine, yapay sinir ağı için Tablo 2'de verilen aralıklarda tanımlar yapılmıştır. Kullanılan makine öğrenme algoritmalarında en iyi parametreleri belirlemek için Izgara Temelli Arama Algoritması kullanılmıştır. Bu arama algoritmasında en iyi parametreler belirlenirken RMSE değeri referans alınmıştır.

Tablo 3'te rastgele orman algoritması için verilen aralıklarda tanımlamalar yapılmıştır. Tablo 4'te kullanılan yükseltme (xgBoost, adaBoost, gradientBoost) algoritmaları için verilen aralıklarda tanımlamalar yapılmıştır.

Tablo 3. Kullanılan Rastgele orman algoritması için parametre ve değer aralıkları

Algoritma	Parametreler	Değer Aralıkları
Rastgele Orman Algoritması	Rasgele Örnek(Bootstrap)	
	Ormandaki ağaç sayısı(n_estimator)	{1, 2, 4, 8, 16, 32, 64, 100, 200}
	Maksimum Derinlik	[1, 32]+8
	Minimum Örnek Sayısı	{2, 5, 10}
	Yaprak Taban Düğümdeki Örnek Sayısı	{3, 4, 5}

Tablo 4. Kullanılan yükseltme algoritmaları parametreleri ve değer aralıkları

Algoritma	Parametreler	Değer Aralıkları
Yükseltme Algoritmaları (XGBoost, AdaBoost, GradientBoost)	Maksimum Derinlik	[3, 10]+2
	Öğrenme Oranı	[0,01, 0,1]+0,02
	Gamma	[0, 30]+10
	Lambda	1
	Alt Örnekleme	[0,1]
	Çocuk Düğüm Min. Ağırlıkları	[0, 30]+10

Tablo 5. Yapay sinir ağı en iyi parametre değerleri

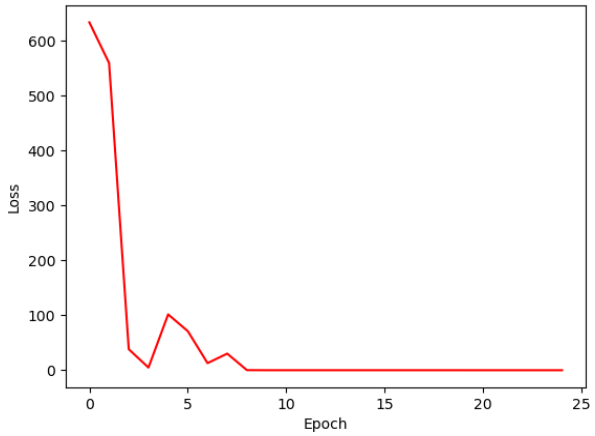
Parametreler	En İyi Sonuç Veren Değerler
Giriş Katmanı Nöron Sayısı	32
1. Gizli Katman Nöron Sayısı	9
2. Gizli Katman Nöron Sayısı	16
Çıkış Katmanı Nöron Sayısı	1
Öğrenme Algoritması	Geriye Yayılım Algoritması
Aktivasyon Fonksiyonu	Relu
Öğrenme Oranı	0.98
Döngü (epoch) Sayısı	25
Momentum Katsayısı	0,5
Alt Örneklerin (batch) Boyutu	40

Bu çalışmada ağ trafiğinde saldırı paketlerini sınıflandırmak için kullanılan yapay sinir ağı için kullanılan parametrelerin en iyi değerleri Tablo 5'te gösterilmiştir.

Nöron sayıları belirlenirken Giriş Katmanı için veri setinde kullanılan parametrelerin sayısına göre belirlenmiştir. 1. ve 2. gizli katman nöron sayıları deneme yanılma ile belirlenmiştir. Çıkış katmanı nöron sayısı ise elde edilmek istenen sınıflandırma değerlerini tespit etmek için tehdit yok (0)– tehdit var (1) seçeneklerinden dolayı 1 olarak belirlenmiştir.

Yapay sinir ağları ile yapılan sınıflandırmada Şekil 3'deki grafikten göründüğü üzere başlangıçtaki hata değerlerinin 600'lerin üzerinde olduğu ve eğitimin

ilerleyen döngülerinde hata değerinin sıfır değerine yaklaştığı görülmektedir. Bu durum YSA'nın bu veri seti için başarılı bir eğitim gerçekleştirdiğinin göstergesidir.



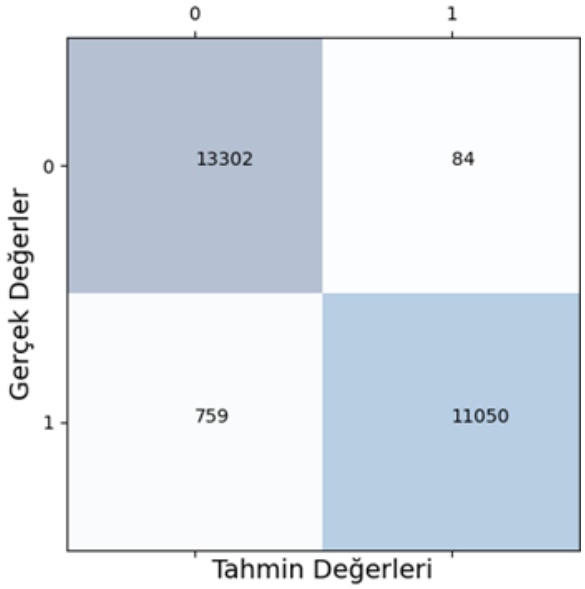
Şekil 3. YSA Sınıflandırmasında veri kaybı

Tablo 6'da yer alan performans değerlendirme ölçütlerine göre sınıflandırma sonucunda kesinlik değerlerinin tehdit olmadığı durumların bazılarının tehdit olarak belirlendiğini göstermektedir. Duyarlılık değerleri incelendiğinde ise tehdit olarak belirlenen değerlerin bazılarının yanlış sınıflandırıldığı görülmektedir. Bu iki durumun harmonik ortalamasını veren F1 Score değeri ise doğruluk puanıyla yaklaşık aynı sonucu verdiği için yapay sinir ağlarıyla yapılan bir sınıflandırma doğru bir model olabilir.

Tablo 6. YSA performans değerlendirme ölçütleri

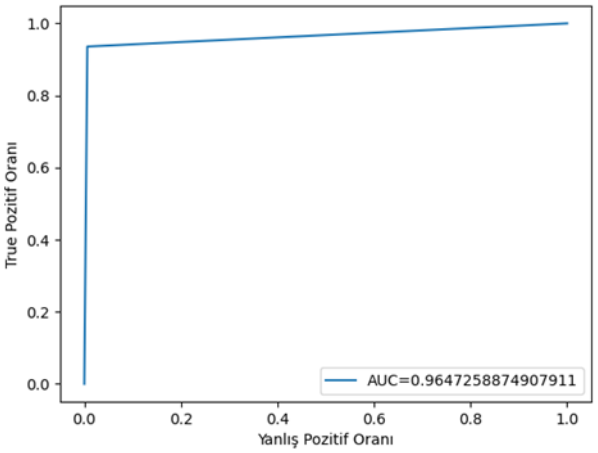
Sınıflandırma Değerleri	Kesinlik	Duyarlılık	F1-Score
0	0,97	1,00	0,98
1	1,00	0,96	0,98

Şekil 4' de YSA modeline ait karmaşıklık matrisine göre 840 adet verinin yanlış sınıflandırıldığı görülmektedir.



Şekil 4. YSA modeline ait karmaşıklık matrisi

Şekil 5'de YSA modelinin AUC değerine göre yaklaşık %96 oranında doğru sınıflandırma yapıldığı gözlenmiştir.



Şekil 5. YSA modeline ait ROC Eğrisi

Bu çalışmada ağ trafiğinde saldırı paketlerini sınıflandırmak için kullanılan Rastgele Orman algoritması için kullanılan parametrelerin en iyi değerleri Tablo 7'de gösterilmiştir.

Tablo 7. Rastgele orman algoritması en iyi parametre değerleri

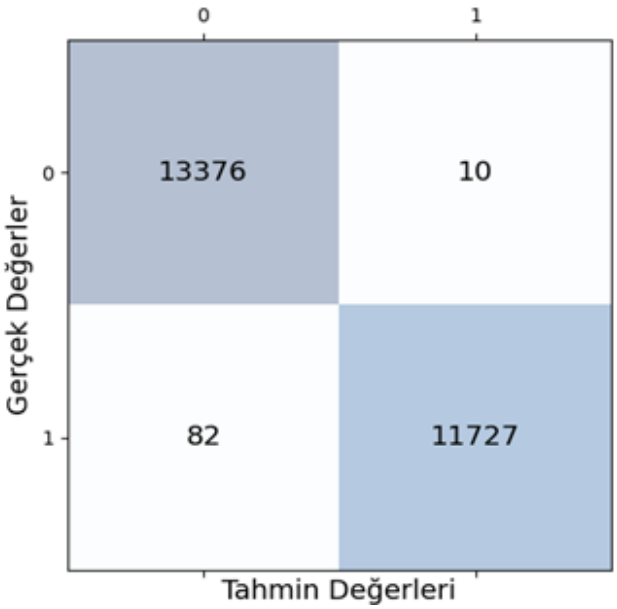
Ormandaki Ağaç Sayısı(n_estimator)	{1, 2, 4, 8, 32, 64, 100, 200}
Maksimum Derinlik	[1, 32]+8
Minimum Örnek Sayısı	{2, 5, 10}
Yaprak Taban Düğümdeki Örnek Sayısı	{3, 4, 5}

Tablo 8'de yer alan performans değerlendirme ölçütlerine göre sınıflandırma sonucunda kesinlik değerlerinin tehdit olmadığı durumların bazılarının tehdit olarak belirlendiğini göstermektedir. Duyarlılık değerleri incelendiğinde ise tehdit olarak belirlenen değerlerin bazılarını yanlış sınıflandırıldığı görülmektedir. Bu iki durumun harmonik ortalamasını veren F1 Score değeri ise doğruluk puanıyla yaklaşık aynı sonucu verdiği için rastgele orman algoritması ile yapılan bir sınıflandırma doğru bir model olabilir.

Tablo 8. RF performans değerlendirme ölçütleri

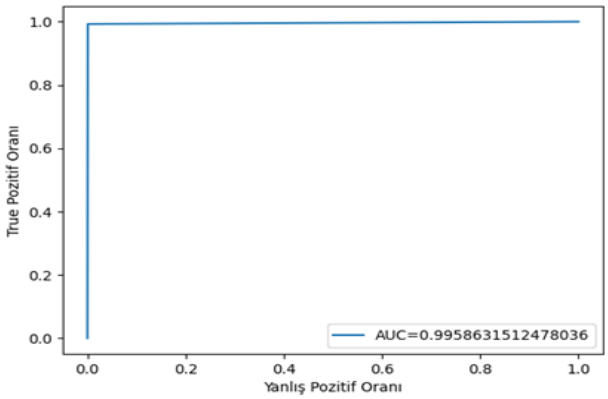
Sınıflandırma Değerleri	Kesinlik	Duyarlılık	F1-Score
0	0,99	1,00	1,00
1	1,00	0,99	1,00

Şekil 6'da RF modeline ait karmaşıklık matrisine göre 92 adet verinin yanlış sınıflandırıldığı görülmektedir.



Şekil 6. RF modeline ait Karmaşıklık Matrisi

Şekil 7'de RF modelinin AUC değerine göre yaklaşık %99 oranında doğru sınıflandırma yapıldığı gözlenmiştir.



Şekil 7. RF modeline ait ROC Eğrisi

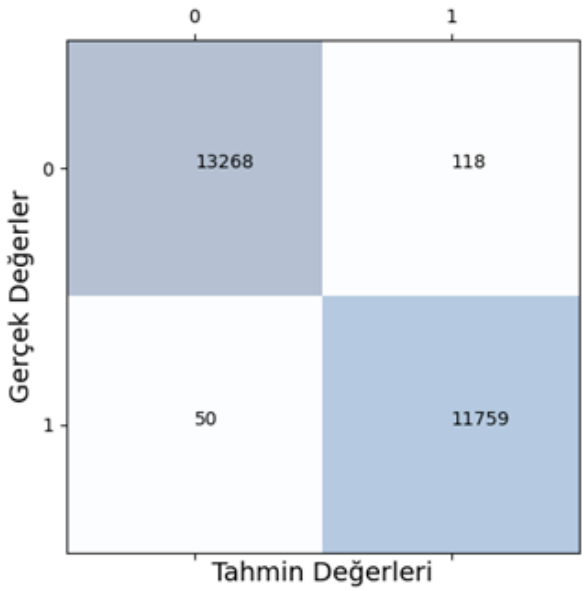
Tablo 9 incelendiğinde YSA için RMSE değerinin 0.1299 olması sınıflandırma sonucunun kabul edilebilir aralıkta olduğunu göstermektedir. R2 değeri sınıflandırma modelinin %93,23 oranında doğru olduğunu göstermektedir. RF algoritması için değerler incelendiğinde RMSE değerinin 0.1078 olması sınıflandırma sonucunun kabul edilebilir aralıkta olduğunu göstermektedir. R2 değeri sınıflandırma modelinin %95,32 oranında doğru olduğunu göstermektedir. Yükseltme algoritmaları ile yapılan sınıflandırmalarda kullanılan algoritmalar; xgBoost algoritması, adaBoost algoritması ve gradientBoost algoritmasıdır. Bu üç algoritma tek tek incelendiğinde ise ortaya çıkan hata oranları incelendiğinde xgBoost için RMSE değerinin 0,0816 olması sınıflandırma sonucunun kabul edilebilir aralıkta olduğunu göstermektedir. R2 değeri sınıflandırma modelinin %97.32 oranında doğru olduğunu göstermektedir. Aynı şekilde adaBoost algoritması için de RMSE değerinin 0,0802 olması sınıflandırma değerinin kabul edilebilir aralıkta olduğunu göstermektedir ve 0 değerine diğerlerinden daha yakın olduğu için tahminin çok güçlü olduğunu göstermektedir. R2 değeri sınıflandırma modelinin %97.41 oranında doğru olduğunu göstermektedir. GradientBoost algoritması için ise RMSE değeri 0,1912olarak çıkmıştır ve kabul edilebilir aralıktadır. Ancak diğer algoritmalara kıyasla RMSE değeri 0' dan daha uzaktır. R2 değerine bakıldığında ise sınıflandırma modelinin %85,32 oranında doğru olduğu görünmektedir.

Tablo 9. Kullanılan algoritmaların hata oranları

Model	MAE	MSE	RMSE	R2	Doğruluk Puanı
ANN	0.031	0.0168	0.1299	0.9323	0.9802
RF Algoritması	**0.0816**	**0.0116**	**0.1078**	**0.9532**	**0.9963**
XgBoost	0,0066	0,0066	0,0816	0,9732	0,9933

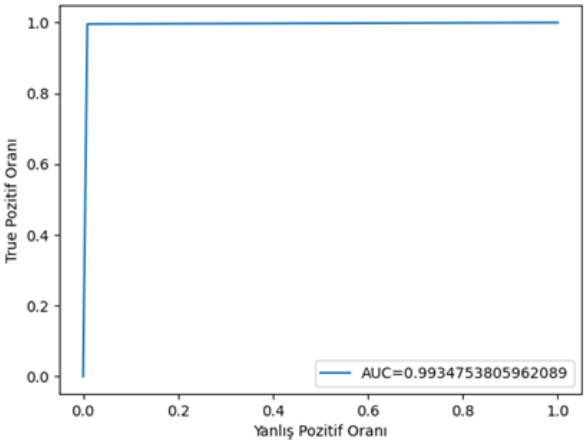
AdaBoost	0,0064	0,0064	0,0802	0,9741	0,9935
gradientBoost	0,0366	0,0366	0,1912	0,8532	0,9634

Bir diğer kıyas için algoritmaların ROC eğrisinden elde edilen AUC değerlerine bakılmaktadır. Şekil 8'de gösterilen xgBoost algoritma modeline ait karmaşıklık matrisine göre 168 adet verinin yanlış sınıflandırıldığı görülmektedir.



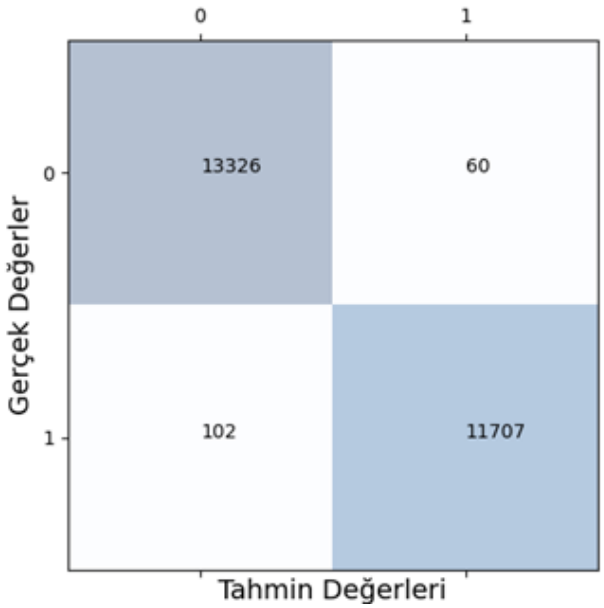
Şekil 8. xgBoost modeli için Karmaşıklık Matrisi

Şekil 9'da xgBoost algoritma modelinin AUC değerine göre yaklaşık %99 oranında doğru sınıflandırma yapıldığı gözlenmiştir.



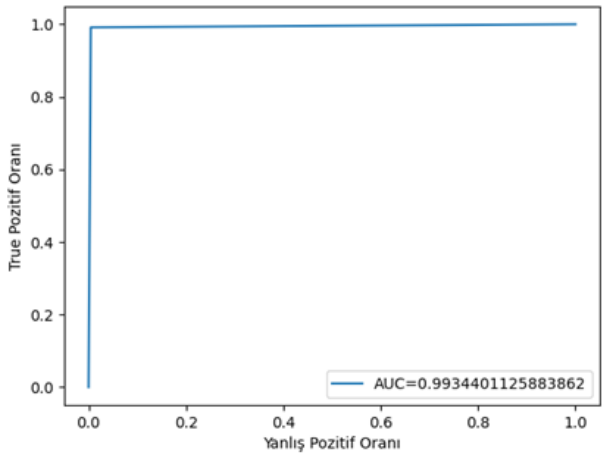
Şekil 9. XgBoost modeline ait ROC Eğrisi

Şekil 10'da gösterilen adaBoost algoritma modeline ait karmaşıklık matrisine göre 162 adet verinin yanlış sınıflandırıldığı görülmektedir.



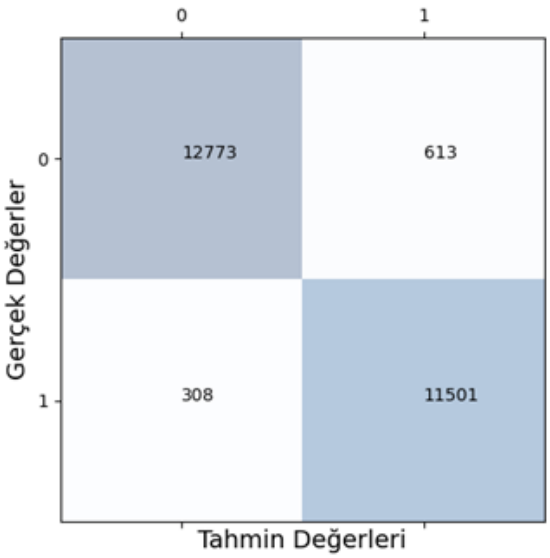
Şekil 10. adaBoost modeli için Karmaşıklık Matrisi

Şekil 11'de adaBoost algoritma modelinin AUC değerine göre yaklaşık %99 oranında doğru sınıflandırma yapıldığı gözlenmiştir.



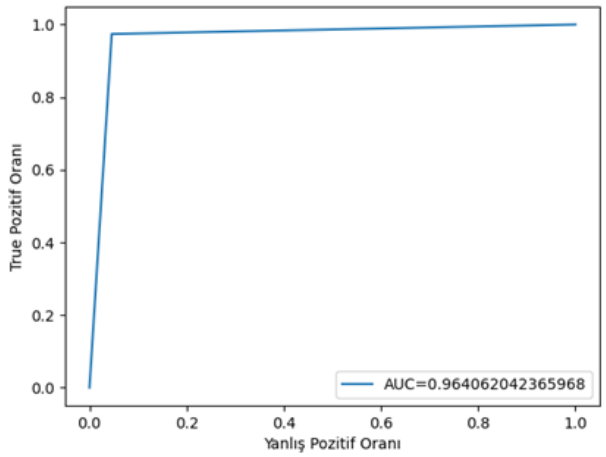
Şekil 11. AdaBoost modeline ait ROC Eğrisi

Şekil 12'de gösterilen gradientBoost algoritma modeline ait karmaşıklık matrisine göre 921 adet verinin yanlış sınıflandırıldığı görülmektedir.



Şekil 12. gradientBoost modeli için Karmaşıklık Matrisi

Şekil 13'te gradientBoost modelinin AUC değerine göre yaklaşık %96 oranında doğru sınıflandırma yapıldığı gözlenmiştir.



Şekil 13. GradientBoost modeline ait ROC Eğrisi

4. Tartışma ve Sonuç

Bu çalışmasında ağ simülasyonu amacı ile mininet ortamında yazılım tanımlı bir ağ tasarlanmıştır. Oluşturulan ağ tasarımında güvenlik tehditlerini tespit etmek için makine öğrenme algoritmalarından Yapay Sinir Ağı, Rastgele Orman, XGBoost, AdaBoost, GradientBoost algoritmaları kullanılmıştır. Ağ trafiğinde saldırı paketlerini sınıflandırmak için 2 sınıflı yapılandırma yapılmıştır. Yapılan sınıflandırma tehdit yok (0)- tehdit var (1) olarak sınıflandırılmıştır.

Makine öğrenme algoritmalarının başarısını tespit edebilmek için ağ üzerinde elde edilen veriler ve internet üzerinde yer alan veri seti birleştirilerek geniş bir veri seti oluşturulmuştur. Kullanılan makine öğrenme algoritmalarında en iyi parametreleri belirlemek için Izgara Temelli Arama Algoritması kullanılmıştır. Bu arama algoritmasında en iyi parametreler belirlenirken RMSE değeri referans alınmıştır.

Bu çalışma kapsamında internet üzerinden alınan veri setine ağ üzerinde oluşturulan veri seti eklenerek birleştirilen veri seti eğitim seti olarak kullanılmıştır. Test verisi olarak ise hping3 ile üretilen farklı parametlere sahip veriler kullanılmıştır. Mininet ortamında DDoS saldırısı oluşturabilmek için hping3 programı ile trafik üretimi yapılmıştır. Bunun sonucunda kullanılan makine öğrenme algoritmaları ile saldırı sınıflandırması yapılmıştır. Elde edilen sonuçlar, uygulanan yükseltme algoritmalarından adaBoost algoritmasının iyi tahminler verdiğini ortaya koymuştur. Çünkü, bu algoritmada en küçük MSE (0,0064) ve MAE (0,0064) değerleri elde edilmiştir. Ancak bu değerler model seçiminin doğruluğunu tek başına sağlamamaktadır. Bunun için modellerin doğruluk puanlarına bakılması gerekmektedir. Doğruluk puanı 1'e en yakın değere sahip modelin daha iyi tahmin verdiğini göstermektedir. AdaBoost algoritması için doğruluk puanı 0,9935' tir. GradientBoost için doğruluk puanı 0,9634' dür. XgBoost için doğruluk puanı 0,9933' dür. Yapay Sinir Ağı için doğruluk puanı 0,9802' dir. Rastgele Orman algoritması için ise doğruluk puanının 0,9963 olduğu görülmektedir. Bu sonuçlar neticesinde Rastgele Orman algoritma modeli sayesinde daha az hata ile saldırıların sınıflandırılması yapılmıştır.

Özetle bu çalışmada, mininet ortamında tasarlanan yazılım tanımlı ağlara yapılan saldırıların çeşitli makine öğrenme algoritmaları ile tehdit olduğu ve tehdit olmadığı durumların tespiti yapılarak, tehdit olan durumların engellenebilmesi sağlanmaktadır. Ayrıca kullanılan makine öğrenme yöntemlerinden en az hata ile saldırı sınıflandırılması yapılabilmesi sağlanmıştır.

5. Kaynaklar

Aidil, A., Citra, W, Hanifa, P.H., Marcello, Y.A., Muhammad, R., Putri, Z., Sherly, A., 2023. Information Security Implementation of DDoS Attack Using HPing3 Tools. JComce - Journal of Computer Science, 1, (4).

Akbaş, M.F., Karaarslan, E., Güngör, C., 2016. Yazılım Tanımlı Ağların Güvenliğinde Yapay Zeka Tabanlı Çözümler: Ön İnceleme. International Artificial Intelligence and Data Processing Symposium, 17-18 September, Malatya, 496-501.

Akyildiz, I., Lee, A., Wang, P., Luo, M., Chou, W., 2014. A Roadmap for Traffic Engineering in SDN-OpenFlow Networks. Computer Networks, 71, 1-30.

Camilleri, M., Neri, F., Papoutsidakis, M., 2014. An Algorithmic Approach to Parameter Selection in Machine Learning Using Meta-Optimization Techniques. WSEAS Transactions on Systems, 13, (1), 203-212.

Ferguson, A.D., Guha, A., Liang, C., Fonseca, R., Krishnamurthi, S., 2013. Participatory Networking: An API for Application Control of SDNs. ACM SIGCOMM Computer Communication Review, 327-338.

Gencal, M.C., Oral M., 2022. Evrimsel Algoritmalar İçin Yeni Bir Meta-İyileştirici: Bipolar Eşleşme Eğilimi. Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 28, (2), 313-323.

Guo, A., Yuan, C., He, G., Xu, L., 2018. Research on SDN/NFV Network Traffic Management and Optimization based on Big Data and Artificial Intelligence. The 18th International Symposium on Communication and Information Technologies, China, 377-382.

Hadi Hadi M.R., 2022. A Proposed Approach Network Intrusion Detection System (NIDS) Using Deep Learning for Software Defined Network (SDN): A Futuristic Approach, Karabük University, Institute of Graduate Programs, Master Thesis, p.80, Karabük.

Heorhiadi, V., Reiter, M.K., 2016. Simplifying Software-Defined Network Optimization Using SOL. Proceedings of the 13th USENIX Symposium on Network Systems Design and Implementation, 16-18 March, Santa Clara, 223-237.

Kaur, S., Singh, J., Ghumman, N.S., 2014. Network Programmability Using POX Controller. International Conference on Communication Computing Systems (ICCCS), 8-9 August 2014, Shaheed Bhagat Singh State Technical Campus, India, 134-138.

Kobayashi, M., Seetharaman, S., Parulkar, G., Appenzeller, G., Little, J., Reijendam, J., Weissmann, P., McKeown, N., 2013. Maturing of OpenFlow and Software-Defined Networking Through Deployments. Computer Networks, 61, 151-175.

Latah, M., Toker, L., 2016. Application of Artificial Intelligence to Software Defined Networking: A Survey. Indian Journal of Science and Technology, 9, (44), 1-7.

Mihai-Gabriel I., Victor-Valerin P., 2014. Achieving DDoS resiliecny in a Software Defined Network by Intelligent Risk Assessment Based on Neural Networks and Danger Theory. Proceedings of IEEE 15th International Symposium on Computational Intelligence and Informatics (CINTI), November, Budapeşt, Hungary, 319-324.

Niyaz, K., Sun, W., Javaid, A.Y., 2016. A Deep Learning Based DDoS Detection System in Software-Defined Networking (SDN). EAI Endorsed Transactions on Security and Safety, 4, (12), 1-12.

Özalp, A.N., 2023. Siber Saldırıların Tespitinde Yapay Zeka Tabanlı Algoritma Tasarımı. Karabük Üniversitesi, Lisansüstü Eğitim Enstitüsü, Doktora Tezi, 84s., Karabük.

Ye, J., Cheng, X., Zhu, J., Feng, L., Song, L., 2018. A DDoS Attack Detection Method Based on SVM in Software Defined Network. Security and Communication Networks, 1-8.