T.C.
GÜMÜŞHANE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKÂ VE AKILLI SİSTEMLER ANABİLİM DALI

ELASTIC STACK VE MAKİNE ÖĞRENMESİ İLE IOT GÜVENLİĞİ: DOS SALDIRILARINA KARŞI AKILLI SAVUNMA SİSTEMİ

YÜKSEK LİSANS

Tuğrul YAĞBASAN

HAZİRAN-2024
GÜMÜŞHANE

T.C.
GÜMÜŞHANE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKÂ VE AKILLI SİSTEMLER ANABİLİM DALI

ELASTIC STACK VE MAKİNE ÖĞRENMESİ İLE IOT GÜVENLİĞİ: DOS SALDIRILARINA KARŞI AKILLI SAVUNMA SİSTEMİ

IOT SECURITY WITH ELASTIC STACK AND MACHINE LEARNING: INTELLIGENT DEFENCE SYSTEM AGAINST DOS ATTACKS

YÜKSEK LİSANS

Tuğrul YAĞBASAN

HAZİRAN-2024
GÜMÜŞHANE

T.C.
GÜMÜŞHANE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ZEKÂ VE AKILLI SİSTEMLER ANABİLİM DALI

ELASTIC STACK VE MAKİNE ÖĞRENMESİ İLE IOT GÜVENLİĞİ: DOS SALDIRILARINA KARŞI AKILLI SAVUNMA SİSTEMİ

IOT SECURITY WITH ELASTIC STACK AND MACHINE LEARNING: INTELLIGENT DEFENCE SYSTEM AGAINST DOS ATTACKS

YÜKSEK LİSANS

Tuğrul YAĞBASAN

Danışman: Dr. Öğr. Üyesi Hayati TÜRE

HAZİRAN-2024
GÜMÜŞHANE

BİLİMSEL ETİĞE UYGUNLUK BEYANI

Yüksek Lisans tezi olarak hazırlamış olduğum "**Elastic Stack ve Makine Öğrenmesi ile IoT Güvenliği: DOS Saldırılarına Karşı Akıllı Savunma Sistemi**" isimli bu tezimin, tamamen kendi çalışmam olduğunu, her alıntıya kaynak gösterdiğimi, alıntı yaptığım tüm çalışmaları kaynakçada belirtiğimi ve Gümüşhane Üniversitesi'nin lisanslı kullanıcısı olduğu intihal yazılım programı ile Lisansüstü Eğitim Enstitüsü'nün belirlediği kıstaslara uygun olarak raporladığımı taahhüt ederim. Tezimin kâğıt ve elektronik kopyalarının Gümüşhane Üniversitesi Lisansüstü Eğitim Enstitüsü arşivinde saklanmasına izin verdiğimi onaylarım.

Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddeleri uyarınca gereğinin yapılmasını arz ederim.

13/06/2024

Tuğrul YAĞBASAN

TEŞEKKÜR

Çalışmalarım sırasında bilgi ve tecrübelerinden faydalandığım tez danışmanım Sayın Dr. Öğr. Üyesi Hayati TÜRE hocama,

Ayrıca tüm bu çalışmalarım sürecinde bana vakit olanağı sağlayan ve desteğini esirgemeyen eşim Özge YAĞBASAN'a ve değerli aileme katkılarından dolayı teşekkür ederim.

Tuğrul YAĞBASAN

GÜMÜŞHANE – 2024

ÖZET

Teknolojinin gelişmesiyle IoT cihazların kullanımı artmaktadır. Bu cihazlar, iletişim kurma ve veri toplama gibi yeteneklerle endüstriyel, sağlık, ulaşım gibi birçok alanda kullanılmaktadır. Ancak, bu cihazların artan kullanımı, siber saldırıların da artmasına neden olmaktadır. Geçmişte Mirai botnet saldırıları gibi örnekler, IoT cihazlarının güvenliğinin ne kadar kritik olduğunu göstermiştir. Bu nedenle, çalışmada Elastic Stack, makine öğrenmesi ve derin öğrenme teknolojilerinin kullanımıyla IoT cihazlarının güvenliğini artırmaya yönelik akıllı savunma sistemi geliştirilmesi amaçlanmıştır. Çalışmada Bot-IoT ve Unsw-IoT-B datasetleri kullanılarak atakların belirlenmesi üzerine yoğunlaşılmıştır. Çalışmada makine öğrenmesi yöntemlerinden random forest, Decision Tree (Information Gain), Decision Tree (Gini Index), Naive Bayes teknikleri kullanılmıştır. Derin öğrenme yöntemlerinden Lstm, Rnn ve Svm kullanılarak datasetleri üzerinde atak ve atak tipleri tespiti gerçekleştirilmiştir.

Tasarlanan modelde makine öğrenmesi yöntemleriyle %0.9998 lik başarım oranları yakalanmış olup derin öğrenme yöntemleriyle %0.98 lik doğruluk oranı elde edilmiştir. Bu modelle birlikte yapay zeka destekli saldırı tespit sistemlerinin IoT cihazlarının güvenliklerinde faydalı olacağını göstermektedir.

Anahtar Kelimeler: Derin öğrenme, Elasticsearch, Makine öğrenmesi,Saldırı Önleme Sistemi, Siber Güvenlik

SUMMARY

With the development of technology, the use of IoT devices is increasing. These devices are used in many areas such as industrial, health, transportation with capabilities such as communication and data collection. However, the increasing use of these devices also leads to an increase in cyber-attacks. In the past, examples such as Mirai botnet attacks have shown how critical the security of IoT devices is. Therefore, this study aims to develop an intelligent defense system to increase the security of IoT devices by using Elastic Stack, machine learning and deep learning technologies. The study focuses on the detection of attacks using Bot-Iot and Unsw-IoT-Botnet datasets. Random forest, Decision Tree (Information Gain), Decision Tree (Gini Index), Naive Bayes techniques were used in the study. Lstm, Rnn and Svm, which are deep learning methods, were used to detect attacks and attack types on datasets.

In the designed model, %0.9998 success rate was achieved with machine learning methods and %0.98 accuracy rate was achieved with deep learning methods. With this model, it shows that artificial intelligence supported intrusion detection systems will be useful in the security of IoT devices.

Keywords: Cyber Security, Deep learning, Elasticsearch, Intrusion Prevention System, Machine Learning,

İÇİNDEKİLER

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

IOT	: Nesnelerin İnterneti (Internet of Things)
DDOS	: Dağıtılmış Hizmet Engelleme Saldırısı
DOS	: Hizmet Engelleme Saldırısı (Denial of Service)
OS	: İşletim Sistemi (Operating System)
DNS	: Alan Adı Sistemi (Domain Name System)
API	: Uygulama Programlama Arayüzü
ELK	: Elasticsearch, Logstash, Kibana
NN	: Yapay Sinir Ağları (Neural Networks)
MLP	: Çok Katmanlı Algılayıcı (Multilayer Perceptron)
CNN	: Konvolüsyonel Sinir Ağları (Convolutional Neural Networks)
RNN	: Tekrarlayan Sinir Ağları (Recurrent Neural Networks)
LSTM	: Uzun Kısa Süreli Bellek (Long Short-Term Memory)
SVM	: Destek Vektör Makineleri (Support Vector Machines)
KNN	: K-En Yakın Komşu (K-Nearest Neighbors)
GB	: Gigabayt
ZB	: Zettabayt
BoT-IoT	: Botnet Internet of Things: Botnet Nesnelerin İnterneti
DNN	: Deep Neural Network: Derin Sinir Ağı
FN	: False Negative: Yanlış Negatif
FNR	: False Negative Rate -Yanlış Negatif Oranı
FP	: False Positive: Yanlış Pozitif
FPR	: False Positive Rate: Yanlış Pozitif Oranı
NB	: Naive Bayes: Naif Bayes
RF	: Random Forest: Rastgele Orman
ROC	: Receiver Operating Characteristic: Alıcı İşletim Karakteristiği
SVM	: Support Vector Machine: Destek Vektör Makinesi
TP	: True Positive: Doğru Pozitif
UNSW	: University of New South Wales: Yeni Güney Galler Üniversitesi

1. GİRİŞ

Siber güvenlik tehditleri, özellikle de hizmet engelleme saldırıları (DDoS), günümüz dijital ekosisteminin belirgin bir riski haline gelmiştir. Bu tehditler, Internet of Things (IoT) cihazlarının hızla artan kullanımıyla birlikte daha da karmaşık hale gelmiştir. IoT cihazları, endüstriyel sistemlerden ev otomasyonuna kadar geniş bir yelpazede kullanılmakta ve bu durum, siber saldırganlar için yeni ve potansiyel olarak savunmasız hedefler sunmaktadır. DDoS saldırıları, bu cihazlar üzerindeki olumsuz etkilerini artırarak, internet erişiminde kesintilere, hizmetlerin çökmesine ve hatta maddi kayıplara yol açabilmektedir. Çalışmada, IoT cihazlarının DDoS saldırılarına karşı savunmasızlıklarını inceleyerek, bu alandaki güvenlik önlemlerinin geliştirilmesine katkıda bulunmayı amaçlamaktadır.

IoT cihazlarının kullanım sayıları günden güne hızlı bir artış göstermektedir. Internet of Thing (IoT) sayıları 2021 yılında yaklaşık 10 milyar adet civarındayken 2030 yılında bu sayının 25 milyar adeti geçeceği beklenmektedir. Kullanım sayısının artmasıyla birlikte IoT cihazları tarafından üretilen data akışının 2025 yılına kadar 70 ZB'ı (zettabayt) geçmesi beklenmektedir. Gerek IoT cihazlarının sayısının artması gerekse kullanım alanlarının genişlemesiyle birlikte bu alandaki siber tehlikelerin riski de gün geçtikçe daha önemli hale gelmektedir.

IoT cihazlarına yönelik siber saldırıların başında gelen hizmet engelleme saldırıları (Dos) bu cihazların kullanım alanlarının artmasıyla birlikte daha da tehlikeli hale gelmektedir. Örneğin 21 Ekim 2016 tarihinde DNS (Domain Name Service) servis sağlayıcılığı hizmeti vermekte olan Dyn firmasına Mirai botnetler vasıtasıyla yapılan saldırılar sonucunda tüm dünyayı etkileyen internet erişim problemleri yaşanmıştır (Eustis, 2019). Bu saldırılardan etkilenen firma ve kişiler gerek itibar kaybı gerekse ciddi maddi kayıplar yaşamıştır. Bu saldırıların büyük çoğunluğunun makineler kullanılarak yapıldığı düşünüldüğünde bunlara karşı savuma sistemlerinin de yapay zeka kullanan makineler tarafından önlenmesi kaçınılmaz bir ihtiyaç doğurmuştur.

2. TEZ KONUSU VE KAPSAMI

Çalışmamızda iki farklı veri seti üzerinde farklı makine öğrenmesi ve derin öğrenme modelleri kullanarak saldırı ve kategori tahmini yapmayı amaçlamaktadır. Çalışmamız makine öğrenimi ve derin öğrenme modellerinin performansını ölçmek ve hangi modelin en iyi sonuçları verdiğini belirlemeyi hedeflemektedir. Performans metrikleri arasında kesinlik, özgünlük, hassasiyet, YP (Yanlış Pozitif) oranı, YN (Yanlış Negatif) ve doğruluk oranları bulunmaktadır. Ayrıca hesaplanan metrikler hakkında nasıl hesaplandığı ve çıkan sonuçların yorumlanması ayrıntılı olarak ele alınmıştır. Belirlenen ataklar opensource elkstack yazılımının web servis entegrasyonu yapılarak attack olarak etiketlenen verilerin önlenmesini sağlayan yöntemle birlikte tehlikeli ataklara karşı bir akıllı savunma sistemi modeli üzerine odaklanmıştır.

Kullanılan modeller arasında Lojistik Regresyon, K-En yakın komşu (KNN), Destek Vektör Makineleri (SVM), Karar Ağaçları (Gini ve Entropy), Naive Bayes, Yapay Sinir Ağlar (NN), Çok Katmanlı Algılayıcı (MLP), Konvolüsyonel Sinir Ağlar (CNN), Tekrarlayan Sinir Ağları (RNN), Uzun Kısa Süreli Bellek (LSTM) bulunmaktadır.

Bu çalışmada, Unsw-IoT-Botnet ve BoT-IoT olmak üzere iki veri seti kullanılmıştır. Unsw-IoT-Botnet veri seti, çeşitli IoT cihazlarına yönelik botnet saldırılarını içermekte olup, veri setinde 45 özellik bulunmaktadır. Bu özellikler arasında kaynak ve hedef IP adresleri, port numaraları, paket ve bayt sayıları gibi ağ akış özellikleri bulunmaktadır. BoT-IoT veri seti ise UNSW Canberra'nın Cyber Range Lab'inde oluşturulmuş olup, normal ve botnet trafiğini içeren gerçekçi bir ağ ortamını temsil etmektedir. Veri setinin pcap dosyaları 69.3 GB boyutunda ve 72.000.000'dan fazla kayıt içerirken, CSV formatında çıkarılan trafik verisi 16.7 GB boyutundadır. Bu veri seti, DDoS, DoS, OS ve Hizmet Taraması, Keylogging ve Veri kaçırma saldırılarını içermektedir. Bu veri setleri, IoT ağlarında saldırı tespiti için kullanılan makine öğrenmesi ve derin öğrenme algoritmalarının performansını değerlendirmek amacıyla işlenmiş ve analiz edilmiştir.

Bu çalışmada elde edilen saldırı tespit sonuçları, Elasticstack açık kaynak yazılımına API vasıtasıyla gönderilmiş ve akıllı bir savunma sistemi önerilmiştir. Bu çalışmanın odak noktası, makine ve derin öğrenme algoritmalarının sonuçlarını kullanarak gerçek zamanlı saldırı tespiti ve önleme yeteneklerine sahip, dinamik ve etkili bir IoT ağ güvenlik çözümü sunmaktadır.

2.1. Önceki Çalışmalar

Shema Alosaimi ve Saad M. Almutairi IoT ağlarının güvenliğini artırmak amacıyla derin öğrenme ve üç seviyeli algoritmaları birleştiren yenilikçi bir saldırı tespit yaklaşımı üzerinde çalışmışlardır. Çalışmalarında BoT-IoT veri seti üzerinde 2 farklı yaklaşım denemişlerdir. Bunlardan bir tanesi mevcut veri setindeki verileri azaltmak diğeri ise veriler üzerindeki dağılım eşitsizliklerine odaklanmıştır. Çalışmasında %99 oranında başarım sağlamıştır (Alosaimi, 2023).

Koroniotis ve arkadaşları BoT-IoT ve diğer 3 veri setinde en iyi 10 özelliği içeren bir alt kümeyle ilgili çalışmalar yapmıştır. Çalışmasında eğitilen SVM modelinden en yüksek verimi sağlamış ve veri setinde en yüksek hassasiyet ve en düşük düşmeyi 10 en iyi özellik veri kümesi sürümünde elde etmiştir (Koroniotis, 2019a) .

Joffrey ve arkadaşları BoT-IoT veri setinde sınıflandırmalar üzerine çalışmıştır. Çalışmasında sınıflandırmada doğru özellik belirmek amacıyla tek özellikli, iki özellikli ve üç özellikli Karar Ağacı modellerinin performanslarını değerlendirmiştir. 3 özellikli BoT-IoT veri seti kümesinden öğrenmesi kolay bir model üretmişlerdir (J. L. Leevy 2021a).

Joffrey ve arkadaşları diğer çalışmasında BoT-IoT veri seti üzerinde alt kategorilerde yer alan veri hırsızlığı kategorisindeki atakları tespit etmeye yönelik bir model üzerinde çalışmışlardır. Çalışmasında çapraz doğrulama yoluyla eğitilen ve test edilen sekiz sınıflandırıcının performansını ölçmüşlerdir. Özellikle CatBoost, LightGBM ve XGBoost modellerinin başarıları ön plana çıkmıştır (J. L. Leevy, 2021b).

Satish Pokhrel ve arkadaşları BoT-IoT veri seti üzerinde KNN algoritmasını kullanarak Botnet tespitindeki başarım oranlarından bahsetmişlerdir. Veri setindeki DDoS saldırılarının tespit oranlarını artırmak için SMOTE teknolojisi kullanarak veri setindeki iyileştirme sonuçlarının etkisi üzerine çalışmalar yapmışlardır (Pokhrel, 2021).

Muhammad Zeeshan ve arkadaşları çalışmasında DDoS attaklarının tespitinde protokol bazlı derin öğrenme tekniği üzerine odaklanmışlardır. Çalışmalarında UNSW-NB15 ve BoT-IoT veri setlerinde kötü niyetli trafiğin belirlenmesi için verilen özelliklerin sayısını neredeyse yarısı kadar azaltarak sonuca ulaşmışlardır. Geliştirdikleri modelle birlikte DoS ve DDoS trafiğini %96.3 oranında doğrulukla tespit etmişlerdir (M. Zeeshan, 2022).

Khalid ve arkadaşları IoT saldırı tespit sistemlerinde kullanılmak üzere DL anomali tabanlı IDS leri kullanmak için kapsamlı bir değerlendirme yapmışlardır. Özellikle DL teknikleriyle çıkarılan öğrenilmiş özelliklerin anomali tabanlı IDS

yaklaşımlarının performanslarını artırdığı üzerine çalışmalar yapmışlardır (Albulayhi, 2021).

Pushparaj ve arkadaşları saldırı tespit sistemlerinde özellik seçimi konusunda çalışma yapmıştır. Çalışmasında 2 farklı veri seti üzerinde özellik seçimi için Information Gain (IG) ve Gain Ratio (GR) özellik seçimi yöntemlerini kullanarak önerdiği modelde saldırı tespit sistemleri üzerinde yüksek doğruluk ve tespit oranı elde etmiştir. Çalışmasında %99 başarım sağlamıştır (Nimbalkar, 2021).

Erman Özer ve arkadaşları çalışmasında veri setlerinde en uygun özellik çiftlerini belirleyecek bir model üzerinde çalışmışlardır. Çalışmalarında en etkili ve en uygun özellik çiftlerinin ML algoritmalarının belirlenmesinde, giriş verisi çiftlerinin dağılımlarının önemli ve kritik olduğunu göstermişlerdir (Özer, 2021).

Tanzila ve arkadaşları IoT ağının performans ve güvenliğini artırma için evrişimsel sinir ağı (CNN) tabanlı bir model sunmuşlardır. IDS için bir IoT ağında derin öğrenme modelini araştırmışlardır. Çalışmalarında önerdikleri yöntemle birlikte NID ve BoT-IoT veri setlerinde %99 ve %95 doğruluk elde etmişlerdir (Saba, 2022).

Shahbaz Ahmad ve arkadaşları çalışmasında DDoS attaklara karşı savunma sistemi için makine öğrenmesi ve derin öğrenme yöntemlerinin hibrit karışımı olan bir model üzerinde çalışmışlardır. Önerilen modelde yazılım tabanlı bir ağ (SDN) ve ara katmanla birlikte pasif saldırı sisteminin uygulama yöntemi üzerine çalışmıştır (Khanday, 2023).

Orieb Abu ve arkadaşları IoT saldırı tespit sistemleri için gelişmiş bir PIO (Pigeon-inspired optimization) özellik seçimi algoritması üzerine çalışmalar yapmıştır. Önerdikleri algoritma olan LS-PIO (Local Search Pigeon-inspired Optimization) özellik seçimi algoritması ile veri setleri üzerindeki özellik sayısını azaltmak için tasarlamışlardır. Çalışma sonuçlarında önerdikleri sistemin daha yüksek doğruluk elde edildiğine dair sonuçlara yer verilmiş ve sistemin çalışma süresini azaltmak için K-Means algoritmasıyla birlikte daha hızlı olduğu sonuçuna varmıştır (Alghanam, 2023).

Zineb ve arkadaşları yapmış olduğu çalışmada Elastic Stack yardımıyla otomatik ağ trafiği analizi çıkarması üzerine bir sistem üzerine yoğunlaşmışlardır. Elastic Stack, ZEEK, Osquery, Kafka ve GeoLocation gibi araçların entegrasyonunu sağlayarak geniş işletme altyapılarına tam olarak uyumlu bir güvenlik analizi platformu oluşturmuşlardır. Önerdikleri modelle birlikte makine öğrenmesi yöntemleriyle entegre bir model oluşturmuşlardır (Maasaouii, 2023).

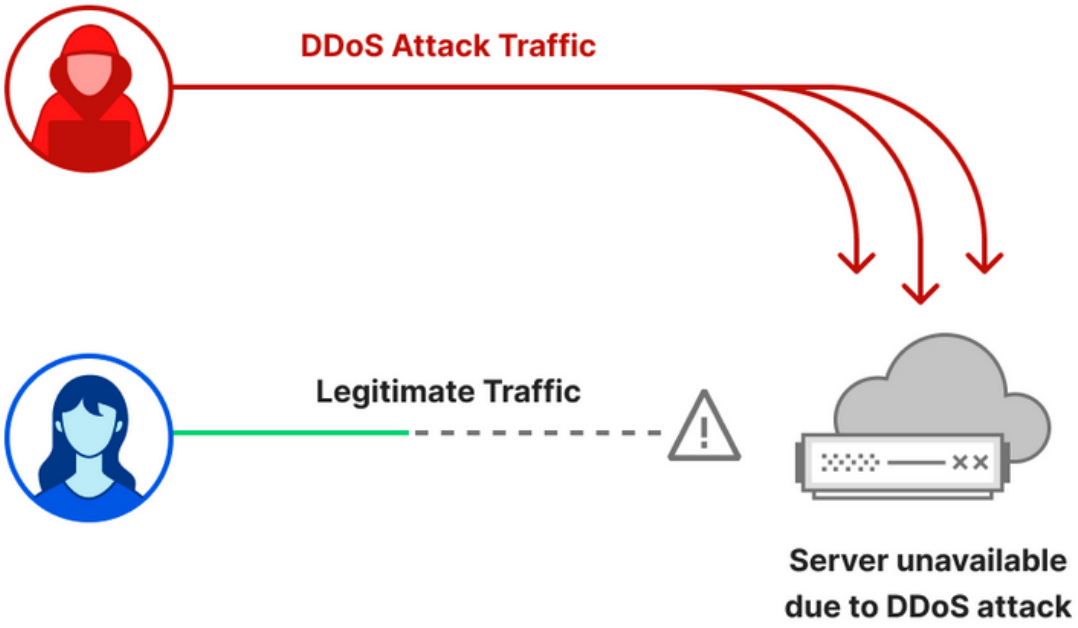
Tianyang ve arkadaşları DDoS ataklarını önlemek için yazılım tabanlı bir model üzerinde çalışma yapmıştır. Çalışmalarında DDoS saldırılarını tespit etmek ve şüpheli

özellikleri çıkarmak için entropi ve denetimsiz makine öğrenimi yöntemlerini birleştirmişlerdir. Ayrıca ek cihazlara ihtiyaç duymadan yeni bir boru hattı filtreleme mekanizması önermişlerdir. Deneysel sonuçlarına göre önerdikleri model saldırı türlerini önceden tanımlamadan hacimsel saldırıları azaltmak üzere yüksek doğruluk ve güçlü uyarlanabilirliğe sahip olduğunu göstermişlerdir (T. Cai, 2023).

Muhammad Ahmad ve arkadaşları UNSW-NB15 veri seti üzerinde uygulama ve aktarım katmanı özelliklerine dayalı denetimli makine öğrenmesi kullanarak IoT cihazlarında izinsiz giriş tespiti üzerine çalışmalar yapmıştır. Çalışmalarında MQTT ve TCP protokollerine odaklanan özellik kümelerinin kullanılmasını önermiştir. Veri setinde ikili sınıflandırmada, ortalama imputasyon kullanarak Random Forest (RF) ile %98 başarım oranı elde etmiştir (Ahmad, 2021).

2.2. DDoS Saldırıları

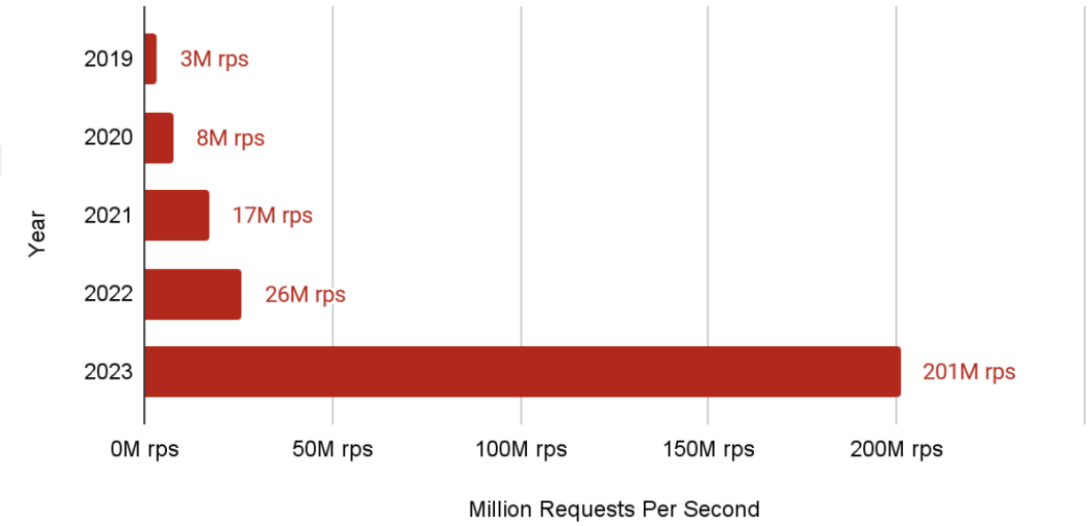
Distributed Denial of Service (DDoS) saldırıları, siber güvenlikte hizmet durdurma ve siber hizmetlerin aksaması konusunda tehditler oluşturan saldırı türlerinin en başında gelir. Bu saldırılar, bir hedef sistemi karşılayabileceğinden çok daha fazla yükleyerek hizmetlerin durmasına veya yavaşlamasına neden olur.



Şekil 1. DDoS ataklarının illüstrasyonu (URL-5).

Saldırılar genellikle birden fazla kaynaktan eşzamanlı olarak gerçekleştirilir. Birden fazla kaynaktan eş zamanlı gelen saldırılar tespit ve engelleme süreçlerini

zorlaştırır. Bu saldırılar gün geçtikçe internet ortamında yaygınlaşmakta ve etkileri daha da yıkıcı olmaktadır. Önemli internet sağlayıcılardan biri konumunda olan Cloudflare, 2023'ün dördüncü çeyreğinde 80 petabayttan fazla ağ katmanı saldırısını engellediğini raporladı ve saatte ortalama 996 ağ katmanı DDoS saldırısını otomatik olarak önlendiği yine raporlar arasındaki dikkat çeken detaylardan birisi konumundadır. Bu, DDoS saldırılarının bir önceki yıla göre %175, çeyrek döneme göre ise %25 artış kaydettiğini göstermektedir (URL-5). Bu saldırıların sadece Cloudflare de hizmet veren siteleri kapsadığı düşünülünce dünya çapındaki etkileri daha yıkıcı durumdadır.



Şekil 2. Cloudflare'e gelen yıllık DDoS ataklarının miktarı (URL-5).

Tarihteki bilinen en büyük zarar Mirai botnet saldırılarıyla başlamış olup vermiş olduğu zararlar birçok sektörde hissedilmiş durumdadır. Mirai botnet saldırıları, hedef sistemlerin çevrimdışı durmasına neden olarak gelir kaybına, müşteri memnuniyetsizliğine ve itibar zedelenmesine yol açmış durumdadır. Örneğin, Dyn DNS saldırısı sırasında, etkilenmiş web siteleri milyarlarca dolarlık potansiyel gelir kaybı yaşadığı bilinmektedir. Ayrıca, bu tür saldırılar, saldırıya uğrayan şirketlerin güvenlik yatırımlarını artırmalarına neden olmuş, saldırı almayan şirketlerinde bu konuda önlem alması gerekliliğini ortaya çıkararak tüm sektörler için ek maliyetler oluşturmuştur.

Ekim 2016'da Dyn DNS sağlayıcısına yapılan saldırıda AirBnB, Amazon, GitHub, HBO, Netflix, PayPal, Reddit ve Twitter gibi birçok popüler web sitesinin kullanıcılar tarafından erişilememesine neden olmuştur. Bu saldırı, 1.2 Tbps'lik bir zirve trafik hacmiyle tarihe geçti ve Mirai'nin gücünü ve etkisini gözler önüne sermiştir (URL-6).

3. MATERYAL VE YÖNTEM

3.1. Çalışmada Kullanılan Veri Setleri

Bu çalışmada, IoT ağlarında saldırı tespiti için Unsw-IoT-Botnet ve BoT-IoT veri setleri kullanılmıştır (URL-7). Unsw-IoT-Botnet veri seti, çeşitli IoT cihazlarına yönelik botnet saldırılarını içermekte olup, kaynak ve hedef IP adresleri, port numaraları, paket ve bayt sayıları gibi 45 farklı özellik sunmaktadır. Bu veri seti, saldırı tespit algoritmalarının performansını değerlendirmek için kullanılmıştır.

BoT-IoT veri seti ise UNSW Canberra'nın laboratuvar ortamında gerçeğe yakın bir ağ ortamı tasarlanarak oluşturulmuştur. Bu veri seti, normal ve botnet trafiğinin bir kombinasyonunu içermektedir (Koroniotis, 2019a). Veri setinin pcap dosyaları 69.3 GB boyutunda olup, 72.000.000'dan fazla kayıt içermektedir. CSV formatında çıkarılan trafik verisi ise 16.7 GB boyutundadır. Bu veri seti, DDoS, DoS, OS ve Hizmet Taraması, Keylogging ve Veri kaçırma saldırılarını içermektedir. Orijinal veri setinin %5'i seçici MySQL sorguları kullanılarak çıkarılmış ve yaklaşık 1.07 GB toplam boyutta, 3 milyon kayıt içeren 4 dosyadan oluşmaktadır.

Bu çalışmada elde edilen saldırı tespit sonuçları, Elasticstack (ELK Stack) açık kaynak yazılımına API vasıtasıyla gönderilmiştir. ELK Stack, Elasticsearch, Logstash ve Kibana bileşenlerinden oluşan bir log yönetim ve analitik yazılımıdır. Elasticsearch büyük veri kümelerinde hızlı arama ve analiz sağlar, entegre olarak çalışan Logstash ile birlikte veri işleme hattı olarak görev yapar, veriyi toplar, dönüştürür ve Elasticsearch'e aktarır, Kibana modülü ise verilerin görselleştirilmesini ve analiz edilmesini sağlayan bir kullanıcı arayüz kümesi hizmeti sağlayarak veriler hakkında detaylı raporlandırmaya olanak sağlar.

Çalışmada kullanılan makine öğrenmesi ve derin öğrenme modelleri Lojistik Regresyon (Logistic Regression), K-En Yakın Komşu (K-Nearest Neighbors), Destek Vektör Makineleri (Support Vector Machines), Karar Ağaçları (Gini ve Entropy), Naive Bayes, Yapay Sinir Ağları (Neural Networks), LSTM (Long Short-Term Memory) ve RNN (Recurrent Neural Network) algoritmalarıdır. Veri setleri üzerinde atak belirleme, kategori belirleme ve alt kategori belirleme çalışmaları yapılmıştır.

Her model, eğitim veri seti üzerinde eğitilmiş ve test veri seti üzerinde değerlendirilmiştir. Performans metrikleri olarak doğruluk (accuracy), precision, recall, F1 skoru, spesifisite (specificity), FPR (False Positive Rate) ve FNR (False Negative

Rate) kullanılmıştır. Ayrıca, her model için ROC eğrileri çizilmiş ve karmaşıklık matrisleri (confusion matrices) görselleştirilmiştir.

3.2. Model Değerlendirme Yöntemleri ve Metrikleri

3.2.1. Karmaşıklık Matrisi

Karmaşıklık matrisi, bir modelin sınıflandırma performansını değerlendirmek için kullanılan önemli bir yöntemdir. Bu matris, modelin yaptığı sınıflandırma tahminlerini ayrıntılı olarak göstermeyi sağlar. İki sınıflı (saldırı ve normal trafik) bir veri seti için karmaşıklık matrisinde dört farklı durum ortaya çıkar: doğru pozitif (DP), doğru negatif (DN), yanlış pozitif (YP) ve yanlış negatif (YN). DP, modelin saldırı olan bir veriyi doğru şekilde saldırı olarak tahmin etmesidir. DN, modelin normal bir veriyi doğru şekilde normal olarak tahmin etmesi anlamına gelir. YP, modelin normal bir veriyi yanlışlıkla saldırı olarak tahmin etmesi durumudur. YN ise modelin saldırı olan bir veriyi normal olarak tahmin etmesi durumunu ifade eder. Bu dört farklı durum, modelin doğruluk ve güvenilirlik analizinde kritik bir rol oynar.

3.2.2. Kesinlik Metriği

Kesinlik (precision) metriği, sınıflandırma modellerinin performansını değerlendirmek için kullanılan önemli bir ölçüttür. Kesinlik, modelin saldırı olarak tahmin ettiği verilerin ne kadarının gerçekten saldırı olduğunu gösterir. Matematiksel olarak, kesinlik DP tahminlerin, doğru pozitif ve YP tahminlerin toplamına bölünmesiyle hesaplanır (Eşitlik 1):

$$\text{Kesinlik} = \frac{\text{DP}}{\text{DP} + \text{YP}} \qquad \text{(Eşitlik 1)}$$

Yüksek kesinlik, modelin saldırı olarak tahmin ettiği verilerin büyük çoğunluğunun gerçekten saldırı olduğunu gösterir, bu da modelin yanlış alarmları (false positives) minimize ettiğini ifade eder. Kesinlik, özellikle yanlış pozitiflerin maliyetinin yüksek olduğu durumlarda önemli bir değerlendirme kriteridir.

3.2.3. Hatırlama (Duyarlılık) Metriği

Hatırlama (recall) metriği, sınıflandırma modellerinin performansını değerlendirmek için kullanılan önemli bir ölçüttür. Hatırlama, modelin gerçekten saldırı

olan verileri ne kadar doğru bir şekilde tespit ettiğini gösterir. Matematiksel olarak, hatırlama DP tahminlerin, doğru pozitif ve YN tahminlerin toplamına bölünmesiyle hesaplanır (Eşitlik 2):

$$\text{Hatırlama} = \frac{\text{DP}}{\text{DP} + \text{YN}} \quad \text{(Eşitlik 2)}$$

Yüksek hatırlama, modelin tüm gerçek saldırıları doğru bir şekilde tespit etme yeteneğini yansıtır ve özellikle kaçırılmaması gereken kritik saldırıların önemli olduğu durumlarda hayati bir değerlendirme kriteridir. Hatırlama, yanlış negatiflerin maliyetinin yüksek olduğu senaryolarda büyük önem taşır.

3.2.4. F1 Skor Metriği

F1 skoru, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir diğer önemli ölçüttür. Kesinlik (precision) ve hatırlama (recall) metriklerinin harmonik ortalamasını temsil eder ve aşağıdaki formül ile hesaplanır (Eşitlik 3):

$$\text{F1 Score} = 2 \times \left(\frac{\text{Kesinlik} \times \text{Hatırlama}}{\text{Kesinlik} + \text{Hatırlama}}\right) \quad \text{(Eşitlik 3)}$$

F1 skoru, kesinlik ve hatırlama arasında bir denge sağlar ve her iki metriğin de önem taşıdığı durumlarda kullanışlıdır. Yüksek bir F1 skoru, modelin hem doğru pozitif oranını artırdığı hem de yanlış negatifleri minimize ettiği anlamına gelir. Bu metrik sayesinde, modelin genel performansını daha kapsamlı bir şekilde değerlendirmemize olanak sağlar.

3.3. Model Eğitimi İçin Özniteliklerin Belirlenmesi

Bu çalışmada, IoT ağlarında saldırı tespiti amacıyla kullanılan makine öğrenmesi ve derin öğrenme modellerinin performansını artırmak için en iyi 10 öznitelik belirlenerek modeller bu özellikler üzerinde uygulanmıştır. Öznitelik seçimi, modelin doğruluğunu ve genel performansını etkileyen kritik bir adımdır. Bu çalışmada kullanılan öznitelik seçimi algoritmaları arasında Karşılıklı Bilgi (Mutual Information), Recursive Feature Elimination (RFE), Öznitelik Önem Puanları (Feature Importance Scores) ve PCA (Principal Component Analysis) bulunmaktadır.

RandomForestClassifier kullanılarak özniteliklerin önem sıralaması belirlenerek en önemli 10 özellik tespit edilmiştir. Bu yöntemler, modelin eğitim süresini azaltmak, aşırı uyumu (overfitting) önlemek ve genel performansı artırmak amacıyla özniteliklerin dikkatli ve doğru bir şekilde seçilmesini sağlamıştır. Çalışmada belirlenen en iyi 10 öznitelik, saldırı tespiti ve kategori sınıflandırma görevlerinde yüksek doğruluk sağlamıştır. Bu öznitelikler, IoT ağlarında saldırı tespiti için kullanılan algoritmaların etkinliğini değerlendirmek ve geliştirmek için kullanılmıştır.

3.3.1. En iyi 10 Özellik Tespiti Hesaplaması

RandomForestClassifier, birden çok karar ağacının oluşturduğu bir topluluk modeli olup sınıflandırma görevlerinde yüksek doğruluk ve genelleme özellikleriyle bilinir. Modelin öznitelik önem sıralaması, veri setinde kullanılan en kritik 10 özelliği tespit etmiştir. En önemli 10 veri hesaplanırken tablodaki "kategori" ve "alt kategori" alanları hesaplama dışı bırakılmıştır. Atak verilerini tespit etmek için en önemli 10 öznitelik ve önem sırası veri seti UNSW-IoT-Botnet için Tablo 1'de gösterilmiştir.

Tablo 1. Atak trafik tespiti için en önemli 10 öznitelik UNSW

Öznitelik	Önem Değeri
dur	0.0433926089780801
pkts	0.0447894669823082
spkts	0.0455206784232945
rate	0.0527575278324633
dport	0.0608639691524380
bytes	0.0665877715858081
proto	0.0721748124878536
saddr	0.0748193779569155
sbytes	0.0820628799501911
state	0.1536316258821463

Tablo 2. Atak trafik tespiti için en önemli 10 öznitelik BoT-IoT

Öznitelik	Önem Değeri
seq	0.13948005172723
stddev	0.00761427615569
sbytes	0.26135751510084
min	0.04330152679775
dbytes	0.08475005735644
mean	0.067975389022026
dur	0.099635099313069
drate	0.067611531602543
srate	0.167182378134021
max	0.061272187309282

Atak tespiti için en önemli 10 öznitelik Bot-IoT veri seti için Tablo 2’ de gösterilmiştir.

3.4. Elastic Stack ve Özellikleri

Elastic Stack, büyük veri analiz ve arama gereksinimlerini karşılamak amacıyla geliştirilmiş açık kaynaklı bir yazılım çözümüdür. Elastic Stack, genellikle "ELK Stack" olarak bilinir ve dört ana bileşenden oluşur: Elasticsearch, Logstash, Kibana ve Beats. Bu bileşenlerin her biri, veri toplama, işleme, depolama ve görselleştirme süreçlerinde kritik rol oynar.

Elasticsearch, dağıtık bir arama ve analiz motorudur. Büyük veri kümeleri üzerinde hızlı arama yapabilme ve gerçek zamanlı analiz gerçekleştirebilme kapasitesi özelliğiyle ön plana çıkmıştır. JSON tabanlı bir RESTful ara yüz sunmakta olup çeşitli veri türlerini destekler. Elasticsearch, verilerin hızlı bir şekilde indekslenmesini, sorgulanmasını, analiz edilmesini ve raporlanmasını sağlar.

Logstash, veri işleme ardışık düzenini yönetmek için kullanılan bir veri toplama, işleme motorudur. Farklı kaynaklardan gelen verileri toplar, filtreler ve dönüştürerek Elasticsearch veya diğer veri depolarına iletir. Logstash, çok sayıda giriş (input), filtre (filter) ve çıkış (output) eklentisi ile veri işleme sürecinde esneklik sunar.

Kibana, verileri görselleştirmek ve analiz etmek için kullanılan bir ara yüzdür. Elasticsearch üzerinde depolanan verilerin grafikler, çizelgeler ve haritalar şeklinde görselleştirilmesini sağlar. Kibana, kullanıcıların veriler üzerinde keşif yapmalarına, özel panolar oluşturmalarına ve interaktif analizler gerçekleştirmelerine olanak tanır.

Beats, hafif veri göndericilerinden oluşan bir platformdur. Çeşitli veri kaynaklarından (sunucu logları, metrikler, ağ verileri vb.) verileri toplayarak Logstash veya Elasticsearch'e iletir. Beats ailesinde Filebeat, Metricbeat, Packetbeat ve Winlogbeat gibi farklı kullanım senaryolarına uygun çeşitli modüller bulunmaktadır.

Elastic Stack, büyük veri setlerini gerçek zamanlı olarak işleyip detaylı analiz edebilir, bu nedenle özellikle hızlı karar alınması gereken durumlar için kritik öneme sahiptir. Elasticsearch'ün dağıtık mimarisi, veri kümeleri büyüdükçe performansın düşmesini engeller ve sistemi yatay olarak ölçeklendirme olanağı sağlar. Ayrıca, Elastic Stack çok çeşitli veri kaynaklarından veri toplayıp işleyebilir, Logstash ve Beats gibi bileşenler veri işleme sürecini ihtiyaçlara göre özelleştirme esnekliği sunar. Kibana, kullanıcıların verileri görselleştirmelerine ve analiz etmelerine yardımcı olan geniş bir araç seti sunar, özelleştirilebilir panolar ve raporlar veri analiz süreçlerini daha verimli hale getirir.

Elastic Stack, siber güvenlik, performans izleme, log yönetimi ve iş analitiği gibi birçok alanda yaygın olarak kullanılmaktadır. Özellikle büyük veri analitiği ve gerçek zamanlı arama gereksinimlerine yanıt veren güçlü ve esnek yapısı sayesinde, veri mühendisliği ve veri bilimi projelerinde tercih edilen bir platform haline gelmiştir.

3.5. Makine Öğrenmesi ve Derin Öğrenme Yöntemleri

Bu çalışmada kullanılan makine öğrenmesi ve derin öğrenme yöntemleri, veri analizi ve sınıflandırma süreçlerinde önemli rol oynamaktadır. Makine öğrenmesi yöntemleri arasında Lojistik Regresyon, K-En Yakın Komşu, Destek Vektör Makineleri, Karar Ağaçları (Gini ve Entropy) ve Naive Bayes yer almaktadır.

3.5.1. Makine Öğrenmesi Yöntemleri

3.5.1.1. Lojistik Regresyon (Logistic Regression)

Logistic regression, makine öğrenmesi ve istatistikte kullanılan temel bir sınıflandırma yöntemidir. Bu yöntem, bağımlı değişkenin ikili (binary) bir sonuç aldığı durumlarda kullanılır. Yani, bağımlı değişken yalnızca iki kategoriye sahip olabilir: örneğin, evet/hayır, başarılı/başarısız gibi. Logistic regression, bağımsız değişkenler ile bağımlı değişken arasındaki ilişkiyi modellemek için sigmoid fonksiyonunu kullanır. Sigmoid fonksiyonu, bağımsız değişkenlerin doğrusal kombinasyonunu alır ve bu kombinasyonu 0 ile 1 arasında bir olasılık değeri olarak haritalar.

Lojistik regresyon, matematikte x ve y arasındaki denklem olarak lojistik fonksiyonu ya da logit fonksiyonu kullanan istatistiksel bir modeldir. Logit fonksiyonu, y'yi x'in sigmoid fonksiyonu olarak eşler. Logistic regression'ın matematiksel fonksiyon denklemi Eşitlik 4'te gösterildiği gibi hesaplanır.

$$\sigma(Z) = \frac{1}{1 + e^{-z}} \qquad \text{(Eşitlik 4)}$$

Çoğu durumda, birden fazla açıklayıcı olan değişken bağımlı olan değişkenin değerini etkiler. Lojistik regresyon formülleri, bu tür giriş veri kümelerini modellemek için bağımsız değişkenler arasında doğrusal bir ilişki olduğunu varsaymıştır. Bu model, bağımsız değişkenlerin doğrusal kombinasyonunu alır ve sigmoid fonksiyonu kullanarak bu kombinasyonu 0 ile 1 arasında bir olasılık değeri olarak ifade eder. Sigmoid fonksiyonunu değiştirebilir ve son çıktı değişkenini şu şekilde işleyebiliriz:

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots\cdots\cdots\cdots + \beta_n x_n$$

(Eşitlik 5)

Eşitlik 5'te gösterilen denklemde β sembolü, regresyon katsayısını temsil etmektedir. Logit modeli, hem bağımlı hem de bağımsız değişkenlerin bilinen değerlerine sahip bir deneysel veri kümesi verdiğinizde bu katsayı değerlerini tersine hesaplayabiliriz.

Lojistik regresyon, kolay anlaşılır ve uygulanabilir bir yöntemdir. Az sayıda parametreye sahip olduğundan, hızlıca uygulanabilir ve sonuçları yorumlanabilir. Modelin çıktısı, bir olasılık değeri olduğundan, karar verme süreçlerinde esneklik sağlar. Bu, bir gözlemin hangi kategoriye ait olma olasılığını ifade eder. Logistic regression, diğer karmaşık makine öğrenmesi algoritmalarına göre daha hızlı eğitim alır. Özellikle büyük veri setleriyle çalışırken avantajlıdır. Ayrıca, basit yapısı sayesinde aşırı uyum riskini minimize eder. Bu, modelin eğitim verilerine fazla uyum sağlama ve yeni verilerde kötü performans gösterme riskini azaltır. Bu, modelin farklı türdeki DDoS saldırılarını tespit etme kapasitesini artırır. Ayrıca, karmaşık makine öğrenmesi algoritmalarına kıyasla daha az hesaplama gücü gerektirir. Bu, özellikle büyük ölçekli ağ trafiği verilerinde avantaj sağlar.

3.5.1.2. K-En Yakın Komşu (K-Nearest Neighbors)

K-En Yakın Komşu (KNN), denetimli öğrenme yöntemlerinden biridir ve sınıflandırma ile regresyon problemlerinde yaygın olarak kullanılır. Bu algoritma, sınıflandırma işlemini, bir veri noktasının en yakın k komşusunun sınıfına bakarak gerçekleştirir. Bir veri noktası, kendisine en yakın k adet komşusunun hangi sınıfa ait olduğuna göre sınıflandırılır. KNN, veri noktalarının sınıflandırılmasında mesafe metriği (genellikle Öklidyen mesafe) kullanır ve bu metrik aracılığıyla en yakın komşuları belirler. Öklidyen mesafe, iki veri noktası arasındaki doğrusal mesafeyi ölçen matematiksel bir hesaplamadır ve Eşitlik 6'da gösterildiği şekilde ifade edilir:

$$\mathrm{d}(\mathrm{x}, \mathrm{x_i}) = \sqrt{\sum_{j=1}^{n} \left(x_i - x_{ij}\right)^2}$$

(Eşitlik 6)

Burada $d(x,x_i)$ iki veri noktası arasındaki öklidyen mesafeyi temsil etmektedir.

KNN, basit ve anlaşılır bir algoritma olması sebebiyle ön plana çıkmıştır. Karmaşık model oluşturma süreci gerektirmez ve doğrudan ham veriler üzerinde çalışır. Ayrıca, parametre ayarlama gerektirmez; yalnızca k değeri ve mesafe metriği seçilir. Bu özellikleri sayesinde hem sınıflandırma hem de regresyon problemlerinde kullanılabilir ve farklı türdeki veri setlerine kolayca uyarlanabilir bir algoritmadır. Küçük ve orta ölçekli veri setlerinde etkili performans göstermektedir. Özellikle iyi yapılandırılmış ve temizlenmiş veri setlerinde başarılı sonuçlar vermektedir.

KNN, yeni veri noktaları eklendiğinde modelin yeniden eğitilmesine gerek kalmadan çalışabilecek esnek bir algoritmadır. Bu özelliği sayesinde, DDoS saldırılarının gerçek zamanlı tespitinde önemli bir fayda sağlamaktadır. Ayrıca, KNN'nin basitliği, algoritmanın kolayca uygulanmasını ve sonuçların hızla elde edilmesini sağlar. DDoS ataklarını tespit etmede zamanın ne kadar kritik olduğu düşünüldüğünde bu büyük bir avantaj sağlamaktadır. Ayrıca KNN, farklı türdeki saldırıları tespit etmek içinde uyarlanabilir ve diğer saldırı tespitleri içinde esneklik sağlamaktadır.

3.5.1.3. Destek Vektör Makineleri (Support Vector Machines)

Destek Vektör Makineleri (SVM), makine öğrenmesi alanında hem sınıflandırma hem de regresyon problemlerinde kullanılan güçlü bir yöntemdir. SVM, verileri yüksek boyutlu bir uzayda ayırmak için en iyi hiper düzlemi (hyperplane) bulmayı amaçlar. Bu yöntem, iki sınıfı en geniş marjinal boşluğu sağlayacak şekilde ayıran bir hiper düzlem oluşturur. SVM, bu ayrımı gerçekleştirirken, yalnızca destek vektörleri olarak adlandırılan ve sınıfları ayırmada kritik olan veri noktalarını kullanır.

$$f(x) = sgn(w.x + b)$$

(Eşitlik 7)

Eşitlik 7'de gösterilen svm karar fonksiyonu hesaplamasında *w* ağırlık vektörü ve b bias terimidir.

SVM, özellikle yüksek boyutlu veri setlerinde etkili performans gösterir. Veri setinde birçok özelliğin bulunduğu durumlarda, SVM'nin doğruluk oranı da artmaktadır. Ayrıca SVM, iyi genelleme yeteneği sayesinde, eğitim verisinden öğrenilen bilgiyi yeni ve görülmemiş verilere başarıyla uygulayabilir. SVM, doğrusal olmayan verileri ayırmak için farklı çekirdek fonksiyonları (kernel functions) kullanabilme özelliğine sahiptir. Bu sayede, veri yapısına uygun olarak polinomial, RBF (Radial Basis Function) gibi çekirdek fonksiyonları seçilebilir. Ayrıca yüksek boyutlu özellik

uzaylarında bile aşırı uyum riskini minimize edecek şekilde çalışmaktadır. Bu, modelin genel performansını artırarak aldığımız sonuçların doğruluk oranlarını artırmaktadır. DDoS saldırıları genellikle büyük miktarda veriyi içerdiğinden, SVM'nin yüksek doğruluğu, saldırıları etkili bir şekilde tespit etmeye yardımcı olur.

3.5.1.4. Karar Ağaçları (Decision Trees)

Karar ağaçları, veriyi dallara ayırarak sınıflandırma yapan bir modeldir. Her dalda veri seti bölünerek, karar ağaçları oluşturulur. Makine öğrenmesi alanında hem sınıflandırma hem de regresyon problemlerinde kullanılan yaygın ve güçlü bir yöntemdir. Karar ağaçları, veriyi belirli kurallara göre dallara ayırarak sınıflandırma veya tahmin işlemini gerçekleştirir. Bu ağaç yapısı, kök düğümden başlayarak her bir iç düğümde veri setini bir özelliğe göre böler ve yaprak düğümlerde sınıflandırma veya regresyon sonuçlarına ulaşır. Karar ağaçları, basit ve anlaşılır bir yapıya sahip olmaları nedeniyle birçok uygulamada tercih edilir. Karar ağaçları, bir düğümü iki veya daha fazla alt düğüme bölme kararını vermek için birden fazla algoritma kullanmaktadır. Alt düğümlerin oluşturulması, oluşturulan alt düğümlerin homojenliğini arttırır. Başka bir ifadeyle, düğümün saflığının hedef değişkenlere göre artış gösterdiğini ifade edebiliriz. Karar ağacımızın entropi değerini minimize etmek isteyen bölünmeler yapmasını isteriz. En iyi bölünmeyi belirlemek içinde bilgi kazancını kullanabiliriz. Bilgi kazancı aşağıdaki eşitlik ile hesaplanır (Eşitlik 8):

$$Gain(S, D) = H(S) - \sum_{V \in D} \frac{|V|}{|S|} H(V)$$

(Eşitlik 8)

Burada, *S* orijinal veri kümesini ifade etmektedir. *D* ise kümenin bölünmüş bir parçasıdır. Her *V*, S'nin bir alt kümesidir olup *V*'nin tümü ayrıktır ve *S*'yi oluşturmaktadır. Bu durumda bilgi kazancı, bölünmeden önceki orijinal veri setinin entropisi ile her bir özniteliğin entropi değeri arasındaki fark olarak tanımlayabiliriz.

Gini endeksi ise veri kümesindeki bölünmeleri değerlendirmek için kullanılan bir maliyet fonksiyonu olarak tanımlayabiliriz. Gini endeksini aşağıdaki eşitlik yardımıyla hesaplayabiliriz (URL-1).

$$\text{Gini(D)} = 1 - \sum_{i=1}^{c} \text{p}_\text{i}{}^2$$

(Eşitlik 9)

Eşitlik 9'da gösterilen hesaplamada p_i, i sınıfının olasılığını temsil etmektedir. Her bir sınıfın olasılıklarının karelerinin toplamının birden çıkarılmasıyla hesaplayabiliriz. Gini daha büyük bölümleri tercih eder ve uygulaması kolaydır.

Karar ağaçları, ağ trafiği verilerinde anormallikleri belirlemek için etkili bir yöntemdir. Her bir düğümde yapılan bölünmeler, normal ve anormal trafik desenlerini ayırt etmede yardımcı olmaktadır.

3.5.1.5. Naive Bayes

Naive Bayes, makine öğrenmesi ve istatistikte kullanılan bir sınıflandırma algoritmasıdır. Bu algoritma, Bayes teoremi temel alınarak geliştirilmiştir ve "naive" (saf) olarak adlandırılmasının nedeni, her özelliğin birbirinden bağımsız olduğunu varsaymasıdır. Naive Bayes, özellikle metin sınıflandırma, spam tespiti ve duygu analizi gibi alanlarda yaygın olarak kullanılmaktadır. Algoritma, eğitim verilerini kullanarak sınıfların koşullu olasılıklarını hesaplar ve yeni bir veri noktası geldiğinde, hangi sınıfa ait olma olasılığının en yüksek olduğuna karar vermektedir. Naive Bayes'in temel matematiksel denklemi şu şekildedir (Eşitlik 10):

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

(Eşitlik 10)

Burada, *P(A|B) B* özellikler kümesi verildiğinde A sınıfının olasılığıdır. *P(B|A), A* sınıfı verildiğinde *B* özellikler kümesinin olasılığıdır. *P(A), A* sınıfının öncelikli olasılığıdır. *P(B), B* ise özellikler kümesinin genel olasılığıdır.

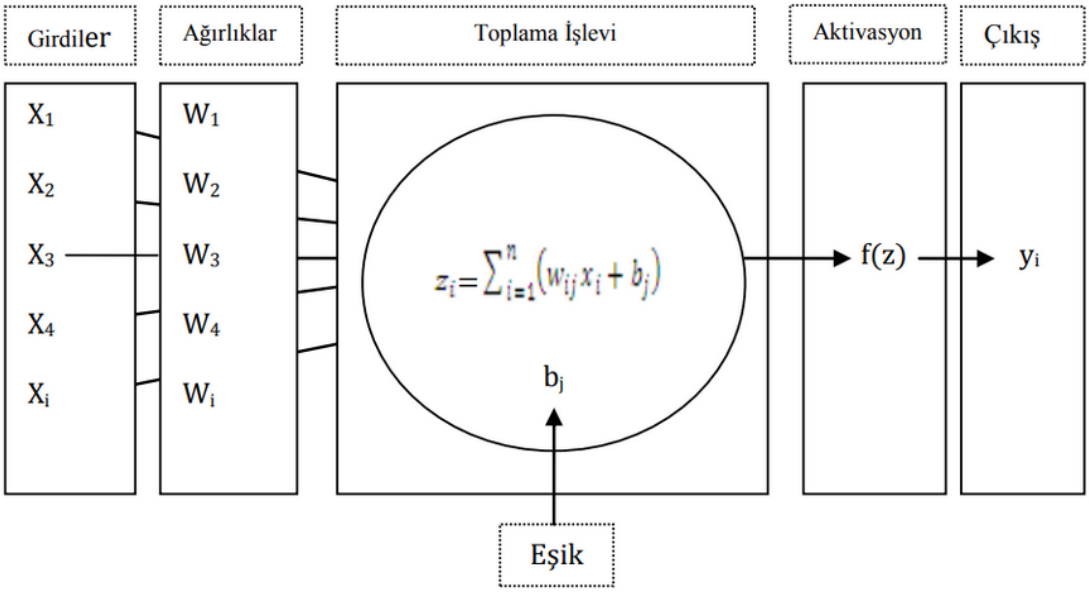
Naive Bayes algoritması, hesaplama açısından çok verimlidir ve büyük veri setlerinde bile hızlı sonuçlar verir. Diğer makine öğrenmesi algoritmalarına kıyasla daha az eğitim verisine ihtiyaç duymaktadır. Bu, DDoS saldırılarının erken evrelerinde bile başarılı tespitler yapmayı sağlar. Ayrıca yüksek boyutlu veri setleriyle etkili bir şekilde çalışabilir ve yeni veriler geldikçe kolayca güncellenebilir olması kullanılması için önemli özelliklerin başında gelmektedir. Naive Bayes modelinin, yeni veriler geldikçe hızlı bir şekilde güncellenmesi mümkündür, bu da DDoS saldırılarının sürekli değişen doğasına karşı etkili bir savunma sağlamaktadır.

3.5.2. Derin Öğrenme Yöntemleri

3.5.2.1. Yapay Sinir Ağları (Neural Networks)

Yapay Sinir Ağları (NN), insan beyninin çalışma prensiplerinden esinlenerek geliştirilmiş çok katmanlı yapıya sahip bir derin öğrenme yöntemidir. Bu algoritmalar, veri setlerinde karmaşık desenleri ve ilişkileri öğrenmek için kullanılır. Sinir ağları, düğümler (nöronlar) ve bu düğümler arasındaki bağlantılardan (ağırlıklar) oluşur. Bu ağlar, girdileri alır, bir dizi hesaplama yapar ve çıktılar üretir. Çok katmanlı yapıya sahip olan sinir ağları, özellikle derin öğrenme (deep learning) alanında önemli bir yer tutar.

Sinir ağlarının matematiksel temeli, her bir nöronun aldığı girişlerin ağırlıklı toplamını hesaplaması ve bu toplamı bir aktivasyon fonksiyonu (örneğin, sigmoid, ReLU) aracılığıyla işleyerek bir çıktı üretmesidir. Temel formül Şekil 3'te gösterilmiştir:



Şekil 3. Yapay sinir hücresi (URL-2).

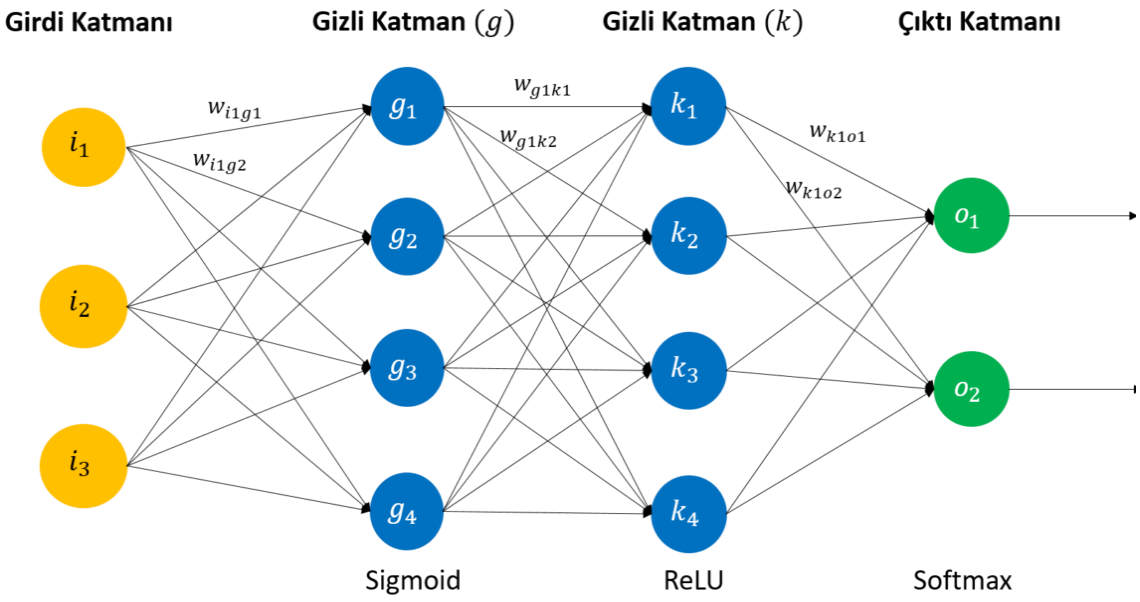
Girdiler, diğer hücrelerden veya dış dünyadan gelen bilgileri temsil etmektedir. Toplama fonksiyonu, hücreye giren net girdiyi hesaplamamızı sağlayan fonksiyondur. Uygulanacak NN modeline göre çeşitli fonksiyonlar kullanılabilir. Genellikle toplama fonksiyonunu hesaplamak için hücreye gelen bilgiler ile o bilgilerin ağırlıkları çarpılarak toplanması işlemi yapılmaktadır.

Aktivasyon fonksiyonu, girdi ile çıktı arasında bağlantıyı kurmaya yarar. Toplama fonksiyonundan gelen bilgileri işleyerek çıktı bilgilerini oluşturabiliriz. Bu fonksiyon da

uygulanacak YSA modeline göre çeşitli fonksiyonlara sahiptir. Çıktı y_i, aktivasyon fonksiyonu sonucunda çıkan değerlerdir.

Yapay sinir ağları tek katmanlı ve çok katmanlı olabilmektedir. Tek katmanlı yapay sinir ağları, genellikle basit yapıda olan ve yalnızca bir giriş katmanı ve bir çıkış katmanından oluşan ağlardır. Bu tür ağlar, en basit formda perceptron olarak bilinir ve genellikle doğrusal sınıflandırma problemlerinde kullanılır. Bu basit yapısı sayesinde doğrusal sınıflandırma problemlerinde etkilidir, ancak karmaşık ve doğrusal olmayan ilişkileri öğrenme kapasitesi sınırlıdır. Basitliği ve hızlı eğitimi sayesinde, hızlı çözümler gerektiren durumlar için idealdir.

Çok katmanlı yapay sinir ağları, giriş ve çıkış katmanlarının yanı sıra bir veya daha fazla gizli katman içerir. Bu gizli katmanlar, verilerin daha yüksek seviyeli özelliklerini öğrenir ve işler. Çok katmanlı yapılar, karmaşık ve doğrusal olmayan desenleri öğrenme kapasitesine sahip olup büyük ve çeşitli veri setleri ile eğitildiğinde yüksek doğruluk ve performans elde etmemizi sağlar. Bu ağlar, görüntü tanıma, doğal dil işleme ve ses tanıma gibi alanlarda yaygın olarak kullanılır. Yapısındaki bu karmaşıklıktan dolayı daha fazla hesaplama gücü ve daha uzun eğitim süreleri gerektirirler. Ayrıca, aşırı uyum riskini minimize etmek için dikkatli bir şekilde düzenlenmelidirler. Çok katmanlı yapay sinir ağları örneği Şekil 4'te gösterilmiştir.



Şekil 4. Çok katmanlı yapay sinir ağları (URL-3).

DDoS saldırılarının tespiti gibi karmaşık problemlerde, çok katmanlı sinir ağları daha etkili olmaktadır. Karmaşık ve doğrusal olmayan trafik desenlerini öğrenme kapasiteleri sayesinde, çeşitli DDoS saldırılarını başarılı bir şekilde tespit edebilirler.

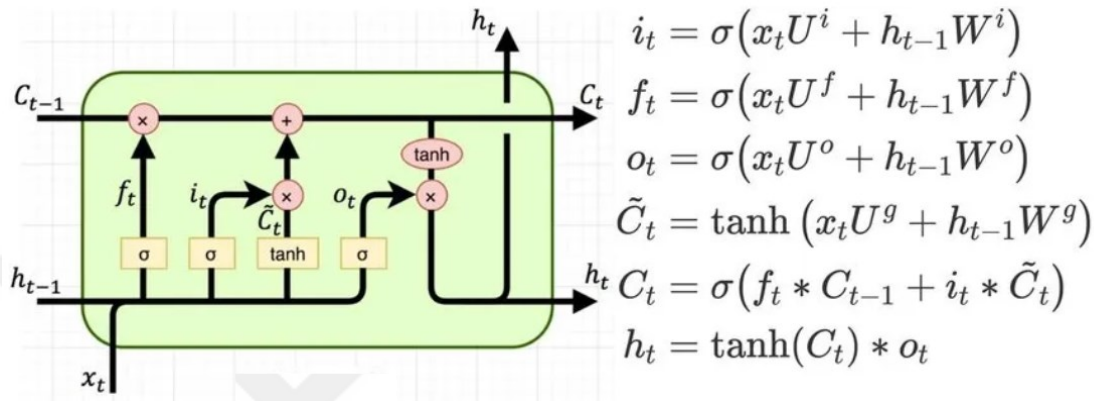
Büyük veri setleri üzerinde eğitildiklerinde, bu ağlar saldırıları yüksek doğrulukla tanımlayabilir ve gerçek zamanlı tespit ve müdahale süreçlerinde kullanılabilirler. Tek katmanlı ağlar ise daha basit saldırı tespit senaryolarında kullanılabilir ancak karmaşık saldırı desenlerindeki sadelik nedeniyle çok katmanlı ağlara göre başarım oranları düşük olmaktadır.

3.5.2.2. LSTM (Long Short-Term Memory)

LSTM (Long Short-Term Memory), tekrarlayan sinir ağlarının (Recurrent Neural Networks - RNN) bir türüdür ve özellikle zaman serisi verilerini ve sıralı bilgileri modellemek için geliştirilmiştir. LSTM, RNN'lerin uzun vadeli bağımlılıkları öğrenme konusundaki sınırlamalarını aşmak için tasarlanmıştır. Geleneksel RNN'ler, uzun dizilerdeki bilgileri taşımakta zorlanırken, LSTM hücreleri, unutma, giriş ve çıkış kapıları kullanarak bilgi akışını kontrol eder ve uzun vadeli bağımlılıkları etkili bir şekilde öğrenir.

LSTM iletişim halinde olan 4 farklı katman içermektedir. Unutma kapısı (forget gate), giriş kapısı (Input gate), hücre durumu (cell state), çıkış kapısı (output gate) olarak bu katmanlar isimlendirilmiştir. Unutma kapısı, hangi bilgilerin tutulacağı hangi bilgilerin unutulacağı konusunda karar sahibi olan kapıdır. Bu bilgilerin kararını vermek için 0 ve 1 değerlerini kullanır. 0 olan bilgi unutulur, 1 olan bilgi ise tutulup cell state ile taşınmaya devam ederler. Hücrelerle ilgili bu kararı verirken bir önceki hücreden gelen veriyle mevcut veri sigmoid fonksiyonuna tabi tutulur ve sonuca göre karar verilir. Giriş kapısı ise güncelleme görevi yapmakla sorumludur. Önceki ve mevcut bilginin sigmoid fonksiyonuna tabi tutulması işleminin sonucuna göre güncelleme kararını verir. İşlem sonucunda 0 olan bilgi önemsiz ve 1 olan bilgi önemli olarak kabul görmektedir. Ayrıca tanh aktivasyon fonksiyonu sayesinde veri -1 ve 1 aralığına sıkıştırılarak ağı düzenleme (regulate) işlemi için kullanılır. Ağ düzenleme işlemleri yapıldıktan sonra sigmoid ve tanh fonksiyonu çıkışları çarpılır ve hangi bilginin güncelleneceği kararına varılır. Çıkış kapısı (output gate), hücre durumundaki bilginin hangi kısmının çıktı olarak verileceğini belirleyen kritik bir bileşendir. Çıkış kapısının temel işlevi, hücre durumundan (cell state) hangi bilginin çıktıya dönüştürüleceğine karar vermektir. Bu kararı vermek için kapının aktivatör fonksiyonu, hücre durumunun aktivasyonu ve çıkış hesaplamalarını yapar. Çıkış kapısı, önceki gizli durum (hidden state) ve mevcut giriş (input) kullanılarak sigmoid aktivasyon fonksiyonu üzerinden geçer. Bu adım, hangi bilginin çıktı olarak kullanılacağına karar veren ağırlıklar ve bias terimleriyle hesaplanır. Sigmoid fonksiyonu (σ), 0 ile 1 arasında bir değer döndürür ve bu, hangi bilginin ne

kadarının çıktıya ekleneceğini belirler. Güncellenmiş hücre durumu (cell state), tanh aktivasyon fonksiyonuna tabi tutulur. Tanh fonksiyonu, hücre durumundaki bilgiyi -1 ile 1 arasında bir değere dönüştürür. Çıkış kapısından gelen değer (output gate value), hücre durumunun tanh aktivasyonundan geçen değerle çarpılarak nihai çıkış (hidden state) elde edilir. Aşağıdaki şekilde örnek bir LSTM hücresi ve kullanılan fonksiyonlar gösterilmiştir.

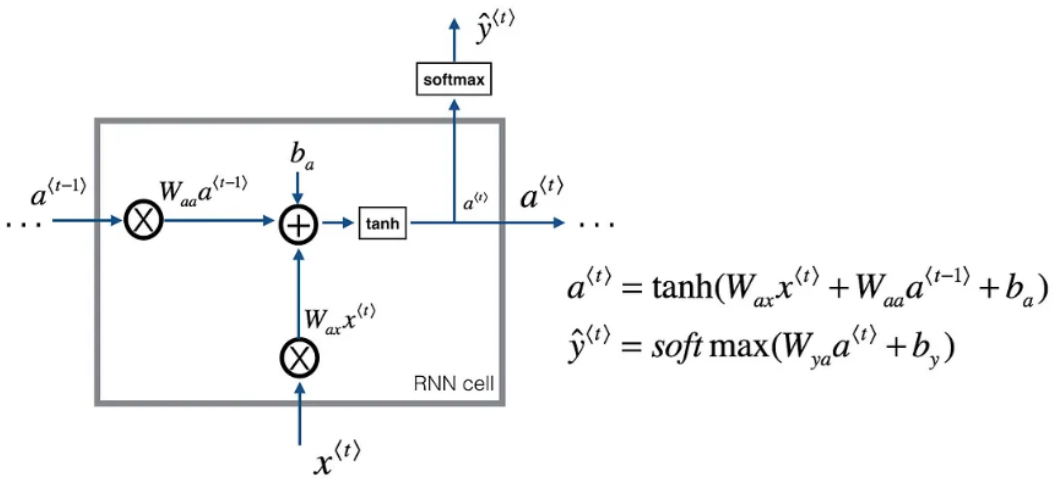


Şekil 5. Lstm hücresi (URL-4).

Şekil 5'te f_t unutma kapısı, i_t giriş kapısı, $\hat{C}$ hücre durumu ve o_t çıkış kapısını ifade etmektedir.

3.5.2.3. RNN (Recurrent Neural Network)

RNN (Recurrent Neural Network), sıralı verilerle çalışmak için tasarlanmış bir yapay sinir ağı türüdür. Geleneksel yapay sinir ağları (ANN) girdileri bağımsız olarak işlerken, RNN'ler girdiler arasındaki zaman veya sıralı bağımlılıkları yakalamak için geliştirilmiştir. Bu ağlar, önceki zaman adımlarından gelen bilgiyi hafızasında tutar ve bu bilgiyi gelecekteki zaman adımlarında kullanarak karar verir. RNN'lerin temel özelliği, her gizli katmanın (hidden layer) bir önceki gizli katmanın durumunu da giriş olarak almasıdır. Bu özellik, RNN'leri dil modelleme, konuşma tanıma ve zaman serisi tahmini gibi uygulamalarda oldukça etkili kılar. Diğer derin öğrenme yapılarından RNN'i ayıran özellikleri hatırlamalarıdır. RNN'ler kurmuş oldukları ilişkilerin kalıcı olmasını sağlamak amacıyla kendi içlerinde dönen döngüye benzeyen bir yapı kullanırlar. Şekil 6'da RNN hücresine ait örnek gösterilmiştir.



Şekil 6. RNN hücresi

RNN'lerin çalışma prensibi, her zaman adımında girdileri işleyerek önceki gizli durum (hidden state) bilgilerini kullanmaktır. Bu süreç girdi alımı, gizli durumun güncellenmesi ve çıktı hesaplaması şeklinde işler. Girdi alımı kısmında her zaman adımında bir girdi alır. Bu girdi x^t ile temsil edilir, burada t zaman adımını gösterir. Gizli durumun güncellenmesi aşamasında mevcut zaman adımındaki gizli durum a^t, önceki zaman adımındaki gizli durum a^{t-1} ve mevcut zaman adımındaki girdi a^t kullanılarak güncellenir. Çıktı aşamasına gelindiği zaman güncellenmiş gizli durum a^t, mevcut zaman adımının çıktısını hesaplamak için kullanılır. Burada, W_y çıkış için ağırlık matrisini ve b_y bias terimini temsil etmektedir.

RNN'lerin hafızaları kısıtlı olması sebebiyle çok uzun verilerde bu kapasite yetersiz kalabilir. Bu durumun önüne geçmek için LSTM ve GRU (Gated Recurrent Units) gibi gelişmiş RNN modelleri geliştirilmiştir. Bu modellerin hafızaları ve çalışma yöntemleri daha büyük verileri tutmaya olanak sağladığı için özellikle büyük veri setlerinde verim oranı artmaktadır.

4. YAPILAN ÇALIŞMALAR

Tez çalışmamızın ana amacı, Distributed Denial of Service (DDoS) saldırılarının tespitinde yenilikçi ve etkili yöntemler geliştirme üzerine kurulmuştur. Bu amaç doğrultusunda, makine öğrenmesi ve derin öğrenme algoritmaları kullanarak ağ trafiği verilerindeki anormallikleri tespit etmeye odaklandık. Kullandığımız veri setlerindeki özniteliklerin belirlenmesi için kullanılan veri sayıları şunlardır:

UNSW-IoT-Botnet veri setindeki kategorilerin sayısal bilgileri Şekil 7'de gösterilmiştir.

category	
DDoS	1926624
DoS	1650260
Reconnaissance	91082
Normal	477
Theft	79

Şekil 7. UNSW-IoT-Botnet kategorilerine ait sayısal değerler.

UNSW-IoT-Botnet veri setindeki alt kategorilere ait sayısal bilgiler Şekil 8'de gösterilmiştir.

subcategory	
UDP	1981230
TCP	1593180
Service_Scan	73168
OS_Fingerprint	17914
HTTP	2474
Normal	477
Keylogging	73
Data_Exfiltration	6

Şekil 8. UNSW-IoT-Botnet alt kategorilerine ait sayısal değerler.

BoT-IoT veri setine ait kategori verilerine ait sayısal bilgiler Şekil 9'da gösterilmiştir.

category	
DoS	4170269
Reconnaissance	2819665
Normal	9283
Theft	1766

Şekil 9. BoT-IoT veri seti kategorilerine ait sayısal değerler.

BoT-IoT veri setine ait alt kategori verilerine ait sayısal bilgiler Şekil 10'da gösterilmiştir.

subcategory	
TCP	4140055
Service_Scan	2461371
OS_Fingerprint	358294
HTTP	30210
Normal	9283
Keylogging	1530
Data_Exfiltration	236
UDP	4

Şekil 10. BoT-IoT veri seti alt kategorilerine ait sayısal değerler.

UNSW-IoT-Botnet datasetindeki tüm öznitelikler Şekil 11'de gösterilmiştir.

```
Tüm Öznitelikler:
['pkSeqID', 'proto', 'saddr', 'sport', 'daddr', 'dport', 'seq', 'stddev',
'N_IN_Conn_P_SrcIP', 'min', 'state_number', 'mean', 'N_IN_Conn_P_DstIP', 'drate', 'srate',
'max', 'attack', 'category', 'subcategory']
```

Şekil 11. UNSW-IoT-Botnet veri setindeki öznitelikler

BoT-IoT veri setinde ait tüm öznitelikler Şekil 12'de gösterilmiştir.

```
Tüm Öznitelikler:
['pkSeqID', 'stime', 'flgs', 'proto', 'saddr', 'sport', 'daddr', 'dport',
'pkts', 'bytes', 'state', 'ltime', 'seq', 'dur', 'mean', 'stddev', 'smac',
'dmac', 'sum', 'min', 'max', 'soui', 'doui', 'sco', 'dco', 'spkts', 'dpkts',
'sbytes', 'dbytes', 'rate', 'srate', 'drate', 'attack', 'category',
'subcategory ']
```

Şekil 12. BoT-IoT veri setindeki öznitelikler

4.1. En Önemli 10 Özniteliğin Belirlenmesi

Veri setlerindeki en önemli 10 özniteliğin belirlenmesi için RandomForestClassifier algoritması kullanılarak modelimiz bu 10 önemli öznitelikler kullanılarak eğitilmiştir. Veri setlerindeki alanların fazla olması ve bazı alanların boş gelmesinden dolayı bu alanlardaki verilere normalizasyon işlemleri uygulandıktan sonra en önemli 10 öznitelik belirleme işlemi yapılmıştır. En önemli 10 özelliğe ait veriler ve önem değerleri Şekil 13'te gösterilmiştir.

```
Top 10 most important features (excluding 'subcategory' and 'category'):
dur: Importance: 0.04339260978008148
pkts: Importance: 0.044789466982308235
spkts: Importance: 0.045520678423294506
rate: Importance: 0.052757527382436366
dport: Importance: 0.06086396915243801
bytes: Importance: 0.06658777158580818
proto: Importance: 0.0721748124878536
saddr: Importance: 0.07481937975691551
sbytes: Importance: 0.08260287995019118
state: Importance: 0.15363316258821463
```

Şekil 13. En önemli 10 öznitelik

En önemli 10 öznitelik belirlerken RandomForestClassifer kullanılmasının amacı bu algoritmanın özellikle aşırı uyum riskini azaltması ve bu, modelin eğitim verisine çok iyi uyup test verisi üzerinde kötü performans göstermesi riskini düşürmesidir.

4.2. Veri Temizleme ve Veri Dengesi Sağlama

Makine öğrenme ve derin öğrenme tekniklerinin performansı veri setindeki verilerin optimizasyonu yapılarak artırılmıştır. Veri setlerindeki eksik veri olan yerlerdeki verilerin yerlerine sütunlardaki verilerin ortalamaları alınarak doldurulmuştur. Veri seti üzerindeki verilerin düzeltilmesi işlemleri uygulanmıştır. Özellikle dport, sport alanlarında bulunan hexadecimal değerler veri bütünlüğü açışından integer değerlere çevrilerek ön temizleme işlemleri gerçekleştirilmiştir. Bu çalışmada, dengesiz veri setinin dengelenmesi amacıyla over-sampling ve under-sampling teknikleri uygulanmıştır. Veri setindeki sınıflar arasındaki dengesizlik, makine öğrenmesi modellerinin performansını olumsuz etkileyebilir. Bu sorunu gidermek için, iki aşamalı bir süreç uygulanmıştır. İlk aşamada, RandomOverSampler kullanılarak azınlık sınıflarının örnek sayısı artırılmıştır. Bu, az sayıda örneğe sahip sınıfların model tarafından daha iyi öğrenilmesini sağlamıştır. İkinci aşamada, RandomUnderSampler kullanılarak tüm sınıfların örnek sayısı eşitlenmiştir. Bu, veri setindeki dengesizliğin tamamen giderilmesini ve her sınıfın eşit temsil edilmesini sağlamıştır. Bu işlemler, makine öğrenmesi modellerinin performansını artırmak ve daha dengeli bir sınıflandırma sağlamak için kritik öneme sahiptir.

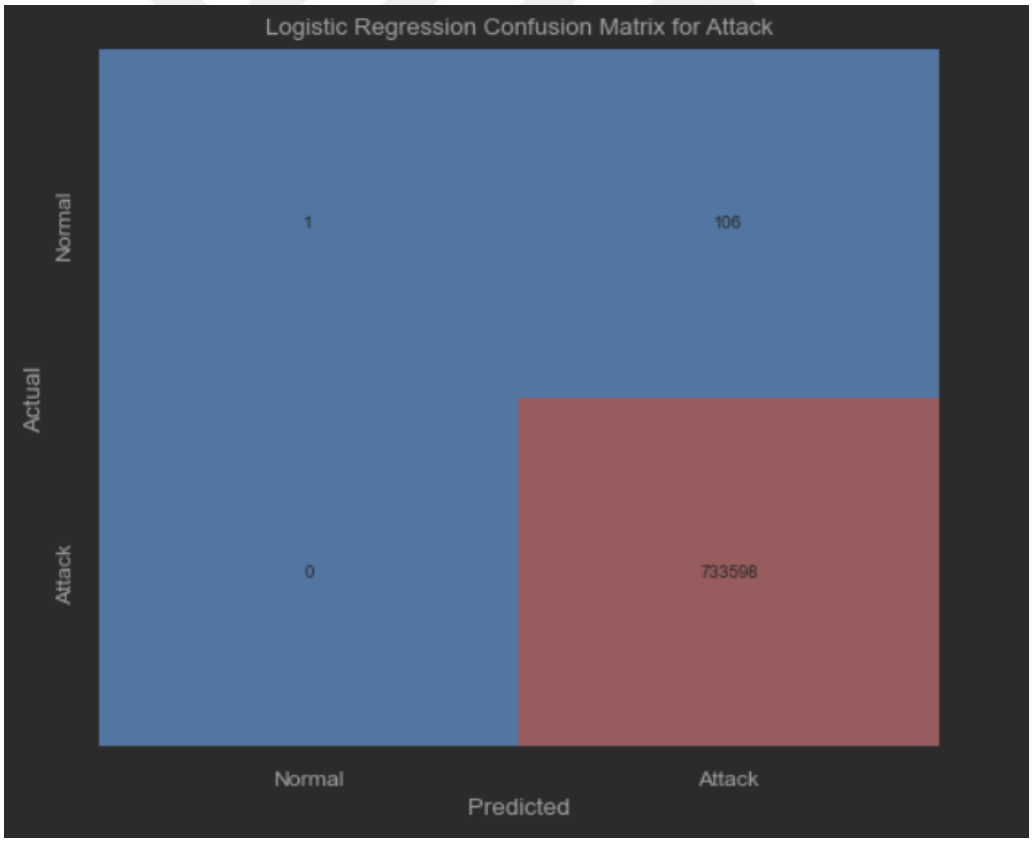
Veri setlerindeki düzenlemelerden sonra veriler analiz için uygun formata One-hot-encoding kullanılarak kategorik veriler sayısal verilere dönüştürülmüştür. Veri seti

gerekli makine ve derin öğrenme modelleri oluşturulduktan sonra eğitim ve test aşamalarına geçilmiştir.

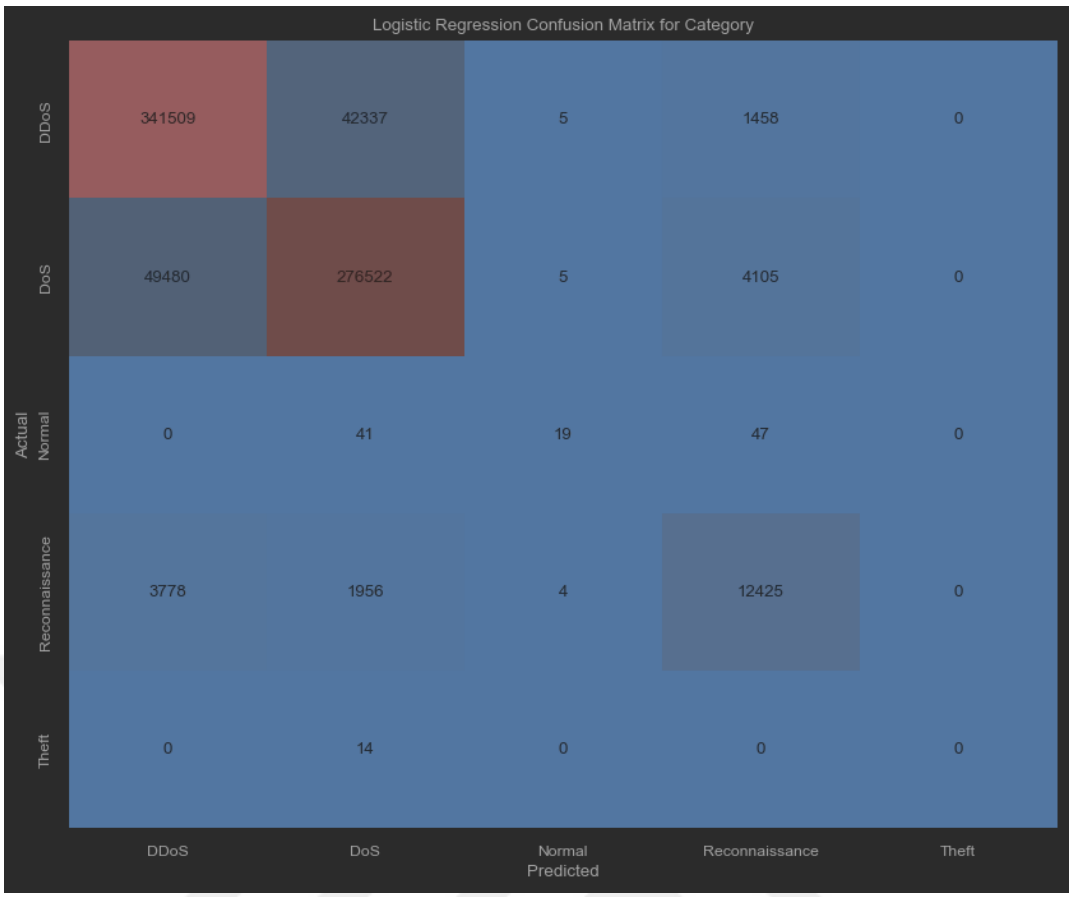
4.3. Model Geliştirme ve Eğitim

4.3.1. Lojistik Regresyon

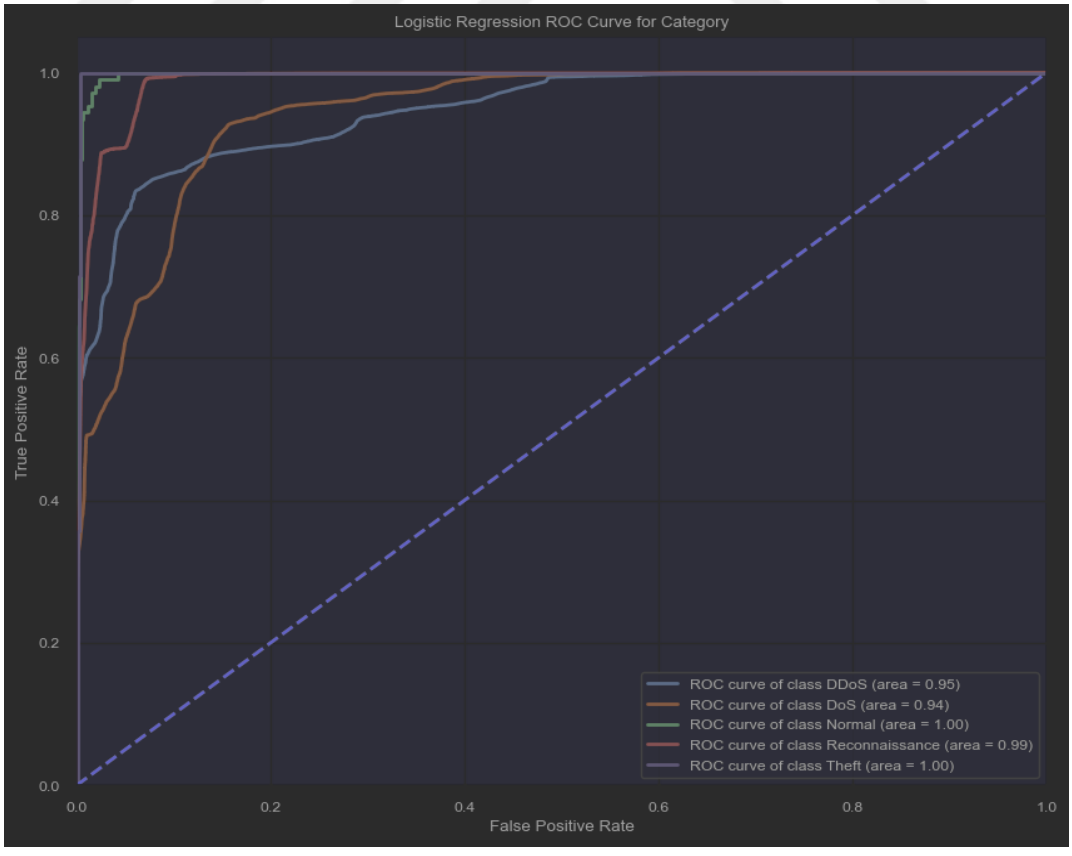
Veri setleri üzerinde gerekli işlemler uygulandıktan sonra uygulama için veri setinin %80'i, eğitim için %20'si ayrılarak gerekli eğitimler uygulanmıştır. Model Lojistik regresyon algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 14'te atak sınıfına ait confusion matrix, Şekil 15'te kategori sınıfına ait confusion matrix, Şekil 16'da kategori sınıfına ait roc curve ve Şekil 17'de alt kategorilere ait confusion matrix gösterilmiştir.



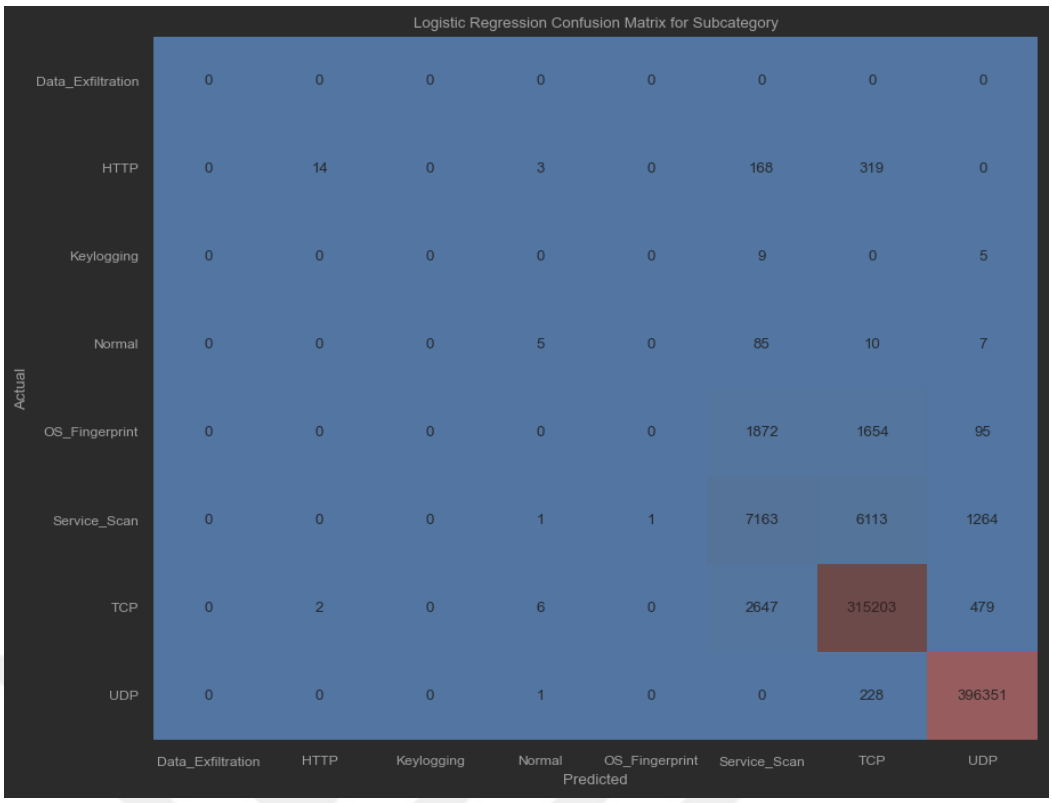
Şekil 14. UNSW Logistic Regression atak confusion matrix



Şekil 15. UNSW Logistic Regression kategori confusion matrix



Şekil 16. UNSW Logistic Regression kategori Roc curve



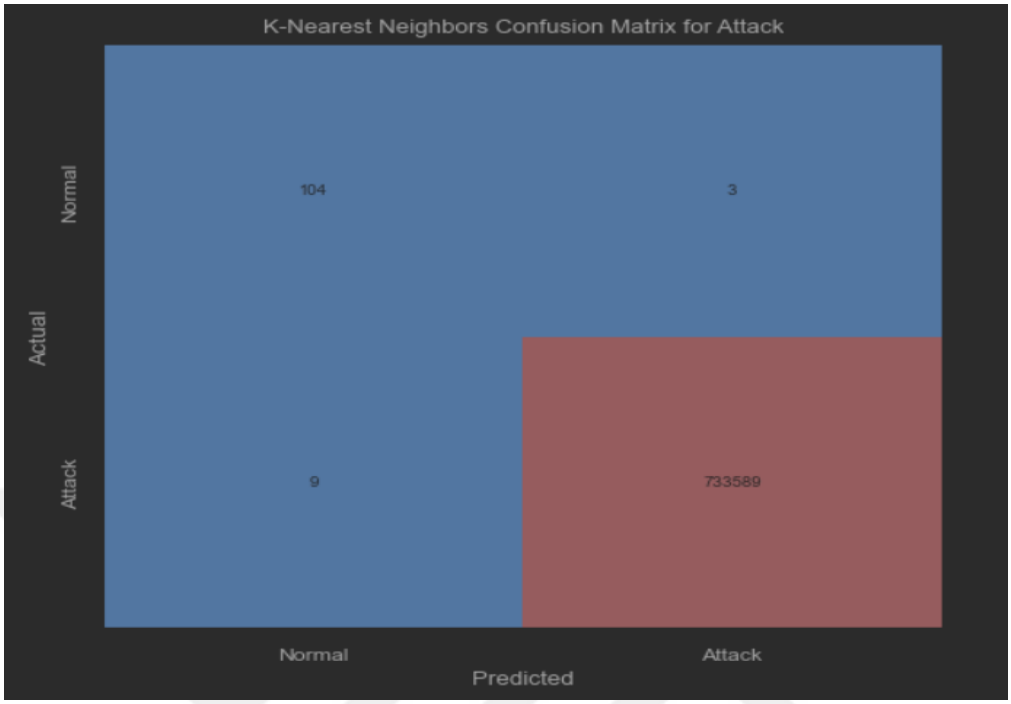
Şekil 17. UNSW Logistic Regression alt kategori confusion matrix

Lojistik regresyon modelimiz DDoS Saldırıları kategorisi için 341509 doğru sınıflandırma ve 42337 yanlış sınıflandırma ile DDoS saldırıları, yüksek doğrulukla tespit edilmiştir. Ancak, DDoS saldırılarının DoS saldırılarıyla karıştırılma oranı da oldukça yüksektir. DoS Saldırıları: 276522 doğru sınıflandırma ve 49480 yanlış sınıflandırma ile DoS saldırıları da genel olarak iyi bir şekilde tespit edilmiştir. DoS saldırıları da DDoS ile karıştırılmıştır. Normal trafik kategorisi için 19 doğru sınıflandırma, ancak diğer kategorilerle karıştırılma oranı yüksektir. Normal trafiğin DDoS ve DoS ile karıştırılması dikkat çekicidir. Reconnaissance (Keşif) saldırıları için 12425 doğru sınıflandırma ve 3778 yanlış sınıflandırma bulunmaktadır. Keşif saldırıları, daha çok DDoS ve DoS saldırıları ile karıştırılmıştır. Theft (Hırsızlık) kategorisi için neredeyse hiç doğru sınıflandırma yoktur. Modelin hırsızlık verilerini diğer kategorilerle karıştırmakta büyük zorluk çektiği görülmektedir.

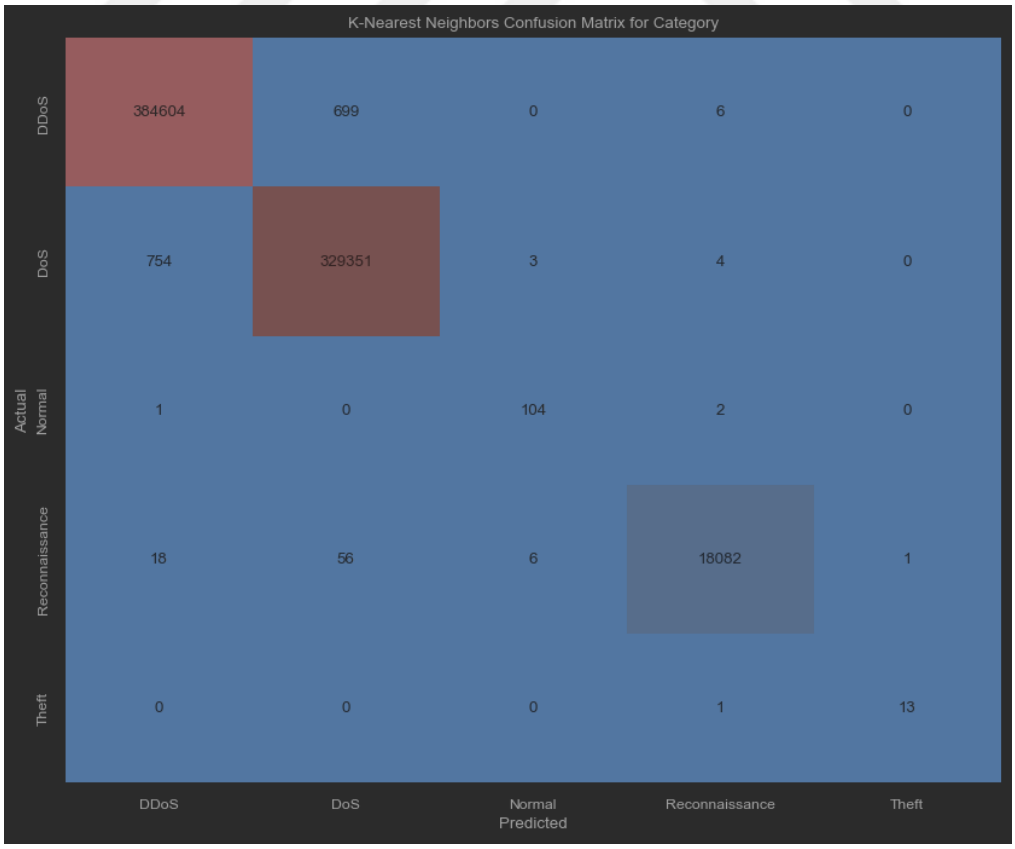
4.3.2. KNN

Model KNN algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 18'de atak sınıfına ait confusion matrix, Şekil 19'da kategori sınıfına ait confusion matrix, Şekil

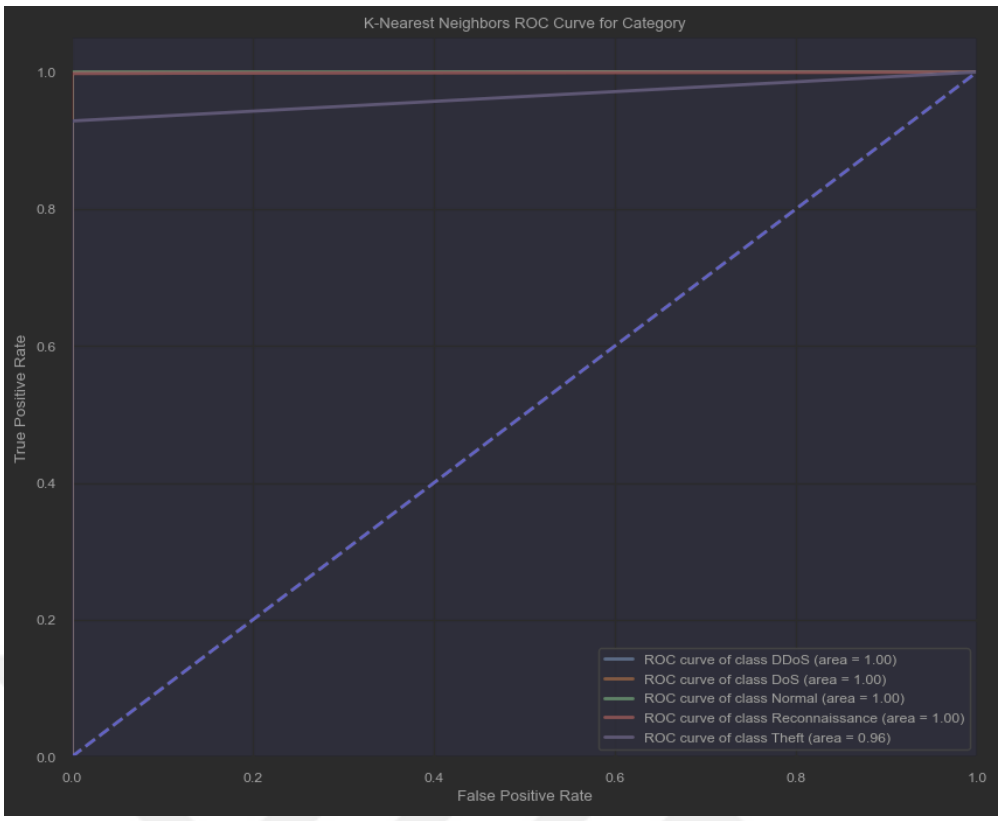
20’de kategori sınıfına ait roc curve ve Şekil 21’de alt kategorilere ait confusion matrix gösterilmiştir.



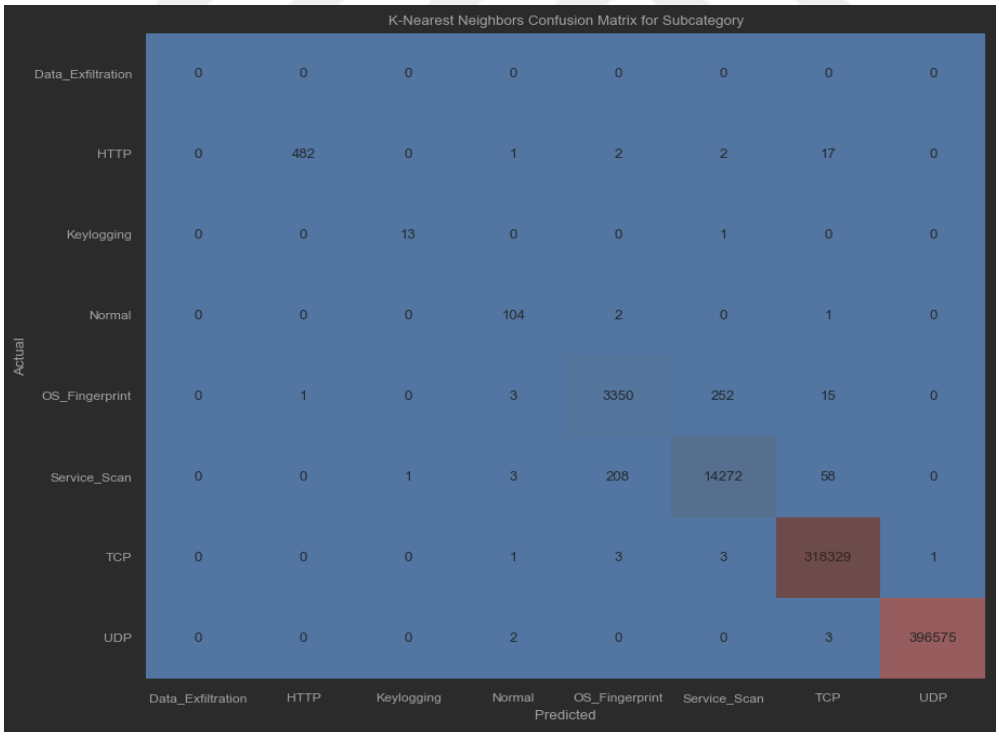
Şekil 18. UNSW KNN atak confusion matrix



Şekil 19. UNSW KNN kategori confusion matrix



Şekil 20. UNSW KNN kategori Roc curve

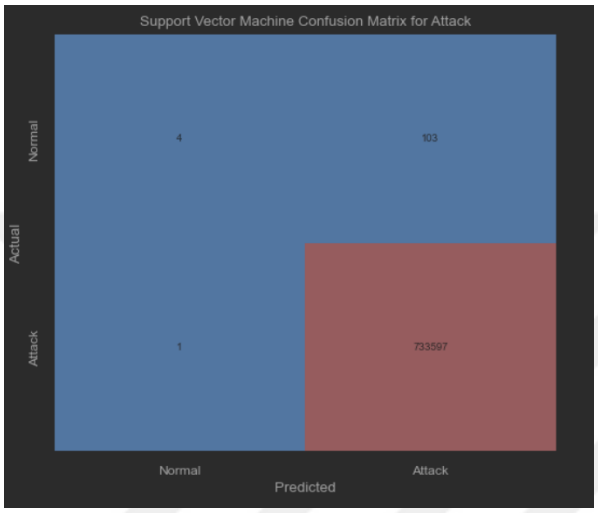


Şekil 21. UNSW KNN alt kategori confusion matrix

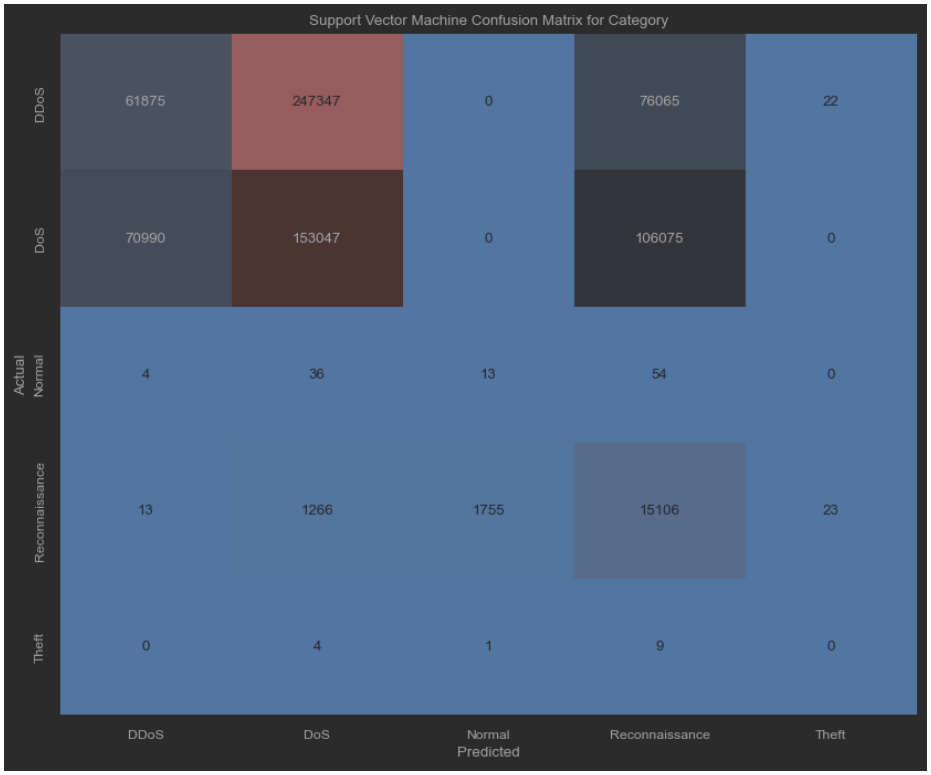
Matrise göre DDoS saldırıları büyük ölçüde doğru sınıflandırılmış olup, 384604 doğru pozitif, 754 yanlış pozitif ve 699 yanlış negatif içermektedir. Ancak, DDoS ve DoS saldırıları arasında belirli bir karışıklık gözlemlenmiştir.

4.3.3. SVM

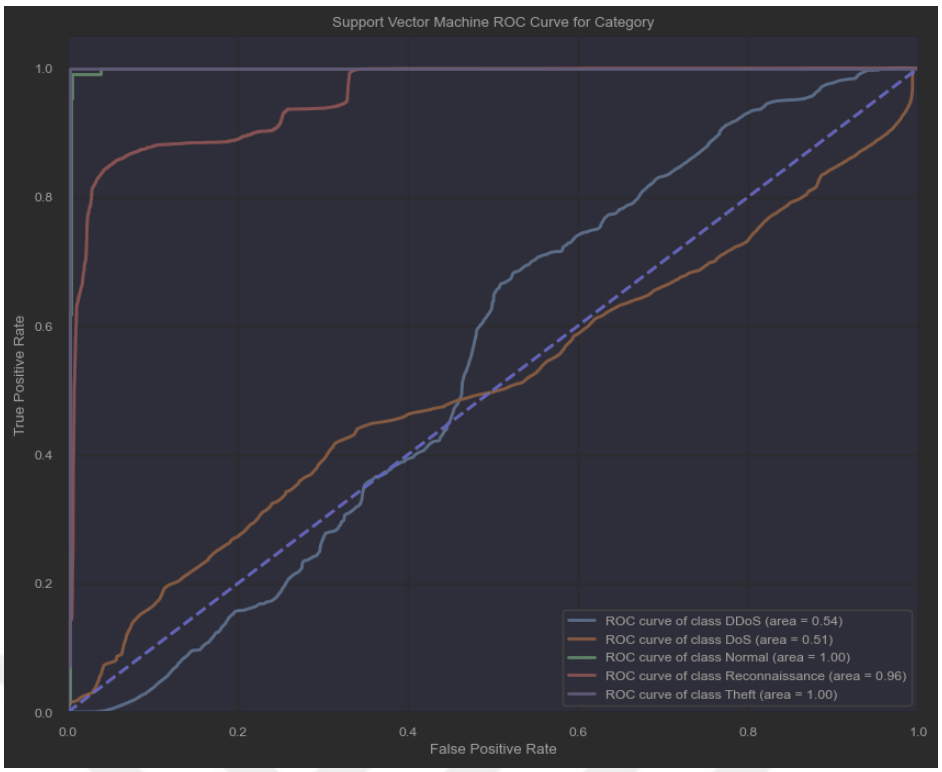
Model SVM algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 22'de atak sınıfına ait confusion matrix, Şekil 23'te kategori sınıfına ait confusion matrix, Şekil 24'te kategori sınıfına ait roc curve ve Şekil 25'te alt kategorilere ait confusion matrix gösterilmiştir.



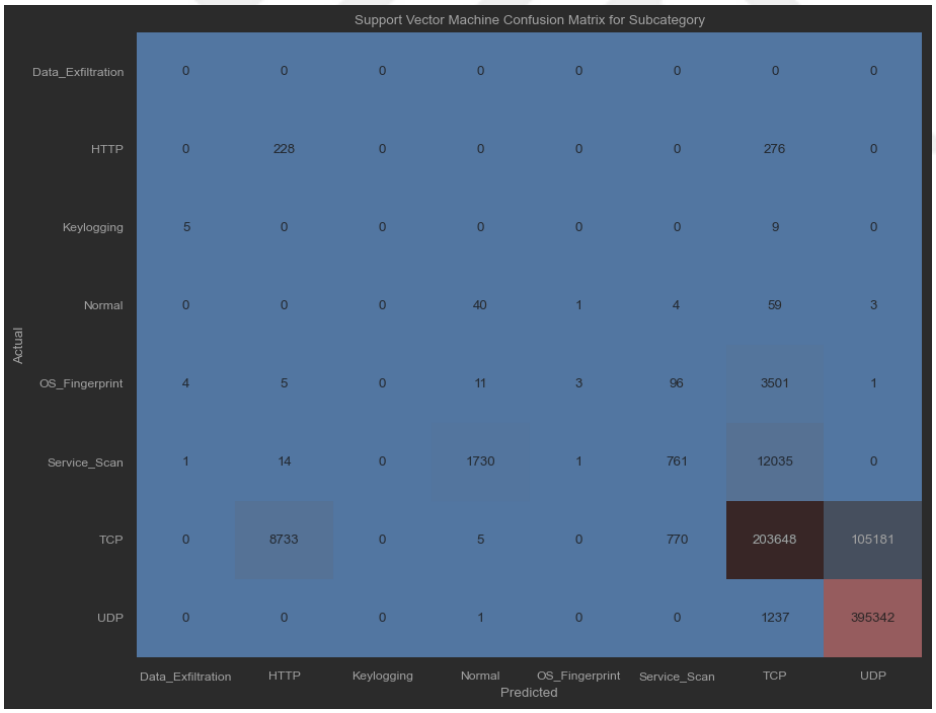
Şekil 22. UNSW SVM atak confusion matrix



Şekil 23. UNSW SVM kategori confusion matrix



Şekil 24. UNSW SVM kategori Roc curve



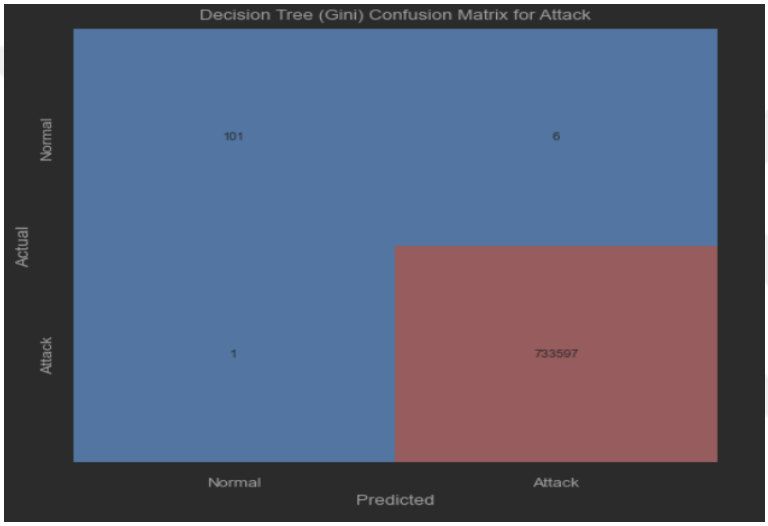
Şekil 25. UNSW SVM alt kategori confusion matrix

Support Vector Machine (SVM) algoritmasının “kategori” hedef değişkeni için performansını gösteren confusion matrix, modelin saldırı türlerini doğru ve yanlış sınıflandırma yeteneğini ortaya koymaktadır. DDoS saldırılarının büyük bir kısmı yanlış olarak DoS ve Reconnaissance olarak sınıflandırılmıştır, bu da modelin DDoS saldırılarını tanımakta zorlandığını göstermektedir. DoS saldırıları, belirgin bir şekilde

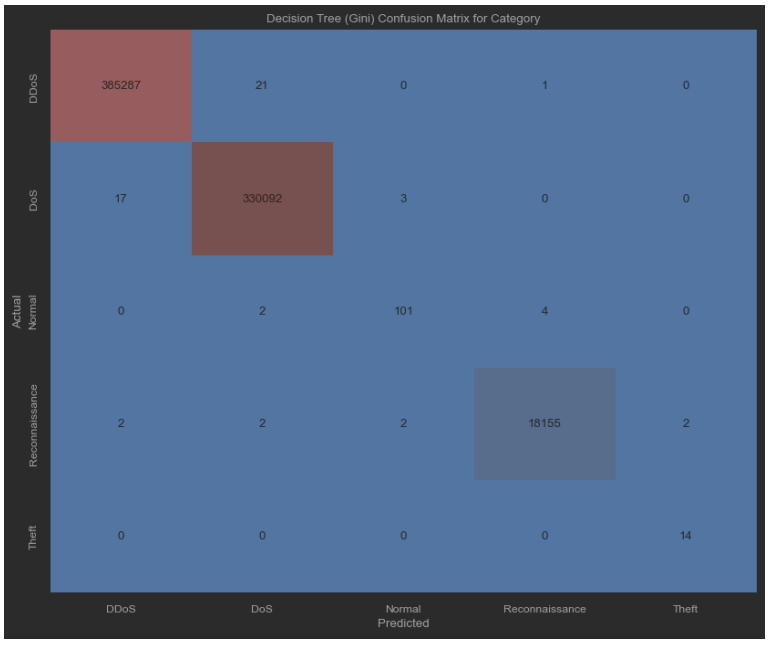
yanlış olarak DDoS saldırılarıyla karıştırılmıştır, bu da modelin bu iki saldırı türünü ayırt etmekte zorlandığını gösterir. Normal trafik ise yüksek bir yanlış sınıflandırma oranına sahiptir ve çoğunlukla diğer kategorilere karıştırılmıştır.

4.3.4. Decision Tree (Gini Index)

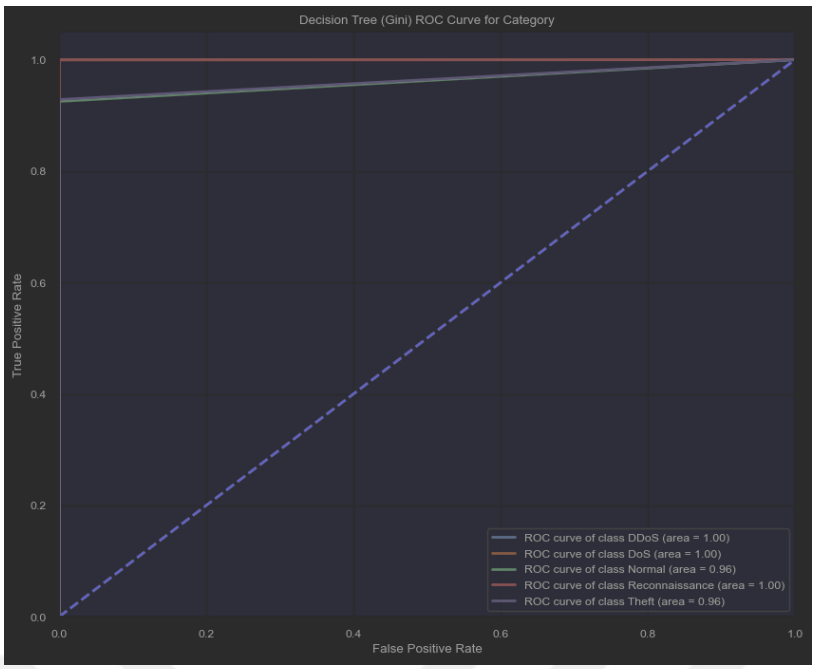
Model Decision Tree (Gini Index) algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 26'da atak confusion matrix, Şekil 27'de kategori confusion matrix, Şekil 28'de kategori roc curve ve Şekil 29'da alt kategorilere ait confusion matrix gösterilmiştir.



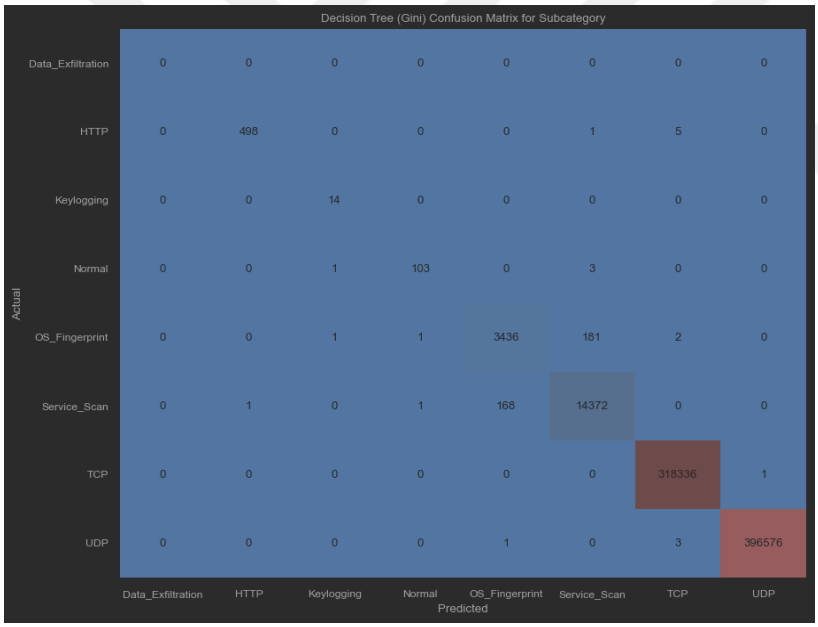
Şekil 26. UNSW Decision Tree (Gini) atak confusion matrix



Şekil 27. UNSW Decision Tree (Gini) kategori confusion matrix



Şekil 28. UNSW Decision Tree (Gini) kategori Roc curve

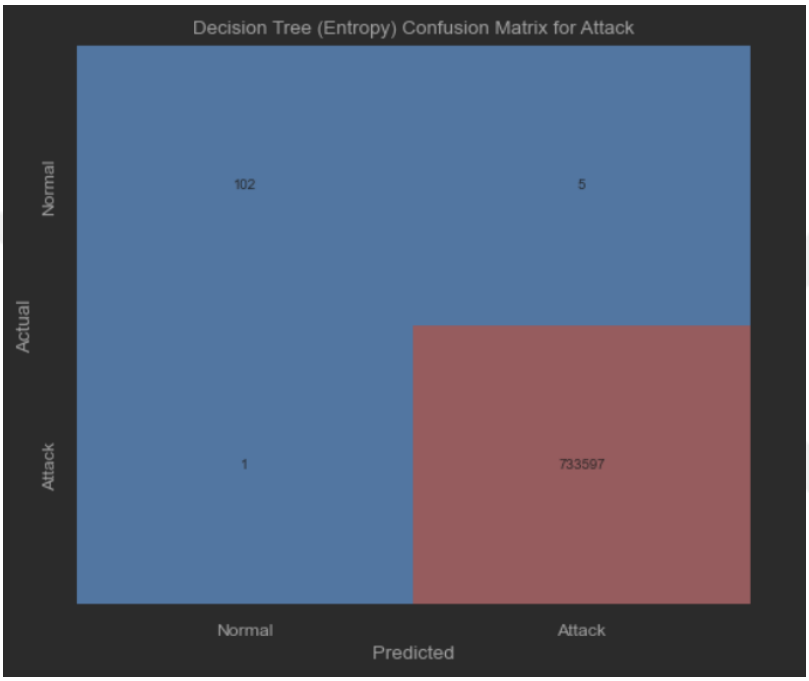


Şekil 29. UNSW Decision Tree (Gini) alt kategori confusion matrix

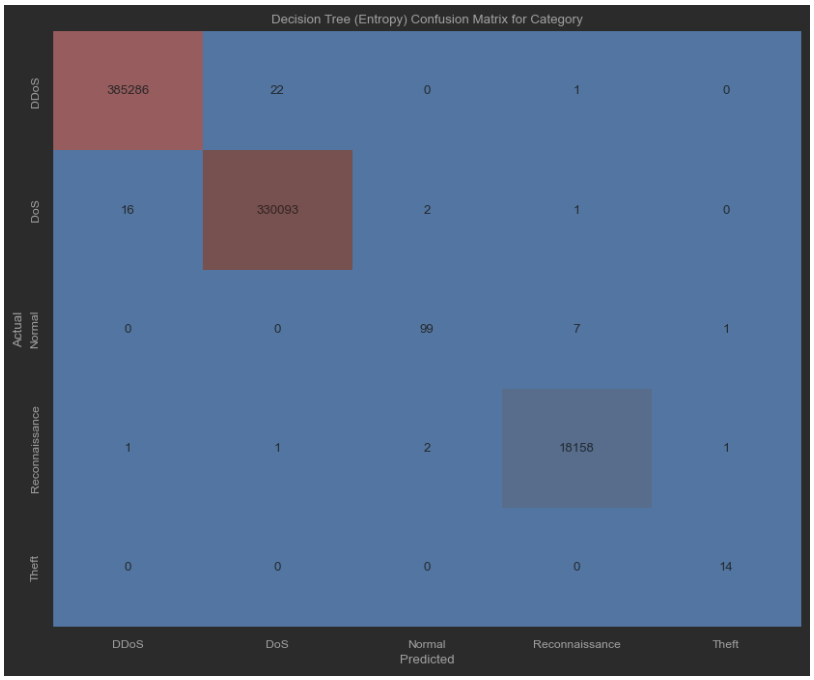
Decision Tree (Gini) DDoS saldırılarının büyük çoğunluğu doğru sınıflandırılmış, 385287 doğru pozitif, 21 yanlış pozitif ve sadece 1 yanlış negatif içermektedir. DoS saldırıları da yüksek doğrulukla sınıflandırılmış olup, 330092 doğru pozitif, 17 yanlış pozitif ve 3 yanlış negatif içermektedir. Normal trafik sınıflandırmasında bazı zorluklar gözlemlenmiştir; 101 doğru sınıflandırma yapılırken, 4 örnek Reconnaissance olarak yanlış sınıflandırılmıştır.

4.3.5. Decision Tree (Entropy)

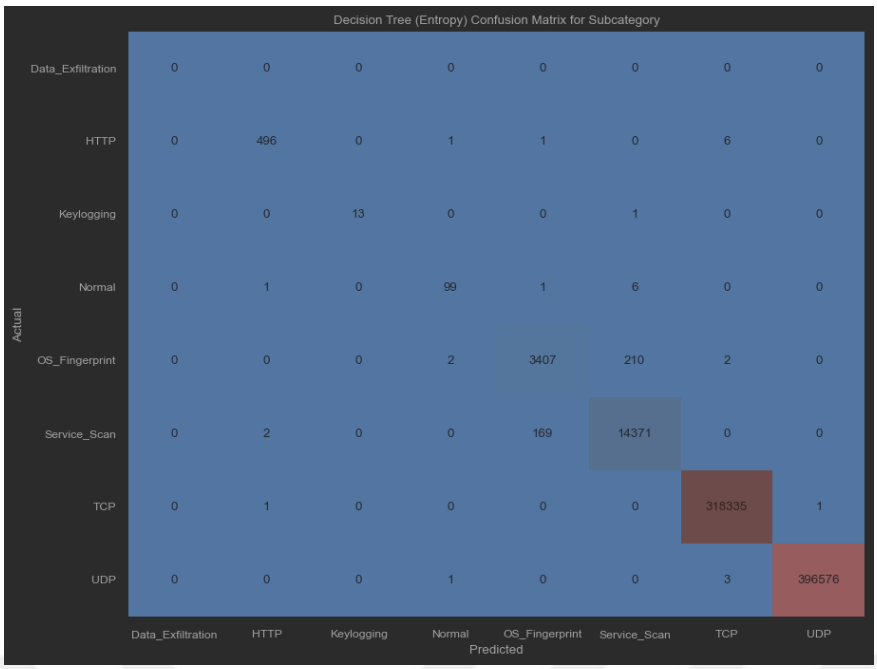
Model Decision Tree (Entropy) algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 30'da atak sınıfına ait confusion matrix, Şekil 31'de kategori sınıfına ait confusion matrix, Şekil 33'te alt kategori sınıfına ait roc curve ve Şekil 32'de alt kategorilere ait confusion matrix gösterilmiştir.



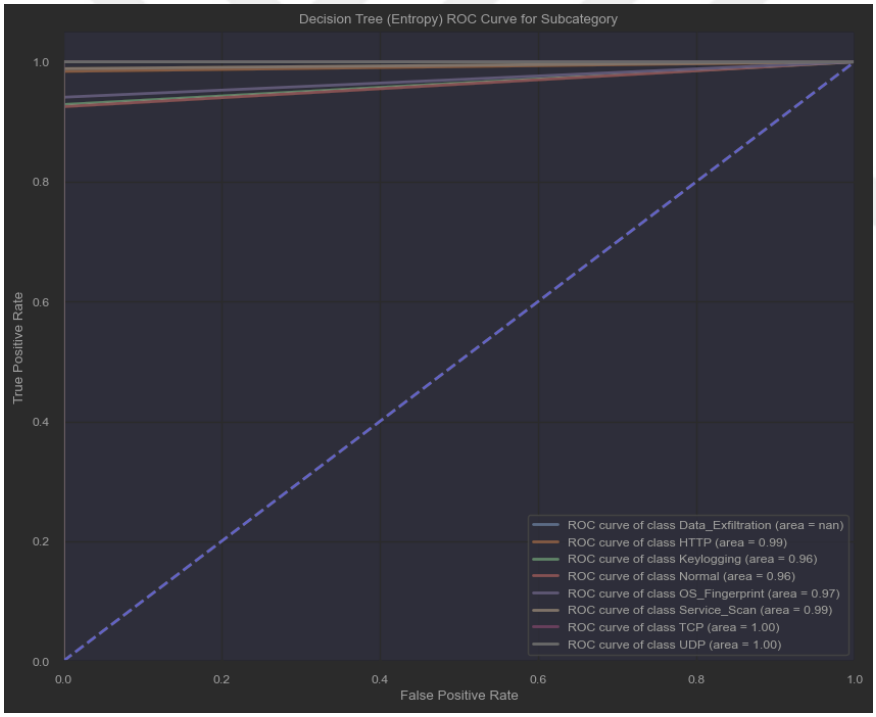
Şekil 30. UNSW Decision Tree (Entropy) atak confusion matrix



Şekil 31. UNSW Decision Tree (Entropy) kategori confusion matrix



Şekil 32. UNSW Decision Tree (Entropy) alt kategori confusion matrix



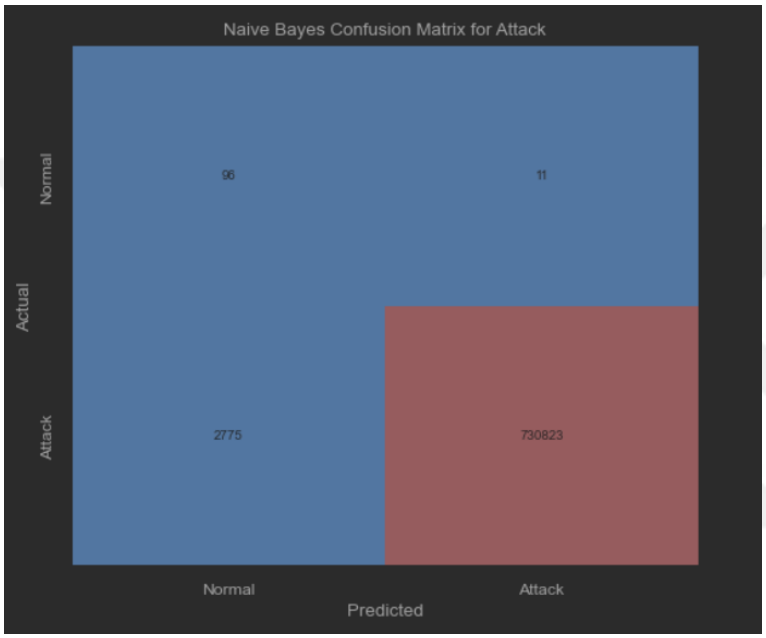
Şekil 33. UNSW Decision Tree (Entropy) alt kategori Roc curve

Decision Tree (Entropy) algoritması DDoS saldırılarının büyük çoğunluğu doğru sınıflandırılmış, 385286 doğru pozitif, 22 yanlış pozitif ve sadece 1 yanlış negatif içermektedir. DoS saldırıları da yüksek doğrulukla sınıflandırılmış olup, 330093 doğru pozitif, 16 yanlış pozitif ve 2 yanlış negatif içermektedir. Normal trafik sınıflandırmasında bazı zorluklar gözlemlenmiştir; 99 doğru sınıflandırma yapılırken, 7 örnek Reconnaissance olarak yanlış sınıflandırılmıştır. Reconnaissance saldırıları büyük

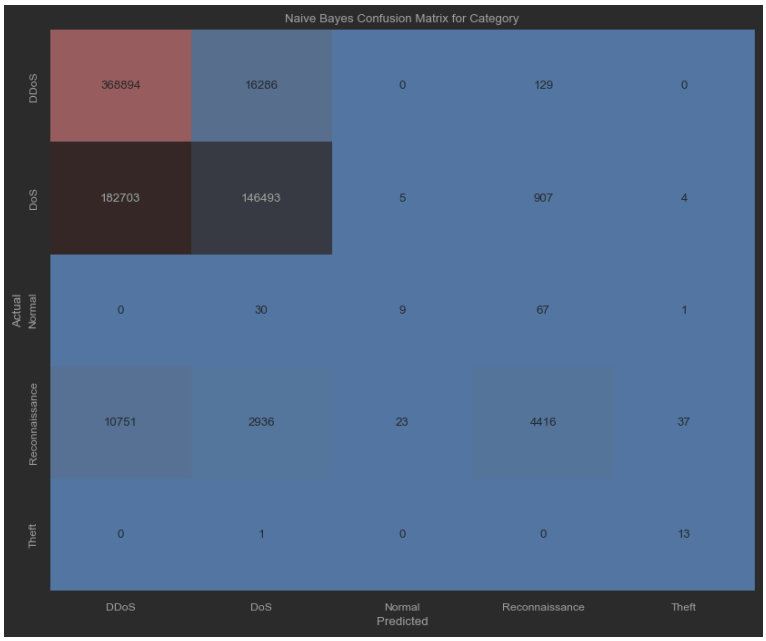
ölçüde doğru sınıflandırılmış, 18158 doğru pozitif, 2 yanlış pozitif ve 2 yanlış negatif içermektedir.

4.3.6. Naive Bayes

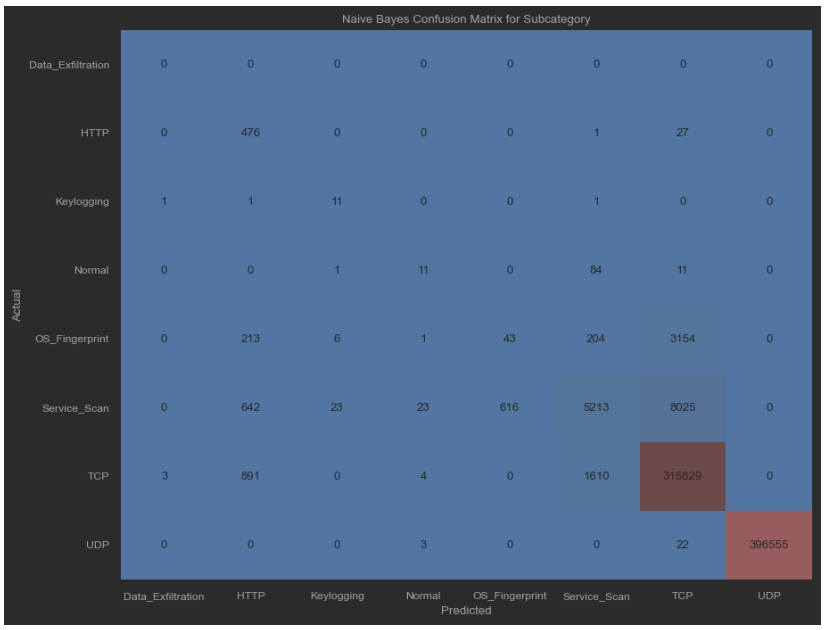
Model Naive Bayes algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 34'te atak sınıfına ait confusion matrix gösterilmiştir.



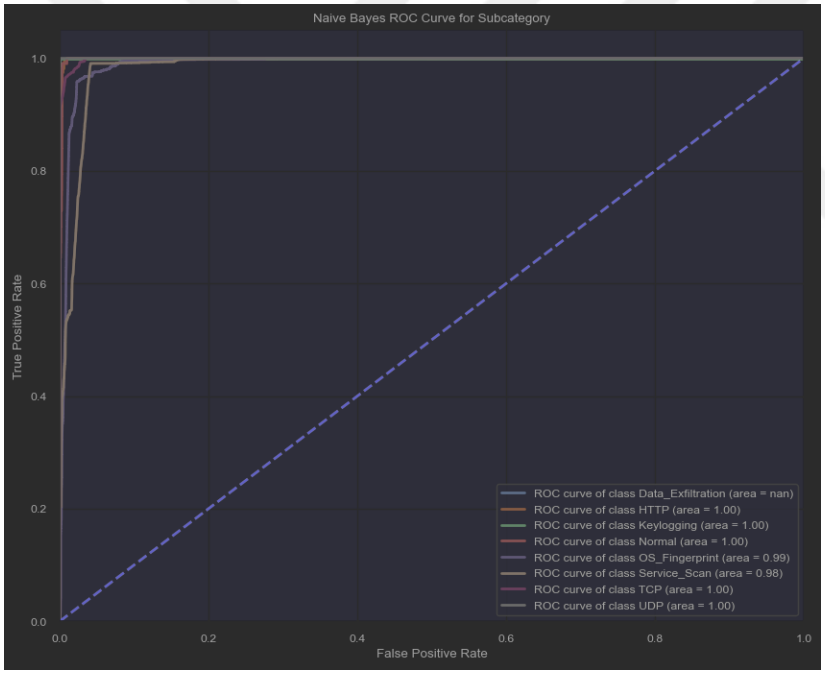
Şekil 34. UNSW Naive Bayes atak confusion matrix



Şekil 35. UNSW Naive Bayes kategori confusion matrix



Şekil 36. UNSW Naive Bayes alt kategori confusion matrix



Şekil 37. UNSW Naive Bayes alt kategori Roc curve

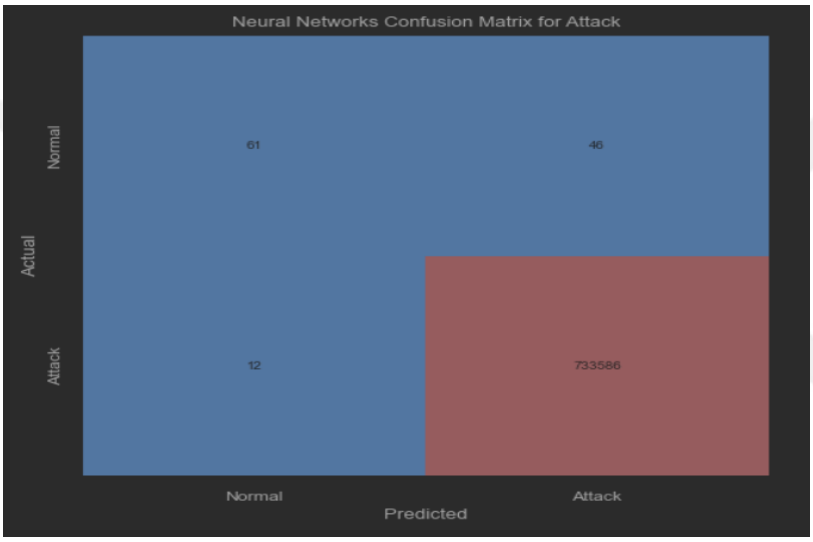
Şekil 35'te kategori sınıfına ait confusion matrix, Şekil 37'de kategori sınıfına ait roc curve ve Şekil 36'da alt kategorilere ait confusion matrix gösterilmiştir.

Naive Bayes algoritması DDoS saldırılarının büyük çoğunluğu doğru sınıflandırılmış, 368894 doğru pozitif, 16286 yanlış pozitif ve 129 yanlış negatif içermektedir. Ancak, DoS saldırıları ile karışıklık yaşanmış ve 182703 DoS saldırısı yanlış olarak DDoS olarak sınıflandırılmıştır. DoS saldırılarının 146493'ü doğru pozitif olarak sınıflandırılmıştır, ancak 907 Reconnaissance ve 4 Theft olarak yanlış sınıflandırılmıştır. Normal trafik tespitinde model önemli zorluklarla karşılaşmıştır;

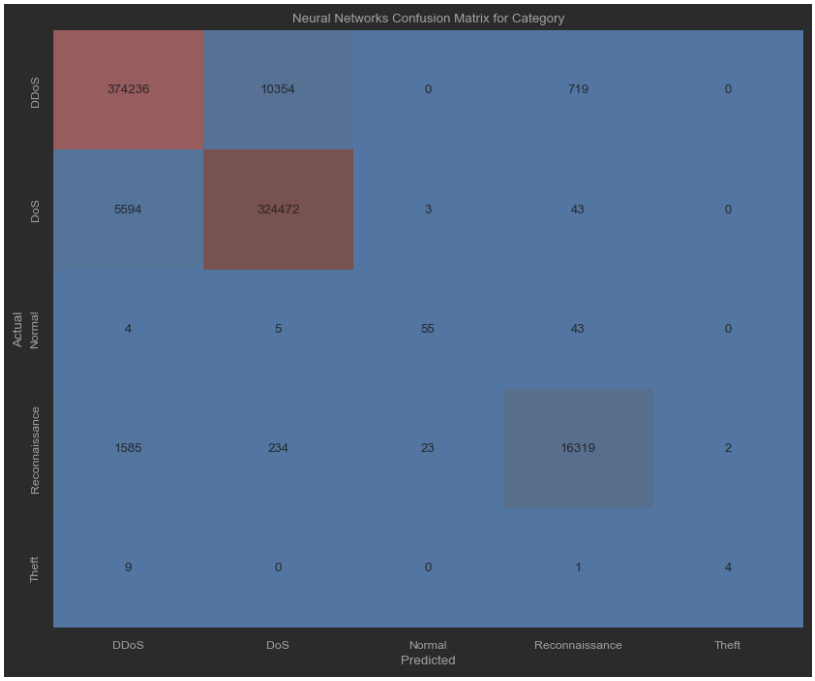
sadece 67 doğru sınıflandırma yapılmış ve bazı örnekler diğer kategorilere karıştırılmıştır. Reconnaissance saldırıları da karışıklık yaşamış, 10751 yanlış pozitif ve 2936 yanlış negatif içermektedir.

4.3.7. Neural Networks

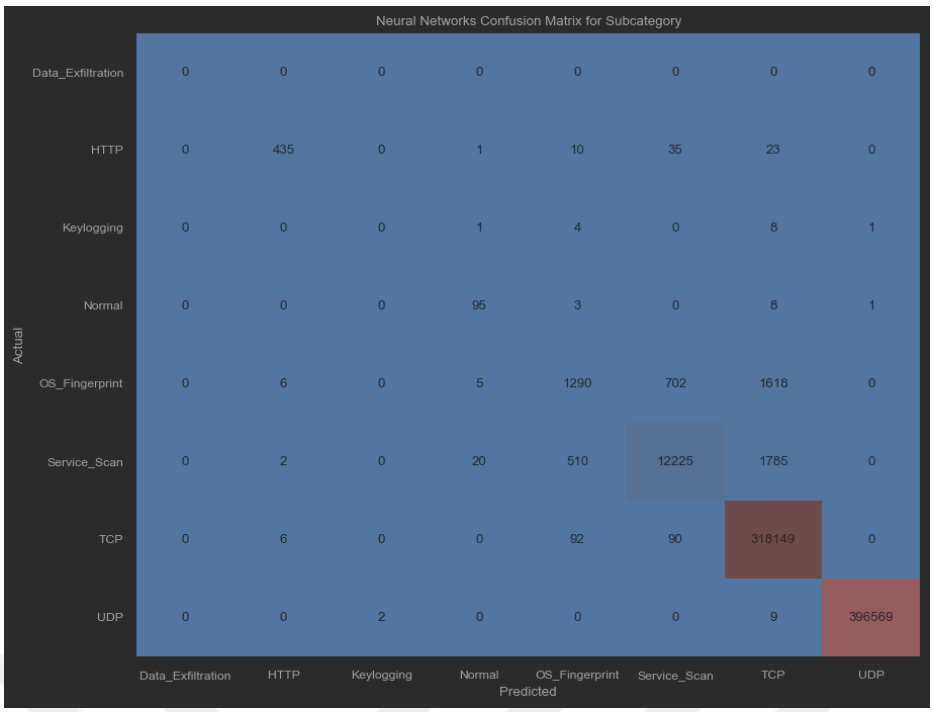
Model Neural Networks algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 38'de atak confusion matrix, Şekil 39'da kategori confusion matrix gösterilmiştir.



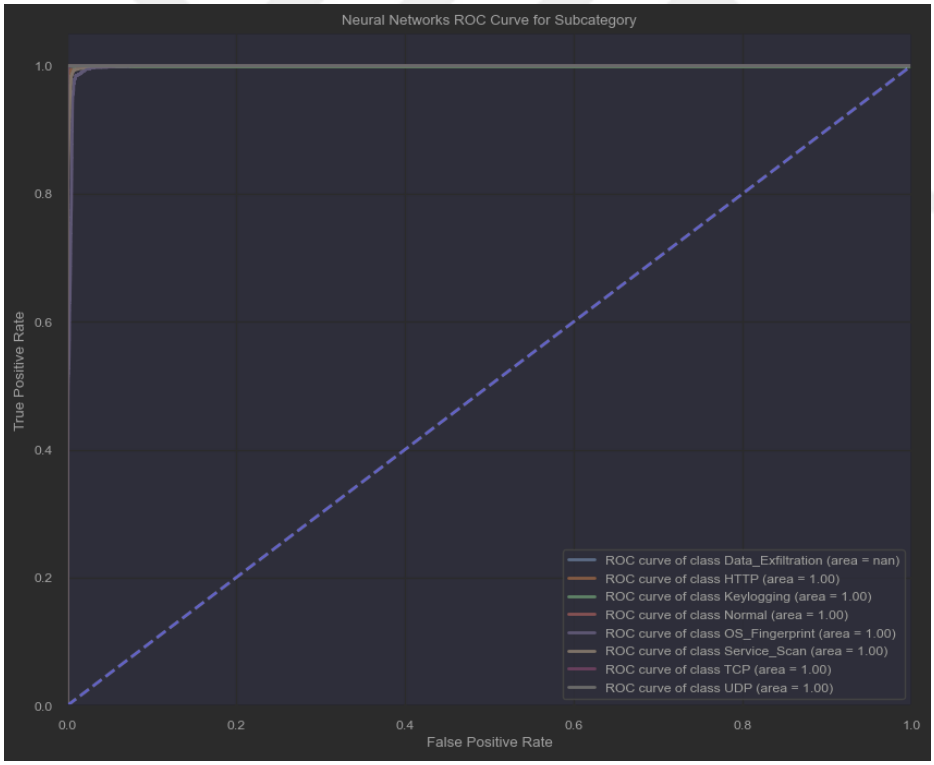
Şekil 38. UNSW NN atak confusion matrix



Şekil 39. UNSW NN kategori confusion matrix



Şekil 40. UNSW NN alt kategori confusion matrix



Şekil 41. UNSW NN alt kategori Roc curve

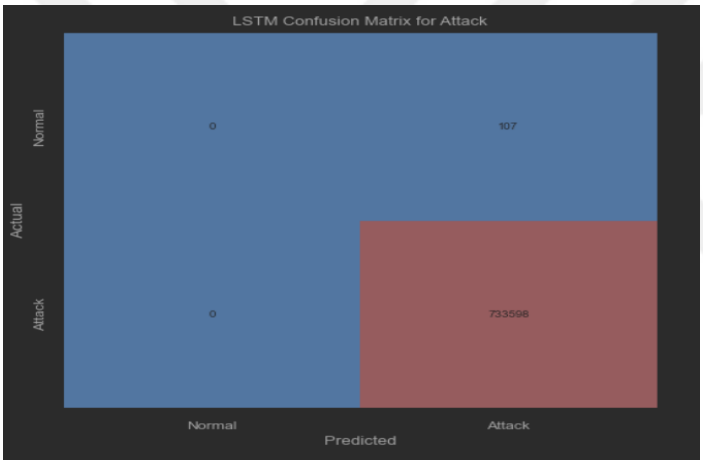
Şekil 41'de kategori sınıfına ait roc curve ve Şekil 40'ta alt kategorilere ait confusion matrix gösterilmiştir.

Neural Networks algoritması DDoS saldırılarının büyük çoğunluğu doğru sınıflandırılmış olup, 374236 doğru pozitif, 10354 yanlış pozitif ve 719 yanlış negatif içermektedir. DoS saldırılarında ise 324472 doğru pozitif bulunmakta, 5594 yanlış

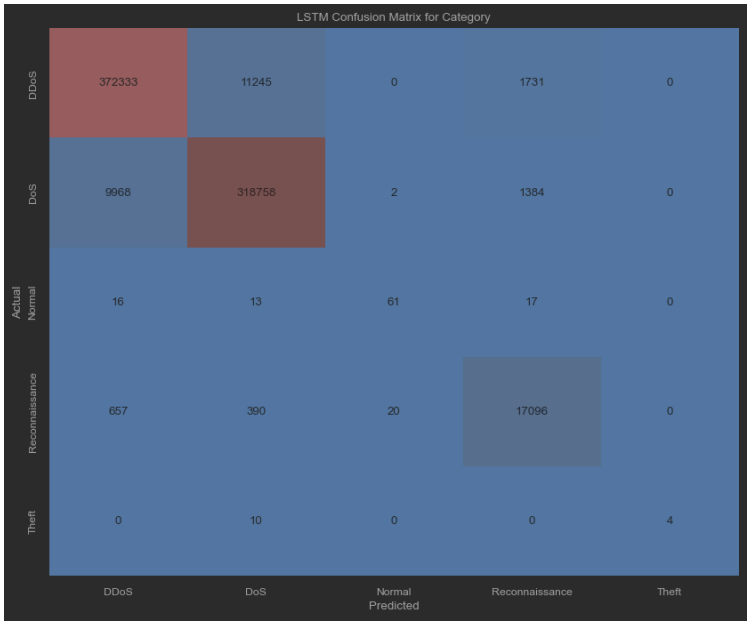
pozitif ve 43 yanlış negatif bulunmaktadır. Normal trafik sınıflandırmasında bazı zorluklar gözlemlenmiştir; sadece 55 doğru sınıflandırma yapılmış ve 43 örnek Reconnaissance olarak yanlış sınıflandırılmıştır. Reconnaissance saldırıları büyük ölçüde doğru sınıflandırılmış olup, 16319 doğru pozitif, 1585 yanlış pozitif ve 234 yanlış negatif içermektedir.

4.3.8. LSTM

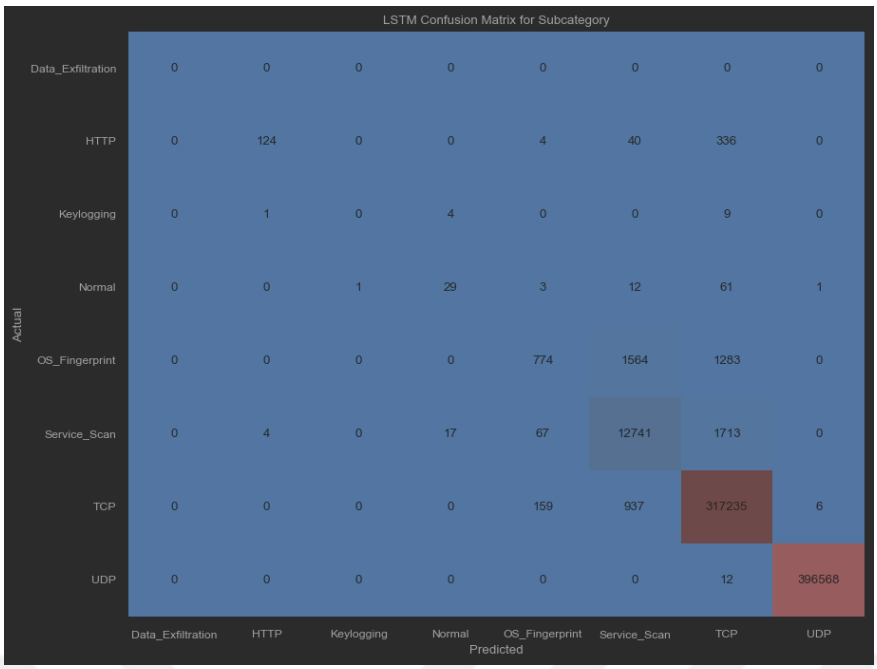
Model LSTM algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 42'de atak sınıfına ait confusion matrix, Şekil 43'te kategori sınıfına ait confusion matrix gösterilmiştir.



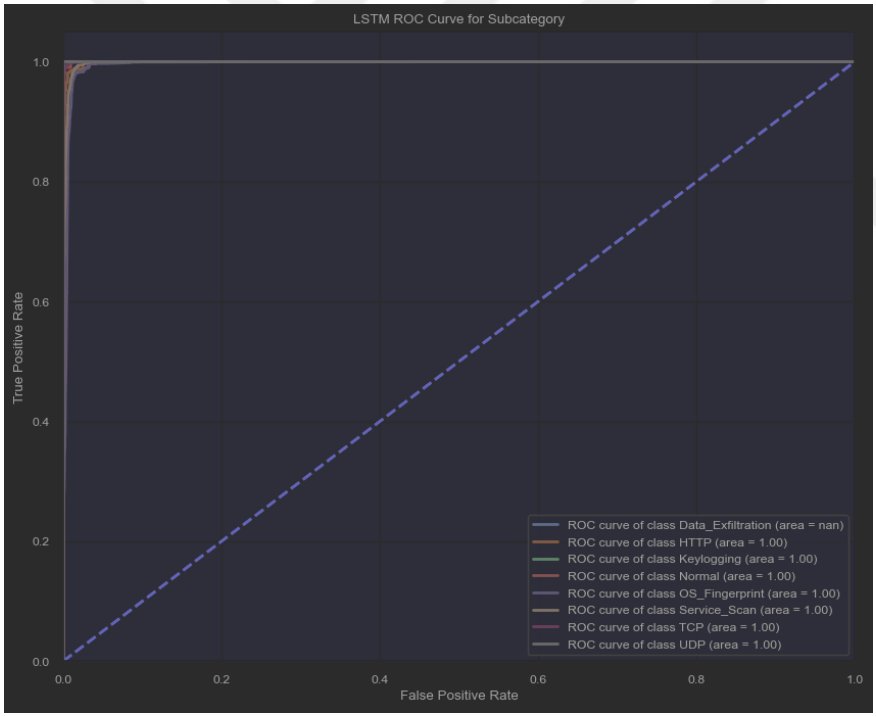
Şekil 42. UNSW Lstm atak confusion matrix



Şekil 43. UNSW Lstm atak confusion matrix



Şekil 44. UNSW Lstm alt kategori confusion matrix



Şekil 45. UNSW Lstm alt kategori Roc curve

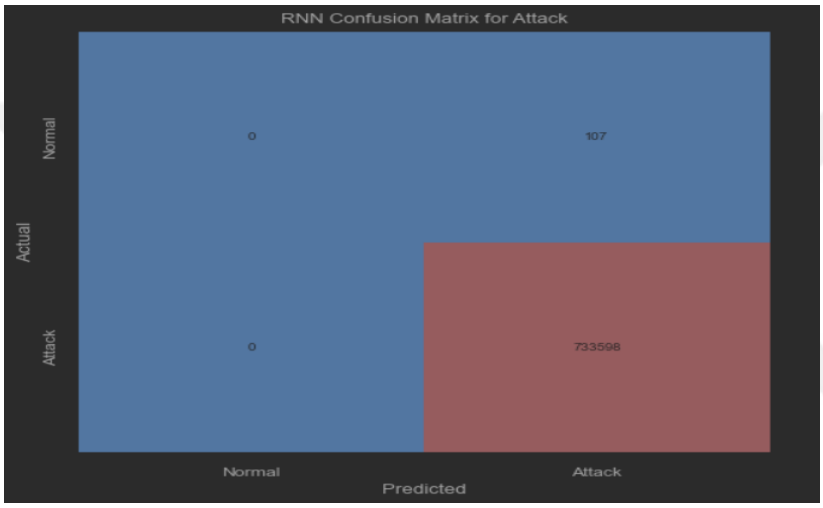
Şekil 45'te kategori sınıfına ait roc curve ve Şekil 44'te alt kategorilere ait confusion matrix gösterilmiştir.

LSTM (Long Short-Term Memory) algoritması DDoS saldırılarının büyük çoğunluğu doğru sınıflandırılmış olup, 372333 doğru pozitif, 11245 yanlış pozitif ve 1731 yanlış negatif içermektedir. DoS saldırılarında ise 318758 doğru pozitif bulunmakta, 9968 yanlış pozitif ve 1384 yanlış negatif bulunmaktadır. Normal trafik sınıflandırmasında bazı zorluklar gözlemlenmiştir; 61 doğru sınıflandırma yapılmış ve

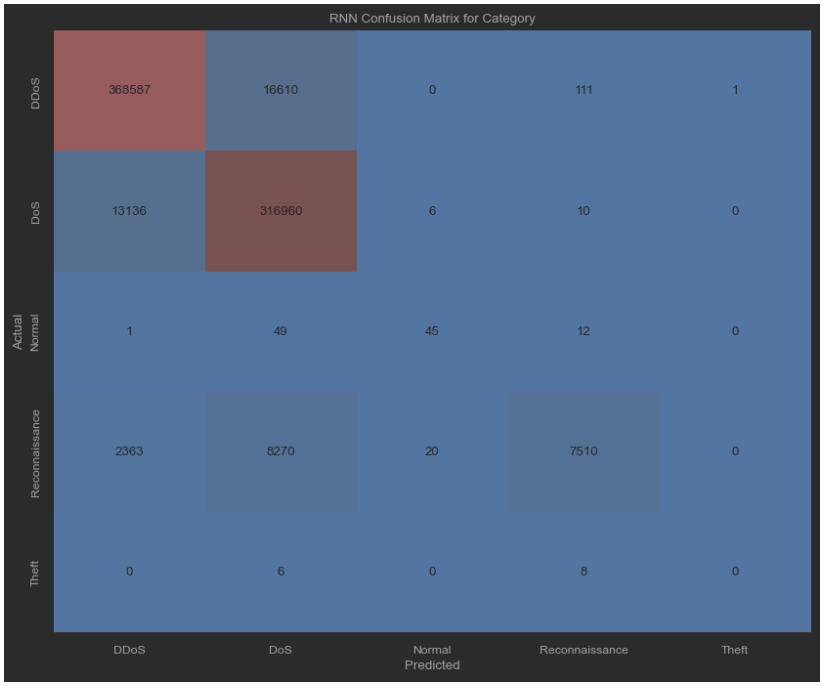
17 örnek Reconnaissance olarak yanlış sınıflandırılmıştır. Reconnaissance saldırıları büyük ölçüde doğru sınıflandırılmış olup, 17096 doğru pozitif, 657 yanlış pozitif ve 390 yanlış negatif içermektedir.

4.3.9. RNN

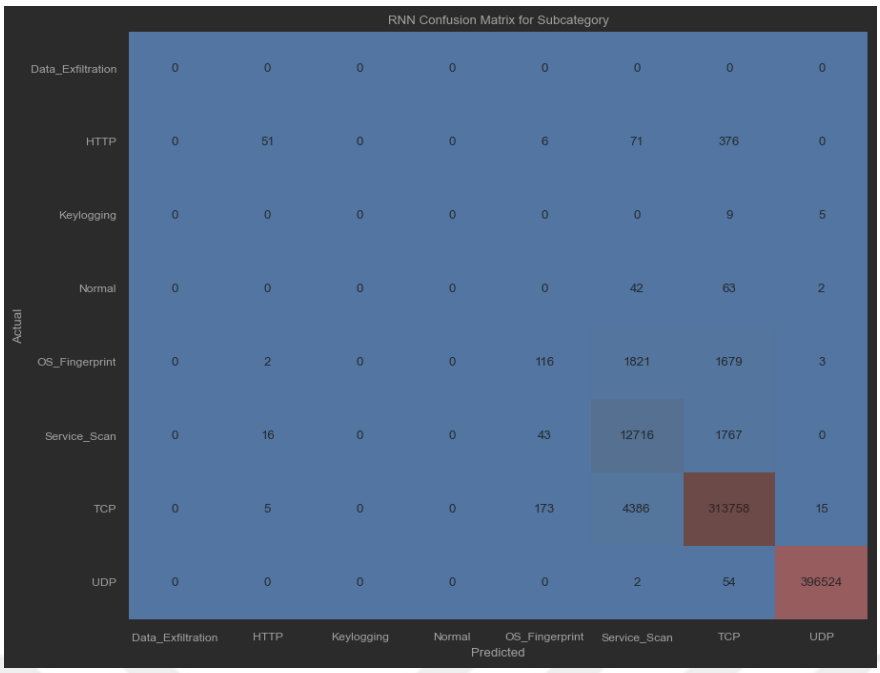
Model RNN algoritmasıyla eğitilip test edilmiştir. Eğitimler için 2 farklı kategori kullanılmıştır. İlk kategoride sadece atak tespiti, ikinci kategoride ise kategori tespiti üzerinde modelimiz test edilmiştir. Test sonuçlarında oluşan çıktılar Şekil 46'da atak sınıfına ait confusion matrix gösterilmiştir.



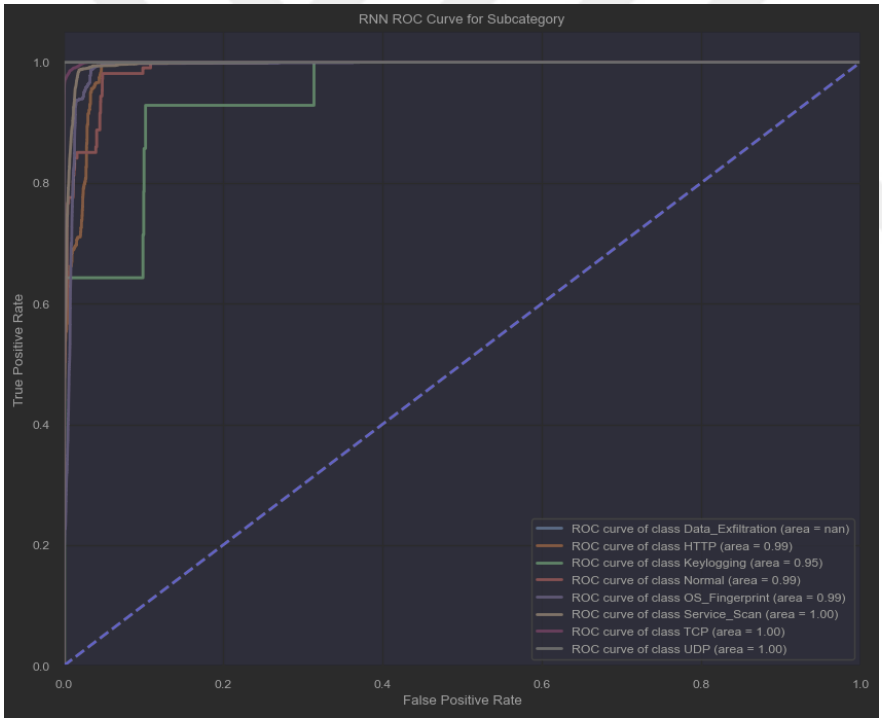
Şekil 46. UNSW RNN atak confusion matrix



Şekil 47. UNSW RNN kategori confusion matrix



Şekil 48. UNSW RNN alt kategori confusion matrix



Şekil 49. UNSW RNN alt kategori Roc curve

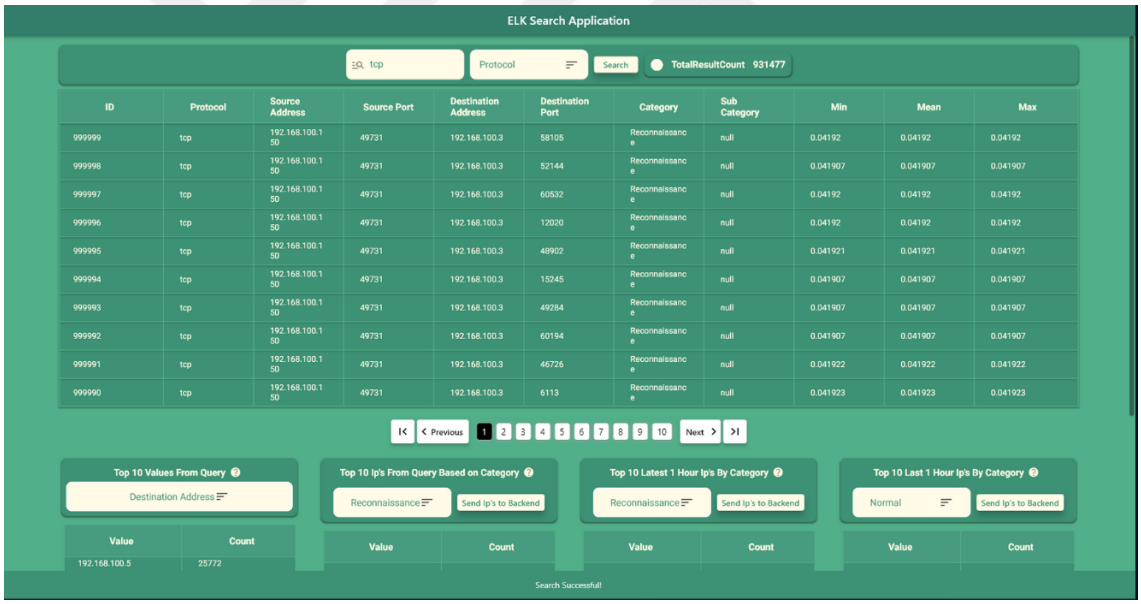
Şekil 47'de kategori sınıfına ait confusion matrix, Şekil 49'da kategori sınıfına ait roc curve ve Şekil 48'de alt kategorilere ait confusion matrix sonuçları gösterilmiştir.

RNN (Recurrent Neural Network) algoritması, DDoS saldırılarının büyük çoğunluğu doğru sınıflandırılmış olup, 368587 doğru pozitif, 16610 yanlış pozitif ve 111 yanlış negatif içermektedir. DoS saldırılarında ise 316960 doğru pozitif bulunmakta, 13136 yanlış pozitif ve 10 yanlış negatif bulunmaktadır. Normal trafik

sınıflandırmasında model belirgin zorluklar yaşamış; sadece 45 doğru sınıflandırma yapılmış ve 12 örnek Reconnaissance olarak yanlış sınıflandırılmıştır. Reconnaissance saldırılarının büyük kısmı doğru sınıflandırılmış olup, 7510 doğru pozitif, 2363 yanlış pozitif ve 8270 yanlış negatif içermektedir.

4.4. Elkstack Entegrasyonu

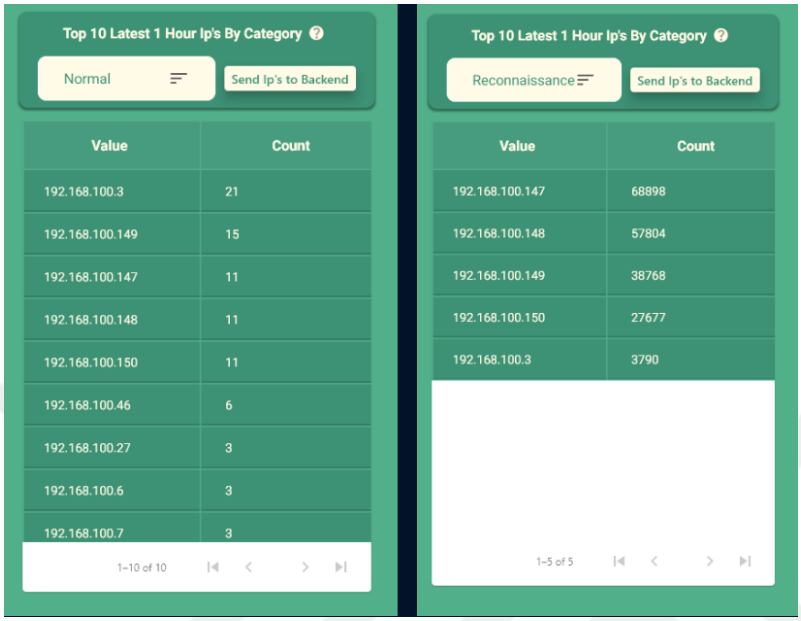
Önerdiğimiz modeli denemek için mevcut veri seti ilk önce elkstack opensource yazılımına import edildi. Import edilen veri seti makine öğrenmesi algoritmalarının çalıştırılması için yerel bilgisayar üzerinde çalıştırıldı ve tespit edilen atack tipleri sınıflandırma işlemleri bittikten sonra tekrar elkstack açık kaynak kodlu yazılıma yüklendi. Bu işlemlerin yapılabilmesi amacıyla Elkstack için api yardımıyla iletişim kuran pyhton flask aracılığıyla backend ve flutter aracılığıyla frontend geliştirilmiştir. Makine öğrenmesi sonucunda işlenmiş sonuçların görselleri ve geliştirilen arayüzün özellikleri aşağıda gösterilmiştir.



Şekil 50. Elkstack uygulama ana ekran görüntüsü

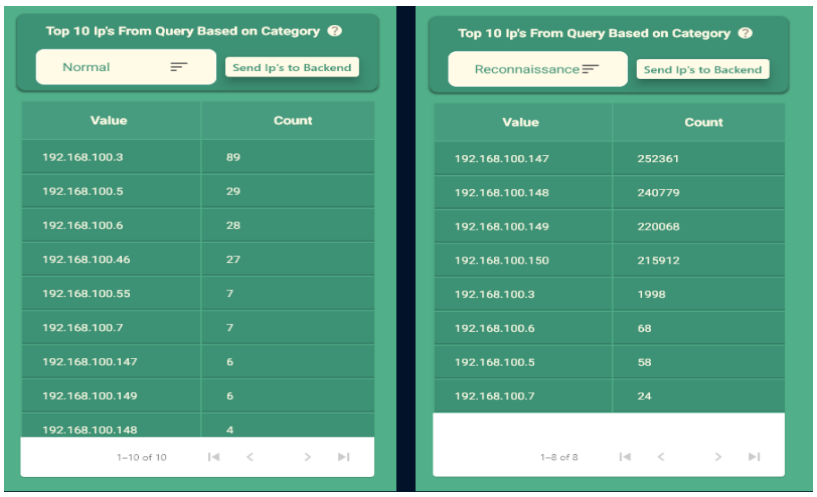
Geliştirilen uygulamadaki ana ekranda elkstack veri tabanında olan değerlerin ip ve protokol bazlı filtreleme işlemlerinin gösterildiği Şekil 50'de mevcut veri setindeki arama işlemleri yapılabilir durumdadır. Ayrıca uygulamanın alt tarafında görünen kısımlarda ise ilgili filtreleme özellikleri eklenerek son kullanıcı için erişim kolaylığı sağlanması amaçlanmıştır. Yine bu ekranda sayfalama işlemlerini kolaylaştırmak amacıysa dinamik olarak güncellenen sayfalama özelliği eklenmiştir. Şekil 51'deki görselde uygulamadaki kategori bazlı arama özelliği gösterilmiştir.

Şekil 51. Elkstack uygulama kategori arama ekran görüntüsü



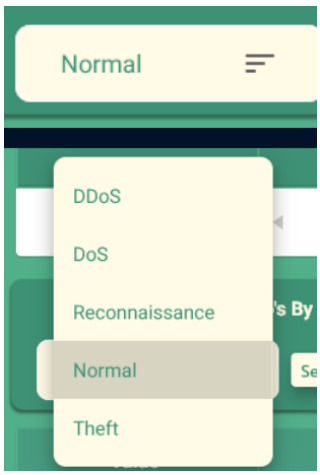
Şekil 52. Elkstack uygulama son saat filtreleme ekran görüntüsü

Şekil 52'de özellikle son 1 saat kategori verisini güncelleme ekranı eklenerek en fazla saldırı yapan ip ve istek sayısının gösterildiği bir ekran eklenmiştir. Ayrıca ilgili ip'nin elkstack ile entegrasyonu yapılacak olan siber güvenlik ürününde otomatik olarak engellenmesi için ilgili güvenlik cihazına otomatik gönderecek özellik eklenmiştir.



Şekil 53. Elkstack uygulama top 10 ip filtreleme ekran görüntüsü

Şekil 53'teki görselde uygulamaya en fazla istek yapan son 10 ip adresini gösterecek şekilde tablo eklenmiştir.



Şekil 54. Elkstack uygulama atak tipine göre filtreleme ekran görüntüsü

Şekil 54'teki görselde uygulamamıza atak tipine göre filtreleme özelliği eklenerek kullanım kolaylığı sağlaması amaçlanmıştır.

5. BULGULAR VE TARTIŞMA

Bu çalışmada, IoT botnet saldırılarının tespiti için çeşitli makine öğrenmesi ve derin öğrenme yöntemlerinin performansları değerlendirilmiştir. UNSW-2018 IoT Botnet veri seti üzerinde Logistic Regression, K-Nearest Neighbors, Support Vector Machines, Decision Trees (Gini ve Entropy), Naive Bayes, Neural Networks, LSTM ve RNN modelleri kullanılarak, saldırı, kategori ve alt kategori sınıfları için model eğitimi yapılmıştır. LSTM modeli için 5 ayrı katman uygulanıp optimizer olarak "adam" seçilmiştir. Ayrıca 50 epoch adımı boyunca modelimiz eğitilmiştir. RNN modelinin eğitiminde 4 ayrı katman kullanılmış ve optimizer olarak "softmax" optimizer seçilmiştir.

Çalışma sonuçlarımız için UNSW-IoT-Botnet veri setinde 3 aşamalı makine ve derin öğrenme yöntemleri uygulanmıştır. BoT-IoT veri setine ise 1 aşamalı makine ve derin öğrenme yöntemi uygulanmıştır. Uyguladığımız ilk aşamada öncelikle verilerin atak ya da atak değil durumu değerlendirildi. 2. Aşamada ise tahmin edilen atakların 5 kategoriden hangi kategoriye ait olduklarını tespit etmeye odaklanarak değerlendirildi. 3. Aşamada ise atak olarak işaretlenen değerlerin alt ana kategorideki 8 atak sınıfına ait atak tipleriyle eşleştirilmesi oranlarına bakıldı. Bu modellerin her bir adıma ait doğruluk, kesinlik, hatırlatma, F1 skoru ve özgüllük gibi performans metrikleri hesaplandı ve ROC eğrileri çizildi. Tablo 3, Tablo 4 ve Tablo 5'te bu aşamalarla ilgili sonuçlara yer verilmiş ve bu sonuçların değerlendirilmesi yapılmıştır.

Tablo 3. Makine öğrenmeleri atak sonuçları karşılaştırma UNSW

	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	Özgüllük	FPR	FNR
Lojistik Regresyon	0.99985	0.999856	0.999785	0.999856	0.009346	0.99065	0.000001
KNN	0.99998	0.99998	0.99998	0.99998	0.97196	0.02803	0.000012
SVM	0.99983	0.99985	0.99979	0.99985	0.03738	0.96261	0.000001
Gini	0.99998	0.99998	0.99998	0.99998	0.93457	0.06542	0.000001
Entropi	0.99999	0.99999	0.99999	0.99999	0.95327	0.04672	0.000001
Naive Bayes	0.99984	0.99620	0.99796	0.99620	0.89719	0.10280	0.003783
NN	0.99988	0.99987	0.99987	0.99989	0.36448	0.63551	0.000012
LSTM	0.99992	0.99993	0.99992	0.99993	0.63551	0.36448	0.000015
RNN	0.99978	0.99985	0.99978	0.99985	0.00934	0.99065	0.000001

Tablo 3'te gösterilen atak tespit sonuçlarında K-En Yakın Komşu ve Karar Ağaçları (Gini ve Entropy) algoritmaları, yüksek doğruluk ve duyarlılık gösterip ayrıca yüksek bir F1 Skoru ve hassasiyet sağlamaktadır. Özgüllük değeri de oldukça yüksek olup bu da düşük bir yanlış pozitif oranı yakalamasını sağlamıştır. Genel olarak, Karar Ağaçları (Gini ve Entropy), Naive Bayes, K-En Yakın Komşu, doğruluk, duyarlılık ve özgüllük açısından en iyi performansı sergileyen sonuçları vermiştir. Bu modeller, düşük yanlış pozitif ve yanlış negatif oranları ile dengeli ve güvenilir sonuçlar almamızı sağlamıştır.

Diğer yandan, Lojistik Regresyon ve Destek Vektör Makinesi de yüksek performans gösterse de, bazı metriklerde (özellikle yanlış pozitif oranı) daha düşük sonuçlar verebilmektedir. RNN ve Neural Networks ise belirli metriklerde iyi performans sergilese de, diğer modeller kadar tutarlı sonuçlar verememiştir.

Sonuç olarak, özellikle saldırı tespitinde yüksek doğruluk ve duyarlılık gerektiren uygulamalar için Karar Ağaçları ve K-En Yakın Komşu algoritmalarını uyguladığımız model çerçevesinde en uygun seçenekler olarak öne çıkmaktadır.

Tablo 4. Makine öğrenmeleri kategori sonuçları karşılaştırma UNSW

	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	Özgüllük	FPR	FNR
Lojistik Regresyon	0.859185	0.859303	0.859084	0.859303	0.945878	0.054122	0.482872
KNN	0.997887	0.997886	0.997886	0.997886	0.999176	0.000824	0.021612
SVM	0.417849	0.313533	0.317076	0.313533	0.784600	0.215400	0.684521
Gini	0.999924	0.999924	0.999924	0.999924	0.999975	0.000025	0.011283
Entropi	0.999925	0.999925	0.999925	0.999925	0.999975	0.000025	0.015021
Naive Bayes	0.762005	0.708493	0.683957	0.708493	0.879078	0.120922	0.468604
NN	0.975237	0.974649	0.974673	0.974649	0.989613	0.010387	0.176381
LSTM	0.976864	0.976621	0.976636	0.976621	0.987599	0.012401	0.307261
RNN	0.957660	0.955533	0.955915	0.955533	0.984615	0.015385	0.344955

Tablo 4'te gösterilen kategorileri tespit etme sonuçlarına baktığımız zaman KNN ve Karar ağaçları algoritmalarının en başarılı sonucu verdiğini gözlemlemekteyiz. Özellikle Karar ağaçları algoritmalarının vermiş olduğu en yüksek F1 skoru ve düşük FPR ve FNR sonuçlarıyla birlikte saldırıların tiplerini belirlemedeki başarım oranıyla dikkat çekmektedir. Karar ağaçları Gini ve Entropi algoritmaları en yüksek doğruluk, duyarlılık ve F1 skoru ile dikkat çekiyor. Ayrıca çok düşük yanlış pozitif ve negatif

oranlarına sahip olması da kullanım açısından öne çıkmaktadır. Destek Vektör Makinesi (SVM) algoritması doğruluk, duyarlılık ve F1 skoru açısından en düşük performansı sergileyen model durumundadır. Ayrıca yüksek yanlış pozitif ve negatif oranları gözlenmekte olup yine atak tiplerini tespit etmedeki başarısı diğer modellere göre düşük kalmıştır. Naive Bayes diğer modellere kıyasla daha düşük doğruluk ve duyarlılık değerleriyle birlikte daha yüksek yanlış pozitif oranına sahip olmuştur. Neural Networks, LSTM ve RNN modelleri de yüksek performans gösterse de bazı metriklerde diğer modellere göre daha düşük performans sergilemiştir.

Tablo 5. Makine öğrenmeleri alt kategori sonuçları karşılaştırma UNSW

	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	Özgüllük	FPR	FNR
Lojistik Regresyon	0.973252	0.979598	0.976003	0.979598	0.995972	0.004028	0.554082
KNN	0.999205	0.999209	0.999207	0.999209	0.999888	0.000112	0.029570
SVM	0.839393	0.817797	0.806105	0.817797	0.974994	0.025006	0.521114
Gini	0.999490	0.999489	0.999489	0.999489	0.999936	0.000064	0.015993
Entropi	0.999442	0.999444	0.999443	0.999444	0.999930	0.007031	0.027699
Naive Bayes	0.974522	0.978783	0.975045	0.978983	0.995873	0.004127	0.350578
NN	0.994380	0.994304	0.993705	0.994304	0.998908	0.001092	0.139349
LSTM	0.995162	0.995228	0.995052	0.995228	0.999332	0.000668	0.103554
RNN	0.985891	0.987635	0.985042	0.987635	0.984615	0.015385	0.344955

Tablo 5'te gösterilen sonuçları değerlendirdiğimiz de Karar ağaçları (Gini ve Entropi), LSTM, K-En Yakın Komşu algoritmaları atak alt kategorilerini belirleme değerlerindeki doğruluk, duyarlılık ve F1 skoru ile en yüksek performansa sahip durumda olup modelimizle çalışan en başarılı algoritma durumundadır. NN ise doğruluk, duyarlılık ve F1 skorundaki yüksek oranlarla diğer en başarılı algoritmalar arasındadır. Ayrıca çok düşük yanlış pozitif ve negatif oranlarına sahip olması modelimizle birlikte kullanılması için tercih sebebidir. SVM ve Naive Bayes modelimizle uyumlu daha düşük performans sergileyen algoritmalar yöntemler olmuşlardır. Lojistik Regresyon ve RNN modelleri yüksek performans gösterse de bazı metriklerde diğer modellere göre daha düşük performans sergiledikleri gözlemlenmiştir. Özellikle FNR oranı diğer modellere göre yüksek çıktığı gözlemlenmiştir. Destek Vektör Makinesi algoritması ise bu kategorideki en düşük değerlere sahip olup özellikle FNR değerinde ki yükseklik dikkat çekmektedir.

Tablo 6. Makine öğrenmeleri atak sonuçları karşılaştırma BoT-IoT

	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	Özgüllük	FPR	FNR
Lojistik Regresyon	0.965876	0.969299	0.967002	0.969299	0.977986	0.022014	0.298041
KNN	0.983603	0.983855	0.983604	0.983855	0.982444	0.017556	0.030785
SVM	0.902446	0.891899	0.885553	0.891899	0.970997	0.029003	0.277057
Gini	0.983995	0.984245	0.984000	0.984245	0.980978	0.019022	0.023746
Entropi	0.984471	0.984722	0.984472	0.984722	0.981465	0.022016	0.029100
Naive Bayes	0.967761	0.970392	0.968273	0.970491	0.979937	0.020063	0.194789
NN	0.980690	0.980902	0.980353	0.980902	0.979954	0.020046	0.086175
LSTM	0.981581	0.981864	0.981536	0.981864	0.980926	0.019287	0.067777
RNN	0.974946	0.976068	0.974521	0.976068	0.972558	0.027442	0.190478

Tablo 6'da BoT-IoT veri seti üzerinde çeşitli makine öğrenimi modellerinin performansları karşılaştırılmıştır. K-En Yakın Komşular (KNN) ve Karar Ağacı (Gini ve Entropi) modelleri, yüksek doğruluk, duyarlılık ve özgüllük ile düşük hata oranları göstererek en yüksek performansı sergiledi. Yapay Sinir Ağları (NN) ve LSTM de oldukça başarılı sonuçlar vermiştir. Lojistik Regresyon ve Destek Vektör Makinesi (SVM) modelleri, özellikle FNR oranları yüksek olduğu için diğer modellere kıyasla daha düşük performans gösterdiği gözlemlenmiştir. Naive Bayes, duyarlılık açısından iyi olsa da kesinlik ve FNR metriklerinde daha düşük performans sergiledi. Genel olarak, KNN ve Karar Ağacı modelleri bu veri seti üzerinde en etkili yöntemler olarak öne çıkmaktadır.

BoT-IoT ve UNSW veri setleri üzerinde farklı modellerin performans sonuçları karşılaştırıldığında, genel olarak K-En Yakın Komşular, Karar Ağaçları ve LSTM modelleri her iki veri setinde de yüksek doğruluk, duyarlılık ve F1 skorlarına ulaşmıştır. BoT-IoT veri setinde, K-En Yakın Komşular modeli en yüksek performansı sergilerken, UNSW veri setinde ise Karar Ağaçları (Gini ve Entropi) modelleri en iyi sonuçları göstermektedir. Logistic Regression ve Naive Bayes modelleri, özellikle UNSW veri setinde düşük hata oranlarıyla dikkat çekmektedir. Bununla birlikte, Destek Vektör Makineleri (SVM), her iki veri setinde de göreceli olarak düşük performans sergilemiştir. Her iki veri setinde de LSTM ve RNN modelleri oldukça başarılı sonuçlar elde etmiştir, ancak IoT-Botnet veri setinde RNN'in performansı daha yüksek olmuştur. Sonuç olarak, BoT-IoT veri setinde K-En Yakın Komşular ve Karar ağaçları modelleri öne çıkarken, UNSW veri setinde Karar Ağaçları ve Logistic Regression modelleri daha iyi performans gösterdiği gözlemlenmiştir.

5.1. Makine Öğrenmesi Modellerinin Performanslarının Karşılaştırılması

Saldırı sonuçlarına göre, tüm modeller oldukça yüksek doğruluk oranlarına sahiptir. Lojistik Regresyon, KNN, SVM, Karar Ağaçları (Gini ve Entropy) ve RNN %99.98 ve üzeri doğruluk oranları ile en iyi performans gösteren algoritmalar olmuşlardır. F1 Score, Precision ve Recall değerleri de bu doğruluk oranları ile uyumludur. Neural Networks ve Naive Bayes modelleri ise nispeten daha düşük performans göstermiştir, ancak yine de %99'un üzerinde doğruluk oranlarına sahiptir.

Kategori sonuçlarına göre, KNN ve Karar Ağaçları (Gini ve Entropy) %99.99 doğruluk oranlarına sahiptir. Bu modeller aynı zamanda en yüksek F1 Score ve Precision değerlerine de sahiptir. Naive Bayes ve SVM modelleri ise daha düşük performans göstermiştir; özellikle SVM %41.78 doğruluk oranı ile en düşük performansı sergilemiştir. Neural Networks ve LSTM modelleri ise %97.52 ve %97.68 doğruluk oranları ile yüksek performans göstermiştir.

Alt kategori sonuçlarına göre, K-En Yakın Komşu (KNN), Karar Ağaçları (Gini ve Entropy) ve LSTM gibi modellerin performansları oldukça yüksektir. KNN, Gini ve Entropy modelleri %99.94 gibi yüksek doğruluk oranlarına sahipken, LSTM %99.52 doğruluk oranı ile diğer modellerden sadece biraz daha düşüktür. Destek Vektör Makineleri (SVM) %83.93 doğruluk oranı ile en düşük performansı göstermiştir. F1 Score, Precision ve Recall değerleri de doğruluk oranları ile paralellik göstermektedir.

5.2. Başka Çalışmalarla Karşılaştırma

Muhammad Shafiq ve arkadaşları 2021 yılında yapmış oldukları bir çalışmalar BoT-IoT dataseti üzerinde IoT ağında kötü niyetli trafik tanımlaması için seçilen özellikleri doğrulamak üzere bijektif bir yumuşak kümeye dayalı entegre TOPSIS ve Shannon entropisini uygulamışlardır (Muhammad Shafiq, 2021). Bot-IoT veri seti üzerinde dört makine öğrenimi algoritmasını kullanarak Tablo 7'de yer alan sonuçları almışlardır.

Tablo 7. BoT-IoT veri setinde yapılmış bir çalışmanın sonuçları

	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru	Özgüllük	FPR	FNR
Karar Ağacı	0.9895	0.9895	0.9999	0.9947	0.9895	0.0105	0.0001
Rastgele Orman	0.9999	0.9999	0.9999	0.9999	0.9999	0.0001	0.0001
Naive Bayes	0.9844	0.9788	0.9788	0.9750	0.9844	0.0156	0.3506
SVM	0.9848	0.8178	0.8178	0.8061	0.9848	0.0152	0.5211

Yine başka bir çalışmada Amin Shahraki ve arkadaşları UNSW-IoT-Botnet veri seti üzerinde önermiş oldukları “Additive Expert” yöntemiyle %95.4 ve random forest yöntemiyle %99.9’luk başarı oranı yakalamışlardır (Amin, 2022).

6. SONUÇLAR VE ÖNERİLER

Bu çalışmada, IoT cihazlar üzerine botnet saldırılarının tespiti için makine öğrenmesi ve derin öğrenme algoritmalarının performansları değerlendirmiştir. K-En Yakın Komşu (KNN), Karar Ağaçları (Gini ve Entropy) ve LSTM gibi modeller, veri setlerimizde yüksek doğruluk oranları ile öne çıkmıştır. Özellikle KNN ve Karar Ağaçları modelleri, %99.99 doğruluk oranları ile en yüksek performansı göstermiştir. Destek Vektör Makineleri (SVM) ise kategori ve alt kategori sınıflandırmalarında daha düşük performans sergilemiştir.

IoT cihazları korumak için planlanan saldırı önleme sistemlerinde KNN ve Karar ağaçları bu modellerin göstermiş olduğu yüksek doğruluk oranları ve düşük negatiflik oranlarıyla iyi performans değerlerinden dolayı tercih edilebilir. Zaman serisi verilerinde ve karmaşık veri yapılarını öğrenmede başarılı olan LSTM ve RNN gibi derin öğrenme modellerinin kullanımı tercih edilmelidir. Bu modellerin, özellikle gerçek zamanlı saldırı tespit sistemlerinde etkili olabileceği görülmüştür. SVM modelinin düşük performans sonuçları verdiği düşünüldüğünde bu modelin optimizasyonu ve iyileştirilmesi üzerine çalışmalar yapılmalıdır. Bu çalışma sonuçlarına baktığımız zaman IoT ağlarında gerçek zamanlı saldırı tespit sistemlerinin geliştirilmesi ve uygulanarak güvenlik önlemleri artırılabilir.

Gelecekteki çalışmalarda farklı veri setleri ve daha geniş özellik setleri kullanarak modellerin genel performansları değerlendirilmelidir. DDoS saldırılarının gelecekte artarak devam edeceği düşünüldüğünde bu alandaki güvenlik stratejilerine dahil edilmesi ve güvenlik açısından önemli katkılar sağlayacaktır.

KAYNAKÇA

Ahmad, M., Rehman, S. U., Rashid, A., Iqbal, R., & Jan, M. A. (2021). Intrusion detection in Internet of Things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. EURASIP Journal on Wireless Communications and Networking, 2021(1), 1-23. https://doi.org/10.1186/s13638-021-01968-9

Alosaimi, S., & Almutairi, S. M. (2023). An intrusion detection system using BoT-IoT. Applied Sciences, 13(9), 5427. https://doi.org/10.3390/app13095427

Albulayhi, K., Alotaibi, F., Alotaibi, R., Alotaibi, M., & Alotaibi, M. (2021). IoT intrusion detection taxonomy, reference architecture, and analyses. Sensors, 21(19), 6432. https://doi.org/10.3390/s21196432

Alghanam, Orieb Abu, et al. "An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning." Expert Systems with Applications 213 (2023): 118745

Amin Shahraki, Mahmoud Abbasi, Amir Taherkordi, Anca Delia Jurcut, A comparative study on online machine learning techniques for network traffic streams analysis,Computer Networks,Volume 207,2022,108836,ISSN 1389-1286,https://doi.org/10.1016/j.comnet.2022.108836.

Eustis, A. G. (2019). The Mirai Botnet and the importance of IoT device security. In *16th International Conference on Information Technology-New Generations (ITNG 2019)* (pp. 85-89)

J. L. Leevy, J. Hancock, T. M. Khoshgoftaar and J. M. Peterson, "An Easy-to-Classify Approach for the Bot-IoT Dataset," 2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI), Atlanta, GA, USA, 2021, pp. 172-179, doi: 10.1109/CogMI52975.2021.00031

Khanday, Shahbaz Ahmad, Hoor Fatima, and Nitin Rakesh. "Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks." Expert Systems with Applications 215 (2023): 119330

Leevy, J. L., Hancock, J., Khoshgoftaar, T. M., & Peterson, J. (2021b, December). Detecting information theft attacks in the bot-iot dataset. In 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 807-812). IEEE.

Koroniotis, Nickolaos, et al. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset." Future Generation Computer Systems 100 (2019a): 779-796.

Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset." *Future Generation Computer Systems* 100 (2019b): 779-796

M. Zeeshan et al., "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," in IEEE Access, vol. 10, pp. 2269-2283, 2022, doi: 10.1109/ACCESS.2021.3137201.

Maasaoui, Zineb, et al. "Design and Implementation of an Automated Network Traffic Analysis System using Elastic Stack." 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA). IEEE, 2023.

M. Shafiq, Z. Tian, A. K. Bashir, X. Du and M. Guizani, "CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques," in *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242-3254, 1 March1, 2021, doi: 10.1109/JIOT.2020.3002255.

Nimbalkar, Pushparaj, and Deepak Kshirsagar. "Feature selection for intrusion detection system in Internet-of-Things (IoT)." ICT Express 7.2 (2021): 177-181

Özer, Erman, Murat İskefiyeli, and Jahongir Azimjonov. "Toward lightweight intrusion detection systems using the optimal and efficient feature pairs of the Bot-IoT 2018 dataset." International Journal of Distributed Sensor Networks 17.10 (2021): 15501477211052202

Pokhrel, Satish, Robert Abbas, and Bhulok Aryal. "IoT security: botnet detection in IoT using machine learning." arXiv preprint arXiv:2104.02231 (2021).

Saba, Tanzila, et al. "Anomaly-based intrusion detection system for IoT networks through deep learning model." Computers and Electrical Engineering 99 (2022): 107810

T. Cai, T. Jia, S. Adepu, Y. Li and Z. Yang, "ADAM: An Adaptive DDoS Attack Mitigation Scheme in Software-Defined Cyber-Physical System," in IEEE Transactions on Industrial Informatics, vol. 19, no. 6, pp. 7802-7813, June 2023, doi: 10.1109/TII.2023.3240586.

URL-1,https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-5-karar-a%C4%9Fa%C3%A7lar%C4%B1-c90bd7593010

URL-2,https://www.gtech.com.tr/yapay-sinir-aglari-ve-uygulamalari/

URL-3,https://www.derinogrenme.com/2018/06/28/geri-yayilim-algoritmasina-matematiksel-yaklasim/

URL-4,https://efecanxrd.medium.com/lstm-nedir-long-short-term-memory-253db6eed147

URL-5,https://blog.cloudflare.com/ddos-threat-report-2023-q4

URL-6,https://www.f5.com/labs/articles/threat-intelligence/mirai-the-iot-bot-that-took-down-krebs-and-launched-a-tbps-attack-on-ovh-22422

URL-7,https://www.kaggle.com/datasets/vigneshvenkateswaran/bot-iot?resource=download&select=data_names.csv

ÖZGEÇMİŞ

Tuğrul YAĞBASAN, lise öğrenimi Malatya Fen Lisesi'nde tamamladıktan sonra lisans eğitimini Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği bölümünde 2010 yılında tamamlamıştır. 2012 yılından itibaren çalışma hayatına Giresun Üniversitesi Bilgi İşlem Daire Başkanlığı'nda sistem ve network sorumlusu olarak devam etmektedir.