ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ VE GENETİK ALGORİTMA KULLANILARAK ANOMALİ TABANLI SALDIRI TESPİT SİSTEMİ

Mustafa Veysel ÖZSARI

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2024

ÖZET

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ VE GENETİK ALGORİTMA KULLANILARAK ANOMALİ TABANLI SALDIRI TESPİT SİSTEMİ

Mustafa Veysel ÖZSARI

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ayhan AYDIN

Teknolojinin hızlı ilerleyişi ve online olanaklarının artışı, günlük yaşantımıza büyük etkiler getirmektedir. Artık bankacılık işlemleri, alışveriş ve daha birçok faaliyet, internet aracılığıyla gerçekleştirilebilmektedir. Ancak, bu dijital dünya, aynı oranda siber saldırılar ve kötüye kullanıma da açık hale getirmiştir. Bu nedenle, verilerimizin güvende olması için ağ güvenliğinin sağlanması son derece kritik bir öneme sahiptir.

Bilgi güvenliği alanında, yapay zeka temelli yaklaşımların kullanımı hızla popülerlik kazanmaktadır. Bu yaklaşımlar, potansiyel tehditleri tanımlama ve engelleme konusunda etkili bir araç olarak öne çıkmaktadır. Ancak, saldırıları tespit etmek amacıyla toplanan veriler genellikle çok sayıda özellik içermektedir. Bu özelliklerin etkili bir şekilde işlenmesi, veri madenciliği ve makine öğrenimi alanlarında karşılaşılan önemli bir zorluktur.

Bu çalışmanın amacı, USB-IDS-1 veri kümesindeki özellikleri genetik algoritma kullanarak azaltmak ve çeşitli sınıflandırıcılarla değerlendirmektir. Karar ağaçları, rastgele orman, kNN, Naive Bayes ve yapay sinir ağları gibi çeşitli sınıflandırıcılar kullanılmıştır. Değerlendirme kriterleri olarak doğruluk oranı, duyarlılık, kesinlik ve F1-skordan faydalanılmıştır.

Elde edilen sonuçlar, genetik algoritmanın Hulk ve Slowloris veri setlerinde oldukça başarılı olduğunu, Slowhttptest verilerinde kısmen etkili olduğunu ancak TCP kümesinde başarılı olamadığını göstermektedir. Bununla birlikte, Slowhttptest ve TCP verilerinde tüm özelliklerin kullanılması sonucunda algoritmaların performansının düşük kaldığı görülmüştür.

Ocak 2024, 46 sayfa
Anahtar Kelimeler: Saldırı Tespiti, USB-IDS-1, Genetik Algoritma, Makine Öğrenmesi.

ABSTRACT

Master Thesis

ANOMALY BASED INTRUSION DETECTION SYSTEM USING MACHINE LEARNING AND GENETIC ALGORITHM

Mustafa Veysel ÖZSARI

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Ayhan AYDIN

The rapid advancement of technology and the increasing online opportunities it brings have had a significant impact on our daily lives. Nowadays, banking transactions, shopping, and many other activities can be carried out through the internet. However, this captivating digital world has also become equally vulnerable to cyberattacks and misuse. Therefore, ensuring the security of our data through network security has become critically important.

In the field of information security, the use of artificial intelligence-based approaches has rapidly gained popularity. These approaches have emerged as effective tools for identifying and mitigating potential threats. However, data collected for intrusion detection often contains numerous features. Effectively processing these features poses a significant challenge in the realms of data mining and machine learning.

The objective of this study is to reduce the dimensionality of features in the USB-IDS-1 dataset using genetic algorithms and evaluate their performance with various classifiers. Various classifiers, including decision trees, random forests, k-NN, Naive Bayes, and artificial neural networks, were employed. Evaluation criteria encompass accuracy rate, sensitivity, precision, and F1-score.

The obtained results indicate that the genetic algorithm demonstrates considerable success in the Hulk and Slowloris datasets, exhibits partial effectiveness in the Slowhttptest data, but falls short of expectations in the TCP dataset. Nevertheless, it is important to note that even when employing all features in the Slowhttptest and TCP datasets, the overall performance of the algorithms remains suboptimal.

January 2024, 46 pages
Key Words: Intrusion Detection, USB-IDS-1, Genetic Algorithm, Machine Learning.

TEŞEKKÜR

Tez çalışmam boyunca, rehberliğini, desteğini, bilgilerini esirgemeyen danışman Hocam Sayın Dr. Öğr. Üyesi Ayhan AYDIN'a,

Tez jürimde bulunan, yorum ve önerileri ile tezimin şekillenmesine yardımcı olan sayın Prof. Dr. İman ASKERBEYLİ'ye ve Dr. Öğr. Üyesi Ayşe Nurdan SARAN'a,

Maddi manevi desteklerini esirgemeyen, her daim yanımda olan anneme, babama, ablam Dr. Şifa ÖZSARI'ya, sevgili eşim Melek'e, neşe ve motivasyon kaynağım olan biricik kızım Mihra'ya en derin duygularla teşekkür ederim.

Mustafa Veysel ÖZSARI
Ankara, Ocak 2024

İÇİNDEKİLER

SİMGELER DİZİNİ

∑	Toplam (Sum)
n	İterasyon sayısı (Time step)

Kısaltmalar

ABC	Artificial Bee Colony
CNN	Convolutional Neural Networks
COLAB	Google Colaboratory
DHR, DDoS	Dağıtılmış Hizmet Reddi (Distributed Denial of Service)
FN	False Negative
FP	False Positive
GA	Genetik Algoritma (Genetic Algorithm)
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
k-NN	k-Nearest Neighbors
MÖ, ML	Makine Öğrenmesi (Machine Learning)
NaN	Not-a-Number
Nİ, IoT	Nesnelerin İnterneti (Internet of Things)
ÖSO	Öznitelik Seçim Ölçütü
PCA	Principal Component Analysis
PSO	Particle Swarm Optimization
STS, IDS	Saldırı Tespit Sistemleri (Intrusion Detection Systems)
TCP	Transmission Control Protocol
TN	True Negative
TP	True Positive
YSA, ANN	Yapay Sinir Ağlar (Artificial Neural Network)
YZ, AI	Yapay Zeka (Artificial Intelligence)

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

1. GİRİŞ

Bilgisayar ağlarının son 10 yılda hızlı bir şekilde ilerleme göstermesi günümüz modern iletişim ve veri paylaşımının temel taşıyıcısı haline getirmiştir. Bu ilerleme ile internet, kurumsal ağlar, kablosuz ağlar ve diğer birçok teknolojik altyapının dünya genelinde faaliyet göstermesine olanak tanımıştır. Bu durum sağlık, eğitim, savunma sanayi, online alışveriş, ticaret ve daha pek çok alanda veri transferinin gerçekleşmesine, aynı zamanda girilen veri sayısının da doğru orantılı olarak artmasına neden olmuştur (Shon ve Moon 2007). Fakat, bununla beraber siber saldırılarda artış göstermiştir. Bilgisayar ağlarını hedef alan siber saldırılar hem bireysel kullanıcılar hem de kurumsal organizasyonlar için ciddi tehditler oluşturmaktadır. Siber saldırılar, bilgisayar sistemlerini ve ağları hedef alarak gizliliği ihlal edebilir, verileri çalabilir, hizmet kesintilerine neden olabilir ve hatta ulusal güvenliği tehlikeye atabilir. Bu nedenle, siber güvenlik, bilgisayar ağlarının işleyişini korumak ve tehditlere karşı savunma mekanizmalarını geliştirmek için önemli bir konu haline gelmiştir. Bu bağlamda, bilgi güvenliği için popüler araçlardan birisi olan saldırı tespit sistemleri büyük bir öneme sahiptir. STS, genel bir ifadeyle, ağ trafiğini veya bilgisayar sistemlerini izleyerek anormal aktivitelerin ve potansiyel tehditlerin tespitini yapan cihaz ya da yazılımlardır. Saldırı tespiti, genel bir ifadeyle legal olmayan kullanıcıların davranışlarının analiz edilmesi varsayımına dayanmaktadır (Stallings 2006).

Birçok alanda olduğu gibi saldırı tespit alanında da yapay zekâ ve özellikle makine öğrenmesi temelli yaklaşımların araştırılması ve uygulanması giderek artmaktadır. YZ, insan beyninin öğrenme yapısından ilham alınarak geliştirilen bir teknoloji olup, bir sistemin verileri doğru bir şekilde yorumlama, bu verilerden öğrenme ve bu bilgileri belirli hedeflere ve görevlere ulaşmak için kullanma yeteneği olarak tanımlanmıştır (Kaplan ve Haenlein 2019). YZ'nin bir alt alanı olan MÖ, ilk olarak Arthur Samuel tarafından ortaya atılmıştır (Samuel 1959). Günümüzdeki makine öğrenmesi uygulamaları genellikle eldeki verileri kullanarak bir sınıflandırıcı geliştirmek ve daha sonra bu modelleri yeni gelen veriler için tahminler üretmek amacıyla yoğun bir şekilde kullanılmaktadır. Siber güvenlik açısından bakıldığında, saldırıların hızla geliştiği ve geleneksel güvenlik önlemlerinin tek başına yetersiz kaldığı bir dönemde, YZ ve MÖ tabanlı yaklaşımlar, tehditleri daha etkili bir şekilde tespit etme ve savunma mekanizmalarını güçlendirme potansiyeli sunmaktadır (Sommer ve Paxson

2010). Literatür incelendiğinde saldırı tespiti için MÖ yaklaşımlarından faydalanılan çok sayıda çalışma yer almaktadır. Bunlar arasından (Aburomman ve Reaz 2016), (Al-Jarrah vd. 2018), (Al-Yaseen vd. 2017), (An vd. 2018), (Belavagi ve Muniyal 2016) atıfları örnek olarak gösterilebilir. Bu bağlamda, KDD Cup99[1], CAIDA (Hick vd. 2007), NSL-KDD (Tavallaee vd. 2009) veri setleri, saldırı tespiti çalışmalarının örnek veri kaynakları olarak sıkça kullanılmıştır.

Bu tez kapsamında, 2021 yılında tanıtılmış olan USB-IDS-1 (Catillo vd. 2021) veri seti üzerinde bir çalışma gerçekleştirilmiştir. USB-IDS-1, büyük boyutu ve 83 farklı niteliğe sahip olmasıyla dikkat çeken önemli bir veri setidir. Büyük boyutlu verilerin işlenmesi sırasında karşılaşılan temel sorunlar arasında donanım kısıtlamaları ve işlem süresi yer almaktadır. İşlem hızını artırmak için yüksek özelliklere sahip cihazlar kullanmak her zaman mümkün olmayabilir. Bu nedenle, bu çalışmada USB-IDS-1 veri seti üzerinde genetik algoritma (Holland 1992) kullanılarak özellik seçimi yapılmış ve daha sonra bu seçilen özellikler ile çalışan karar ağacı, rastgele orman (Breiman 2001), k-en yakın komşu (Cover ve Hart 1967), Naive Bayes ve yapay sinir ağları kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Yapılan deneyler ile GA'nın özellik seçimi üzerinde ki performansı gözlemlenmiştir.

GA bir optimizasyon algoritması olup bir dizi kısıt altında en iyi çözümü bulma veya bir hedef fonksiyonunu maksimize veya minimize etme amacı taşıyan matematiksel veya bilgisayar tabanlı tekniklerdir. Optimizasyon algoritmaları, birçok farklı uygulama alanlarında kullanılmış olup, birçok gerçek dünya problemi etkili çözümler sunmuştur.

GA bu alanda oldukça yaygın kullanılan, biyolojik evrim süreçlerini taklit ederek potansiyel çözümleri bir nesil denemesi ve doğal seçilim yoluyla iyileştirerek en iyi çözümü bulmayı amaçlayan basit bir optimizasyon yöntemidir. Bu çalışmada GA kullanılarak özellik azaltma işlemi yapılmış olup etkisi olmayan nitelikler tespit edilmiştir. Böylece yeni gelen veriler için bu özellikler kullanılmadan tahmin edebilecek bir sınıflandırma sistemi tasarlanmıştır.

[1]KDD., The 1999 KDD intrusion detection, 1999. Available: http://kdd.ics.uci.edu/databases/kddcup99/task.html.

Tezin kalan şu şekilde organize edilmiştir: İkinci bölümünde kullanılmış olan veri seti ve algoritmalar hakkında detaylı bilgi verilmiştir. Üçüncü bölümde ise parametre ayarları, yapılan deneyler ve deney sonuçları açıklanmıştır. Son olarak dördüncü bölümde alınan sonuçlar değerlendirilmiş ve sonrasında yapılabilecek çalışmalar için öneriler sunulmuştur.

2. GEREÇ VE YÖNTEMLER (MATERIAL AND METHODS)

2.1 Veri Seti (Data Set)

STS üzerinde yapılan akademik çalışmalar ve incelemeler için oldukça yaygın olarak kullanılan farklı veri setleri literatürde bulunmaktadır. Araştırmamızda, USB-IDS-1 veri seti kullanılmıştır (Catillo vd. 2021). Söz konusu veri seti, 83 farklı özelliğe sahip ve toplamda 16 sınıf içermektedir. Bu özellikler: "Flow ID, Src IP, Src Port, Dst IP, Dst Port, Protocol, Timestamp, Flow Duration, Total Fwd Packet, Total Bwd packets, Total Length of Fwd Packet, Total Length of Bwd Packet, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std, Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std, Flow Bytes/s ,Flow Packets/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min, Bwd IAT Total ,Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Length, Bwd Header Length, Fwd Packets/s, Bwd Packets/s, Packet Length Min, Packet Length Max, Packet Length Mean, Packet Length Std, Packet Length Variance, FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, CWR Flag Count, ECE Flag Count, Down/Up Ratio, Average Packet Size, Fwd Segment Size Avg, Bwd Segment Size Avg ç, Fwd Bytes/Bulk Avg, Fwd Packet/Bulk Avg, Fwd Bulk Rate Avg, Bwd Bytes/Bulk Avg, Bwd Packet/Bulk Avg, Bwd Bulk Rate Avg, Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes, FWD Init Win Bytes, Bwd Init Win Bytes, Fwd Act Data Pkts, Fwd Seg Size Min, Active Mean, Active Std, Active Max, Active Min, Idle Mean, Idle Std, Idle Max, Idle Min, Label" şeklinde verilmektedir.

Araştırma sürecinde, "Flow ID", "Fwd Header Length", "Src IP", "Src Port", "Dst IP", "Dst Port" ve "Timestamp" gibi nitelikler, aslında sınıflandırma işlemine katkı sağlamayan ve veri seti üzerinde meta veri görevi gören özelliklerdir. Bu nedenle, sınıflandırma ve özellik seçimi sonuçlarını etkilememek adına, bu özellikler veri setinden çıkarılmıştır. Ayrıca, veri setinde NaN değerler bulunan satırlar da veri setinden çıkarılmıştır. Bu şekilde ön işlem

uygulanan veri seti, çalışmanın temel veri kaynağı olarak kullanılmıştır. Sınıf etiketleri içerisinde savunma modülünü temsil eden gruplar da bulunmaktadır. Bu gruplar aşağıdaki gibi sınıflandırılmıştır:

- Hulk-NoDefense
- Hulk-Reqtimeout
- Hulk-Evasive
- Hulk-Security2
- TCPFlood-NoDefense
- TCPFlood-Reqtimeout
- TCPFlood-Evasive
- TCPFlood-Security2
- Slowhttptest-NoDefense
- Slowhttptest -Reqtimeout
- Slowhttptest -Evasive
- Slowhttptest -Security2
- Slowloris-NoDefense
- Slowloris-Reqtimeout
- Slowloris-Evasive
- Slowloris-Security2

Burada "–" işaretinden önceki kısım saldırı tipini, sonraki kısım ise savunma modelini ifade etmektedir. Saldırı tipleri kısaca aşağıdaki gibi açıklanabilir:

- Hulk: Bu saldırı türü bir web sitesine veya uygulamaya karşı gerçekleştirilen bir tür DDoS saldırısıdır. Bu tür bir saldırıda, birçok istemci tarafından eşzamanlı olarak istekte bulunulan web sunucusu veya uygulama, isteklerin yoğunluğu nedeniyle yanıt veremeyebilir ve bu da hizmet kesintisine yol açabilir. Genellikle çok sayıda bot veya köle bilgisayarlar kullanılarak gerçekleştirilen bu saldırıda, sunucuya çok sayıda istek gönderilerek sunucunun kaynaklarını tüketmek ve böylece gerçek kullanıcıların erişimini engellemek amaçlanmaktadır.
- TCPFlood: Bir bilgisayar veya ağa, özellikle bir ağdaki TCP bağlantılarını aşırı yükleyen bir DDoS saldırı türüdür. Bu tür saldırılar, birçok TCP bağlantısının açılması ve hızlıca kapatılması yoluyla gerçekleştirilir. Bu bağlantıların açılıp

kapatılması sunucunun kaynaklarını tüketir ve hizmet kesintisine yol açabilir. TCPFlood saldırıları, saldırganlar tarafından aynı anda çok sayıda sahte TCP bağlantısı isteği göndererek veya mevcut TCP bağlantılarını tüketerek gerçekleştirilebilir. Bu, sunucunun kaynaklarını tüketir ve sunucunun yanıt veremez hale gelmesine neden olabilir. Saldırganlar bu tür saldırıları, hedef sunucunun ağ bant genişliğini aşırı yükleyerek, servis reddi durumu oluşturarak veya hedef sunucunun işlemci ve bellek kaynaklarını tüketerek gerçekleştirebilirler.

- Slowhttptest: Bu tür saldırılar, sunucunun kaynaklarını tüketmek veya sunucunun hizmetini yavaşlatmak amacıyla özellikle HTTP trafiği üzerinde gerçekleştirilir. Saldırganlar, sunucuya uzun süreli istekler gönderir ve yanıt almadan veya eksik yanıtlarla sunucunun kaynaklarını tüketirler. Bu, sunucunun hizmetini yavaşlatır ve kullanıcıların web sitesine erişimini engelleyebilir.
- Slowloris: Bu saldırı türü web sunucularına karşı gerçekleştirilen bir tür DDoS saldırısıdır. Sunucunun kaynaklarını aşırı yüklemeyi amaçlayarak sunucuyu hizmet veremez hale getirmeyi hedefler. Slowloris saldırısı, sunuculara çok sayıda yavaş bağlantı açarak erişimi kesebilir ve kullanıcı deneyimini olumsuz etkileyebilir.

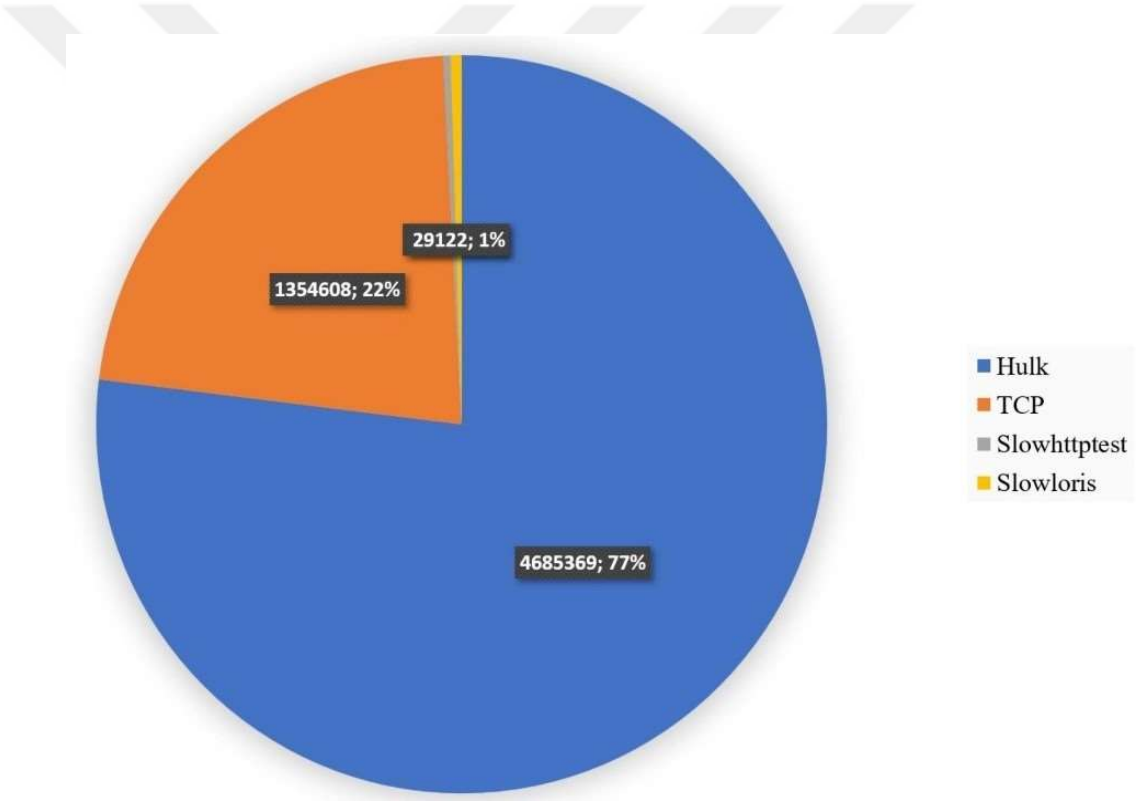
Savunma tipleri ise şu şekilde ifade edilebilir:

- Reqtimeout: Bir web sunucusunu Slowloris ve benzeri yavaş saldırılara karşı korumak için kullanılan bir savunma yöntemidir. Reqtimeout, gelen isteklerin işlenme sürelerini sınırlayan bir mekanizma sağlar. Bu, sunucunun belirli bir isteği kabul etme ve yanıtlama süresini kontrol etmesine yardımcı olur. Eğer bir istek belirli bir süre içinde işlenmezse veya yanıt verilmezse, bu istek reddedilir ve sunucu kaynakları serbest bırakılır.
- Evasive: Bu savunma türü, Hulk ve benzeri çok sayıda istekle sunucu kaynaklarını tüketerek kullanılamaz hale getirmeyi amaçlayan saldırılara karşı korunmak üzere geliştirilmiş bir mekanizmadır. Bu teknik, gelen HTTP isteklerini izler ve şüpheli IP adreslerini ve benzeri faaliyetleri tespit eder. Örneğin, aynı sayfaya kısa süre içerisinde ardışık olarak gönderilen çok sayıda isteği şüpheli aktivite olarak algılar ve bu tür olaylar algılandığında, 403 HTTP hata kodu ile yanıt verilerek şüpheli IP adreslerini belirli bir zaman dilimi boyunca kara listeye alınmasını sağlar (Catillo vd. 2020). Bu, sunucu kaynaklarının kötü niyetli kullanımını sınırlar ve sunucunun

performansını korur.

- Security2: Bu savunma modeli, web uygulamalarının güvenliğini artırmak ve saldırılara karşı korumak amacıyla kullanılan bir ModSecurity2 sunucusu modülüdür. Bu modül, bir dizi özelleştirilebilir kural ve filtre ile donatılmış olup bilinen saldırı yapılarına karşı koruma sağlar. Böylece kötü niyetli istemcilerin web sunucusuna erişimini sınırlar.

Bu veri setinde tüm gruplardan eşit sayıda veri bulunmadığından dolayı Hulk, TCPFlood, Slowhttptest ve Slowloris saldırı tipleri ayrı ayrı grup olarak alınmıştır. **Şekil 2.1**'de veri setindeki grupların dağılımı gösterilmiştir.



Şekil 2.1 Veri dağılımı (Data distribution)

Literatürde bu veri seti üzerinde yapılan olan çalışmalar incelenmiş olup, az sayıda araştırma bulunmuştur. Bu yayınlardan ilki, Catillo ve diğerleri tarafından gerçekleştirilen (Catillo vd. 2022) çalışmasıdır. Burada yazarlar, veri seti üzerinde karar ağacı, rastgele orman ve derin sinir ağları gibi algoritmalar kullanarak deneyler gerçekleştirmiştir. Eğitim işlemi başka veri

kümeleri kullanılarak gerçekleştirilmiş, test işlemi ise bu özel veri seti üzerinde uygulanmıştır.

Diğer önemli bir çalışma, Kalutharage ve arkadaşları tarafından yürütülen (Kalutharage vd. 2022) çalışmasıdır. Bu çalışmada, IoT güvenlik izlemesi için derin otomatik kodlayıcı modellerini açıklayıcı yapay zekâ ile birleştiren yeni bir yaklaşım sunulmuştur. Bu yaklaşımın temel amacı, MÖ tabanlı saldırı tespit sistemlerinin güvenilirliğini ve kesinliğini doğrulamaktır. Önerilen yöntem, USB-IDS-1 veri seti üzerinde test edilmiştir.

2.2 Makine Öğrenmesi (Machine Learning)

Makine Öğrenmesi kavramı, yapay zekâ alanının bir alt dalıdır ve bilgisayarların verileri analiz edip örüntüleri tanımasını ve öğrenmesini sağlayan bir dizi algoritma ve istatistiksel teknikler bütününü ifade eder. Makine öğrenimi, bilgisayarların görevleri yerine getirmeleri için programlamak yerine, verilerden öğrenerek ve deneyimlerden bilgi çıkararak işlevlerini geliştirebilmelerini amaçlar. Makine öğrenmesi sınıfları, temel öğrenme yaklaşımları ve problemlerine dayalı olarak çeşitlilik gösterir.

- Denetimli Öğrenme (Supervised Learning): Denetimli öğrenme, etiketli verilerle çalışır. Bu tür veriler, girdi ve hedef çıktılar arasındaki ilişkiyi gösterir. Temel amaç, bir modelin girdiyi alıp doğru çıktıyı üretebilmesini sağlamaktır. Örnekler arasında sınıflandırma (classification) ve regresyon (regression) problemleri bulunur.
- Denetimsiz Öğrenme (Unsupervised Learning): Denetimsiz öğrenme, etiketsiz verilerle çalışır ve veri içindeki doğal yapıları keşfetmeye odaklanır. Bu tür öğrenme, kümeleme (clustering) ve boyut azaltma (dimensionality reduction) problemleri gibi alanları içerir.
- Takviyeli Öğrenme (Reinforcement Learning): Takviyeli öğrenme, bir ajanın bir ortamda belirli bir görevi en iyi şekilde yerine getirmeyi öğrendiği bir öğrenme yaklaşımıdır. Ajan, çevresiyle etkileşimde bulunur ve ödüller veya cezalar alır.

Ödülleri maksimize etmek için ajan, eylem-strateji (action-policy) öğrenme yoluna gider.

Bu çalışmada denetimli öğrenme yaklaşımı ile çalışan karar ağaçları, rastgele orman (Breiman 2001), k-en yakın komşu (Cover ve Hart 1967), Naive Bayes ve yapay sinir ağları yöntemleri kullanılmıştır.

Karar ağacı hem sınıflandırma hem de regresyon görevleri için kullanılan en güçlü denetimli öğrenme algoritmalarından biridir. Akış şemasına benzer bir ağaç yapısı oluşturur; her iç düğüm bir özniteliğe yapılan bir testi temsil eder, her dal testin sonucunu gösterir ve her yaprak düğüm, bir sınıf etiketi içerir. Ağaç, eğitim verilerini, durdurma kriterine (ağacın maksimum derinliği veya bir düğümü bölme için gereken minimum örnek sayısı gibi) ulaşılıncaya kadar öznitelik değerlerine göre alt kümelere böler. Eğitim sırasında, karar ağacı algoritması, veriyi bölme işlemini entropy veya Gini impurity gibi ölçüm yöntemi kullanarak gerçekleştirir. Bu ölçümler, alt kümelerdeki belirsizlik veya rastlantı düzeyini ölçer. Amaç, bölme sonrası bilgi kazancını veya belirsizlikteki azalmayı maksimize eden özniteliği bulmaktır. Bu süreç, her türetilmiş alt küme üzerinde tekrarlanan, recursive partitioning olarak adlandırılan rekürsif bir şekilde gerçekleştirilir. Rekürsiyon, bir düğümdeki alt kümenin hedef değişkenin aynı değerine sahip olduğunda veya bölme artık tahminlere değer katmadığında tamamlanır. Karar ağaçları yüksek boyutlu veriler üzerinde etkili sonuçlar çıkarabilir.

Entropi, veri kümesindeki rastgelelik veya belirsizlik derecesini ölçen bir metriktir. Sınıflandırmada, sınıf etiketlerinin veri kümesindeki dağılıma dayalı olarak rastgeleliği ölçer. İlk veri kümesinin bir alt kümesi için entropi, i^{th} düğümündeki K sayısındaki sınıflar için şu şekilde tanımlanabilir:

$$H(x) = -\sum_1^n p(x_i) \log_2 p(x_i) \quad (2.1)$$

Gini belirsizliği, sınıflandırılmış gruplar arasındaki bölünmenin ne kadar doğru oluğunu değerlendiren bir skordur. Gini belirsizliği, 0 ile 1 arasında bir skoru değerlendirir; 0, tüm gözlemlerin bir sınıfa ait olduğu durumu, 1 ise sınıflar içindeki elemanların rastgele

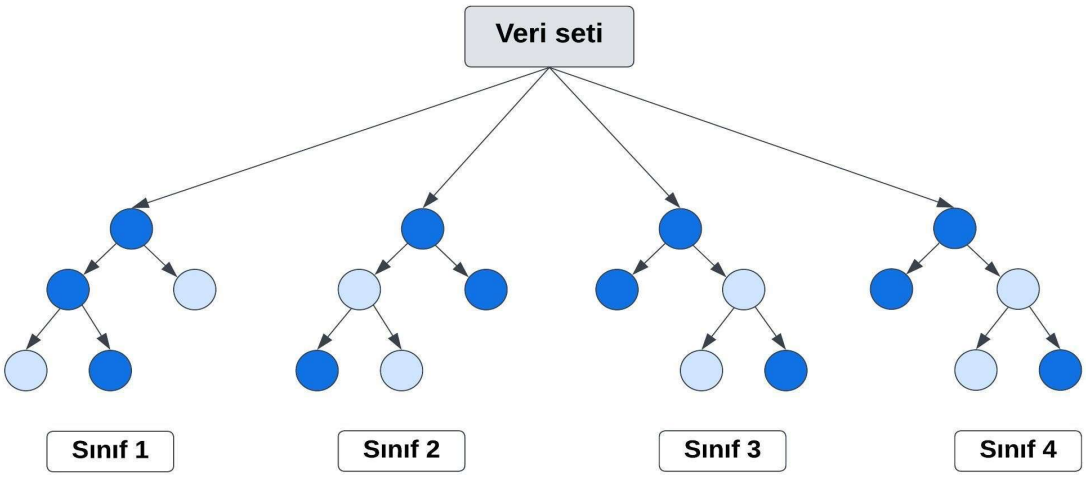
dağılımını temsil eder. Bu durumda, mümkün olan en düşük Gini indeks skoruna sahip olması beklenmektedir. Gini indeksi, karar ağacı modelimizi değerlendirmek için kullanacağımız değerlendirme metriğidir. Gini belirsizliği aşağıdaki formül ile tanımlanabilir.

$$Gini = 1 - \sum_{i=1}^{j} p(i)^2 \quad (2.1)$$

Rastgele ormanlar, makine öğrenimi paradigması içerisinde sınıflandırma ve regresyon problemlerini çözmek amacıyla tasarlanmış bir ensemble öğrenme yöntemidir (Resende ve Drummond 2018). Temelde, birbirinden bağımsız çok sayıda karar ağacının birleştirilmesi prensibine dayanır ve genellikle yüksek boyutlu ve karmaşık veri setlerinde etkili performans sağlar.

Her bir karar ağacı, rastgele seçilmiş özellikler ve veri noktaları üzerinde eğitilir. Bu rastgele seçim, her ağacın birbirinden bağımsız olmasını sağlar. Ağaçların eğitim sürecinde, bilgi kazancını maksimize etmek ve aşırı uyuma karşı dirençli modeller elde etmek amacıyla, her iç düğümde bir alt küme oluşturmak için özellikler rastgele seçilir. Tahmin aşamasında, her bir karar ağacının verdiği sınıflandırma veya regresyon tahminleri bir araya getirilir.

Sınıflandırma durumunda genellikle oy çokluğu ilkesi kullanılır, regresyon durumunda ise tahminlerin ortalaması alınır. Rastgele ormanlar, özellikle veri setlerinin çok boyutlu ve gürültülü olduğu durumlarda etkili olup, aşırı uyma eğilimini azaltarak genel model performansını artırma özelliği ile bilinir. Bu algoritma, otomatik özellik seçimi, açıklanabilirlik ve dayanıklılık gibi avantajlar sağlayarak birçok uygulama alanında tercih edilen bir makine öğrenimi yaklaşımıdır. **Şekil 2.2**'de örnek bir orman yapısı gösterilmiştir.



Şekil 2.2 Rastgele orman (Random forest)

k-NN algoritması, örnek tabanlı bir makine öğrenimi yaklaşımıdır ve sınıflandırma ile regresyon problemlerini çözmek amacıyla kullanılan en basit ve geleneksel yöntemlerden birisidir (Bishop 1995), (Manocha ve Girolami 2007). Temel fikir, bir veri noktasının sınıflandırılması veya değerlendirilmesi için ona en yakın komşularının etrafındaki etiket veya değerleri kullanmaktır.

Eğitim aşamasında, veri setindeki her bir örnek, öznitelik değerleri ve hedef etiket veya değeri ile birlikte temsil edilir. Eğitim aşamasında, algoritma veri setini ezberlemez, sadece veri noktalarını içeren uzayı temsil eder. Tahmin aşamasında, bir test veri noktasının sınıflandırılması veya değerlendirilmesi istendiğinde, bu noktanın uzaydaki konumu belirlenir. Belirlenen konumda, k-NN algoritması en yakın k adet eğitim veri noktasını bulur. Sınıflandırma problemlerinde, en yakın komşuların sınıfları arasında çoğunluk oyu kullanarak test noktasının sınıfını belirler. Regresyon problemlerinde, en yakın komşuların değerlerinin ortalaması veya ağırlıklı ortalaması kullanılarak test noktasının tahmini değeri hesaplanır.

Parametre k değeri, en yakın komşuların sayısını temsil eder. Bu parametre, algoritmanın performansını etkiler. Küçük bir k değeri, modele daha fazla esneklik katar ancak gürültüye daha hassas hale getirebilir. Büyük bir k değeri, daha düzenli ve genelleştirilebilir modellere yol açabilir, ancak lokal örüntüleri kaçırma eğiliminde olabilir. k-NN, basit ve anlaşılır bir yapısı olması ve özellikle küçük boyutlu veri setlerinde etkili

sonuçlar vermesi nedeniyle tercih edilen bir algoritmadır. Ancak büyük boyutlu veri setlerinde ve yüksek boyutlu uzaylarda performans sorunlarına neden olabilir.

En yakın komşuların bulunması için çok sayıda mesafe ölçüm metrikleri bulunmaktadır. En çok kullanılan mesafe ölçüm metrikleri, Öklid, Minkowski ve Manhattan yöntemleridir. Öklid mesafe ölçümü aşağıdaki denklem ile ifade edilebilir:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.3}$$

Minkowski mesafe ölçümü aşağıdaki denklem ile ifade edilebilir:

$$D = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p} \tag{2.4}$$

Manhattan mesafe ölçümü aşağıdaki denklem ile ifade edilebilir:

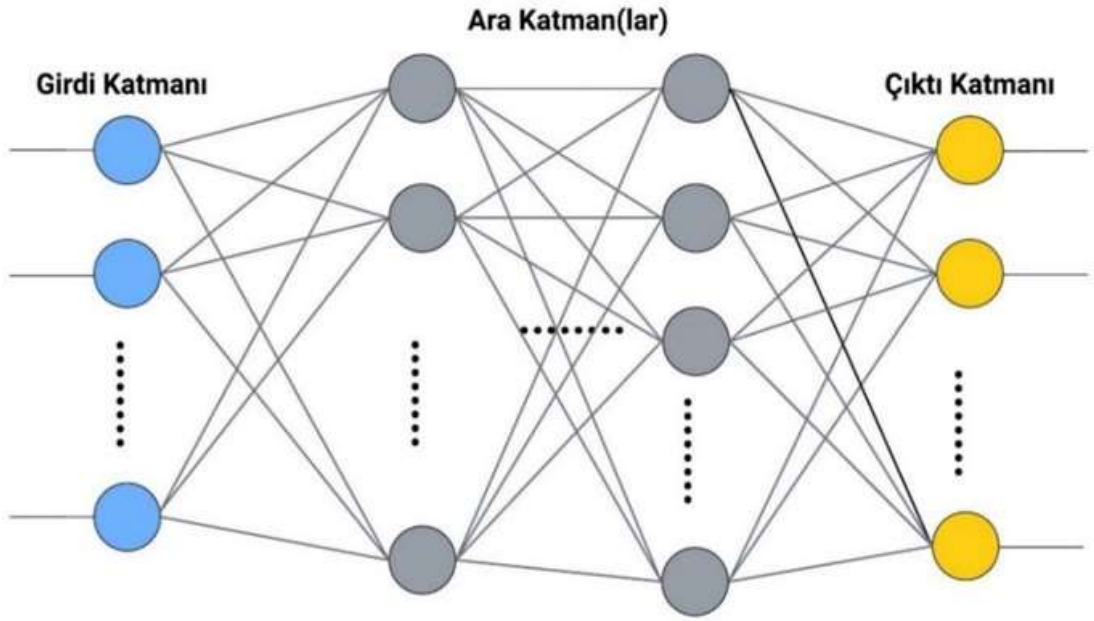
$$D = \sum_{i=1}^{n} |x_i - y_i| \tag{2.5}$$

Naive Bayes sınıflandırma algoritması, adını matematikçi Thomas Bayes'ten alan bir sınıflandırma ve kategorizasyon algoritmasıdır. Bu algoritma, olasılık temellerine dayalı olarak tanımlanmış bir dizi hesaplama kullanarak, sisteme sunulan verilerin sınıfını veya kategorisini belirlemeyi amaçlamaktadır. Bayes teoremi matematiksel olarak aşağıdaki denklemle ifade edilir:

$$P(A|B) = \frac{P(B|A)*P(A)}{P(B)} \tag{2.6}$$

Yapay sinir ağları, sinir biliminden ilham alarak oluşturulan, paralel ve dağınık hesaplama mimarilerini temel alan bir bilgisayar bilimi alanıdır (Haykin 1999). YSA'lar, karmaşık bilişsel görevleri gerçekleştirebilen adaptif sistemler olarak tasarlanmıştır. Temel yapıları, matematiksel işlemleri gerçekleştiren yapay nöronlar ve bu nöronlar arasındaki ağırlıklı bağlantılardan oluşur. Bu bağlantılar, öğrenme sürecinde uyarlanabilir ve ağırlıkların optimize edilmesi, modelin belirli bir görevi başarmasına olanak tanır.

Yapay sinir ağları genellikle giriş, gizli ve çıkış katmanlarından oluşan hiyerarşik bir yapıya sahiptir. Girdi katmanı, sistem tarafından işlenmek üzere dış dünyadan gelen verileri içerir. Gizli katmanlar, bu girdileri daha yüksek düzeyde temsil etmek için kullanılır ve çıkış katmanı, belirli bir görevi gerçekleştirmek için nihai sonuçları üretir. YSA' da genellikle öğrenme süreçlerinde kullanılan geriye yayılım (backpropagation) gibi optimizasyon algoritmaları kullanılır. Bu algoritmalar, ağın çıktılarını gerçek etiketlerle karşılaştırarak hata fonksiyonunu minimize etmeye yönelik ağırlık güncellemelerini gerçekleştirir. Bu süreç, ağın girdi-çıkış ilişkilerini anlamasına ve genelleme yeteneğini artırmasına olanak tanır. YSA, geniş bir uygulama yelpazesiyle birlikte, özellikle derin öğrenme modelleri aracılığıyla karmaşık görevlerde yüksek performans elde etmek amacıyla yaygın olarak kullanılmaktadır. **Şekil 2.3**'de örnek bir YSA modeli gösterilmiştir.



Şekil 2.3 Temel bir YSA yapısı (A basic ANN structure)

2.3 Genetik Algoritma (Genetic Algorithm)

Genetik algoritma, evrimsel hesaplama alanında ortaya çıkan ve biyolojik evrimin temel prensiplerine dayanan bir optimizasyon ve arama tekniğidir (Koza 1992). Bu algoritma,

doğal seçilim, çaprazlama, mutasyon ve popülasyon tabanlı yaklaşımları içeren evrimsel işlemleri taklit ederek çözüm uzayında en iyi sonucu bulma amacı taşır. Genetik algoritmanın temel bileşenleri şunlardır:

- Birey (Chromosome): Potansiyel bir çözümü temsil eden genetik algoritmanın temel birimidir. Birey, genetik materyali içerir, genellikle bir dizi gen veya parametre kümesi şeklinde temsil edilir.
- Popülasyon (Population): Bir dizi bireyin bir araya gelmesiyle oluşan kümedir. Algoritmanın her iterasyonunda popülasyon, çaprazlama veya mutasyon gibi çeşitli genetik operatörler aracılığıyla evrilir.
- Doğal Seçilim (Natural Selection): Her iterasyonun sonunda, popülasyondaki bireylerin uygunluğuna dayanarak bir seçim yapılır. Uygunluk, çözümün belirli bir hedefe ne kadar yakın olduğunu belirten bir değerlendirmedir.
- Çaprazlama (Crossover): Seçilen bireyler arasında genetik materyalin değiştirilmesini sağlayan bir operatördür. İki birey arasında rastgele bir noktada kesilerek yeni bireyler oluşturulur.
- Mutasyon (Mutation): Popülasyon içindeki bireylerin genetik materyalinde rastgele değişiklikler yaparak çeşitliliği artıran bir işlemdir.

Genetik algoritma, karmaşık arama ve optimizasyon problemlerini çözmek, genetik programlama, öznitelik seçimi, planlama ve tasarım gibi birçok alanda kullanılmaktadır. Bu algoritma, büyük ve çok boyutlu arama uzaylarında etkili bir şekilde çalışabilme özellikleri ile bilinir. Genetik Algoritmanın başlangıcında, belirli bir problem alanına uygun şekilde rastgele seçilmiş bireylerden oluşan bir popülasyon oluşturulur. Bu bireyler genellikle çözüm uzayındaki potansiyel çözümleri temsil eden genetik yapıları içerir. Sonlanma şartı sağlanana kadar, popülasyondaki bireylerin problem için belirlenen amaç fonksiyonu (fitness fonksiyonu) kullanılarak fitness değerleri hesaplanır. Fitness değeri, bir bireyin çözümünün ne kadar başarılı olduğunu gösteren bir ölçüdür. Elitizm adı verilen bir strateji kullanılarak, en iyi fitness değerine sahip bireyler doğrudan bir sonraki nesle aktarılır ve böylece bu bireylerin genetik materyali korunur.

Çaprazlama işlemi, seçilmiş bireyler arasında genetik materyal değişimi gerçekleştiren bir süreçtir. Belirli bir çaprazlama yöntemine göre, iki birey arasında genlerin yer değiştirmesi

yapılır ve yeni bireyler oluşturulur. Bu çaprazlama işlemi, popülasyondaki çeşitliliği artırmaya ve potansiyel çözümleri keşfetmeye yardımcı olur. Belirli bir olasılıkla uygulanan mutasyon işlemi, popülasyondaki bireylerin genetik materyalinde küçük rastgele değişiklikler yapar. Mutasyon, popülasyondaki çeşitliliği sürdürmeye yardımcı olur ve çözüm uzayında daha geniş bir alanı keşfetme olasılığını artırır. Ancak, genellikle düşük bir oranda uygulanır, çünkü yüksek mutasyon oranları iyi çözümlerin kaybedilmesine yol açabilir. Genetik algoritma, belirli bir duruma ulaşıldığında sonlanır. Sonlanma şartları, belirli bir iterasyon sayısı, fitness eşiği veya çözüm doğruluğu gibi kriterlere dayanabilir. Bu şekilde, GA problem alanına uygun bir şekilde uyarlanabilir ve çeşitli optimizasyon problemlerine etkili bir şekilde uygulanabilir.

3. TARTIŞMA (DISCUSSIONS)

Bu bölümde, parametre ayarları, gerçekleştirilen deneyler ve elde edilen sonuçlara dair bilgiler sunulmaktadır. Deneylerin tamamı, Google Research tarafından geliştirilen ve çevrimiçi olarak Python kodlarının yazılmasına ve çalıştırılmasına olanak tanıyan Google Colaboratory ortamında gerçekleştirilmiştir. COLAB, araştırmacılara GPU gibi donanım kaynaklarına erişim sağlama ve birçok kütüphaneyi otomatik olarak kullanma imkânı tanıyan bir platformdur. Bu nedenle, özellikle makine öğrenmesi çalışmaları için oldukça kullanışlıdır.

3.1 Metrikler ve Parametre Ayarları (Metrics and Parameter Settings)

Yöntemlerin performansını değerlendirmek amacıyla doğruluk oranı, duyarlılık, kesinlik ve F1-skor gibi ölçütler kullanılmıştır. Bu metriklere ait formüller sırasıyla eşitlik (3.1), (3.2), (3.3) ve (3.4)'te sunulmuştur:

$$Doğruluk = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

$$Kesinlik = \frac{TP}{TP+FP} \tag{3.2}$$

$$Duyarlılık = \frac{TP}{TP+FN} \tag{3.3}$$

$$F1 = \frac{2\ x\ Duyarlılık\ x\ Kesinlik}{Duyarlılık + Kesinlik} \tag{3.4}$$

Eşitlik (3.1), (3.2), (3.3) ve (3.4) için kullanılan TP, FP, TN, FN terimleri genellikle bir sınıflandırma modelinin performansını değerlendirmek amacıyla kullanılan kavramlardır. Bu terimleri daha detaylı bir şekilde aşağıdaki gibi açıklanabilir:

- True Positive (TP): Modelin doğru bir şekilde pozitif olarak sınıflandırdığı durum

sayısını temsil eder. Yani, gerçekte pozitif olan örnekleri doğru bir şekilde tanımlamış olan durumdur.

- False Positive (FP): Modelin yanlış bir şekilde pozitif olarak sınıflandırdığı durum sayısını temsil eder. Yani, gerçekte negatif olan bir örneği pozitif olarak yanlış bir şekilde tanımlamış olan durumdur.
- True Negative (TN): Modelin doğru bir şekilde negatif olarak sınıflandırdığı durum sayısını temsil eder. Yani, gerçekte negatif olan örnekleri doğru bir şekilde tanımlamış olan durumdur.
- False Negative (FN): Modelin yanlış bir şekilde negatif olarak sınıflandırdığı durum sayısını temsil eder. Yani, gerçekte pozitif olan bir örneği negatif olarak yanlış bir şekilde tanımlamış olan durumdur.

Çalışmada kullanılan yöntemlere ait parametreler yapılan ön deneyler ile belirlenmiştir. Test edilen parametre değerleri arasında şunlar yer almaktadır:

- GA için:
  - Popülasyon sayısı: 50, 100, 200, 300, 400, 500
  - İterasyon sayısı: 50, 100, 150, 200
  - Mutasyon oranı: 0.1, 0.2, 0.3
- Rastgele orman:
  - Ağaç sayısı: 25, 50, 100, 150
- k-NN
  - En yakın komşu: 10, 25, 50, 75, 100, 125, 150
  - Mesafe ölçümü: Öklid, Minkowski ve Manhattan
- YSA
  - Ara katman: 1, 2, 3, 4, 5
  - Ara katmandaki düğüm sayısı: 50, 100, 150, 200, 250, 300
  - İterasyon sayısı: 100, 200, 300, 400, 500
  - Aktivasyon fonksiyonu: Sigmoid, tanh, identity

Deneyler sonucunda parametre ayarları şu şekilde yapılmıştır:

- Label encoding: Temelde sayısal olmayan verilerin sayısal verilere dönüştürülmesi için bu yöntem tercih edilmiştir.

- GA: Popülasyon sayısı 300, iterasyon sayısı 100 ve mutasyon oranı 0,1 olarak ayarlanmıştır. En iyi 2 birey bir sonraki nesle aktarılarak elitizm yapılmıştır. Turnuva seçim yöntemi, iki nokta çaprazlama ve değer değişimi mutasyon yöntemleri kullanılmıştır.
- Karar ağaçları: Gini formülü kullanılarak kök düğüm belirlenmiştir.
- Rastgele orman: Ağaç sayısı 100 olarak alınmıştır.
- k-NN: En yakın 100 komşuya bakılmıştır.
- YSA: Solver olarak ADAM (Kingma ve Jimmy 2014), aktivasyon fonksiyonu olarak "identity" kullanılmıştır. 4 ara katman (her birisinde 100-500 arası düğüm) eklenmiştir.

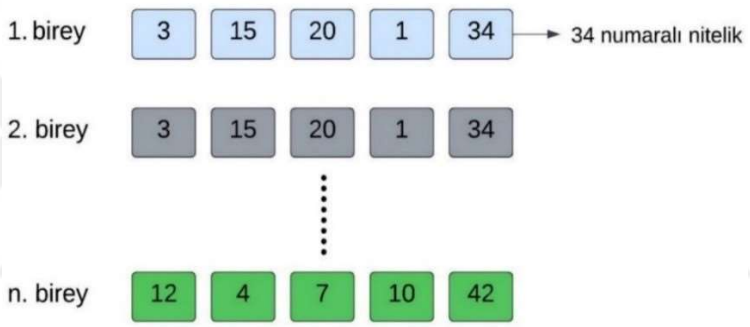
Burada değinilmesi gereken önemli bir nokta parametre ayarında çalışma süreside dikkate alınmıştır. Örneğin GA'da popülasyon sayısının büyük olması veya YSA'da ara katman ve ara katmandaki nöron sayısının fazla olması daha etkili sonuç sağlayabilir. Fakat aynı zamanda hesaplama maliyetinin de yüksek olmasına neden olmaktadır.

3.2 Deneyler (Experiments)

Parametre değerleri belirlendikten sonra, ilk aşamada, herhangi bir özellik seçimi yapılmadan sınıflandırma işlemi gerçekleştirilmiştir; yani algoritmalar, tüm nitelikleri kullanarak sınıflandırma yapmıştır. Daha sonra, Genetik Algoritmanın özellik azaltma yeteneği incelenmiştir. İlk aşamada, her saldırı tipi için ayrı ayrı deneyler gerçekleştirilmiştir. Bu deneylerin temel sebebi, veri sayılarının eşit olmamasıdır ve **Şekil 2.1'de** veri dağılımı gösterilmiştir. Burada, sınıflara ait veri sayıları oldukça dengesizdir. Makine öğrenmesi temelli yaklaşımlarda, veri dağılımının dengeli olması önemlidir çünkü veri dengesizliği, algoritmaların yanlış çıkarımlar yapmasına neden olabilir. Örneğin, burada "Hulk" grubu baskındır ve sınıflandırmada diğer kategorilerin etkisini azaltabilir. Bu nedenle, her sınıf için kendi veri sayısı dikkate alınarak dengeli bir veri seti oluşturulmuş ve ayrı ayrı deneyler yapılmıştır. Ardından, tüm grupların bir arada olduğu bir yaklaşımın performansını gözlemlemek için en düşük veri sayısı temel alınarak her gruptan eşit sayıda veri alınarak

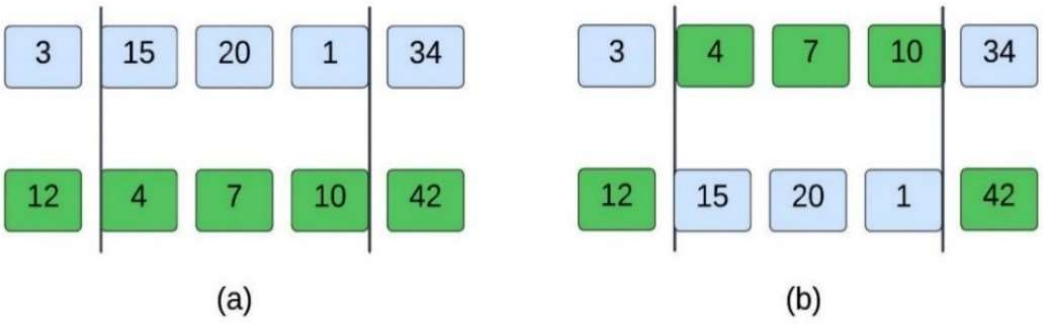
aynı deneyler tekrar edilmiştir. Burada, sadece 10 özellik için sonuçlar sunulmuştur. Ancak, veri sayısının azlığı ve sınıf sayısının 4 katına çıkması nedeniyle 5 özellik için elde edilen sonuçlar olumsuz etkilenmiş ve çok düşük değerler elde edilmiştir. Bu nedenle, tabloda 5 nitelik için elde edilen çıktılara yer verilmemiş, sadece 10 özellik için elde edilen sonuçlara odaklanılmıştır.

Özellik seçiminde ilk olarak nitelik sayısı 5 olarak belirlenmiştir. 5'te çok düşük sonuç elde edilmesi durumunda 10 için algoritmaların performans incelemesi yapılmıştır. Genetik Algoritma' ya uygun olarak bir popülasyon **Şekil 3.1'de** gösterildiği gibi oluşturulmuştur. Burada, özellik sayısının 5'e indirilmesine ait örnek bireyler gösterilmiştir. Bundan dolayı bir birey 5 tane genden oluşmaktadır. Her bir gen bir niteliğe karşılık olarak verilmiş bir sayıdır.



Şekil 3.1 Örnek popülasyon (An example of population)

Her bir birey için fitness değeri hesaplanmakta, daha sonra turnuva yöntemi ile seçilmiş olan bireylere çaprazlama yapılmaktadır. **Şekil 3.2'de** örnek bir çaprazlama gösterilmiştir.



Şekil 3.2 Çaprazlama (a) ata birey (b) çocuk birey (Crossover (a) parents (b) children)

Şekil 3.2 (a) daki iki ata bireyin çizgi olarak belirtilen aralıktaki genleri yer değiştirilerek **Şekil 3.2 (b)** deki yeni iki çocuk birey elde edilmiştir. Daha sonra mutasyon aşamasına geçilmektedir. Değer değişimi mutasyon sürecinin tamamlanmasından sonra yeni popülasyon elde edilmektedir. Bu işlemler (elitizm, çaprazlama, mutasyon) 100 iterasyon sayısına kadar sırasıyla yapılmaktadır. Algoritmanın sonlanması ile en başarılı birey problemin çözümü olarak alınmaktadır.

Çizelge 3.1, 3.2, 3.3 ve 3.4' te her sınıf için ayrı ayrı sırasıyla Hulk, TCP, Slowhttptest, Slowloris için özellik azaltma işlemi yapılmadan ve özellik azaltma işlemi yapılarak elde edilen sonuçlar verilmiştir. **Çizelge 3.5'** de ise tüm grupların bir arada olduğu deneylere ait çıktılar sunulmuştur.

Çizelge 3.1 Hulk için deney sonuçları (Experimental results for Hulk)

Özellik sayısı	**Sınıflandırıcı**	**Doğruluk oranı**	**Duyarlılık**	**Kesinlik**	**F1-skor**
5	Karar ağacı	0,71	0,72	0,72	0,72
Tüm	Karar ağacı	0,72	0,73	0,73	0,73
5	Rastgele orman	0,72	0,73	0,73	0,73
Tüm	Rastgele orman	0,73	0,73	0,74	0,74
5	k-NN	0,65	0,65	0,66	0,65
Tüm	k-NN	0,66	0,66	0,68	0,66
5	Bayes	0,71	0,72	0,72	0,72
Tüm	Bayes	0,72	0,73	0,73	0,73
5	YSA	0,62	0,62	0,64	0,61
Tüm	YSA	0,64	0,63	0,65	0,63

Çizelge 3.1 incelendiğinde hem tüm özellikler alınarak hem de özellik azaltma bakımından en iyi sonucu tüm metriklerde rastgele orman algoritmasının verdiği görülmektedir. 0,72 ve üzeri sonuçlar ile GA'nın özellik azaltmada başarılı olduğu görülmektedir. Sınıflandırma başarısına etkisi bakımından en etkili 5 özellik:

- Down/up ratio

- Fwd IAT Min
- Bwd IAT Mean
- Bwd IAT Max
- Bwd Packet Length Mean olarak bulunmuştur.

Diğer yaklaşımlar incelendiğinde karar ağacının ve Bayes' inde rastgele orman kadar başarılı olduğu görülmektedir. Daha sonra, k-NN gelmekte ve en az etkili yöntem ise YSA olmuştur. Tablodaki tüm değerler incelendiğinde %61 ve üzeri sonuçlar kabul edilebilirdir.

Çizelge 3.2 TCPFlood için deney sonuçları (Experimental results for TCP)

Özellik sayısı	**Sınıflandırıcı**	**Doğruluk oranı**	**Duyarlılık**	**Kesinlik**	**F1-skor**
5	Karar ağacı	0,25	0,25	0,25	0,25
10	Karar ağacı	0,25	0,26	0,25	0,25
Tüm	Karar ağacı	0,25	0,25	0,25	0,25
5	Rastgele orman	0,26	0,26	0,26	0,26
10	Rastgele orman	0,25	0,25	0,25	0,25
Tüm	Rastgele orman	0,25	0,25	0,25	0,25
5	k-NN	0,25	0,25	0,25	0,25
10	k-NN	0,25	0,26	0,25	0,14
Tüm	k-NN	0,25	0,25	0,25	0,25
5	Bayes	0,26	0,27	0,26	0,15
10	Bayes	0,26	0,29	0,26	0,14
Tüm	Bayes	0,25	0,26	0,25	0,15
5	YSA	0,25	0.16	0,25	0,1
10	YSA	0,25	0,25	0,19	0,18
Tüm	YSA	0,25	0,25	0,25	0,23

Çizelge 3.2 incelendiğinde hem tüm özellikler alınarak hem de özellik azaltma bakımından algoritmaların başarılı sonuçlar üretemediği görülmektedir. 0,25 ve altı değerler oldukça yetersiz kalmaktadır. Özellikle F1-skorda 0,1'e kadar düşme olduğu dikkat çekmektedir. Tüm yöntemler aşağı yukarı aynı sonuçları üretmiştir. Bu durumun algoritmalardan ziyade, veri

setinden kaynaklı olduğu düşünülmektedir. Fakat, tablodan yine karar ağacı ve rastgele orman algoritmasının diğer algoritmaları geride bıraktığı çıkarımı yapılabilir.

Çizelge 3.3 Slowhttptest için deney sonuçları (Experimental results for Slowhttptest)

Özellik sayısı	**Sınıflandırıcı**	**Doğruluk oranı**	**Duyarlılık**	**Kesinlik**	**F1-skor**
5	Karar ağacı	0,39	0,39	0,43	0,4
10	Karar ağacı	0,52	0,53	0,55	0,54
Tüm	Karar ağacı	0,53	0,54	0,57	0,55
5	Rastgele orman	0,4	0,4	0,43	0,41
10	Rastgele orman	0,5	0,5	0,53	0,51
Tüm	Rastgele orman	0,51	0,54	0,55	0,54
5	k-NN	0,4	0,51	0,43	0,45
10	k-NN	0,42	0,52	0,45	0,47
Tüm	k-NN	0,38	0,49	0,41	0,44
5	Bayes	0,38	0,5	0,41	0,33
10	Bayes	0,37	0,41	0,39	0,32
Tüm	Bayes	0,35	0,23	0,39	0,27
5	YSA	0,38	0,5	0,41	0,34
10	YSA	0,35	0,25	0,38	0,27
Tüm	YSA	0,37	0,55	0,4	0,34

Çizelge 3.3 incelendiğinde tüm nitelikler ile sınıflandırma da en iyi sonucun karar ağacından alındığı görülmektedir. Rastgele orman dışındaki yaklaşımların başarı oranlarının düşük olduğu görülmektedir. Kalan algoritmalar sıralandığında k-NN, YSA ve son olarak Bayes gelmektedir.

Özellik sayısı 5 olarak belirlendiğinde sonuçlar yine çok düşüktür. 5 nitelikte en iyi sonuçları k-NN vermiştir. Rastgele orman algoritması da benzer çıktılara sahiptir. Fakat diğer algoritmaların %40 ve altı sonuçlar ile oldukça başarısız oldukları görülmektedir. Burada tüm niteliklerde olduğu gibi yine Bayes (YSA dan az bir farkla) en başarısız algoritma olmuştur.

Özellik sayısı 10' a düşürülerek tekrar deneyler yapılmıştır. 10' da tüm özelikler ile yapılan deneylere daha yakın çıktılar elde edilmiştir. Elde edilen çıktılar çok iyi olmasada Slowhttptest için kısmen daha düşük sayıda veri vardır. Bundan dolayı, sonuçlar %50 ve üzeri olup kabul edilebilirdir. En başarılı yaklaşım karar ağacı olarak alınmıştır. Diğer yaklaşımlar incelendiğinde rastgele ormanında benzer performans sağladığı görülmektedir. Daha sonra k-NN algoritması, Bayes ve son olarak YSA gelmektedir. Tüm nitelikler ve 5 özelliğin aksine burada Bayes YSA' dan daha etkili olmuştur. Slowhttptest' i sınıflandırmada en başarılı mimari tarafından (karar ağacı) en etkili 10 özellik:

- Bwd IAT Mean
- Flow Bytes/s
- RST Flag Count
- Bwd IAT Max
- Fwd Packet Length Min
- Total Fwd Packet
- Flow Duration
- Bwd PSH Flag
- Active Min
- FWD Init Win Bytes olarak tespit edilmiştir.

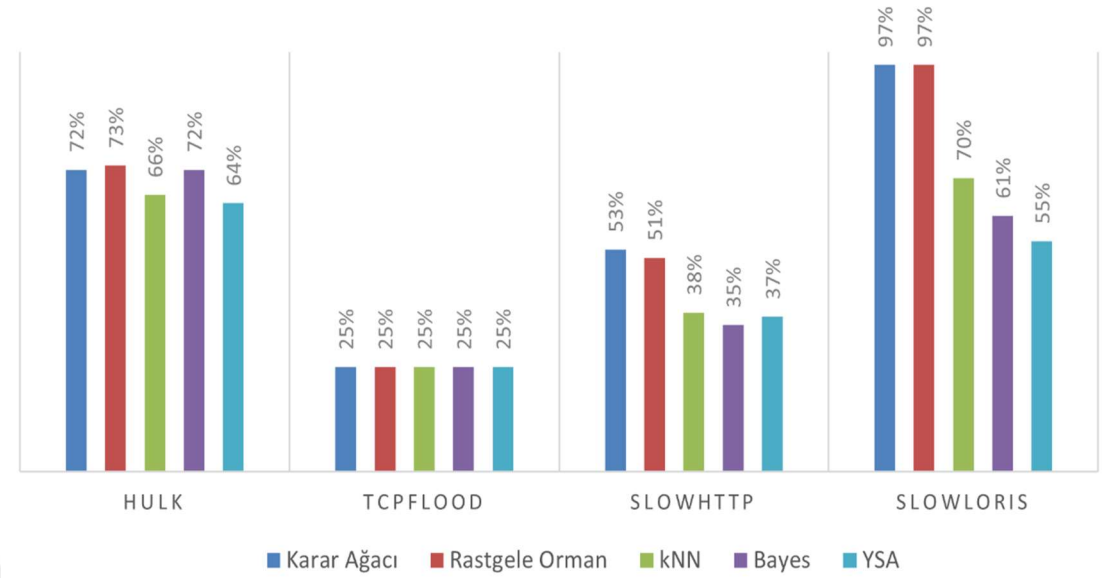
Çizelge 3.4 Slowloris için deney sonuçları (Experimental results for Slowloris)

Özellik sayısı	**Sınıflandırıcı**	**Doğruluk oranı**	**Duyarlılık**	**Kesinlik**	**F1-skor**
5	Karar ağacı	0,96	0,94	0,94	0,94
Tüm	Karar ağacı	0,97	0,96	0,96	0,96
5	Rastgele orman	0,97	0,96	0,96	0,96
Tüm	Rastgele orman	0,97	0,96	0,95	0,96
5	k-NN	0,7	0,5	0,61	0,54
Tüm	k-NN	0,7	0,5	0,61	0,54
5	Bayes	0,62	0,41	0,5	0,4
Tüm	Bayes	0,61	0,5	0,33	0,37
5	YSA	0,62	0,41	0,5	0,4
Tüm	YSA	0,55	0,33	0,4	0,31

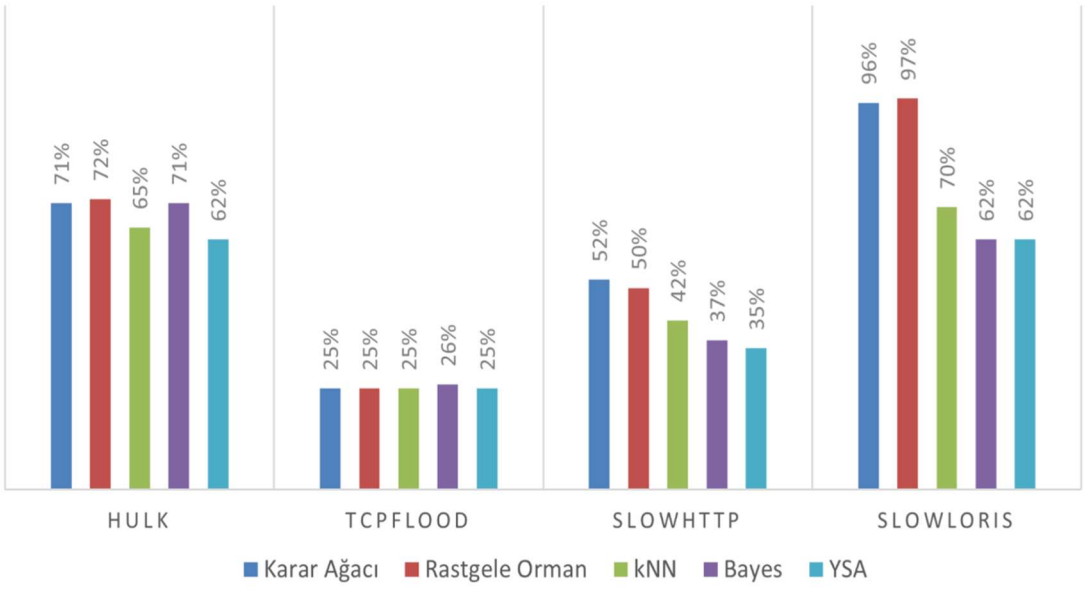
Çizelge 3.4 incelendiğinde yine en etkili algoritmaların tüm metrikler dikkate alındığında karar ağacı ve rastgele orman olduğu görülmektedir. %96 ve üzeri değerler başarı oranlarının yüksek olduğu sonuçlardır. Diğer yaklaşımlara bakıldığında k-NN, Bayes ve son olarak YSA gelmektedir. Nitelik seçimi için sonuçlar incelendiğinde diğer deneylere benzer şekilde rastgele orman algoritması ön plana çıkmaktadır. Aynı şekilde, karar ağacının da başarısı dikkat çekmektedir. 0,95 ve üzerinde performans ele alındığında GA' nın da sınıflandırmaya en etkili 5 özelliği tahmin edebildiği çıkarımı yapılmaktadır:

- Bwd IAT Total
- Active Max
- Bwd IAT Max
- Fwd IAT Std
- Bwd Packet Length Std

Karar ağacı ve rastgele orman algoritmalarından sonra k-NN, daha sonra YSA ve son olarak Bayes algoritmaları gelmektedir. **Şekil 3.3, Şekil 3.4, Şekil 3.5 ve Şekil 3.6' da, Çizelge 3.1, 3.2, 3.3 ve 3.4' deki** değerler incelendiğinde Genetik Algoritma uygulanmadan önceki (tüm nitelikler ile) doğruluk oranları ve Genetik Algoritma uygulandıktan sonra (nitelik seçimi ile) doğruluk oranları gösterilmiştir. Burada, tüm çıktılar dikkate alınarak Genetik Algoritma' nın başarılı olduğu yorumu yapılabilir.

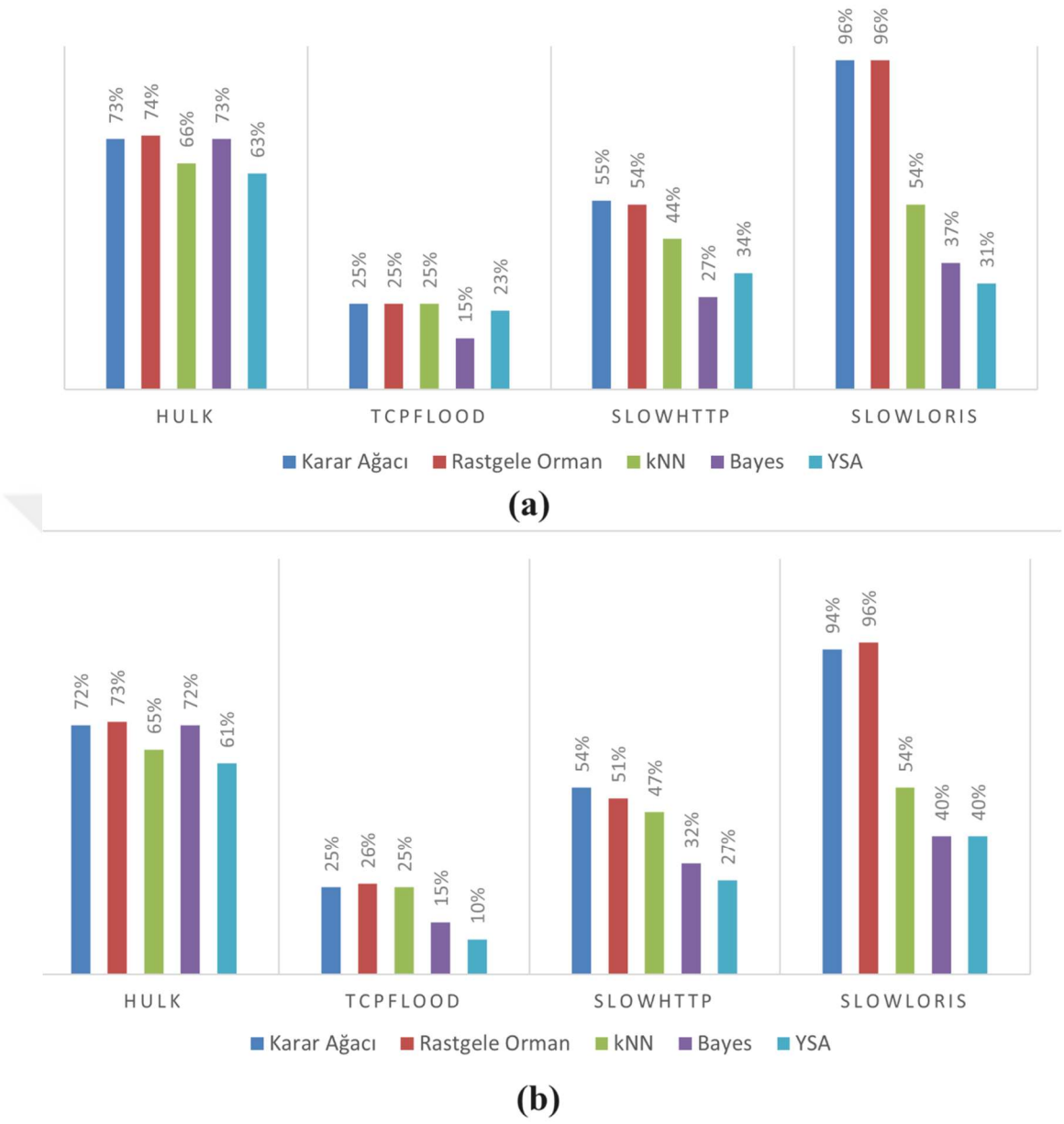


(a)

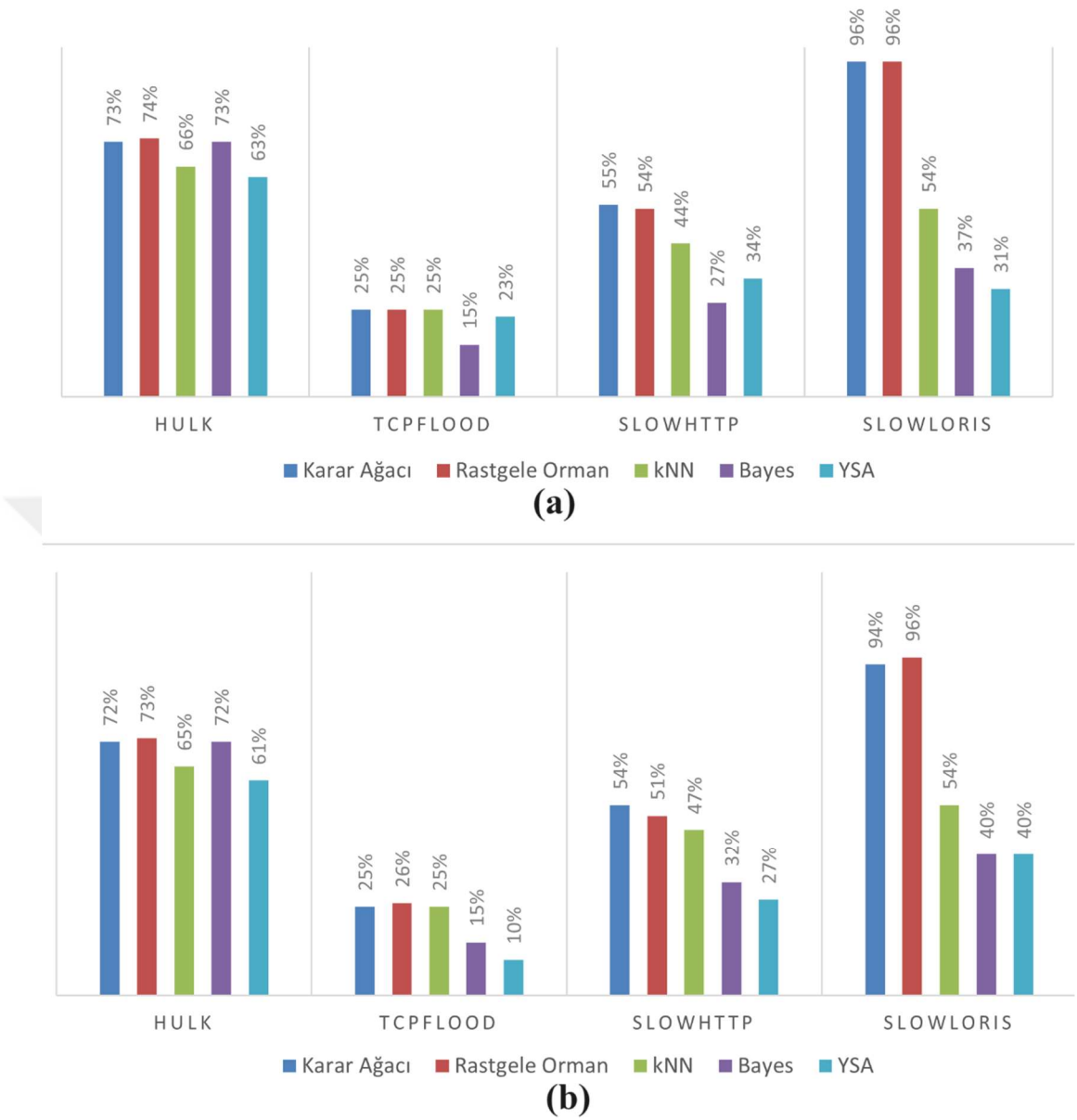


(b)

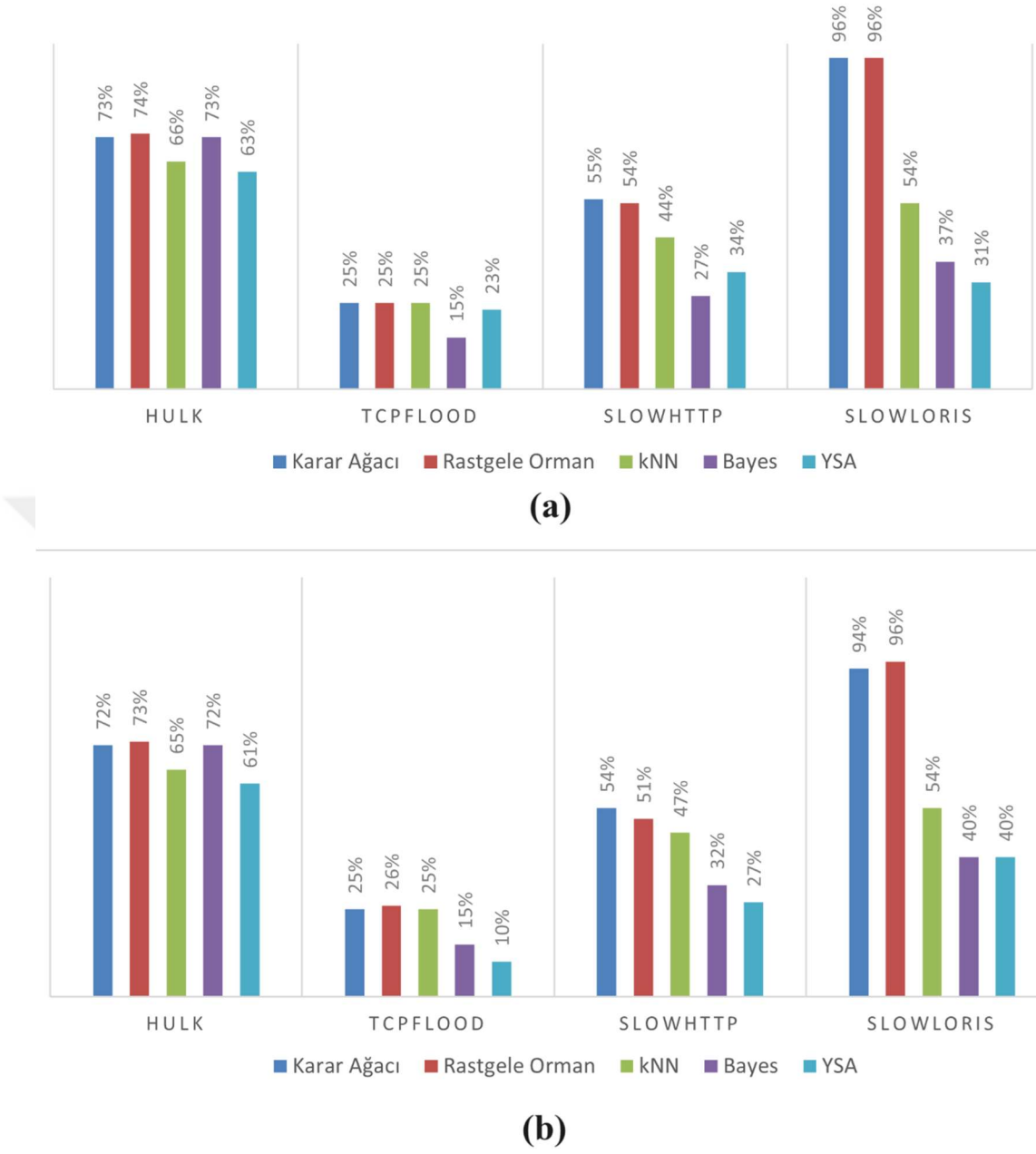
Şekil 3.3 Doğruluk oranları (a) GA uygulanmadan önce (tüm özelliklere göre) (b) GA uygulandıktan sonra (Accuracy rates (a) before applying GA (according to all features) (b) after applying GA)



Şekil 3.4 Duyarlılık oranları (a) GA uygulanmadan önce (tüm özelliklere göre) (b) GA uygulandıktan sonra (Recall rates (a) before applying GA (according to all features) (b) after applying GA)



Şekil 3.5 Kesinlik oranları (a) GA uygulanmadan önce (tüm özelliklere göre) (b) GA uygulandıktan sonra (Presicion rates (a) before applying GA (according to all features) (b) after applying GA)



Şekil 3.6 F1-skor oranları (a) GA uygulanmadan önce (tüm özelliklere göre) (b) GA uygulandıktan sonra (F1-score rates (a) before applying GA (according to all features) (b) after applying GA)

Çizelge 3.5 Tüm gruplar için deney sonuçları (Experimental results for all groups)

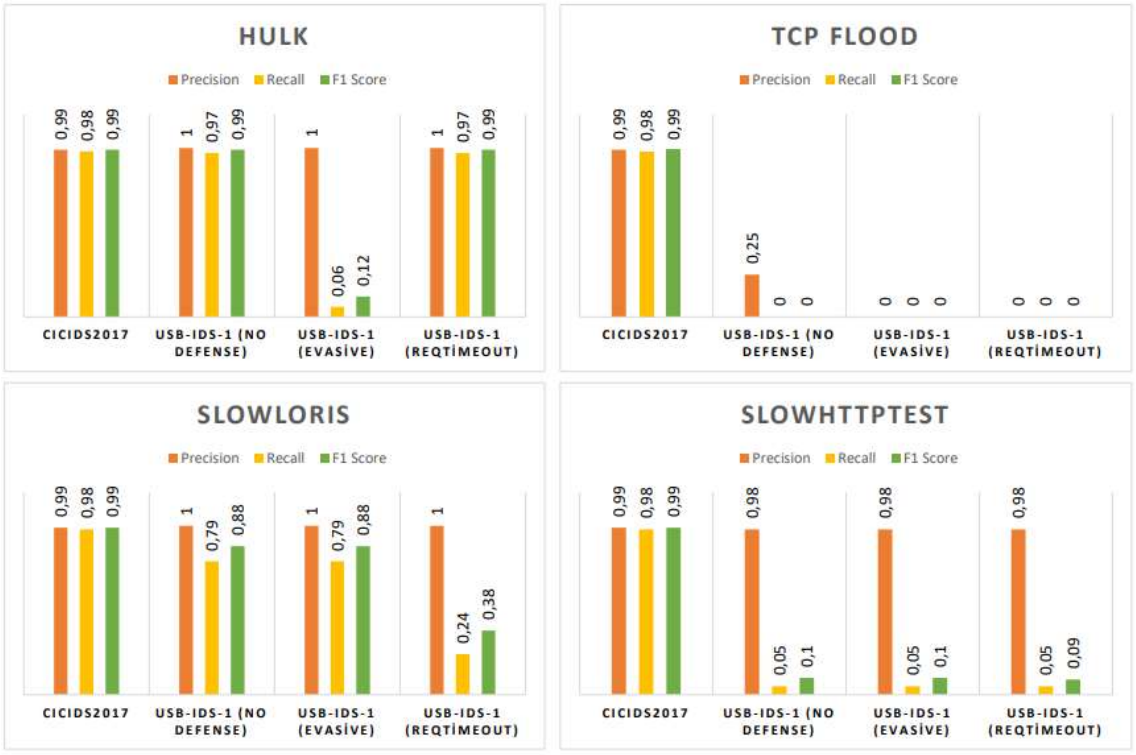
Özellik sayısı	**Sınıflandırıcı**	**Doğruluk**	**Duyarlılık**	**Kesinlik**	**F1-skor**
10	Karar ağacı	0,54	0,58	0,56	0,55
Tüm	Karar ağacı	0,58	0,61	0,6	0,59
10	Rastgele orman	0,55	0,59	0,57	0,56
Tüm	Rastgele orman	0,58	0,64	0,6	0,58
10	k-NN	0,45	0,44	0,44	0,38
Tüm	k-NN	0,48	0,42	0,46	0,41
10	Bayes	0,44	0,35	0,42	0,32
Tüm	Bayes	0,41	0,34	0,39	0,28
10	YSA	0,28	0,23	0,25	0,21
Tüm	YSA	0,43	0,41	0,4	0,33

Çizelge 3.5'ten tüm grupların bir arada olduğu deneyler incelendiğinde, ayrı ayrı olan sonuçlarda olduğu gibi hem tüm özelliklerde hem de 10 tane nitelikte rastgele orman algoritmasının en başarılı algoritma olduğu görülmektedir. Karar ağacı da yine rastgele orman ile yaklaşık sonuçları üretmiştir. Tüm nitelikler bakımından kalan algoritmalar için k-NN, YSA ve Bayes sıralaması yapılırken, 10 nitelik açısından sıralama k-NN, Bayes ve YSA' dır. Tahmin edilen (rastgele orman sınıflandırıcı için) en etkili 10 özellik ise şu şekildedir:

- Bwd Header Length
- Fwd Segment Size Avg
- Fwd IAT Total
- URG Flag Count
- Bwd Packet Length Min
- Bwd Packet Length Mean
- Bwd IAT Std
- Fwd IAT Mean
- Bwd PSH Flags
- Bwd IAT Min

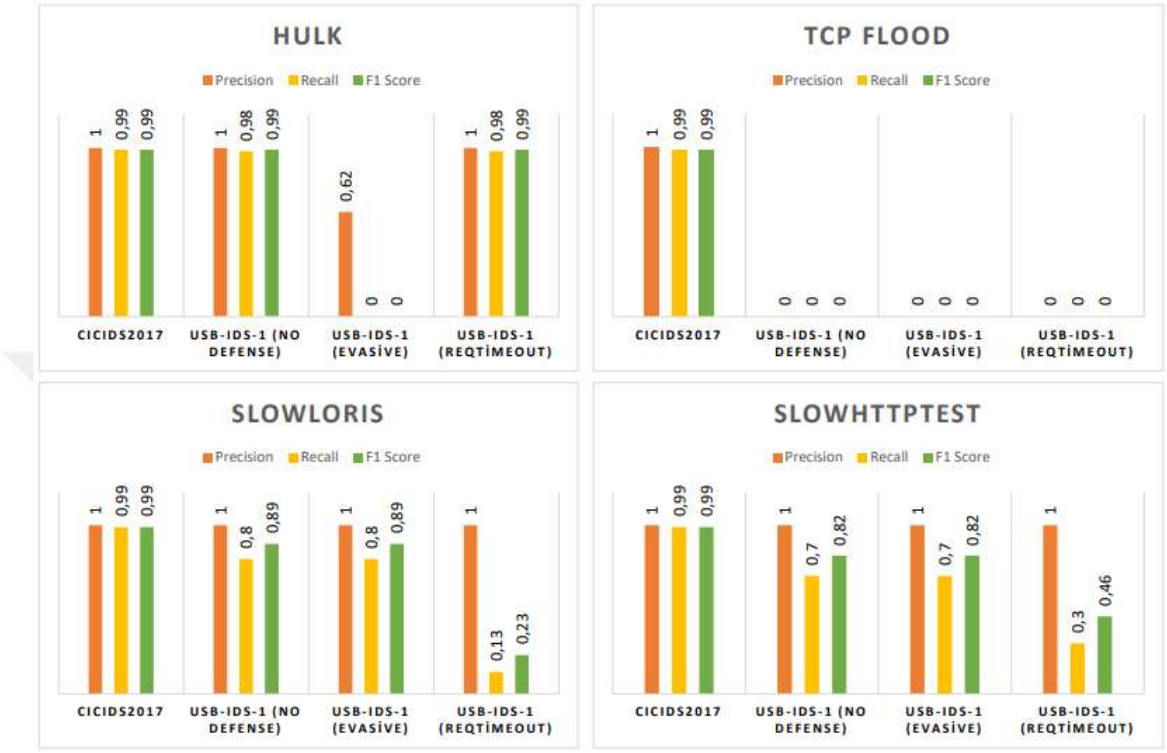
Sonuçların ortalama değerlerde olması ayrı ayrı deneyler incelendiğinde beklenen bir durumdur. Slowhttptest ve özellikle TCP'den kaynaklı olarak başarı oranı aşağı çekilmektedir. TCP için ayrı şekilde testlerde çıktılar %25 civarındadır. Daha çok sayıda veri olduğu halde sonucun bu kadar düşük çıkması, sınıf sayısının artması ve veri sayısının azalması halinde çok daha düşük değerlere neden olacaktır. Slowloris için olan deneyler de ise az sayıda örnek olmasına rağmen sonuçların çok iyi olduğu gözlemlenmiştir. TCP'nin aksine bu kategori totalde başarı oranının yukarı çıkmasını sağlayacaktır.

Şekil 3.7 'de Catillo ve diğerleri tarafından karar ağaçları algoritması ile gerçekleştirilen deney sonuçları gösterilmiştir. Yazarlar çalışmalarında eğitim için farklı bir seti kullanmışlardır. Ayrıca her veri setinde her saldırı tipi için bir uygulama gerçekleştirmişlerdir. Bu bağlamda birebir karşılaştırma yapmak mümkün olmasada, benzer şekilde yazarların TCP Flood veri setinde başarılı sonuç alamadığı açıkça görülmektedir.



Şekil 3.7 Her saldırı için CICIDS2017 ve USB-IDS-1 veri setleri üzerinde karar ağaçlarının değerlendirilmesi (Catillo vd. 2021)

Benzer şekilde**, Şekil 3.8 'de** Catillo ve diğerleri tarafından rastgele orman algoritması ile gerçekleştirilen deney sonuçları gösterilmektedir. Şekilden yine TCP veri setinden olumlu sonuç alınamadığı yorumu yapılabilmektedir.



Şekil 3.8 Her saldırı için CICIDS2017 ve USB-IDS-1 veri setleri üzerinde rastgele orman modeli değerlendirilmesi (a-d) (Catillo vd. 2021)

4. SONUÇ

Teknolojik gelişmeler hayatı kolaylaştırıcı birçok yeniliğin yanı sıra beraberinde kötü amaçla kullanılan işlemleri de getirmiştir. Online dolandırıcılık tehlikesine karşı verilerin güvende tutulması oldukça önem arz etmektedir. Bundan dolayı sıkı önemler alınmakta, saldırıları önlemek için sistemler geliştirilmektedir. Yapay zeka tabanlı yaklaşımların çıkarım yapmadaki başarısı, günlük hayatta ki problemlere uygulanabilirliğini arttırmıştır. Makine öğrenmesi yapay zekanın bir alt alanı olup, algoritmaları yaygın bir kullanıma sahiptir.

Bu çalışmada saldırı tespiti üzerine hazırlanmış olan USB-IDS-1 veri seti üzerinde karar ağacı, rastgele orman, k-NN, Bayes ve YSA yaklaşımları uygulanarak sınıflandırma yapılmıştır. Daha sonra GA ile özellik azaltma işlemi yapılarak GA'nın başarısı incelenmiştir. Veri setinde 4 ayrı saldırı tipine ait (Hulk, TCP, Slowhhtptest, Slowloris) 83 nitelikten oluşan satırlar yer almaktadır. Fakat, burada gruplar arasındaki veri oranı oldukça dengesizdir. Bu şekilde deney yapılması durumunda algoritmalar baskın olan gruba göre sınıflandırma yapacaktır. Bu durumda performansların yanlış yorumlanmasına neden olabilir. Bundan dolayı her grup için ayrı ayrı deneyler yapılmıştır. Fakat, tüm gruplar ile de algoritmaların performansları incelenmiştir. Bunun için en düşük veri sayısı dikkate alınarak her sınıftan eşit sayıda örnek alınmış ve deneyler gerçekleştirilmiştir.

Elde edilen sonuçlara göre Hulk ve Slowloris veri setinde karar ağacı ve rastgele orman algoritması başarılı olmuştur. Özellikle Slowloris' te %95 ve üzeri çıktılar ile oldukça iyi performans göstermişlerdir. Genel olarak tüm gruplarda bu iki algoritmanın diğerlerine göre daha etkili olduğu görülmektedir. Slowhttptest verisinde ise bu iki algoritma ortalama bir başarı sergilemiş, fakat diğer algoritmalar yine daha kötü sonuç ortaya koymuştur. TCP verilerini hem tüm özelliklere göre hem de azaltılmış özelliklere göre sınıflandırılmada tüm algoritmalar başarısız olmuştur.

Tüm grupların dahil olduğu sonuçlar incelendiğinde ise, en iyi değerlerin %55-%64 aralığında olduğu görülmüştür. Bu durum ayrı ayrı sınıflandırmalar göz önüne alındığında beklenen bir neticedir.

İleri ki çalışmalarda farklı optimizasyon algoritmaları PSO (Kennedy ve Eberhart 1995), ABC (Karaboga 2005) vb.) ve farklı veri setleri ile deneyler yapılabilir. Ayrıca farklı sınıflandırma yöntemlerinin ve YSA' nın gelişmiş versiyonları olan derin öğrenme yaklaşımlarının da performansları değerlendirilebilir.

KAYNAKLAR

Aburomman, A. A., Reaz, M. B. I. 2016. *Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection*, Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 636–640.

Al-Jarrah, O. Y., Al-Hammdi, Y., Yoo, P. D., Muhaidat, S., Al-Qutayri, M. 2018. *Semi-supervised multi-layered clustering model for intrusion detection*, Digital Communications and Networks, 4(4), 277–286.

Al-Yaseen, W. L., Othman, Z. A., Nazri, M. Z. A. 2017. *An hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system*, Expert Systems with Applications, 67(1), 296–303.

An, X., Su, J., Lü, X., Lin, F. 2018. *Hypergraph clustering model-based association analysis of DDOS attacks in fog computing intrusion detection system*, EURASIP Journal on Wireless Communications and Networking, 249 (1), 1–9.

Belavagi, M. C., Muniyal, B. 2016. *Performance evaluation of supervised machine learning algorithms for intrusion detection*, Procedia Computer Science, 89(1), 117–123.

Breiman, L. 2001. *Random forests*, Machine Learning, 45, 5—32.

Bishop, C. M. 1995. *Neural networks for pattern recognition*. Oxford University Press, 502, England.

Catillo, M., Del Vecchio, A., Ocone, L., Pecchia, A., Villano, U. 2021. *USB-IDS-1: a Public Multilayer Dataset of Labeled Network Flows for IDS Evaluation*, 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 1–6, doi: 10.1109/DSN-W52860.2021.00012.

Catillo, M., Del Vecchio, A., Pecchia, A., Villano, U. 2022. *Transferability of machine learning models learned from public intrusion detection datasets: the CI-CIDS2017 case study*, Software Quality Journal, 1–27.

Cover, T., Hart, P. 1967. *Nearest neighbor pattern classification*, IEEE transactions on information theory, 13(1), 21–27.

Haykin, S. 1999. *Neural networks: A comprehensive foundation (2nd ed.)*. Prentice Hall, 842, New Jersey.

Holland, J. H. 1992. *Genetic algorithms*. Scientific American, 267(1), 66–73.

Kalutharage, C.S., Liu, X., Chrysoulas, C. 2022. *Explainable AI and Deep Autoencoders Based Security Framework for IoT Network Attack Certainty (Extended Abstract)*,

International Workshop on Attacks and Defenses for Internet-of-Things, Springer, 41–50. https://doi.org/10.1007/978-3-031-21311-3_8

Kaplan, A., Haenlein, M. 2019. *Siri, Siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of Artificial Intelligence*, Business Horizons, 62(1), 15–25.

Karaboga, D. 2005. *An idea based on honey bee swarm for numerical optimization* (Technical report-tr06), Erciyes university, engineering faculty, computer engineering department, 200, 1–10.

Kennedy, J., Eberhart, R. 1995. *Particle Swarm Optimization*, Proceedings of IEEE International Conference on Neural Networks, 4, 1942—1948.

Kingma, D. P., Jimmy, Ba. 2014. *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980.

Koza, J. R. 1992. *Genetic programming: On the programming of computers by means of natural selection*. Bradford Books, 840, Massachusetts.

Manocha, S., Girolami, M. A. 2007. *An empirical analysis of the probabilistic K-nearest neighbour classifier*. Pattern Recognition Letters, 28, 1818–1824.

Resende, P. A. A., Drummond, A. C. 2018. *A survey of random forest based methods for intrusion detection systems*. ACM Computing Surveys, 51, 48.

Samuel, A. L. 1959. *Some Studies in Machine Learning Using the Game of Checkers*, IBM Journal of Research and Development, 3(3), 210–229.

Shon, T., Moon, J. 2007. *A hybrid machine learning approach to network anomaly detection*. Information Sciences, 177, 3799–3821.

Sommer, R., Paxson, V. 2010. *Outside the closed world: On using machine learning for network intrusion detection*. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy, IEEE Computer Society, Washington, DC, USA, SP '10, pp 305–316

Stallings, W. 2006. *Cryptography and network security principles and practices*. Prentice Hall, 680, USA.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A. A. 2009. *A detailed analysis of the KDD CUP 99 data set*, IEEE symposium on computational intelligence for security and defense applications, 1–6.

Vaishali S. 2023. *CAIDA UCSD DDoS 2007 Attack Dataset*, IEEE Dataport. doi: https://dx.doi.org/10.21227/dvp9-s124.