T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ

AĞ ANOMALİSİ TESPİTİNDE EMÜLATÖR ORTAMI TASARIMI VE MAKİNE ÖĞRENMESİ İLE SALDIRI TESPİTİ

Serkan KESKİN

Danışman: Dr. Öğr. Üyesi Ersan OKATAN

BURDUR, 2024

ETİK KURALLARA UYGUNLUK BEYANI

Burdur Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili hükümleri uyarınca **Yüksek Lisans** Tezi olarak sunduğum **"Ağ Anomalisi Tespitinde Emülatör Ortamı Tasarımı ve Makine Öğrenmesi ile Saldırı Tespiti"** başlıklı bu tezin;

- Kendi çalışmam olduğunu,
- Sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi,
- Bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi,
- Kullandığım verilerde değişiklik yapmadığımı,
- Tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını,
- Bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı,
- Bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı,

bildirir, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

03/06/2024

(İmza)

Serkan KESKİN

ÖNSÖZ

Bu araştırma için beni yönlendiren, karşılaştığım zorlukları bilgi ve tecrübesi ile aşmamda yardımcı olan değerli Danışman Hocam Dr. Öğr. Üyesi Ersan OKATAN'a teşekkürlerimi sunarım. Araştırmamın her aşamasında yardımlarını esirgemeyen çalışma arkadaşım Öğr. Gör. Dr. Enes AÇIKGÖZOĞLU'na teşekkür ederim.

Ayrıca bu yolda beni yalnız bırakmayan, sevgi ve destekleri ile her zaman yanımda olan eşim ve oğluma sonsuz sevgi ve saygılarımı sunarım.

Haziran, 2024 **Serkan KESKİN**

İÇİNDEKİLER

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

ATSTS : Anormallik Tabanlı Saldırı Tespit Sistemi

CIA : Confidentiality Integrity Availability

CSV : Comma Separated Values - Virgülle Ayrılmış Değeler

DDoS : Dağıtılmış Hizmet Reddi

DN : Doğru Negatif

DNS : Domain Name System – Alan Adı Sistemi

DoS : Denial Of Service - Servis Hizmet Reddi

DP **:** Doğru Pozitif

DVM : Destek Vektör Makinesi

EVE-NG : Emulated Virtual Environment - Next Generation

FTP : File Transfer Protocol – Dosya Transfer Protokol

IP : İnternet Protokol

KA : Karar Ağacı

KNN : En Yakın Komşu

RFE : Recursive Feature Elemination - Özyinelemeli Özellik Seçimi

RO : Rastgele Orman

SDN : Software Define Network – Yazılım Tanımlı Ağ

SMOTE : Sentetik Azınlık Aşırı Örnekleme Tekniği

SÖS : Saldırı Önleme Sistemi

STÖS : Saldırı Tespit ve Önleme Sistemi

STS : Saldırı Tespit Sistemi

UDP : User Datagram Protocol

XSS : Cross-site scripting

YN **:** Yanlış Negatif

YP **:** Yanlış Pozitif

YSA : Yapay Sinir Ağı

ÖZET

Yüksek Lisans Tezi

Ağ Anomalisi Tespitinde Emülatör Ortamı Tasarımı ve Makine Öğrenmesi ile Saldırı Tespiti

Serkan Keskin

Burdur Mehmet Akif Ersoy Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ersan OKATAN

Haziran, 2024

İnternetin ortaya çıkışı ve bununla bağlantılı cihazların sayısının hızla artması birçok avantajı beraberinde getirmiştir. Bu avantajların yanı sıra bize önemli zorluklar da sunmuştur. Bu zorluklar arasında siber tehditler, acil müdahale gerektiren en önemli konu olarak öne çıkıyor. Bireyleri, kuruluşları, hatta ulusların tamamını hedef alan bu saldırılar, maddi kayıpların yanı sıra duygusal ve psikolojik zarara da yol açma potansiyeli taşımaktadır.

Bu olumsuz durumların ışığında bu araştırmanın amacı, ağ saldırılarını doğru bir şekilde tanımlayıp tespit edebilen anomali tabanlı tespit sistemleri geliştirmek amacıyla çeşitli makine öğrenme tekniklerinin kapsamlı bir analizini yapmaktır. Bu tez çalışması kapsamında emülatörden beş farklı veri seti oluşturmak için yararlanılmıştır. Deneysel çalışmalar üç farklı öznitelik seçimi yöntemi kullanılarak gerçekleştirilmiştir. Bu yöntemler özyinelemeli, ileri yönelimli ve manuel özellik seçimidir. Dengesiz veri kümelerini ele almak için Sentetik Azınlık Aşırı Örnekleme Tekniği uygulanarak veri artırma gerçekleştirildi. Tüm potansiyel sonuçlara ilişkin doğruluk oranları, Rastgele Orman, Kara Ağaçları, K En Yakın Komşu ve Destek Vektör Makinesi olmak üzere sınıflandırma algoritmaları kullanılarak doğruluk oranları hesaplandı. Bu hesaplamalar Python programlama dili ve Sklearn, NumPy, Pandas ve Matplotlib gibi kütüphaneler kullanılarak gerçekleştirildi.

Saldırı tespit sistemlerinde, makine öğrenmesi algoritmalarından özellikle Rastgele Orman, Karar Ağaçları ve En Yakın Komşu algoritmaları, performans ve başarı açısından en yüksek düzeyde sonuçlar vermiştir. Bu üç algoritmanın doğruluk oranları genel olarak %99.92 ile %100 arasında gerçekleşmiştir. Bu durum söz konusu algoritmaların saldırı tespiti için son derece etkili olduğunu ortaya koymaktadır. Sentetik veri artırımının uygulanmasının sonuçlar üzerinde önemli bir etki yaratmadığı gözlemlenmiştir.

Yapılan bu tez çalışması sayesinde, ağdaki veri modellerinin ve hacminin derinlemesine incelenmesi, izlenmesi ve analiz edilmesi, ağın verimli bir şekilde işlev görmesini sağlayacak ve bilgi paylaşımının güvenli bir şekilde gerçekleşmesine olanak tanıyacak güvenilir bir Saldırı Tespit Sistemi'nin geliştirilmesine katkıda bulunacaktır.

Anahtar Kelimeler: ağ anomalisi, emülatör, makine öğrenmesi, öznitelik seçimi, saldırı tespit

ABSTRACT

M.Sc. Thesis

Emulator Environment Design for Network Anomaly Detection and Attack Detection with Machine Learning

Serkan Keskin

Burdur Mehmet Akif Ersoy University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Ersan Okatan

June, 2024

The emergence of the Internet and the rapid increase of connected devices have brought many advantages. In addition to these advantages, it has also presented us with significant challenges. Among these challenges, cyber threats stand out as the most important issue that requires urgent intervention. These attacks, which target individuals, organisations and even entire nations, have the potential to cause emotional and psychological damage as well as material losses. In the light of these adverse situations, the aim of this research is to conduct a comprehensive analysis of various machine learning techniques in order to develop anomaly-based detection systems that can accurately identify and detect network attacks.

In this thesis, the emulator was used to generate five different data sets. Experimental studies were carried out using three different feature selection methods. These methods are recursive, forward-oriented and manual feature selection. Data augmentation was performed by applying Synthetic Minority Oversampling Technique to handle unbalanced datasets. Accuracy rates for all potential results were calculated using classification algorithms including Random Forest, Black Trees, Nearest Neighbour and Support Vector Machine. These calculations were performed using Python programming language and libraries such as Sklearn, NumPy, Pandas and Matplotlib.

Among the machine learning algorithms, especially Random Forest, Decision Trees and Nearest Neighbour algorithms gave the highest level of performance and success in intrusion detection systems. The accuracy rates of these three algorithms were generally between 99.90% and 100%. This shows that these algorithms are highly effective for intrusion detection. It was observed that the application of synthetic data augmentation did not have a significant effect on the results.

Thanks to this thesis study, in-depth examination, monitoring and analysis of the data patterns and volume in the network will contribute to the development of a reliable Intrusion Detection System that will enable the network to function properly and allow information sharing to take place securely.

Keywords: emulator, feature selection, intrusion detection, machine learning, network anomaly

1. GİRİŞ

Günümüzde, bilgi ve iletişim teknolojileri çağdaş yaşamın ayrılmaz bir parçası haline gelmiştir. Bilgi ve iletişim teknolojilerindeki hızlı ilerleme, modern toplumları, iletişim ve işbirliği yollarını kökten değiştiren geniş bir dijital ağın içine sokmuştur. Bu dijital ağlar, dünya çapında veri ve bilgi alışverişini kolaylaştırmakla kalmayıp aynı zamanda iş süreçlerini optimize etmek için de hayati bir rol oynamaktadır. Bu teknolojik gelişmeler, bilgi sistemlerini güvenlik tehditlerine daha da duyarlı hale getirmiştir. Gelişmiş siber saldırı yöntemleri ve saldırganların sürekli evrim geçiren taktikleri, geleneksel güvenlik önlemlerini aşmakta ve ağ altyapılarına zarar verebilmektedir. Şu an yaklaşık olarak iki milyar insan interneti kullanmaktadır ve Microsoft'un tahminlerine göre bu rakamın 2025 yılında dört milyarı aşması beklenmektedir (Karaman vd., 2021). İnternet kullanımının bu muazzam genişlemesi göz önüne alındığında, Bilgi Teknolojileri Güvenliği olarak da bilinen Siber Güvenliğin sağlanması giderek daha da önem kazanmaktadır.

Bilgi günümüzün en değerli kaynağıdır. Bilginin yönetimi, analizi ve kullanımı çok önemlidir. Çeşitli alanlarda güvenliği korumaya yönelik çok sayıda kurum ve kuruluş bulunmaktadır. Teknoloji öncesi dönemde güvenliğe ilişkin bilgiler sözlü veya yazılı araçlarla nesilden nesile aktarılıyordu. Günümüz dünyasında bilgi aktarımı bilgisayarlar ve internet sayesinde zahmetsiz hale gelmiştir. Bu aktarım farklı toplumlar arasında gerçekleşebilir (Güler vd., 2017). Bilgiye ulaşmanın kolaylığı kuşkusuz faydalı olmakla birlikte, bu kolaylıkla elde edilen bilgilerin hem doğru hem de güvenilir olmasını sağlamak da büyük önem taşımaktadır.

Pek çok çalışmanın güncel vurgusu bilgiye erişim ve bilginin korunması üzerinedir. Bilginin korunması merkezi bir endişe noktası olarak konumlandırılmıştır. Daha önce bilgiler dolaplarda saklanıyordu. Günümüzde bulut bilişimin kullanılabilirliği, geleneksel dolapların ötesine geçerek büyük ölçekli sistemler de dâhil olmak üzere her boyuttaki bilgisayarı kapsayacak şekilde genişlemiştir. Veriler, hem bireysel bilgisayarlarda hem de bilgisayar ağlarında yaygın bir uygulama olan sunucularda depolanmaktadır (Yazar vd., 2022).

İşlenen bilgilerin korunmasının önemi, günümüzde pek çok kurum ve kuruluşun bilgi güvenliğine öncelik vermesiyle ortaya çıkmıştır. Bu amaca ulaşmak için bir dizi önlemler kullanılmaktadır. Bilginin ve bilgi sistemlerinin korunması, yetkisiz erişim, bilginin kullanımı, bilginin değiştirilmesi ve bilginin imhasının önlenmesi bilgi

güvenliğinin temel bileşenleridir. Bu tanım, hassas verilerin korunmasının ve bütünlüğünün sağlanmasının önemini vurgulamaktadır. Bilginin işlendiği tüm platformlarda gizliliğin, bütünlüğün ve erişilebilirliğin korunması bilgi güvenliğinin sürdürülmesi için zorunludur. Bilgi güvenliğinin temeli, CIA (Confidentiality-Integrity-Availability / Gizlilik-Bütünlük-Erişilebilirlik) üçgeni olarak adlandırılan üç temel bileşenden oluşur. Şekil 1.1'de CIA üçgeninin temel görünümü verilmiştir. CIA'nın 3 temel bileşeni gizlilik, bütünlük ve kullanılabilirliktir. Bu bileşenlere eşlik eden diğer unsurlar ise güvenilirlik, inkar edilemezlik, kimlik doğrulama, yetkilendirme ve izlenebilirliktir. Şekil 1.2'de CIA ya eşlik eden unsurlar gösterilmiştir.

Şekil 1.1. CIA Üçgeni

Gizliliğin amacı, kişinin özel bilgilerini korumak, bu bilgilerin yetkisiz kişilerin erişimine kapalı kalmasını ve her türlü yetkisiz kullanıma karşı korunmasını sağlamaktır. Söz konusu kişi, gizli bilgilerinin başkaları tarafından kullanılmasını engellemek ve bu bilgilerin yetkisiz kişilerin eline geçmesini engellemek istemektedir.

Bütünlüğün amacı, bilgilerin değiştirilmeden ve bozulmadan kalmasını sağlayarak herhangi bir değişiklik veya silme işleminin gerçekleşmesini önlemektir. Bilgilerin hiçbir değişiklik veya eksiklik olmaksızın orijinal haliyle korunması esastır.

Erişilebilirlik kavramı bilginin sürekli olarak kullanılabilirliğini gerektirir. Amacı, kullanıcıların istedikleri verileri ulaşabilecekleri yerden almalarını sağlamaktır (H. Aydın, 2023).

Şekil 1.2. Bilgi Güvenliği Unsurları

Güvenilirlik, bir bilgi sistemine veya sürecine güvenebilme yeteneği anlamına gelir. Bu yetenek, sistemin veya sürecin istikrarlı bir şekilde çalışmasını, veri bütünlüğünü korumasını ve beklenen performans seviyelerini sağlamasını içerir.

İnkâr edilemezlik ilkesi, iletilen verinin göndericisi ile alıcısı arasında ortaya çıkabilecek iletişim sorunlarının ve çatışmaların azaltılması amacına hizmet etmektedir. İki sistem arasında veri alışverişi yapıldıktan sonra taraflardan hiçbiri aktarıma dâhil olduklarını inkâr edemez. Bu prensip, özellikle anlık işlemlere dayanan finansal sistemlerde hayati öneme sahiptir. Bu prensibi sürdürmek için "Kimlik Doğrulama" ve "Bütünlük" ilkelerini oluşturmak esastır. Ayrıca inkâr edilemezliği desteklemek için "Yetkilendirme" ve "İzlenebilirlik" ilkelerinin de sağlanması gerekmektedir.

Kullanıcı kimlik doğrulaması, güvenilirlik oluşturmanın ve kimliği doğrulamanın ilk adımıdır. İşlem sırasında cihazın ya da kullanıcının bu özel eylemi gerçekleştirmesi gerekir. Kullanıcı kimlik doğrulaması, kimliğin doğrulanmasını içerir. Sistem, kullanıcı adının kayıtlı olup olmadığını doğrular ve kimlik doğrulama için parolayı doğrulamaya devam eder. Doğrulama onaylandıktan sonra kullanıcıya sistem erişimi için izin verilir.

Yetkilendirme, kullanıcılara kullanıcı adı ve şifreleri doğrulandıktan sonra bir sisteme, programa veya ağa giriş izni verilip verilemeyeceğini belirleyen mekanizmadır.

Kullanıcılar sistem içerisinde kategorilere ayrılır ve farklı yetki seviyelerine sahip olmaktadırlar. Kullanıcı, atandığı grupla ilişkili tüm ayrıcalıkları devralır. Bir kullanıcı birden fazla gruba aitse genellikle her gruba verilen tüm ayrıcalıklara sahiptir. Örnek olarak kısıtlı yetkilendirme yapılmış bir kullanıcı, dosyaları görüntülemesine olanak tınılabilir ancak silmelerine izin vermez; aynı şekilde sisteme giriş yapabiliyorlar ancak hiçbir kaynağa erişemiyorlar. Optimum güvenliği sürdürmek için kullanıcı izinlerini yalnızca gerekli olanlarla sınırlamak çok önemlidir.

Denetim olarak da bilinen izlenebilirlik ve kayıt tutma süreci, ortaya çıkabilecek sorunları belirleme ve ele alma aracı olarak hizmet eder. Sistem içerisindeki her işlem ve kullanıcı erişimi titizlikle kayıt altına alınmaktadır. Bu kayıtlar yasal, düzenleyici ve teknik amaçlarla tutulur. Siber saldırı gibi bir sorun yaşanması durumunda bu kayıtlar incelenerek iç görü elde edilir ve sorun giderilir (Gönen vd., 2016).

Ağ saldırılarının karmaşıklığı arttıkça, Saldırı Tespit ve Önleme Sistemleri (STÖS), vazgeçilmez savunma araçları haline gelmektedir. Bir Saldırı Tespit Sistemi (STS), potansiyel güvenlik tehditlerini hızlı bir şekilde tanımlayarak ağı izler. Algılama durumunda, sistem hemen bir alarm gönderir ve yöneticiyi bilgilendirir. Sonrasında kaynak adresinden gelen trafiği engelleyen Saldırı Önleme Sistemlerini (SÖS) tetikler. Bir STS, bugün birçok kuruluş için popüler bir tercih olmuştur. STS genellikle, firewall üreticileri için vazgeçilmez bir sistemdir. STS'lerin geliştirilmesinde bir dizi yaklaşım görülmektedir. Bu yaklaşımlar kural tabanlı sistemlerden istatistiksel yöntemlere, eşik değerlerine, yapay sinir ağlarına, veri madenciliğine, bulanık mantığa ve yapay bağışıklık sistemlerine kadar uzanmaktadır (Karaman vd., 2021).

STS'ler imza tabanlı ve anormallik tabanlı olmak üzere iki türe ayrılır. İmza tabanlı STS'ler, bilinen saldırıları bir veri tabanında saklar ve gelen paketleri bu veri tabanıyla eşleştirirken, anormallik tabanlı STS'ler önceden mevcut saldırı bilgilerine dayanmazlar. İmza tabanlı STS'ler bir saldırı algıladığında, eğer veri tabanında mevcutsa saldırı engellenir. Ticari uygulamalar genellikle imza tabanlı STS'lere ağırlık verirler.

Yapay Zekâ tekniklerini kullanarak Anormallik Tabanlı Saldırı Tespit Sistemleri (ATSTS) geliştirilmekte ve sistemi normal ve anormal verilerle eğitmek suretiyle sağlanmaktadır(Baykara ve Daş, 2019). ACARM-ng, AIDE, Bro NIDS, Fail2ban, Samhain, Snort, Suricata vb. gibi birçok sistem, bu yöntemleri kullanmaktadır. Karşılaşılan benzersiz saldırılara karşı, ATSTS güvenilir bir çözümdür. İstatistiksel yöntemler ve model sınıflandırma teknikleri, önceden belirlenmiş bilgiye dayanarak ATSTS'yi modellemek için kullanılır. Veri kümeleri, bir örneği temsil etmek için birçok

özellik içerebilir. İstenen en iyi performans için büyük bir özellik sayısına sahip bir model tasarlamak önerilmez (Fitni ve Ramli, 2020). Bu yaklaşım, yüksek bir hesaplama maliyetine ve sistem için daha yüksek bir hata oranına yol açabilir. Bu maliyet ve hata oranları nedenle, Fitni ve Ramli'nin yapmış olduğu çalışmada saldırı modelini iyileştirmek için farklı özellikler kullanılmıştır.

Ağ güvenliği araştırmacıları ve uzmanları, ağ saldırılarını daha etkin bir şekilde tespit etmek ve önlemek için yeni ve yenilikçi yaklaşımlar geliştirmektedirler. Bu çabaların bir parçası olarak, emülatör ortamı tasarımı ve makine öğrenmesi tekniklerinin birleştirilmesiyle ilgili araştırmalar giderek daha fazla ilgi çekmektedir. Yapılan çalışmada, ağ anomalilerinin tespiti ve saldırıların etkili bir şekilde belirlenmesi için emülatör ortamı tasarımı ve makine öğrenmesi tekniklerinin birleştirilmesi ele alınmıştır. Makine öğrenmesi algoritmalarının kullanımı, ağ trafiği verilerindeki desenleri ve anormallikleri belirlemek için daha esnek ve adaptif bir yaklaşım sunmuştur. Emülatör ortamı, gerçek dünya senaryolarını simüle etmek için kullanılarak, bu algoritmaların performansını test etmek ve iyileştirmek için bir platform sağlamaktadır.

2. GENEL BİLGİLER

2.1. Saldırı Tespit Sistemleri

Saldırı tespit sistemleri, bilgisayar ağlarında veya bilgi sistemlerinde meydana gelebilecek zararlı etkinlikleri tespit etmek ve bu etkinliklere yanıt vermek için tasarlanmış önemli bir güvenlik bileşenidir. Bu sistemler, ağ trafiğini izleyerek, kullanıcı etkinliklerini analiz ederek veya sistem kayıtlarını inceleyerek normal ve anormal davranışları ayırt etmeye çalışmaktadır (Okay vd., 2021).

Saldırı tespit sistemlerinin temel olarak iki ana kategorisi bulunmaktadır: imza tabanlı (signature-based) ve davranış tabanlı (anomaly-based) sistemler.

2.1.1. İmza Tabanlı Saldırı Tespit Sistemi

İmza tabanlı saldırı tespit sistemleri, bilinen saldırı kalıplarını tanımlayan önceden tanımlanmış imzaları kullanarak saldırıları tespit etmeyi amaçlar. Bu sistemler, belirli bir saldırı türünün karakteristik özelliklerini barındıran imzaları içeren bir veritabanını kullanır ve ağ trafiğini veya sistem kayıtlarını bu imzalarla eşleştirir (Masdari ve Khezri, 2020).

Bu sistemlerin işleyişi aşağıda verilen adımları içermektedir:

- **Veri Toplama**: İmza tabanlı saldırı tespit sistemleri ağ trafiği, sistem logları veya diğer kaynaklardan gelen verileri toplamaktadır. Bu veriler, potansiyel saldırı aktivitelerini içermektedir.
- **İmza Tanımlama**: Sistem, belirli saldırı türlerini temsil eden imzaları içeren bir veri tabanına sahiptir. Bu imzalar, saldırının karakteristik özelliklerini ve izlerini içerir. Örneğin, bir DDoS saldırısının belirli bir ağ trafiği deseni veya bir kötü amaçlı yazılımın belirli bir dosya yapısı olabilir.
- **Eşleştirme ve Tanımlama**: İmza temelli saldırı tespit sistemleri, toplanan verileri veri tabanındaki imzalarla eşleştirir. Eşleşen bir imza bulunduğunda, sistem saldırıyı tanımlar ve uygun bir yanıt üretir.
- **Uyarı ve Yanıt**: Sistem, bir saldırı tespit edildiğinde, bir uyarı üretir ve belirli bir yanıt mekanizması devreye girer. Bu yanıt, saldırıyı durdurmak veya saldırının etkilerini minimize etmek için alınan önlemleri içerebilir.

İmza temelli saldırı tespit sistemlerinin avantajları aşağıdaki şekilde sıralanabilir:

- **Etkinlik**: Belirli saldırı türlerini tanımlayan önceden tanımlanmış imzalar sayesinde etkili bir şekilde çalışır.
- **Hız**: İmzaların doğrudan eşleştirilmesi, hızlı bir saldırı tespit süreci sağlar.
- **Doğruluk**: İmzalar genellikle doğru saldırı tespiti sağlar, çünkü tanımlanmış ve bilinen saldırı kalıplarına dayanır.

İmza temelli sistemlerin bazı sınırlamaları da vardır. Bu sınırlamalar:

- **Yenilikçilik**: Yeni ve bilinmeyen saldırı türlerini tespit etme konusunda zayıf olabilirler.
- **Güncelleme Gereksinimi**: İmza veri tabanlarının düzenli olarak güncellenmesi gerekebilir, aksi takdirde yeni saldırılar tespit edilemeyebilir.
- **Yanlış Pozitifler**: İmzaların sınırlı kapsamı nedeniyle, bazen normal trafiği yanlışlıkla saldırı olarak tanımlayabilirler.

Bu nedenlerle, günümüzde genellikle imza temelli saldırı tespit sistemleri, diğer tekniklerle birlikte kullanılarak daha kapsamlı bir güvenlik çözümü oluşturmak için tercih edilmektedirler.

2.1.2. Davranış Tabanlı (Anomali) Saldırı Tespit Sistemi

Davranış tabanlı saldırı tespit sistemleri, normal sistem davranışlarını modelleyerek, anomali olarak nitelendirilebilecek davranışları tespit etmeyi amaçlamaktadır. Bu sistemler, ağ trafiği, kullanıcı etkinlikleri veya sistem kayıtları gibi verileri analiz ederek, normal işletme durumlarından sapmaları belirlemeye çalışırlar (Dina ve Manivannan, 2021).

Davranış temelli saldırı tespit sistemlerinin işleyişi aşağıda verilen adımları içermektedir:

Model Oluşturma: Sistem, normal işletme durumunu temsil eden bir model oluşturur. Bu model, ağ trafiği desenleri, kullanıcı etkinlikleri, sistem kayıtları ve diğer gözlemlerden elde edilen verilere dayanmaktadır.

Anomalilerin Tanımlanması: Oluşturulan model, normal davranışları temsil eden bir referans noktası olarak kullanılır. Sistem, normalden sapmaları tespit etmek için bu referans noktasıyla gerçek zamanlı verileri karşılaştırır. Normalden sapmalar, olası saldırı girişimlerini veya diğer anormal aktiviteleri işaret edebilir.

Anomalilerin Değerlendirilmesi: Sistem, tespit edilen her bir anormalliği değerlendirir ve bu anormalliğin gerçek bir saldırı girişimi olup olmadığını belirlemeye çalışır. Bu

değerlendirme genellikle insan denetimi veya otomatik karar verme algoritmaları tarafından yapılır.

Uyarı ve Yanıt: Sistem, potansiyel bir saldırı tespit edildiğinde, bir uyarı üretir ve uygun bir yanıt mekanizması devreye girer. Bu yanıt, saldırıyı durdurmak, etkilerini azaltmak veya diğer koruma önlemlerini uygulamak için alınabilir.

Davranış temelli saldırı tespit sistemlerinin avantajları aşağıda verilmiştir:

Esneklik: Yeni ve bilinmeyen saldırı türlerini tespit etme konusunda daha esnek olabilirler, çünkü belirli saldırı kalıplarına dayanmazlar.

Yenilikçilik: Anomalileri tanımlamak için model tabanlı yaklaşımlar kullanırlar, bu da daha yenilikçi ve gelişmiş saldırıları algılama yeteneği sağlar.

Doğruluk: Normal işletme durumlarının modellemesi, genellikle doğru saldırı tespiti sağlamaktadır.

Verilen avantajlara rağmen davranış temelli sistemlerin bazı sınırlamaları da vardır. Bu sınırlamalar:

Eğitim ve Ayarlama Gereksinimi: Model oluşturma ve ayarlama süreci, doğru sonuçlar elde etmek için dikkatli bir şekilde yapılmalıdır.

Yanlış Alarm Riski: Normalden sapmaların yanlışlıkla saldırı olarak tanımlanması riski vardır. Bu yanlış tespitler, gereksiz uyarılar ve güvenlik ekiplerinin zamanının boşa harcanmasına neden olabilir.

Sonuç olarak davranış temelli saldırı tespit sistemleri, bilinen saldırı imzalarıyla sınırlı kalmadan daha geniş bir saldırı yelpazesini tespit etme yeteneği ile önemli bir güvenlik aracı olmaktadır.

Günümüzde, saldırı tespit sistemleri olarak genellikle imza temelli ve davranış temelli yaklaşımların bir kombinasyonu kullanılmaktadır. Bu kombinasyon, saldırı tespit yeteneğini artırırken, aynı zamanda yanlış pozitif sonuçları azaltmaya yardımcı olmaktadır. Saldırı tespit sistemlerinin performansını artırmak için makine öğrenimi ve yapay zeka teknikleri de yaygın olarak kullanılmaktadır. Bu teknikler, büyük veri kümelerini analiz ederek daha karmaşık ve gelişmiş saldırı kalıplarını tespit etmeye yardımcı olmaktadır (Özger, 2023).

2.2. Siber Saldırı Türleri

2.2.1. Kötü Amaçlı Yazılım

Kötü amaçlı yazılım ifadesi, kullanıcının zararına ve saldırganın faydasına olan programları kapsayan genel bir terimdir (A. Aydın vd., 2018). Bu yazılımlar, cihazların güvenlik açıklarını kullanarak izinsiz giriş yaparlar ve çeşitli tekniklerle güvenlik kontrollerini atlayarak kendilerini bir sisteme ya da cihaza yüklerler. Kötü amaçlı yazılımın çeşitli türleri bulunmaktadır. Bunlar yazılımlar sadece kullanıcıları aldatmakla kalmaz, aynı zamanda güvenlik önlemlerini aşmak için özel olarak tasarlanmış teknikleri kullanırlar (Tüfekci ve Önal, 2024).

- Fidye yazılımı olarak bilinen şantaj yazılımı, kullanıcının bilgisayarını kilitlemek ve bilgisayara yeniden erişim sağlamak için zorla ödeme almak üzere tasarlanmıştır.
- Truva atları, genellikle e-posta ekleri veya ücretsiz indirmeler yoluyla gizlice iletilir ve daha sonra kullanıcının cihazına yüklenir. Bu kötü amaçlı yazılım, hassas kullanıcı verileri de dâhil olmak üzere çeşitli türde bilgileri toplama yeteneğine sahiptir.
- Bir yazılım biçimi olan casus yazılım, saldırganın kurban bilgisayarın işlemleriyle ilgili gizli ayrıntıları elde etmek için bilgisayarın sabit diskinden gizlice veri almasına olanak tanımaktadır. Ayrıca bu casus yazılım, tuş vuruşu kaydedici olarak çalışma ve hassas bilgilerin ekran görüntülerini yakalama yeteneğine sahip olabilmektedir.

2.2.2. Servis Hizmet Reddi (DoS – Denial of Service) Saldırıları

Bir DoS saldırısının amacı, ağ bağlantılı bir hizmeti aşırı talep ile yükleyerek kullanılamaz hale getirmektir. Hizmet sağlayıcıya yönlendirilen bu sayıca fazla kötü niyetli istekler, sunucunun belirli bir noktadan sonra yanıt veremez hale gelmesine neden olmaktadır (Hasan vd., 2021). Bu saldırılar, hedeflenen sistem üzerindeki kaynakları tüketerek, normal kullanıcıların erişimini engelleyerek veya önemli ölçüde yavaşlatarak, hizmetin kesintiye uğramasına neden olabilir.

DOS saldırıları, siber güvenlik araştırmacıları ve sistem yöneticileri için önemli bir endişe kaynağı olmaktadır. Çünkü etkili bir saldırı, hizmetin erişilemez hale gelmesine ve hizmet sağlayıcının itibarına zarar vermesine neden olur. Bu nedenle, DOS saldırılarını önlemek ve etkisini en aza indirmek için çeşitli güvenlik önlemleri ve siber savunma stratejileri geliştirilmiştir.

2.2.3. Dağıtılmış Hizmet Reddi (DDoS) saldırıları

Dağıtılmış Hizmet Reddi (DDoS) saldırısı, belirli bir hedef servisi, ele geçirilmiş kullanıcılar ile erişilemez hale getirmeyi amaçlayan bir saldırı biçimidir. Bu saldırı türü, bir sunucuya, web sitesine veya diğer ağ kaynaklarına yönelik, güvenliği ihlal edilmiş çok sayıda bilgisayar sisteminin planlı kullanımını içerir. Saldırı, hedef sistemi aşırı miktarda mesaj, bağlantı isteği veya hatalı biçimlendirilmiş paketlerle doldurarak sistemi etkili bir şekilde yavaşlatır veya çökertir. Sonuç olarak, meşru kullanıcılar veya sistemler hizmete erişememekte ve dolayısıyla onları hizmetin faydalarından mahrum bırakmaktadır (Banitalebi Dehkordi, Soltanaghaei ve Boroujeni, 2021).

2.2.4. Kimlik Avı

Bir dolandırıcılık planı olan kimlik avı eylemi, bir saldırganın e-posta gibi çeşitli iletişim kanalları aracılığıyla banka, tanınmış şirket ve hatta bir birey gibi saygın bir varlığın kimliğini üstlenmesini içermektedir. Saldırganın amacı, zararlı bağlantıları veya eklentileri yaymak ve nihai hedefi, şifreler, kredi kartı bilgileri, fikri mülkiyet ve daha fazlası dâhil olmak üzere hassas verileri ifşa ederek hedefi kandırmaktır (Savaş ve Savaş, 2022).

Belirli bireyleri veya kuruluşları hedef alan dolandırıcılık girişimleri, hedef odaklı kimlik avı saldırıları olarak bilinir. Balina avcılığı saldırıları ise yalnızca bir kuruluş içindeki üst düzey yöneticilere odaklanmaktadır. Balina avcılığı saldırısının özel bir biçimi, iş e-postalarının kontrolünün ele geçirilmesini içerir. Failler, finansal işlemleri yetkilendirme yetkisine sahip belirli çalışanları tespit edip hedef alarak, fonları kendi kontrolleri altındaki bir hesaba yönlendirmeleri için onları kandırırlar.

2.2.5. SQL Enjeksiyon (SQL Injection) Saldırıları

SQL injection saldırıları yöntemi, veri tabanındaki belirli komutları yürütmek amacıyla kullanıcılar tarafından sağlanan giriş alanlarına SQL komutlarının eklenmesini içerir. Bir bilgisayar korsanı, bu veri tabanı sorgularından alınan yanıtı analiz ederek veri tabanının yapısına ilişkin bilgi edinir ve bir uygulama içinde yer alan kısıtlı bilgilere yetkisiz erişim elde eder.

Web sitelerinin çoğunluğu veri tabanlarını içeren işlemlerde bulunduğu için bu durum onları SQL injection saldırılarına karşı savunmasız hale getirmektedir. SQL sorgusu, veri tabanına belirli eylemleri gerçekleştirme talimatı veren özel bir istektir. Dikkatlice oluşturulmuş kötü niyetli istekler, veri tabanında saklanan verilere erişebilir,

bunları değiştirebilir, silebilir veya dışarıya çıkarabilir (Kara vd., 2023). Bu veriler, fikri mülkiyet, müşteri kişisel bilgileri, yönetici kimlikleri veya ticari gizlilik gibi çeşitli hassas bilgileri içerebilmektedir.

SQL injection güvenlik açıkları MySQL, Oracle ve SQL Server gibi çeşitli veri tabanı yönetim sistemlerinde bulunabilir. Bu güvenlik açıkları, ağırlıklı olarak SQL veri tabanlarına dayanan uygulamaları etkileme potansiyeline sahiptir. Kötü niyetli kişiler, SQL injection kullanarak uygulamalar tarafından uygulanan güvenlik önlemlerini ve kimlik doğrulama protokollerini atlayabilir, böylece veri tabanlarına yetkisiz giriş yapabilirler.

SQL injection saldırısı gerçekleştirmek için SQL yöneticisi ile aynı düzeyde uzmanlığa sahip olmak gerekli değildir. Ancak veri tabanının altında yatan mantığın yanı sıra saldırıda yer alan komutların ve adımların arkasındaki mantığın da tam olarak anlaşılması çok önemlidir. Bir saldırı sırasında durumu analiz edebilmek bilgi ve beceri gerektirmektedir.

2.2.6. Siteler arası komut dizisi (XSS) saldırıları

Siteler arası komut dosyası çalıştırma (XSS) saldırıları olarak bilinen zararlı, güvenilmeyen bir kaynak tarafından kötü amaçlı kodun bir web uygulamasına sızması yaygın bir olay olarak görülmektedir. Bu XSS saldırıları, dinamik içeriğe zararlı kodların eklenmesini ve daha sonra kurbanın tarayıcısına gönderilmesini içermektedir. Bu yöntemle saldırganlar, JavaScript, Java, Ajax, Flash ve HTML gibi programlama dillerinde yazılmış kötü niyetli kodları başka bir kullanıcının tarayıcısında çalıştırma yeteneği kazanmış olurlar (Kaur, Garg ve Bathla, 2023).

XSS, saldırganın oturum çerezlerini ele geçirerek, saldırganın bir kullanıcı gibi davranmasına izin verir. Bununla birlikte, kötü niyetli yazılımın yayılması, web sitelerinin içeriğinin değiştirilmesi, sosyal ağlarda karmaşa yaratılması, kimlik bilgilerinin ele geçirilmesi ve sosyal mühendislik teknikleriyle birlikte daha fazla zararlı saldırının gerçekleştirilmesi gibi amaçlarla da kullanılabilir.

2.2.7. Botnet

Botnet, "robot ağı" teriminin kısaltmasıdır ve genellikle siber suçlular tarafından kullanılan bir tür kötü niyetli yazılım ağıdır. Bir botnet, uzaktan kontrol edilen ve yönetilen bir dizi enfekte edilmiş bilgisayar veya cihazın bir araya gelmesiyle oluşur

(Zhang vd., 2020). Bu bilgisayarlar genellikle sahiplerinin haberi olmadan enfekte edilmiştir ve genellikle kötü amaçlı yazılım (bot) tarafından kontrol edilirler.

Botnetler, çeşitli amaçlarla kullanılabilirler. En yaygın kullanımlardan bir kaçı, bilgisayarları spam e-posta göndermek, kimlik avı saldırıları yapmak veya kötü niyetli yazılım dağıtmak gibi eylemler için kullanmaktır (Soe vd., 2020). Botnetler ayrıca tıklama dolandırıcılığı, fidye yazılımlarının dağıtılması veya DDoS saldırıları gibi saldırıları gerçekleştirmek için de kullanılabilirler.

Bir botnet oluşturmanın ana hedefi, mümkün olduğunca çok sayıda cihaza bulaşmak, bu cihazların işlem gücünü ve kaynaklarını kötü amaçlı faaliyetleri gerçekleştirmek için kullanmaktır. Botnetler büyük çaplı ve karmaşık yapılar oluştururlar, genellikle siber suçlular tarafından organize edilir ve yönetilirler.

2.2.8. Kaba Kuvvet Saldırısı

Kaba Kuvvet saldırısı, kriptografi ve şifreleme sistemlerini hedef alarak gerçekleştirilen bir saldırı türüdür. Bu saldırı türünde saldırganlar, şifrelenmiş bir veriyi veya mesajı çözmek için tüm olası anahtarları veya şifreleme seçeneklerini deneyerek kaba kuvvet yöntemini kullanırlar. Kaba Kuvvet saldırısı genellikle güçlü bir bilgi işlem altyapısına ve hesaplama kaynaklarına ihtiyaç duyarlar. Çünkü saldırganlar tüm olasılıkları sırayla denemek zorundadırlar (Kara vd., 2023).

Kaba Kuvvet saldırıları, çeşitli şifreleme algoritmalarına ve anahtar uzunluklarına karşı etkili olabilir. Özellikle, zayıf şifreleme algoritmaları veya kısa anahtar uzunluklarıyla korunan veriler daha savunmasızdır (Zhang vd., 2020). Güçlü şifreleme algoritmaları ve uzun anahtar uzunlukları kullanıldığında, kaba kuvvet saldırılarının etkisi önemli ölçüde azaldığı görülmektedir.

Kaba kuvvet saldırılarına karşı korunmanın en etkili yolu, güçlü şifreleme algoritmaları ve uzun anahtar uzunlukları kullanmaktır. Çok faktörlü kimlik doğrulama gibi ek güvenlik önlemleri de kaba kuvvet saldırılarının etkisini azaltabilir. Kaba kuvvet saldırılarının potansiyel tehdidi göz önünde bulundurularak, şifreleme anahtarlarının ve parolaların güçlendirilmesi ve düzenli olarak güncellenmesi önem arz etmektedir.

2.2.9. DNS Amplification Saldırısı

DNS amplification saldırısı, bir saldırganın, DNS (Alan Adı Sistemi) sunucularını hedef alarak yoğun miktarda istek göndermek suretiyle gerçekleştirilen bir tür DDoS

saldırısıdır. Bu saldırı türünde saldırganlar, genellikle halka açık olan ve büyük miktarda trafik yönlendiren DNS sunucularını hedef almaktadırlar.

Saldırganlar, sahte bir hedef IP adresi kullanarak DNS sunucusuna istek gönderirler ve bu istekler, genellikle DNS sunucularının yanıtlarını büyütmek için istismar edilebilecek genişletilebilir bir protokol olan UDP (User Datagram Protocol) kullanılarak gönderilir (Sanjay ve D, 2020). Saldırganlar, gönderilen her isteğin yanıtının hedefe yönlendirilmesini sağlamak için IP adreslerini sahte olarak değiştirirler.

DNS amplification saldırısı, saldırganların küçük boyutlu istekler göndererek büyük boyutlu yanıtlar almasını sağlar. Bu saldırganların, hedefe karşı büyük miktarda trafik oluşturmasına ve hedefin aşırı yüklenmesine neden olurlar (Adiwal vd., 2023). Sonuç olarak, hedefin internet hizmetlerinin normal işleyişini bozabilir ve kullanıcılara hizmet verme yeteneğini azaltabilir veya tamamen engelleyebilir.

2.3. Siber Güvenlik Tehditlerine Karşı Bulut Bilişiminin Önemi

Son yıllarda, donanım ve fiziksel mekânlardan bağımsız veri saklama imkânı sunan bulut bilişim teknolojisi giderek daha fazla benimsenmektedir. Bu teknolojinin yaygınlaşmasıyla birlikte, iş akışlarında maliyet tasarrufu, erişim hızı, esneklik, verimlilik ve performans açısından birçok avantaj elde edilmektedir. Örneğin, mobilite gerekliliğinin arttığı günümüzde, her yerden erişim önem kazanmakta ve bulut sistemleri üzerinde iş akışlarının yüksek verimlilikle sürdürülmesi kolaylaşmaktadır. Kullandığınız kadar öde prensibiyle çalışan bulut bilişim sistemleri sayesinde, kaynakların kullanımıyla ilgili maliyet tasarrufu sağlanmakta ve bu kullanım kolayca raporlanabilmektedir. Tüm bu avantajların yanı sıra, bulut sistemlerinin sürdürülebilirliğini en az riskle ve en yüksek performansla sağlamak için bulut güvenlik çözümleriyle korunmaları önem arz etmektedir. Bulut bilişimin siber güvenlik açısından önemli avantajları aşağıdaki gibi sıralayabiliriz:

- Profesyonel Güvenlik Uygulamaları: Bulut sağlayıcıları genellikle büyük ölçekli ve uzmanlaşmış bir şekilde güvenlik önlemleri alırlar. Alınmış olan güvenlik önlemleri sayesinde, kullanıcıların kendi başlarına sağlayamayacakları seviyede güvenlik sağlamaktadır.
- Sürekli Güncelleme ve İyileştirmeler: Bulut sağlayıcıları, sistemlerini sürekli olarak günceller ve geliştirirler. Bu iyileştirmeler neticesinde, yeni ortaya çıkan tehditlere karşı daha hızlı ve etkili bir şekilde tepki verebilirler.

- Yedekleme ve Kurtarma Özellikleri: Bulut bilişim, veri yedekleme ve felaket kurtarma işlemlerini otomatikleştirir ve kolaylaştırır. Bu özellik ile veri kaybı durumunda hızlı bir şekilde sistemlerin eski haline getirilmesi mümkün olmaktadır.
- Ölçeklenebilirlik: Bulut bilişim, ihtiyaca göre kaynakların ölçeklendirilmesine imkân tanır. Ani trafik artışları veya yoğunluklarına karşı daha etkili bir şekilde mücadele edilebilme ve saldırıların etkileri en aza indirilebilme imkânı sağlanmış olmuştur.
- Kapsamlı Yetkilendirme ve Erişim Kontrolü: Bulut sağlayıcıları genellikle kapsamlı yetkilendirme ve erişim kontrolü mekanizmalarına sahiptir. Bulut bilişimin bu özelliği sayesinde, yetkisiz erişim girişimleri engellenebilir ve veri güvenliği sağlanabilir.

Bulut bilişim kullanımının getirdiği güvenlik avantajlarına rağmen, kullanıcıların dikkatli olmaları ve güvenlik politikalarına uygun hareket etmeleri önemlidir. Bilinçsiz kullanım veya güvenlik açıklarının ihmali, siber güvenlik risklerini artırabilir ve ciddi sonuçlara yol açabilir. Bu riskler nedeni ile bulut bilişim hizmetlerinin kullanımıyla birlikte güvenlik bilincinin artırılması ve etkili güvenlik politikalarının oluşturulması gerekmektedir.

2.4. Makine Öğrenmesi

Makine öğrenmesi, bilgisayar sistemlerinin veriler üzerinden öğrenme yeteneğine sahip olduğu ve deneyimlerini geliştirebildiği bir yapay zekâ alt alanıdır. Temel amacı, belirli bir görevi gerçekleştirmek için belirli verilerden örüntüler, ilişkiler ve kurallar çıkarmaktır (Janiesch vd., 2021). Bu süreç, bilgisayarların doğrudan programlanmasına gerek kalmadan veri analizi yapabilmesini sağlamaktadır.

Makine öğrenmesi, geniş bir uygulama alanına sahiptir ve otomatik görüntü tanıma, doğal dil işleme, ağ saldırıları, saldırı tespit ve önleme sistemleri, tıbbi teşhis, finansal tahminler ve daha birçok alanda kullanılmaktadır (Meng vd., 2020). Doğru sonuçlar elde etmek için doğru veri ön işleme, model seçimi ve model değerlendirme gibi önemli adımların dikkatlice ele alınması gerekmektedir.

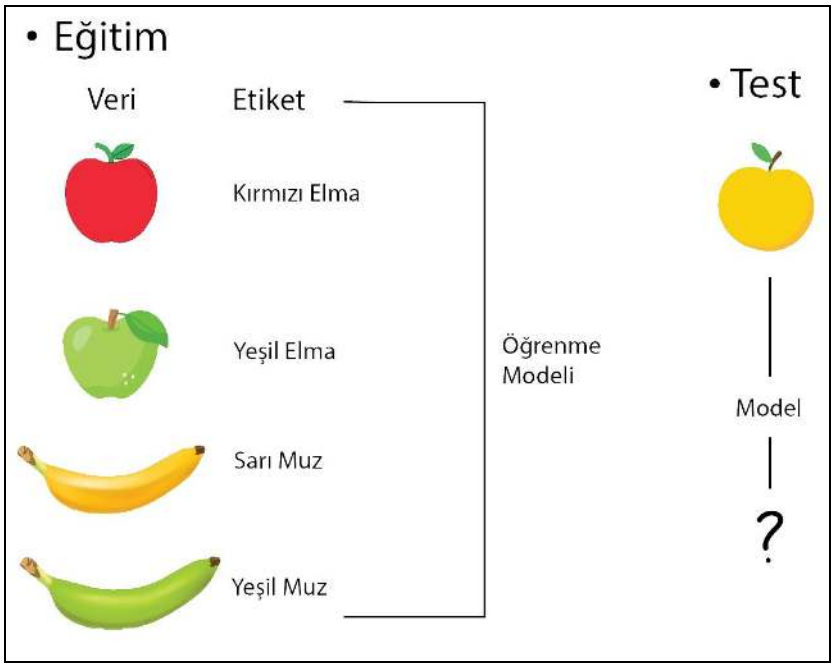
Makine öğrenmesi genellikle denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve güçlendirme öğrenmesi olmak üzere dört ana kategori altında incelenir.

2.4.1. Denetimli Öğrenme

Denetimli öğrenme algoritmaları, veri ve bilgi arasındaki yazışmanın önceden belirlendiği etiketli veri kümelerinin kullanımına dayanmaktadır. Bu algoritmaların

amacı, etiketli eğitim verilerine dayanarak girdi özelliklerini belirli çıktılara etkili bir şekilde bağlayabilen en uygun işlevi tanımlamaktır. Bu algoritmalar, girdi ve çıktı arasındaki ilişkiyi yakalayan en doğru fonksiyonu ortaya çıkararak, yeni girdi verileri için potansiyel çıktıların tahmin edilmesini sağlamaktadır (Shurrab ve Duwairi, 2022). Bu makine öğrenimi modeli, bilinen bir girdi kümesiyle ilişkili çıktılardan yararlanarak, daha önce algoritmayı eğitmek için hiç kullanılmamış verileri en uygun çıktıları üretmek üzere tasarlanmıştır.

Sistem, algoritmayı kullanarak kapsamlı eğitim verilerini toplar ve analiz eder. Bu analiz sonucunda kalıpları, farklılıkları ve mantıksal bağlantıları kademeli olarak ayrıştırır. Örneğin meyve tahmini yapmak için ayrıştırma işlemi sonrasında, "elma mı yoksa muz mu" sorularına benzer sorgulara verilecek doğru yanıtları özerk bir şekilde tahmin edebileceği bir noktaya ulaşır. Şekil 2.1'de meyve tahmini için kullanılan denetimli öğrenme model tasarımı örneklenmiştir. Bu gerçekleşen süreç, çocuğa bir dizi problemin yanı sıra bir cevap anahtarı verilmesine ve daha sonra ondan problem çözme yaklaşımını göstermesinin ve muhakemesini ifade etmesinin istenmesine benzetilebilir. Günün farklı zamanlarına göre en verimli rotayı tahmin eden Waze ve trafik analizi uygulamalarında kullanılan öneri motorları gibi denetimli öğrenme modelleri günlük hayatımızın birçok alanında uygulama alanı bulmaktadır.



Şekil 2.1. Meyve tahmini denetimli öğrenmesi

2.4.2. Denetimsiz Öğrenme

Denetimsiz öğrenme, etiketsiz veriler arasındaki gizli yapıları veya örüntüleri keşfetmeye yönelik bir makine öğrenmesi algoritmasıdır. Bu algoritma, verilerin etiketlenmemiş olduğu ve dolayısıyla elimizdeki verilerin hangi kategorilere veya sınıflara ait olduğu hakkında bilgi sahibi olmadığımız durumlarda kullanılır. Sadece verilerin kendisi bulunmaktadır ve bu verilerden anlamlı sonuçlar çıkarmaya çalışılır. Bu bağlamda, denetimsiz öğrenme algoritmaları, çıktılarını kesin doğrulukla değerlendirmek yerine, keşif ve örüntü tanıma süreçlerinde kullanılır (Alloghani vd., 2020).

Denetimsiz öğrenme modellerinin cevap anahtarı yoktur. Bunun yerine, bu modeller, ellerindeki tüm ilgili bilgileri kullanarak kalıpları ve bağlantıları ortaya çıkarmak için etiketlenmemiş ve yapılandırılmamış girdi verilerine dayanır. Denetimsiz öğrenme birçok açıdan insan gözlemini yansıtır. Benzer şeyleri bir arada gruplamak için sezgi ve deneyime güvendiğimiz gibi, makineler de kategorize etme ve tanımlama yeteneklerini geliştirmek için verilerden yararlanır. Ne kadar çok veriye erişirlerse sınıflandırmaları da o kadar doğru olur. Denetimsiz öğrenme, yüz tanıma, gen dizisi analizi, pazar araştırması ve siber güvenlik gibi çeşitli alanlarda kullanılmaktadır.

2.4.3. Yarı Denetimli Öğrenme

Yarı denetimli öğrenme sıkça kullanılan bir tekniktir. Bu teknikte, bir sınıflandırıcı öncelikle küçük bir etiketlenmiş veri kümesiyle eğitilir. Daha sonra bu sınıflandırıcı, etiketlenmemiş veri üzerinde kullanılarak sınıflandırma denemeleri yapılır. Genellikle en güvenilir görünen etiketlenmemiş noktalar, tahmin edilen etiketleriyle birlikte eğitim kümesine eklenir. Bu süreç tekrarlanarak sınıflandırıcı eğitilmeye devam edilir (Van Engelen ve Hoos, 2020).

Tüm verileri bir sisteme girmeden önce organize etmek ve kategorize etmek ideal olsa da, bu her zaman mümkün değildir. Ancak büyük miktarlarda yapılandırılmamış verilerle uğraşırken geçerli bir çözümüme ulaşmak istiyor isek bunu yarı denetimli öğrenme ile yapmak mümkündür. Bu yaklaşım, etiketlenmemiş veri kümelerini geliştirmek için az miktarda etiketli veri kullanmayı içerir. Etiketli verilerle başlayarak sistem bir avantaj elde eder ve öğrenme hızını ve doğruluğunu büyük ölçüde artırır. Yarı denetimli öğrenme algoritması, makinenin etiketlenmemiş verilere uygulanabilecek ilgili özellikler için etiketli verileri analiz etmesini sağlar. Yarı denetimli öğrenmeyi etkili bir şekilde kullanan şirketler, en iyi uygulama protokollerini oluşturduklarından emin olurlar.

Bu yöntem, konuşma ve dil analizi, protein kategorizasyonu gibi karmaşık tıbbi araştırmalar ve gelişmiş sahtekârlık tespiti dâhil olmak üzere çeşitli alanlarda uygulanmaktadır.

2.4.4. Güçlendirme Öğrenmesi

Bu kategoride, algoritma çevresel etkileşimlerden yola çıkarak belirli bir görevi en etkin biçimde yerine getirmek üzere öğrenme sürecine girmektedir. Algoritma, çevreyi gözlemleyerek ve geribildirimler alarak doğru karar verme yetisini kazanmaktadır. Bu tür öğrenme, çoğunlukla oyun teorisi ve robotik gibi disiplinlerde tercih edilmektedir (Ladosz ve ark. 2022).

Denetimli öğrenme, makinenin doğru sonuçlar arasında ilişkiler kurabilmesi için gerekli cevap anahtarını sunmaktadır. Buna karşın güçlendirme öğrenmesi, bir cevap anahtarına ihtiyaç duymadan çalışmaktadır. Güçlendirme öğrenmesi yöntemi, belirlenmiş bir dizi izin verilen eylem, kural ve potansiyel son durumlar üzerine kuruludur. Algoritmanın hedefi sabit veya ikili olduğunda, makineler örnekler aracılığıyla öğrenme kapasitesine sahiptir. Fakat arzu edilen sonuçların değişebilir olması durumunda, deneyim ve ödül mekanizması yoluyla bilgi edinmeleri gereklidir. Güçlendirme öğrenmesi modellerinde, sistemin toplam verimini artırmak amacıyla algoritmaya sayısal bir ödül entegre edilmiştir. Satranç öğretimi, bu modele birçok yönden benzemektedir. Her olası hamleyi göstermek pratik olmadığından, odak noktası kuralları açıklamak ve pratik yaparak yeteneklerinin geliştirilmesine olanak tanımaktır. Bu sürecin ödüllerinden biri sadece oyunda zafer kazanmakla kalmaz, aynı zamanda rakibin taşlarını ele geçirmek de olabilir.

2.5. Makine Öğrenmesi Algoritmaları

2.5.1. Rastgele Orman

Rastgele Orman (RO) algoritması, denetimli sınıflandırma problemleri için kullanılan bir algoritmadır ve regresyon ile sınıflandırma problemlerini çözmek için uygun bir yapıya sahiptir. RO algoritması, birçok karar ağacı oluşturarak sınıflandırma işleminde doğruluğu artırmayı amaçlamaktadır. RO algoritması, bağımsız olarak çalışan birçok karar ağacının bir araya gelerek en yüksek puanı alan değeri seçtiği bir yöntemdir (Akar ve Güngör, 2012). Ağaç sayısı arttıkça, doğruluk oranı artar ve kesin sonuç elde etme olasılığı yükselir. RO algoritması ile karar ağaçları algoritması arasındaki temel

fark, RO algoritmasında kök düğümü bulma ve düğümleri bölme işleminin rastgele olarak gerçekleştirilmesidir. RO algoritması, büyük miktarda ağaç kullanıldığında aşırı öğrenme problemini azaltma eğilimindedir ve sınırlı bir veri ön işleme gereksinimi duymaktadır.

RO algoritması, farklı veri kümeleri üzerinde eğitim yoluyla genellikle karar ağaçlarıyla ilişkilendirilen aşırı uyum (overfitting) veya aşırı varyans sorununu gidermektedir. Bu yaklaşım, bootstrap yöntemiyle oluşturulan alt veri kümelerinde aykırı değerlerle karşılaşma olasılığını etkili bir şekilde azaltır. RO algoritması, öznitelik önemine ilişkin bilgiler de sunmayı amaçlamaktadır. Belirli sayıda özelliği (x) belirleyip, en değerli olanların (y) seçilmesini isteyerek, istenirse bu bilgiyi diğer modellerde de kullanabiliriz (S. Han vd., 2020). Bu algoritmanın dikkate değer bir avantajı, veri kümenizi daha derinlemesine inceleme ve çeşitli modeller oluşturarak gizli kümeleri ortaya çıkarma yeteneğidir.

2.5.2. K En Yakın Komşu

K En Yakın Komşu (KNN), denetimli öğrenme algoritmalarından biridir. Sınıflandırma ve regresyon problemlerinde kullanılmaktadır. Temel prensibi, bir örneğin sınıflandırılması veya tahmin edilmesi için, ona en yakın olan k adet eğitim örneğinin etrafındaki sınıf veya değerlerin kullanılmasıdır (Kemalbay ve Alkış, 2021).

Algoritmanın çalışma prensibi oldukça basittir. Bir veri noktası geldiğinde, bu noktanın etrafındaki k en yakın veri noktası belirlenir. Sınıflandırma problemlerinde, bu k en yakın noktaların etiketleri kullanılarak çoğunluk oyu alınır ve test örneği bu oya göre sınıflandırılır (Patro vd., 2020). Regresyon problemlerinde ise, k en yakın noktaların değerleri kullanılarak ortalama veya ağırlıklı bir ortalama hesaplanır ve bu değer tahmin olarak kullanılır.

KNN algoritması, basit ve anlaşılması kolay olmasının yanı sıra, doğrusal olmayan veri yapılarında ve karmaşık karar sınırlarında da iyi performans gösterebilir. Büyük veri kümelerinde ve yüksek boyutlu veri kümelerinde hesaplama maliyeti yüksek olabilir ve bu durum bellek kullanımı artmasına neden olabilir.

2.5.3. Karar Ağacı

Karar Ağacı (KA) algoritması, denetimli öğrenme yöntemlerinden biridir ve sınıflandırma ve regresyon problemlerini çözmek için kullanılır. Temel prensibi, bir veri kümesindeki özelliklerin değerlerine göre karar düğümleri oluşturarak veri kümesini parçalamak ve sonunda belirli bir sonuca (sınıf veya değer) ulaşmaktır. KA algoritması,

bir ağaç yapısı şeklinde modellenir. Her iç düğüm, bir özelliğin değerini test ederken, her yaprak düğüm ise bir sınıf etiketi veya regresyon tahmini içerir. KA algoritması, veri kümesini en uygun şekilde bölen ve kararları en iyi şekilde yapan düğümleri belirlemek için bir öğrenme süreci uygular (Çelik vd., 2022).

KA algoritması, veri kümesindeki özelliklerin bilgisi kazanma oranına göre en önemli özelliği seçer ve bu özelliği kullanarak veriyi ikiye böler. Ardından, her alt veri kümesi için aynı işlemi tekrar ederek ağacı derinleştirir. Bu bölme işlemi, veri kümesi tamamen sınıflandırıldığında veya belirli bir ağaç derinliğine ulaşıldığında sonlanır. KA algoritması, yüksek derecede yorumlanabilir bir model üretmektedir, Bunun nedeni ise her karar düğümü insan tarafından anlaşılabilir bir şekilde yorumlanabilmesidir (Taha Jijo ve Mohsin Abdulazeez, 2021). Veri ön işleme gereksinimini minimumda tutar ve doğrusal olmayan ilişkileri de modelleyebilir. Aşırı uyuma (overfitting) eğilimli olabilir ve bazı durumlarda karmaşık modeller oluşturabilir.

2.5.4. Destek Vektör Makinesi

Destek Vektör Makinesi (DVM), denetimli öğrenme algoritmalarından biridir ve sınıflandırma ve regresyon problemlerini çözmek için kullanılır. Temel prensibi, veri noktalarını sınıflar arasında bir karar sınırı oluşturarak ayırmaktır. Bu karar sınırı, veri noktalarının en iyi şekilde ayrılmasını sağlayacak şekilde seçilir ve bu sınırın iki tarafında bulunan destek vektörleri arasındaki mesafe maksimize edilir. Bu sınır, veri noktalarının bulunduğu çok boyutlu uzayda bir hiperdüzlem olarak düşünülebilir. DVM, bu hiperdüzlemi bulmak için özellik vektörleri arasındaki ilişkileri analiz eder ve en iyi sınıflandırma sonucunu veren hiperdüzlemi belirler (Altuntas vd., 2020).

DVM, yüksek boyutlu veri kümeleriyle iyi çalışabilir ve çoklu sınıflandırma problemlerine uygun olarak genişletilebilir. DVM algoritmasının çekirdek yöntemleri sayesinde doğrusal olmayan ilişkileri modelleyebilir ve veriye dayalı karmaşık karar sınırları oluşturabilir.

2.6. Öznitelik Seçimi

Öznitelik seçimi, analiz edilen örneği doğru bir şekilde temsil etmek için belirli kriterlere göre belirli bir kümeden en uygun özelliklerin seçilmesini içermektedir. Bir başka açıdan özellik çıkarımı, uygulama sürecinin bir parçası olarak yeni bir özellik alanı oluşturmak için mevcut bilgilerin dönüştürülmesini gerektirir (Küçüksille ve Ateş, 2016).

Çalışmada toplam 79 farklı özellik içeren bir veri kümesi kullanılmıştır. Amacımız, çalışmada kullanıma sunulacak saldırı tespit sistemindeki ağ trafiğini yakından izleyerek saldırı tespitinin etkinliğini arttırmaktır. Çalışma süresini en aza indirmek ve daha az öznitelikle yüksek doğruluk oranına ulaşmak için toplam 79 farklı özelliğin sayısını azaltmak gerekmektedir. Bu azaltma prosedürü boyunca, sınıflandırma sürecinde en iyi sonuçların sağlanması için sınıflandırma için en önemli özelliklerin korunmasına öncelik verilecektir. Sınıflandırma üzerinde minimum etkiye sahip olan nitelikler veri kümesinden çıkarılacak ve bunların hariç tutulması sağlanacaktır.

Önceki araştırmalar, saldırı tespitinin etkinliğinin, saldırı türüne göre belirli özellik gruplarının seçilmesiyle artırılabileceğini göstermiştir. Örneğin, belirli öznitelikler bir DDoS saldırısının belirtilerini gösterirken, farklı öznitelik grupları bir Kaba Kuvvet saldırısının belirtilerini ortaya koymaktadır. Bu çalışma öncelikle belirli saldırı türlerini sınıflandırmaktan ziyade saldırının oluşumunu belirlemeye odaklanmıştır. Eldeki verilere yoğun bir ilgiyle incelenen amaç, veri kümesindeki tüm saldırı türlerinde çok çeşitli ortak belirtiler sergileyen veya saldırı türlerinin çoğunda belirtiler gösteren özellikleri ortaya çıkarmaktır. Bu strateji doğrultusunda, çeşitli özellik grupları dikkatle seçilmiş ve hem makine öğrenimi metodolojileri hem de manuel yaklaşımlar kullanılarak kapsamlı sınıflandırma analizleri gerçekleştirilmiştir.

2.6.1. Özyinelemeli Özellik Seçimi

Araştırmada, saldırıları tespit etmek için en önemli özellikleri belirlemek amacıyla çeşitli yöntemler kullanıldı ve bunların sonuçları değerlendirildi. Başlangıçta bir özellik seçme yöntemi olan Özyinelemeli Özellik Seçimi (Recursive Feature Elemination-RFE) tekniği kullanıldı. Bu teknik, tercih edilen denetimli öğrenme yaklaşımını kullanarak veri kümesini alt kümelere bölmeyi içermektedir.

RFE süreci, her bir alt kümenin değerlendirilmesini ve en az değerli özelliğin belirlenmesini ve daha sonra kümeden çıkarılmasını içerir. Bu yinelemeli süreç istenilen sayıda özellik elde edilene kadar devam etmektedir. Bu tekniğin temel prensibi, veri kümesindeki tüm nitelikleri alt kümelere bölmek, önem derecelerini puanlamak ve en anlamlı olanları belirlemek için en az önemli olan nitelikleri ortadan kaldırmaktır. Süreç boyunca, yalnızca en önemli özellik kalana kadar tekrarlanır ve önem derecesini belirlemek için denetimli bir öğrenme algoritması kullanılır.

2.6.2. İleri Yönelimli Seçim

İleri Yönelimli Seçim tekniği (Forward Selection-FS), önceden belirlenmiş bir eşik değeri p'ye dayalı olarak başlangıçta boş bir kümeden kademeli olarak özelliklerin seçilmesini içermektedir. Özellik seti, özelliklerin değerlerinin sıradan en küçük kareler yöntemi kullanılarak hesaplanmasıyla belirlenir. Kullanılan hesaplama yöntemi, bir veya daha fazla açıklayıcı değişkeni birleştiren ve bunları hataların karelerinin toplamı ile birleştiren sıradan En Küçük Kareler (Ordinary Least Squares-OLS) modelidir (Bro vd., 2002). FS süreci, sürekli veya aralıklı bir sonuç değişkeninin tahmin edilen ve gerçek değerleri arasındaki farklılığın en aza indirilmesini ve böylece bir bağlantı kurulmasını içermektedir. Bu adım adım prosedür, eşik değerini aşanları vurgulayarak veri kümesindeki en önemli özellikleri tanımlar (Acito, 2023).

FS tekniğinin daha yakından incelenmesi üzerine, başlangıçtaki özellik kümesinin boş kabul edilebileceği açıkça ortaya çıkmaktadır. Bu boş kümenin ilk elemanı, tüm nitelikler ile hedef değişken veya sınıf etiketi arasındaki ilişkinin değerlendirilmesi ve en büyük etkiye sahip olan veya hedef değişkenin en doğru tanımını sağlayan özelliğin seçilmesiyle belirlenir. Daha sonra, seçilen bu özelliğe hangi ek özelliğin eklenmesi gerektiği, özellikle seçilen kümeden hangi özelliğin hedef değişkeni en etkili şekilde tamamladığı sorusu ortaya çıkmaktadır. Bu süreç, en iyi performansı sağlayan özelliğin belirlenmesi ve eklenmesi amacıyla tüm özellikler için tekrarlanır. Bu yinelemeli süreç, istenen nitelik kümesi elde edilene kadar devam eder. Özellik kümesine üçüncü öğenin eklenmesini ve hedef değişkeni etkili bir şekilde tanımlayan üç özelliğin optimal kombinasyonunun belirlenmesini içermektedir. Bu yinelemeli süreç, tüm özellikler "p" olarak belirtilen eşik değerinin altına düşene kadar devam eder.

"p" olarak adlandırılan kullanıcı tanımlı değişken, bahsedilen eşik değerini temsil etmektedir. Bu değeri belirlemek için evrensel olarak kabul edilen bir yöntem olmasa da istatistiksel öneme sahiptir. Hesaplamalar dikkate alınınca genel kabul gören değer 0,05 olarak belirlendi. Bu aynı zamanda SPSS (Statistical Package for the Social Sciences) yazılımında ilgili olasılığı belirlemek için kullanılan standart kriterdir. Bu eşiği aşan herhangi bir değişkenin kabul edilebilir aralığın üzerinde olduğu kabul edilir. 0,01'in altındaki değerler istatistiksel olarak anlamlı kabul edilir, ancak istatistiksel önem açısından istatistiksel olarak anlamsızdır (Alfa, 2024). FS tekniği kullanılarak özellik seçme işlemi 30 nitelik üzerinde bir kez daha gerçekleştirildi. Eşik değeri p'nin 0,05'e ayarlanmasıyla nitelik sayısı daha sonra 30'dan 26'ya düşürüldü.

2.6.3. Manuel Seçim Yöntemi

Sınıflandırma başarısı, hem araştırmada önerilen hibrit yaklaşımlı özellik seçme yönteminden hem de daha fazla sayıda deney yapılarak farklı özellik gruplarının seçilmesinden etkilenebilir. Bu etkiyi incelemek için belirli özellik grupları seçildi ve hem manuel seçim hem de makine öğrenme teknikleri kullanılarak sınıflandırma testlerine tabi tutuldu. Manuel özellik seçimi, önceki çalışmalarda yaygın olarak kullanılan özelliklerin, ağ bilgilerine göre önemli sayılan özelliklerin ve ön çalışma aşamasında nispeten daha yüksek başarı oranları gösteren özelliklerin kullanılmasını içermektedir.

Niteliklerin önemi, dikkate değer niteliklerin bir araya getirilmesiyle oluşturulan grupları dikkate alan çeşitli çalışmalarda incelenmiştir. Ayrıca CICIDS 2017 veri kümesindeki sınıf etiketlerine göre en öne çıkan özellikleri belirleyen bir çalışma yürütmüştür (Shams, 2020). Çizelge 2.1'de atak sınıflarına göre önemli öznitelikler görülmektedir.

Çizelge 2.1. Sınıflara göre en önemli öznitelikler (Shams, 2020)

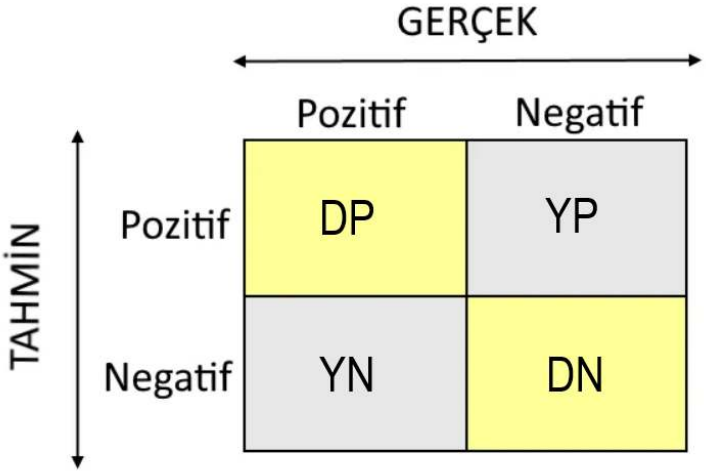
Sınıf	Öznitelik
DDoS	Bwd Packet Length Std, Total Backward Packets, Fwd IAT Total, Total Length of Fwd Packets
DoS Hulk	Flow Bytes/s, Total Length of Fwd Packets, Fwd Packet Length Max, Flow IAT Mean
DoS GoldenEye	Flow IAT Max, Bwd Packet Length Std, Flow IAT Min, Total Backward Packets
DoS slowloris	Flow IAT Mean, Total Length of Bwd Packets, Bwd Packet Length Mean, Total Fwd Packets
DoS Slowhttptest	Flow IAT Mean, Fwd Packet Length Std, Fwd Packet Length Mean, Fwd Packet Length Min
DDoS	Bwd Packet Length Std, Total Backward Packets, Fwd IAT Total, Total Length of Fwd Packets
FTP-Patator	Fwd Packet Length Max, Fwd Packet Length Std, Fwd Packet Length Mean, Bwd Packet Length Std
SSH-Patator	Fwd Packet Length Max, Flow IAT Mean, Flow IAT Max, Total Length of Fwd Packets
Bot	Bwd Packet Length Mean, Flow Duration, Flow IAT Std, Flow IAT Min
Web Attacks	Bwd Packet Length Std, Flow IAT Min, Total Length of Fwd Packets, Bwd Packet Length Max
Infiltration	Fwd Packet Length Mean, Total Backward Packets, Total Length of Fwd Packets, Flow Duration
Heartbleed	Bwd Packet Length Mean, Bwd Packet Length Max, Total Length of Fwd Packets, Fwd Packet Length Max

Araştırmacılar manuel olarak seçilen özellik gruplarını oluştururken ilgili çalışmalardan elde edilen verileri dikkate almışlardır. Çalışmanın odak noktası ikili sınıflandırma olduğundan özellik grupları bu amaç için özel olarak seçilmiştir. Öznitelik grupları aynı şekilde kullanılmasa da, bu verilerin de gösterdiği gibi, gruplar içindeki paylaşılan özellikler ve deneylerde olumlu sonuçlar veren özellikler, manuel özellik gruplarının belirlenmesinde önemli bir rol oynamıştır.

2.7. Performans Değerlendirme Kriterleri

2.7.1. Karmaşıklık Matrisi

Bu tez çalışmasında kullanılan sınıflandırıcıların performansı, karışıklık matrisi olarak da bilinen karmaşıklık matrisi kullanılarak değerlendirilmiştir. Karmaşıklık matrisi, farklı sınıfların birbirinden ne kadar etkili bir şekilde ayırt edildiğini değerlendiren ve algoritmanın sınıf etiketlerini doğru bir şekilde tahmin etme becerisine dair fikir veren bir tablodur. Sınıfların "pozitif" ve "negatif" olarak etiketlendiği ikili sınıflandırma senaryosunda, karmaşıklık matrisi Şekil 2.2'de gösterilen yapıyı takip etmektedir.



Şekil 2.2. Karmaşıklık matrisi

Her satır tahmin edilen sınıfı temsil ederken, her sütun gerçek sınıfı temsil eder. Matris, pozitif sınıf için doğru tahminlerin sayısını gösteren doğru pozitifler (DP), pozitif sınıf için yanlış tahminleri temsil eden yanlış pozitifler (YP), negatif sınıf için yanlış tahminleri ifade eden yanlış negatifler (YN) ve doğru tahmin ettiği negatifler içinde doğru negatifleri (DN) içermektedir. Karmaşıklık matrisi, bir sınıflandırma algoritmasının

etkinliğini anlamada değerli bir kaynak görevi görmektedir ve doğru veya yanlış sınıflandırılmış örneklerin sayısının net bir dökümünü sağlamaktadır.

Karmaşıklık matrisinden, algoritma performansının değerlendirilmesine yardımcı olan ve farklı modeller arasındaki karşılaştırmaları kolaylaştıran çeşitli değerlendirme ölçümleri türetilebilir. Çizelge 2.2'de karmaşıklık matrisine dayalı olarak bu ölçümleri hesaplamak için kullanılan formüller özetlenmektedir.

Çizelge 2.2. Karmaşıklık matrisi formülizasyonu

Ölçüt	**Formül**
Doğruluk (Accuracy)	$\frac{\mathrm{DP} + \mathrm{DN}}{\mathrm{DP} + \mathrm{DN} + \mathrm{YP} + \mathrm{YN}}$
Kesinlik (Precision)	$\frac{\mathrm{DP}}{\mathrm{DP} + \mathrm{YP}}$
Duyarlılık (Recall)	$\frac{\mathrm{DP}}{\mathrm{DP} + \mathrm{YN}}$
F1 Skoru (F1 Score)	$2 * \frac{\mathrm{Kesinlik} * \mathrm{Duyarlılık}}{\mathrm{Kesinlik} + \mathrm{Duyarlılık}}$

- Doğruluk ölçüsü, yapılan tahminlerin toplam sayısına göre doğru tahminlerin oranıyla ilgilidir. Gerçek pozitiflerin sayısının gerçek negatiflerin ve tahminlerin toplamına bölünmesiyle belirlenir.
- Kesinlik, gerçek pozitiflerin, gerçek pozitiflerin ve yanlış pozitiflerin toplamına oranıyla belirlenir.
- Duyarlılık olarak bilinen gerçek pozitif vakaları tahmin etmedeki doğruluk, gerçek pozitiflerin sayısının, gerçek pozitiflerin ve yanlış negatiflerin toplamına bölünmesiyle belirlenir.
- F1 Skroru kesinlik ile duyarlılık arasındaki dengeyi ifade eder. Bu iki faktörün harmonik ortalaması ile bulunur.

2.7.2. Çapraz Doğrulama

Çapraz doğrulama olarak bilinen yöntem, sınıflandırmanın güvenilirliğini artıran bir araç olarak hizmet etmekte ve ayırt edicilik açısından daha yüksek bir başarı düzeyi sağlamaktadır. Bu yöntemde, veri kümesi parçalara ayrılarak, bu parçalardan farklı eğitim

ve test veri kümeleri oluşturulur. Oluşturulan her bir veri kümesi için modelin doğruluğu hesaplanır. Bu hesaplama işlemi, veri kümesinin farklı bölümlerinden eğitim ve test kümeleri seçilerek tekrarlanır. Elde edilen sonuçlar daha genelleştirilebilir yorumlar yapmak için daha uygun hale gelmektedir.

2.8. Sentetik Azınlık Aşırı Örnekleme Tekniği

Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE), azınlık sınıfı için sentetik örnekler üreten ve rastgele aşırı örneklemenin ortaya çıkardığı aşırı uyum sorununun üstesinden gelmeye yardımcı olan bir aşırı örnekleme tekniğidir (Wang, 2020). Bu algoritma özellik uzayında çalışır ve birbirine yakın olan mevcut pozitif örnekler arasında enterpolasyon yoluyla yeni örnekler üretir. Enterpolasyon terimi, veri noktaları arasındaki bilinmeyen bir değeri tahmin etmek veya bulmak için kullanılan bir yöntemdir.

SMOTE algoritması, tipik olarak ikili sınıf dağılımını dengelemek için seçilen toplam aşırı örnekleme gözlem sayısını (N) ayarlayarak başlar. Algoritma daha sonra iteratif olarak rastgele bir pozitif sınıf örneği seçer, uygun bir mesafe metriği kullanarak K adet en yakın komşusunu belirler ve seçilen örnek ile komşuları arasında enterpolasyon yaparak sentetik örnekler oluşturur.

SMOTE dengesiz veri kümelerini ele almak için güçlü bir teknik olsa da, avantajlarını ve sınırlamalarını anlamak önemlidir. SMOTE kullanmanın önemli avantajlarından biri, dengeli veri kümeleri oluşturmaya yardımcı olarak makine öğrenimi modellerinin doğruluğunu artırmasıdır. SMOTE tekniği, azınlık sınıfının gerçekçi temsilleri olan sentetik örnekler üreterek modelin önyargılı olmamasını sağlar. SMOTE, özellikle gürültülü verilerle uğraşırken veya azınlık sınıfı önemli ölçüde küçük olduğunda tüm senaryolar için uygun olmayabilir (Kaur ve Gupta, 2023). SMOTE uygulanırken dikkatli değerlendirme ve veri kümesi özelliklerinin dikkate alınması çok önemlidir.

2.9. Kaynak Özetleri

Saldırı tespit sistemleri ve bu sistemlerde yapay zekanın kullanımı konusunda yapılmış çalışmalar literatürde yer almaktadır.

Sarkar ve arkadaşlarının çalışmasında ağa izinsiz veri girişinin tespitinde makine öğrenmesine dayalı toplu öğrenme yöntemi önerilmiştir ve hiperparametre optimizasyonu temelli çalışmalara örnek olarak gösterilebilir. NSL-KDD ve KDD Cup99 veri kümeleri üzerinde gerçekleştirilen çalışmada, veri dengeleme yöntemlerinin karşılaştırması yapılmıştır. Bu bağlamda, hibrit Destek Vektör Makinesi-SMOTE ve

Rasgele Azınlık Örnekleyici (Random Under Sampler) tekniklerinin en etkili veri dengeleme teknikleri olduğu belirlenmiştir. Elde edilen bulgulara göre veri düzenleme tekniklerinin, hiperparametre optimizasyonu gibi sistem performansını artıran unsurlar olduğunu vurgulanmaktadır. Araştırmada, sinir ağları için katman sayısı, döngü sayısı, demet büyüklüğü, optimize edici gibi kritik hiperparametreler üzerinde optimizasyon işlemi gerçekleştirilmiştir. Sonuç olarak, çalışma %89,32 tahmin başarısı elde etmiştir (Sarkar vd., 2023).

Han ve arkadaşları tarafından yürütülen araştırmada, Yapay Sinir Ağı (YSA) özellik çıkarıcısı kullanılarak ve kontrol edilebilir K değişkeni ile entegre bir şekilde eğitilen bir model geliştirilmiştir. Bu model, ağ saldırılarını tespit etmek amacıyla hiperparametre optimizasyonunu kullanmaktadır. Söz konusu model, UNSW-NB15 ve CICIDS-2017 veri kümeleri üzerinde değerlendirilmiş, hiperparametre optimizasyonu ile sağlanan ayarlamaların model performansını önemli ölçüde iyileştirdiği gözlemlenmiştir. Gerçekleştirilen testler sonucunda, UNSW-NB15 veri kümesi için %94,26 ve CICIDS-2017 veri kümesi için %96,55 F1 skoru elde edilmiştir. Bu sonuçlar, önerilen modelin tahmin başarısının yüksek olduğunu ve pratik uygulamalar için etkili bir alternatif sunabileceğini ortaya koymaktadır. (Han vd., 2022).

Araç özel ağlarında DDoS saldırılarını tespit etmek amacıyla yapılan araştırmada, verimli ve konforlu bir trafik deneyimi sağlamak için çeşitli makine öğrenmesi yöntemleri kullanılmıştır. Çalışma kapsamında, özellik seçimi için Minimum Artıklık Maksimum Uygunluk algoritması ve hiperparametre ayarları için Bayes optimizasyon yöntemi uygulanmıştır. Bu süreçler, farklı makine öğrenmesi algoritmalarının performansını artırmayı amaçlamaktadır. Özellikle, Karar Ağaçları yöntemi kullanılarak gerçekleştirilen özellik seçimi ve hiperparametre optimizasyonu sonucunda %99,35 oranında tahmin başarısı elde edilmiştir. Bu bulgular ile söz konusu tekniklerin DDoS saldırılarının tespitindeki etkinliğini göstermektedir ve araç ağlarının güvenliğini artırmada potansiyel bir yol sunmaktadır. (Türkoğlu vd., 2022).

Siva ve arkadaşları tarafından gerçekleştirilen çalışmada, ağ saldırılarını tespit etmek amacıyla yapay zekâ tekniklerine dayanan bir model önerilmiştir. Bu modelin performansını artırmak için, veri kümesinin hazırlanması, öznitelik seçimi ve hiperparametre optimizasyonu gibi çeşitli adımlar uygulanmıştır. Önerilen sistem, öz yapımı destekleyen bulanık sistemler kullanarak ağ saldırılarını tespit etmeyi amaçlamaktadır. Çalışma sonuçlarına göre, UNSW-NB15 ve NSL-KDD veri kümelerinde sırasıyla %99,97 ve %99,99 gibi yüksek doğruluk oranları elde edilmiştir.

Bu sonuçlar, modelin ağ saldırılarını tespit etmede son derece etkili olduğunu göstermektedir (Siva Shankar vd., 2024).

Ye ve arkadaşları, kontrol cihazı aracılığıyla ağ trafiği bilgilerini toplamış ve bu verileri kullanarak DVM algoritmasıyla DDoS saldırılarına ilişkin altı karakteristik özellik çıkarmışlardır. Bu özellikler, saldırı tespiti için kullanılmıştır. Araştırmanın sonuçları, saldırı tespitinde yüksek bir başarı oranı elde edildiğini göstermiştir. Çalışmada özellikle ICMP protokolü tabanlı DDoS saldırılarının tespitindeki doğruluk oranının, diğer saldırı türlerine kıyasla daha düşük olduğu vurgulanmıştır. Bu durum, belirli saldırı türlerine yönelik tespit stratejilerinin optimizasyonu için önemli bir gösterge olarak değerlendirilmektedir. (Ye vd., 2018).

Sanchez ve arkadaşlarının çalışmasında, IoT cihazlarına yönelik DDoS saldırılarını tespit etmek amacıyla CICIDS veri kümelerini kullanarak bir araştırma gerçekleştirmiştir. Çalışmada, birden fazla makine öğrenme algoritmasını içeren modeller üzerinde detaylı bir hiperparametre optimizasyonu uygulanmıştır. Grid arama yöntemi ile gerçekleştirilen hiperparametre optimizasyonu sonucunda, Rasgele Orman ve Karar Ağaçları algoritmaları ile %99,98 tahmin başarısı elde edilmiştir. Bu bulgular ışığında, hiperparametre optimizasyonunun makine öğrenme algoritmalarının performansını önemli ölçüde artırdığını göstermektedir. Hiperparametre seçeneği bulunmayan Naive Bayes algoritması, diğer algoritmalarla karşılaştırıldığında daha düşük performans sergilemiştir. (Sanchez vd., 2021).

Kim tarafından yürütülen bu araştırmada, DDOS saldırılarını tespit etmek amacıyla Yapay Sinir Ağları (YSA) ve Uzun Kısa Vadeli Bellek ile Rekürrent Sinir Ağları (LSTM-RNN) teknikleri kullanılmıştır. Çalışma, model performansını etkileyen hiperparametre ayarları ve veri ön işleme yöntemlerini incelemiştir. CAIDA ve DARPA veri kümeleri üzerinde gerçekleştirilen deneyler, YSA için tek katmanlı yapının, maliyet ve eğitim süresi açısından birden fazla katmanlı yapıya göre daha etkili olduğunu ortaya koymuştur. Optimize edicinin öğrenme hızını doğrudan etkilediği belirlenmiştir. LSTM-RNN kullanımı, süre maliyeti açısından YSA'ya göre daha yüksek olmasına rağmen, modeldeki gerçek pozitif oranlarının daha kabul edilebilir düzeylerde olduğunu göstermiştir. Pozitif tahmin değerleri YSA'ya kıyasla daha düşük bulunmuştur. Araştırma sonuçları, LSTM-RNN algoritmasında dönem sayısının artırılmasıyla pozitif tahmin oranının yükseltilebileceğini ortaya koymaktadır. (Kim, 2019).

Tank ve arkadaşları tarafından gerçekleştirilen çalışmada, veri ön işleme teknikleri arasında min-max normalizasyonu, kategorik verilerin one hot encoding yöntemiyle

kodlanması ve LightGBM algoritmasının uygulanması yer almaktadır. Bu ön işlemler gerçekleştirildiğinde, Autoencoder (AE) tabanlı derin öğrenme modeli kullanılarak oluşturulan veri kümesi üzerinde yapılan deneyler, modelin %89,82 oranında bir doğruluk değerine ulaştığını göstermiştir. Alınan bu sonuç, söz konusu yöntemlerin veri kümesinin işlenmesindeki etkinliğini ve AE algoritmasının yüksek doğruluk potansiyelini ortaya koymaktadır. (Tang vd., 2020).

Manzini ve arkadaşları tarafından yürütülen çalışmada, NSL-KDD ve ISCXIDS2012 veri kümelerine yönelik olarak kategorik veri kodlaması ve ölçeklendirme ön işlemleri uygulandıktan sonra, özellik seçimi için yapay arı kolonisi (ABC) algoritması geliştirilmiştir. Seçilen özelliklerin değerlendirilmesinde ise AdaBoost.M2 algoritması tercih edilmiştir. Yaptıkları bu çalışmada kullanılan AdaBoost.M2, standart AdaBoost algoritmasından farklı olarak, birden fazla sınıflı durumlar için daha uygun olan sahte kayıp kavramını kullanmaktadır. Bu kavram, zayıf hipotezlerin performansını ölçmek için kullanılır ve her yinelemede numune ağırlıklarının sahte kayıplara göre ayarlanmasıyla temel öğrenenlerin tahmin yeteneklerinin arttığı gözlemlenmiştir. Çalışma sonucunda elde edilen verilere göre algılama oranı %99,61, yanlış algılama oranı %0,01 ve doğru tespit oranı ise %98,90 olarak belirlenmiştir. (Mazini vd., 2019).

Emanet ve arkadaşları tarafından yapılan araştırmada, saldırı tespit sistemlerindeki başarı faktörleri özellikle öznitelik seçimi bağlamında incelenmiştir. Çalışma, çeşitli sınıflandırma algoritmaları üzerinde detaylı analizler yaparak, ki-kare testi ve özyinelemeli öznitelik eliminasyonu gibi yöntemlerin etkinliğini değerlendirmiştir. Bu yöntemlerden elde edilen verilere göre, en yüksek başarı oranı %98,79 ile ekstra ağaçlar modeli kullanılarak sağlanmıştır. Yapılan çalışma neticesinde öznitelik seçiminin, saldırı tespit sistemlerinin performansını önemli ölçüde iyileştirebileceğine dair değerli bir kanıt sunmaktadır (Emanet vd., 2021).

Kanimozhi ve Prem tarafından gerçekleştirilen çalışma, Botnet saldırılarının tespitine yöneliktir. Araştırmacılar, CICIDS-2018 veri kümesi üzerinde yapay zeka tekniklerini kullanarak Botnet saldırılarını tespit etme sürecini incelemişlerdir. Yapılan analizler sonucunda, Yapay Sinir Ağları (YSA) tekniği ile %99,97 oranında doğruluk elde edildiği bildirilmiştir. Bu bulgu, YSA'nın Botnet saldırılarını tespit etme konusunda yüksek etkinlik gösterdiğini göstermektedir. (Kanimozhi ve Prem Jacob, 2019).

Panigrahi ve arkadaşları tarafından yürütülen araştırmada, saldırı tespit sistemlerinde kullanılan çeşitli sınıflandırma algoritmaları ve bu algoritmaların farklı veri kümeleri üzerindeki performansları detaylı bir şekilde analiz edilmiştir. NSL KDD,

ISCXIDS2012 ve CICIDS2017 veri kümeleri üzerinden elde edilen sonuçlar, her bir algoritma için ayrı ayrı değerlendirilmiştir. Özellikle NSL KDD veri kümesinde yapılan analizlerde, Çok Katmanlı Algılayıcı algoritması %80,46, Lojistik Regresyon algoritması %78,04, k-Yakın Komşu (KNN) algoritması %95,01 ve Karar Ağacı algoritması %97,28 doğruluk oranında sınıflandırma başarısı göstermiştir. Bu sonuçlar doğrultusunda, kullanılan sınıflandırma tekniklerinin etkinliğini ve belirli veri kümeleri üzerindeki performans farklılıklarını ortaya koymaktadır. (Panigrahi vd., 2021).

Ekici ve Takcı tarafından CSE-CIC-IDS-2017 veri kümesi üzerinde gerçekleştirilen araştırma, yinelemeli öznitelik elemesi ve ileri yönelimli seçim tekniklerinin kullanımını ele almıştır. Bu teknikler, özellikle Rastgele Orman algoritmasının performansını değerlendirmek üzere uygulanmıştır. Çalışmanın sonuçları, Rastgele Orman algoritmasının, uygulanan öznitelik seçim teknikleri sayesinde %94.00 oranında bir başarı sağladığını göstermiştir. Yinelemeli öznitelik elemesi ve ileri yönelimli seçim tekniklerinin yapılan çalışma sayesinde, etkili bir sınıflandırma performansı elde etmede önemli role sahip olduğunu ortaya koymaktadır. (Ekici ve Takcı, 2022).

Jumabek ve arkadaşları tarafından CSE-CIC-IDS-2018 veri kümesi üzerinde yürütülen araştırmada, veri dengesizliğinin giderilmesi sonrasında elde edilen performans iyileştirmeleri incelenmiştir. Yapılan çalışmada, CatBoost algoritması kullanılarak yapılan analizler %92,41 oranında başarı sağlarken, Rastgele Orman algoritması %89,88 başarı oranı ile performans sergilemiştir. Bu sonuçlar, veri dengesizliğinin giderilmesinin, sınıflandırma algoritmalarının etkinliğini artırabileceğini göstermektedir. (Jumabek vd., 2021).

Dey ve Rahman tarafından yürütülen araştırma, yazılım tanımlı ağ denetleyicilerine yönelik saldırıların tespiti amacıyla geleneksel makine öğrenme algoritmaları ve derin öğrenme modellerinin kullanılmasını içermektedir. Araştırmada ilk olarak, NSL-KDD veri kümesine Rastgele Orman algoritması uygulanmış ve öznitelik seçimi ile doğruluk oranı artırılmıştır. İkinci aşamada ise, Geçitli Tekrarlayan Birim (Gated Recurrent Unit, GRU) tabanlı bir derin öğrenme modeli kullanılarak saldırı tespiti gerçekleştirilmiştir. Elde edilen sonuçların karşılaştırılmasıyla, derin öğrenme tabanlı saldırı tespit yönteminin, geleneksel makine öğrenme yaklaşımlarına kıyasla daha etkili olduğu tespit edilmiştir. (Dey ve Rahman, 2019).

Nanda ve arkadaşları tarafından yapılan çalışmada, güvenlik tehditlerini en aza indirmek amacıyla ağ saldırılarına yönelik veriler üzerinde eğitilmiş çeşitli makine öğrenimi algoritmalarının kullanılmasını önermişlerdir. Çalışmada, DVM, C4.5, Bayes

Ağı, Karar Ağacı ve Naive Bayes gibi algoritmaların etkinliği test edilmiştir. Bu algoritmaların kullanılması, veri düzleminde kötü niyetli kullanıcıların etkin bir şekilde tespit edilmesine olanak tanımaktadır. Yapılan bu tespitler sonucunda DVM kontrolcüsü tarafından ağın etkinliğini ve sürekliliğini korumak için gereken yeni kuralların hızlı ve etkili bir şekilde oluşturulması mümkün hale gelmektedir. (Nanda vd., 2017).

Hsider ve arkadaşları tarafından yürütülen araştırma, dijital çağın yeni teknolojilerle şekillenen dünyasında, gelişmekte olan bilgi işlem ağlarının karşılaştığı siber tehditlerin önemini vurgulamaktadır. Özellikle, DDoS saldırıları gibi ciddi tehditlere karşı, Yazılım Tanımlı Ağ (SDN) yapısının merkezi kontrol mantığı, yapay zeka yöntemleriyle entegre edilerek bu tür saldırıların etkili bir şekilde tespit edilmesi ve önlenmesi sağlanmıştır. Araştırma kapsamında, akış tabanlı bir veri kümesi kullanılarak Eş Zamanlı Akış Analizi (ESA) tabanlı bir sistem önerilmiştir. Bu sistem, DDoS saldırılarının tespit edilmesi ve doğrulanmasında önemli avantajlar sunmaktadır. (Haider vd., 2020).

Bıçakcı ve Toklu tarafından gerçekleştirilen çalışmada, öznitelik seçimi için Yığılmış Otomatik Kodlayıcı ve SelectBest metodolojilerinin uygulandığı belirtilmiştir. Yaptıkları bu çalışmada, sınıflandırma modelleri olarak Rastgele Orman ve Destek Vektör Makineleri kullanılmıştır. NSL-KDD veri kümesi üzerinde yapılan deneysel çalışmalar sonucunda, uygulanan modeller %99,67 doğruluk oranı elde etmiştir. Bu yüksek doğruluk oranı, öznitelik seçimi metodolojilerinin ve kullanılan sınıflandırma algoritmalarının etkinliğini ortaya koymaktadır (Bıçakcı ve Toklu, 2022).

Fitni ve Ramli tarafından CSE-CIC-IDS-2018 veri kümesi üzerinde yürütülen deneysel bir çalışmada, veri özniteliklerinin belirginleştirilmesi amacıyla Spearman'ın sıra korelasyon katsayısının kullanıldığı bildirilmiştir. Araştırma, verilerin analizinde bu istatistiksel yöntemin uygulanmasını detaylandırmaktadır. Rastgele orman algoritması kullanılarak yapılan sınıflandırma işlemleri sonucunda %98,8 oranında bir başarı elde edildiği ifade edilmiştir. Bu bulgular doğrultusunda, Spearman korelasyon katsayısının ve rastgele orman algoritmasının, söz konusu veri kümesi üzerinde etkili bir şekilde kullanılabileceğini göstermektedir. (Fitni ve Ramli, 2020).

Emhan ve Akın tarafından KSL-KDD veri kümesi üzerinde gerçekleştirilen araştırmada, öznitelik boyutunun azaltılması amacıyla filtreleme ve korelasyon tabanlı öznitelik seçimi teknikleri uygulanmıştır. Bu çalışma kapsamında öznitelik seçimi, öznitelikleri sıralama yöntemiyle ve seçme yaklaşımıyla gerçekleştirilmiştir. Çalışma sonuçları, Rastgele Orman algoritmasının uygulanan öznitelik seçimi yöntemleri ile

entegre edildiğinde %93,40 doğruluk oranıyla en başarılı sınıflandırma yöntemi olduğunu göstermiştir. Yapılan çalışma ile öznitelik seçiminin sınıflandırma performansı üzerindeki önemli etkisini ve Rastgele Orman yönteminin bu kontekstteki etkinliğini vurgulamaktadır. (Emhan ve Akın, 2019).

Pehlivanoğlu ve arkadaşları tarafından gerçekleştirilen çalışmada, sentetik azınlık aşırı örnekleme tekniği (SMOTE) kullanılarak iki seviyeli bir hibrit yöntem önerilmiştir. Bu önerilen yöntem, özellikle veri dengesizliklerinin giderilmesine yönelik olarak tasarlanmıştır. Çalışma kapsamında, CNN (Evolüsyonel Sinir Ağı) ile Rastgele Orman algoritması kombinasyonu kullanılmış ve yapılan değerlendirmeler sonucunda bu birleşimin %98 başarı oranıyla en iyi performansı sergilediği belirlenmiştir. Önerilen hibrit yöntemin, veri dengesizliği sorunlarını çözmede ve yüksek doğruluk oranları elde etmede etkili olduğunu göstermektedir. (Pehlivanoğlu vd., 2019).

Hamid ve arkadaşları tarafından gerçekleştirilen çalışmada, KDD CUP-99 veri kümesinin %10'luk bir bölümüne denk gelen 494020 veri kullanılmıştır. Yapılan bu çalışmada, WEKA yazılım platformunda bulunan çeşitli denetimli öğrenme algoritmaları test edilmiştir. İlk aşamada, bu veri kümesi üzerinde denetimli öğrenme algoritmaları çalıştırılarak sonuçlar elde edilmiştir. Elde edilen bulgulara göre, Karar Ağacı algoritması %99,96, Çok Katmanlı Algılayıcı algoritması %98,75, Lojistik Regresyon algoritması %99,94 ve k-Yakın Komşu algoritması %99,94 doğruluk oranları ile sınıflandırma yapmıştır. Bu sonuçlar neticesinde, denetimli öğrenme algoritmalarının yüksek doğruluk oranları ile etkili sınıflandırma performansı sergilediğini göstermektedir (Hamid vd., 2016).

Musa ve arkadaşları tarafından gerçekleştirilen çalışmada, 2010 ile 2015 yılları arasında yayımlanan araştırma makalelerinde kullanılan veri kümelerine yönelik bir analiz yapılmıştır. Bu çalışmada toplam 28 araştırma makalesi incelenmiş ve NSL KDD veri kümesinin 14 makalede kullanıldığı tespit edilmiştir. NSL KDD veri kümesini, dört makalede kullanılan KDD CUP-99 veri kümesi izlemektedir. Araştırma sonuçlarına göre diğer yaygın kullanılan veri kümeleri arasında CICIDS2017, UNSWNB-15, UGR-16 ve Kyoto2006+ yer almaktadır. Yapılan bu çalışma, siber güvenlik araştırmalarında kullanılan veri kümelerinin dağılımı ve popülerliğine dair önemli bir farkındalık sunmaktadır. (Musa vd., 2020).

3. MATERYAL VE YÖNTEM

3.1. Simülasyon, Emülasyon ve Sanallaştırma

Simülasyon terimi, genel olarak bir şeyin orijinaliyle benzerlik gösteren veya taklit edilen bir versiyonunu ifade etmek için kullanılır. Ancak teknik anlamda, gerçek dünya süreçlerinin veya sistemlerinin zaman içindeki işleyişinin kopyalanması anlamına gelmektedir. Bu işleyiş, tanımlanmış ilişkilere sahip olan sistem veya süreçlerin bir modelini oluşturmayı içermektedir.

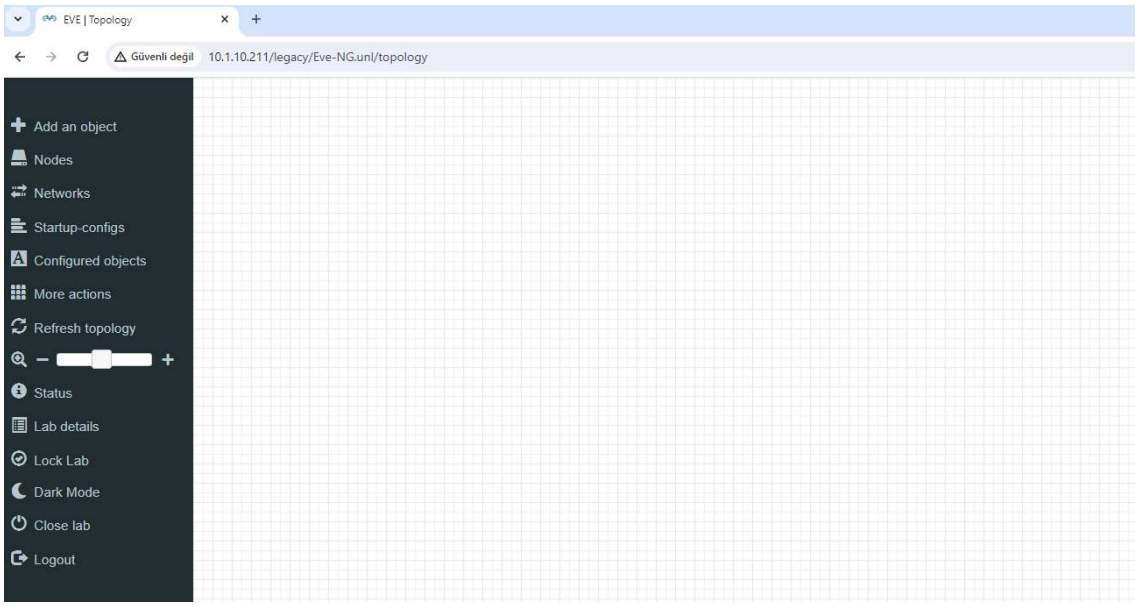
Emülasyon, bir bilgisayar ortamında bir başka bilgisayar ortamını donanımsal olarak simüle etme sürecini ifade etmektedir. Emülatörlerin amacı, yazılımı donanıma benzeterek normalde sadece belirli bir sistemde çalışabilen çeşitli verilerin ve cihazların farklı ortamlarda da çalıştırılmasını sağlamaktır.

Sanallaştırma, bilgisayar kaynaklarının kullanıcılar tarafından soyutlanması sürecini ifade etmektedir. Bu süreci gerçekleştirmek için kullanılan yöntemler, bilgisayar kaynaklarının paylaştırılmasını veya birleştirilmesini sağlar. Sanallaştırmanın temel amacı, kullanıcıya bir soyutlama katmanı sunarak onları kaynaklardan izole etmektir. Bu şekilde, kullanıcı ile kaynaklar arasına bir mantıksal tabaka eklenmiş olur.

3.2. EVE-NG Emülatör

EVE-NG (Emulated Virtual Environment - Next Generation), ağ ve sistem mühendisleri tarafından kullanılan bir sanal laboratuvar platformudur. Bu platform, ağ cihazları, sunucular, kişisel bilgisayarlar ve diğer ağ bileşenlerini sanal olarak çalıştırmak ve simüle etmek için kullanılır. EVE-NG, ağ tasarımı, testi, eğitimi ve sertifikasyon hazırlığı gibi çeşitli amaçlarla kullanılabilir (Aung, 2020).

EVE-NG'nin kullanımı oldukça esnektir ve kullanıcı dostu bir web tabanlı ara yüzü vardır. Şekil 3.1’de Eve-NG emülatörün web görünümü verilmiştir.

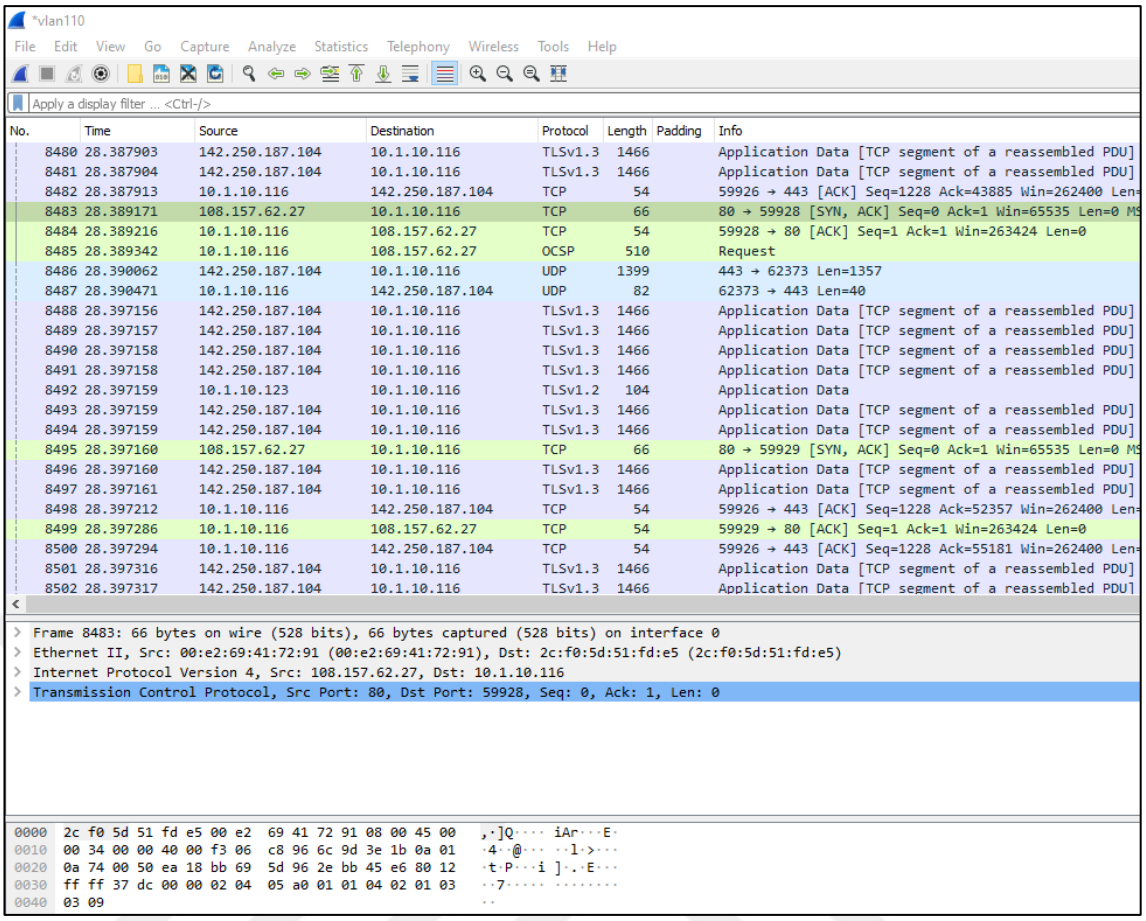


Şekil 3.1. Eve-NG Emülatör ortamı

Web ara yüzündeki menüler aracılığı ile kullanıcılar sanal ağ cihazlarını kolayca ekleyebilir, bunları birbiriyle bağlayabilir ve istedikleri ağ topolojilerini oluşturabilirler. Ayrıca, her bir sanal cihazın yapılandırması ve davranışı tamamen özelleştirilebilir, bu da kullanıcıların gerçek dünya senaryolarını mümkün olan en iyi şekilde simüle etmelerini sağlamaktadır (Harahus vd., 2023).

3.3. Wireshark

Wireshark, özgür ve açık kaynak kodlu bir ağ paket çözümleyicisi olarak bilinmektedir. Proje, başlangıçta Ethereal adıyla bilinmekteydi ancak ticari marka sorunları nedeniyle Mayıs 2006'da Wireshark olarak yeniden adlandırıldı. Günümüzde, Wireshark en önde gelen paket çözümleyicilerinden biri olarak kabul edilmektedir (Sasi vd., 2020). Bu yazılım, ağ üzerinden iletilen verilerin paketlerini yakalar ve bu paketleri çeşitli ağ protokollerine göre ayrıştırarak kullanıcıya sunar (Li vd., 2021). Wireshark, genellikle ağ sorunlarını teşhis etme, ağ yapılarını analiz etme, yazılım veya iletişim protokollerini geliştirme ve eğitim amaçları için kullanılmaktadır. Şekil 3.2'de wireshark programının bir ekran görüntüsü verilmiştir.



Şekil 3.2. Wireshark ekran görüntüsü

Bir ağ paket analizörü olan wireshark, elektrik kablolarının içindeki akımı ölçmek için kullanılan bir voltmetreye benzer şekilde, ağ kablolarının içindeki verileri incelemek için bir ölçüm cihazı olarak düşünülebilir. Geçmişte, bu tür araçlar ya yüksek maliyetliydi ya da tescilli olarak sunuluyordu. Ancak Wireshark'ın ortaya çıkmasıyla bu durum değişti. Wireshark, ücretsiz olarak erişilebilir ve açık kaynak kodludur. Günümüzde, mevcut en etkili paket analizörlerinden biri olarak kabul edilir.

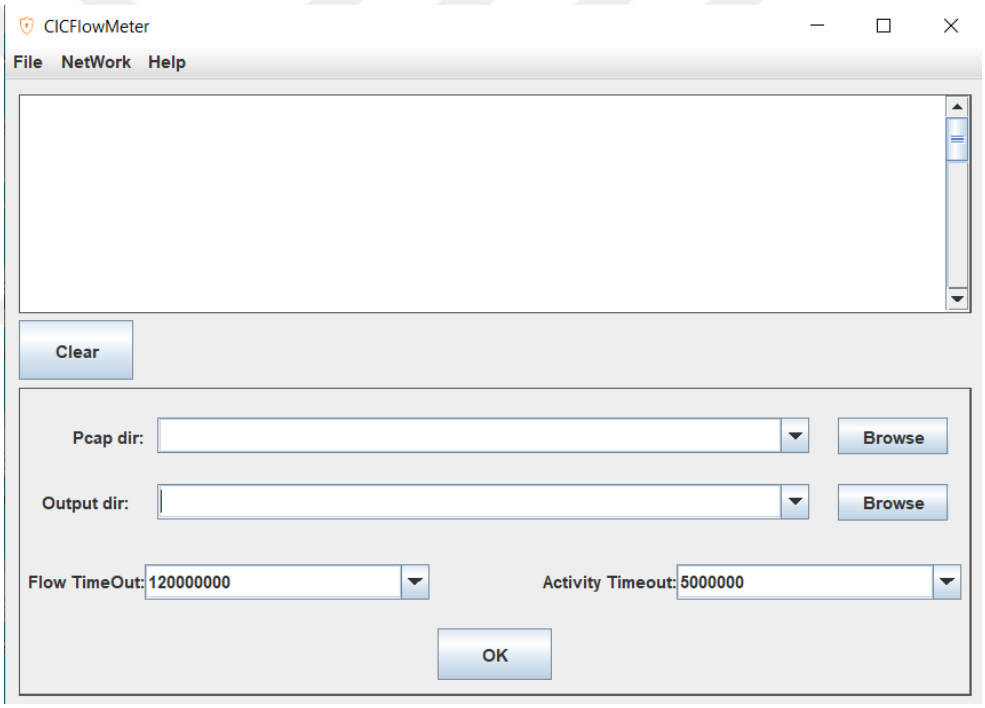
3.4. CICFlowMeter

CIC Flowmeter, ağ trafiğini analiz etmek ve sınıflandırmak için kullanılan bir araçtır. Bu araç, özellikle siber güvenlik alanında ağ trafiğinin izlenmesi ve tehditlerin tespit edilmesi amacıyla geliştirilmiştir. CIC Flowmeter, ağ üzerindeki veri akışlarını detaylı bir şekilde inceleyerek, farklı uygulamalar, protokoller ve hizmetler arasındaki ilişkileri belirlemeyi amaçlamaktadır.

CIC Flowmeter, ağ trafiğinin yoğunluğunu, doğasını ve zaman içindeki değişimlerini analiz ederek, ağda gerçekleşen olayları anlamak için kullanılır (Ali vd.,

2022). Bu analizler, ağdaki normal ve anormal davranışları tanımlamak, anormal aktiviteleri tespit etmek için önemli bir temel sağlamaktadır. Özellikle, kötü niyetli yazılımların ve siber saldırıların belirlenmesi için ağ trafiğinin örüntülerini ve davranışlarını izlemek amacıyla kullanılan bir araçtır.

CICFlowMeter, *pcap* dosyalarından flow'lar üretme ve bu akışlardan özellikler türetme işlevine sahip olan açık kaynaklı bir araçtır. CICFlowMeter, bir ağ trafiği akış oluşturucusu olarak kabul edilebilir ve çift yönlü akışlar oluşturmak için kullanılabilir. Bu akışlar, ilk paketin ileri (kaynaktan hedefe) ve geri (hedeften kaynağa) yönlerini belirler. Dolayısıyla istatistiksel zamanla ilgili özellikler, süre, paket sayısı, bayt sayısı, paketlerin uzunluğu gibi 80'den fazla istatistiksel ağ trafiği özelliği ileri ve geri yönlerde ayrı ayrı hesaplanabilir (Dung, 2023).



Şekil 3.3. CICFlowMeter ara yüzü

CICFlowMeter'ın ortam görünümü Şekil 3.3'de verilmiştir. *Pcap dir* kısmından Wireshark'dan aldığımız *pcap* dosyalarını ekleyip *output dir* kısmından da *csv* uzantılı dosyayı nereye çıkaracağını seçiyoruz.

3.5. Emülatör Tasarımı ve Saldırı Tespiti İçin Kullanılan Bilgisayar

Tasarlanacak emülatör ortamı için kullanılacak bilgisayar bir iş istasyonudur. Intel Xeon E-2236 CPU 3.40GHz bir işlemci, 128 GB Ram bellek, 1 TB 2000 MB/s okuma

yazma hızı olan M2 sata disk, Nvidia GeForce GTX 1660 Ti 6GB ekran kartı konfigürasyona sahip bir bilgisayardır. Bu iş istasyonuna Windows 10 PRO kurulmuştur. Sanallaştırma olarak ise VMware Workstation Pro kullanılmış ve emülatör bu sanallaştırma platformuna kurulmuştur.

3.6. VMware Workstation Pro

VMware Workstation Pro, sanal makine (virtual machine) yazılımıdır ve bilgisayarınızda sanal ortamlar oluşturmanıza izin vermektedir. Bu yazılım, bir ana bilgisayar üzerinde birden fazla işletim sistemi çalıştırmanızı sağlamaktadır. Sanallaştırma yazılımı sayesinde, bir bilgisayarın donanım kaynaklarını sanal makineler arasında bölüştürerek birden fazla işletim sistemini tek bir fiziksel bilgisayarda çalıştırabilme avantajı sağlar. Yazılım geliştirme, test, eğitim ve sistem yönetimi gibi birçok alanda kullanışlıdır. Emülatör veya similatör ortamları içinde kullanımı yaygın olduğu görülmektedir.

3.7. Saldırı Tespiti ve Emülatör Sistemini Tasarlanması

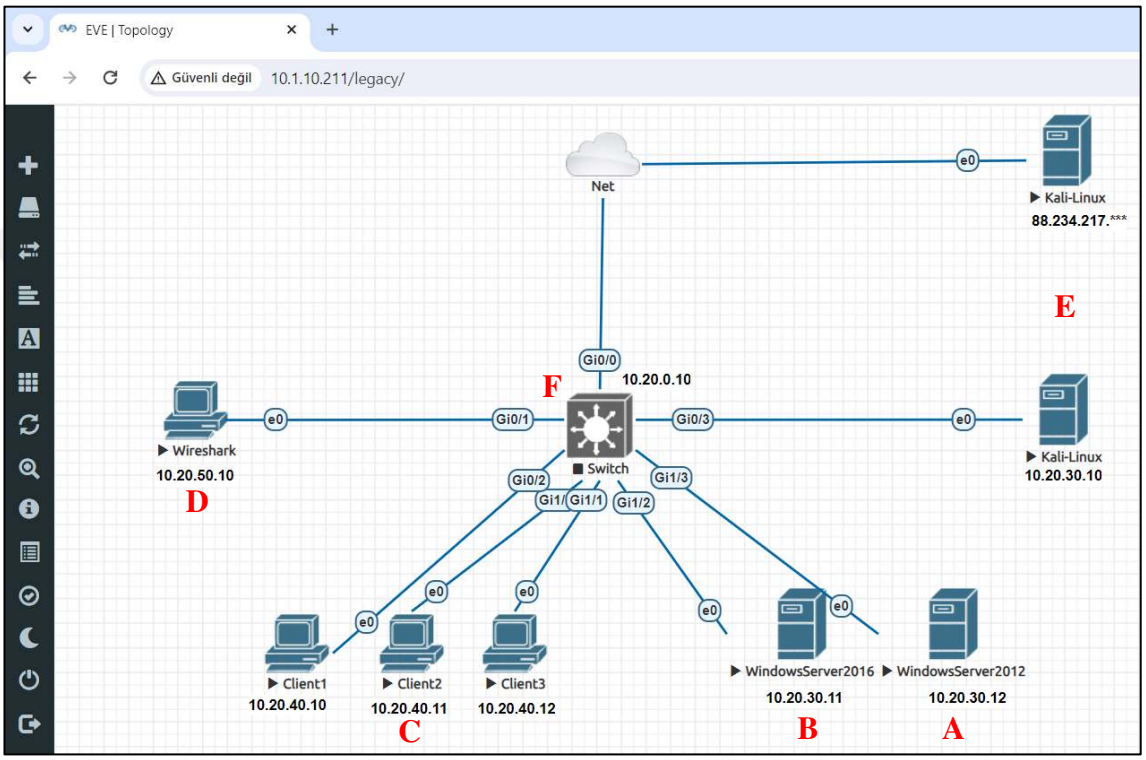
Yapılan çalışmalar incelendiğinde genellikle hazır veri kümeleri üzerinde çalışmalar yapıldığı gözlenmektedir. Bu çalışmada hazır veri kümesi kullanmak yerine emülatör ortamında bir ağ sistemi tasarlanmıştır. Tasarlanan sistemde normal zamanlar ve saldırı zamanlarında toplanan verilerden elde edilen veri kümeleri oluşturulmuştur. Oluşturulan veri kümeleri kullanılarak saldırı tespiti yapılmıştır. Bu yönü ile diğer çalışmalardan ayrışmaktadır.

Bu çalışmada, karmaşık ağlarda saldırı tespiti gerçekleştirmek amacıyla öncelikle bir emülatör ortamı tasarlanmıştır. Tasarlanan bu emülatör ortamında, Windows işletim sistemleri, Windows sunucuları, Cisco anahtarlama cihazları ve Kali-Linux sistemleri kurulumu yapılmıştır. Saldırı esnasında ağ trafik kayıtlarının toplanması amacıyla, Wireshark yüklü bir bilgisayar ağa entegre edilmiştir. Toplanan loglar, daha sonra Wireshark ile elde edilen *pcap* dosyalarından, CICFlowmeter kullanılarak *csv* formatına dönüştürülmüştür. Araştırmanın son aşamasında ise, Python ile gerçekleştirilen makine öğrenmesi algoritmaları kullanılarak saldırı tespit süreci tamamlanmıştır.

3.7.1. Emülatör Ortamında Ağ Sisteminin Tasarımı

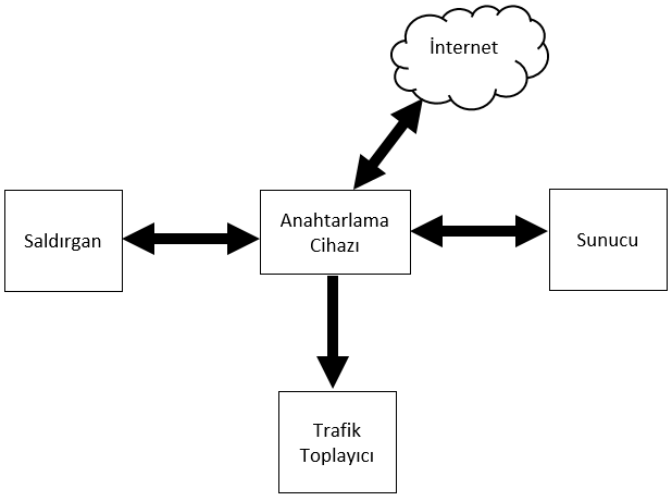
Emülatör ortamın tasarlamak için ilk önce Windows 10 Pro kurulu bilgisayarda VMware Workstation Pro yükleme işlemi yapılmıştır. Amacımız EVE-Ng

emülatörümüzü sanal bir otamada çalıştırmak istememizdir. Böylelikle emülatörümüzü kurduktan sonra web tarayıcısı üzerinden istediğimiz yerden ulaşmamızı sağlanmış olacaktır. Emülatörümüzü VMware Workstation PRO üzerine kurduktan sonra, EVE-Ng'nin web sitesinde ilgili dokümantasyona bakarak Windows 10 PRO, Windows Server 2012-2016, Kali-Linux ve Cisco anahtarlama cihazını Şekil 3.3'deki gibi tasarım ve kurulumlarını yapıyoruz.



Şekil 3.4. Emülatör ortamı

Şekil 3.4'de saldırı yapılan SQL ve FTP sunucu A, DNS ve kaba kuvvet saldırısı yapılan sunucu B, kurban olarak kullanılan istemciler C, wireshark uygulaması ile paketlerin toplandığı bilgisayar D, saldırı yapmakta kullanılan Kali-Linux sistemler E, tüm cihazların birbirine bağlanması için kullanılan anahtarlama cihazı F ile gösterilmiştir. Cihazlar arası bağlantılar emülatör ortamında yapılmıştır.



Şekil 3.5. Uygulama ortamının blok diyagramı

Şekil 3.5'de uygulama ortamının bir blok diyagramı verilmiştir. Blok diyagramda saldırgan rolündeki cihazların bir ağ cihazı üzerinden geçerek kurban rolündeki cihaza erişimi tasvir edilmiştir. Trafik toplayıcı vasıtası ile ağ üzerinden trafik akışları alınmaktadır. Tüm bu cihazların internet erişimi bulunmaktadır.

3.7.2. Saldırı ve Wireshark ile Paketlerin Toplanması

Tasarladığımız emülatör ortamında, Kali-linux kurulu bilgisayarları kullanarak kullanıcı ve sunucu bilgisayarlara DDOS, Kaba Kuvvet, DNS Amplification, SQL Injection'dan oluşan saldırıları gerçekleştiriyoruz. Kurban ve saldırganların yapmış oldukları trafikleri toplamak için anahtarlama cihazı üzerinde port aynalama özelliği kullanılarak, farklı bir bilgisayarda kurulu olan Wireshark vasıtası ile gerçekleştiriyoruz. Wireshark uygulası ile toplanan tüm trafik *pcap* dosya formatı kullanılarak kayıt edilmiştir.

3.7.3. CICFlowmeter ile Flow Analizi

CICFlowMeter ile gerçekleştirilen flow analizi, ağ trafiğini sürekli olarak gözlemleyerek bu trafiğin ayrıntılı veri akışları (flows) üzerine öznitelikler elde etme sürecini kapsamaktadır. Flow, bir ağda belirlenen zaman dilimi içerisinde, bir kaynak ve hedef arasında transfer edilen paketlerin toplamını tanımlar. CICFlowMeter, bu flow'ları detaylı bir şekilde analiz eder ve onların çeşitli istatistiksel özelliklerini tespit etmektedir. Bir flow, kaynak IP, hedef IP, kaynak port, hedef port ve kullanılan protokol gibi benzersiz tanımlayıcılarla karakterize edilir. Flow'lar, ağ trafiğini daha yönetilebilir ve anlaşılır segmentlere ayırmaktadır. CICFlowMeter, her bir flow için toplam paket sayısı,

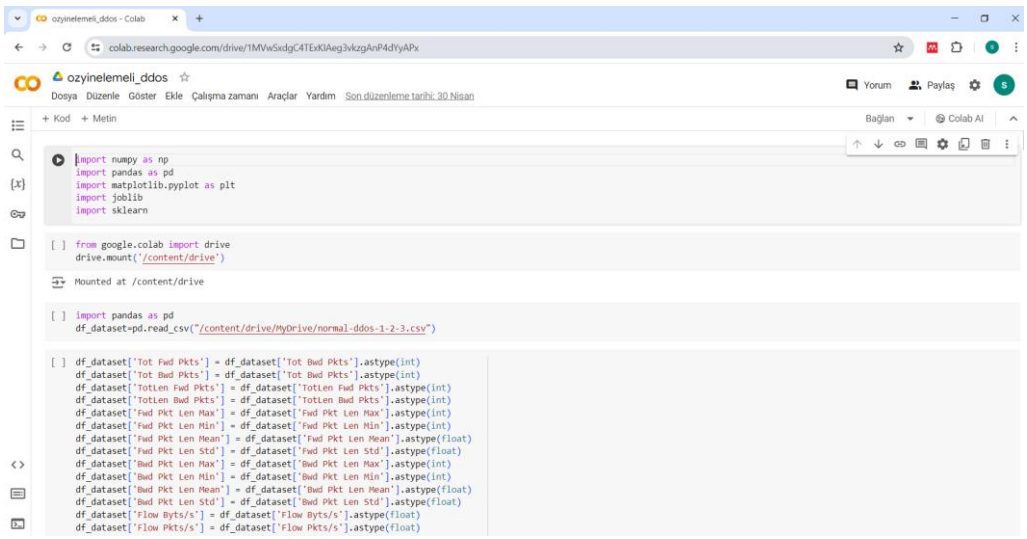
toplam bayt sayısı, flow süresi, paketler arası zaman aralığı gibi çeşitli istatistiksel öznitelikleri hesaplar. Flow analizi, ağ performansını değerlendirme ve ağ güvenliği ile ilgili potansiyel sorunları belirleme amacıyla sıklıkla kullanılır. Ağ trafiğini analiz etmek ve sınıflandırmak için CICFlowmeter uygulaması sayesine ile 80'den fazla istatistiksel ağ trafiği özelliği oluşturulmaktadır.

Siber güvenlik alanında, flow analizi özellikle saldırı tespiti, ağ izleme ve anormal davranışların saptanması gibi kritik roller üstlenir. Ağ üzerindeki normal ve anormal davranışlar arasındaki farkları ayırt etmeye yönelik detaylı veriler sağlamaktadır.

3.8. Yazılım Geliştirme Ortamı

Çalışmada, son zamanlarda özellikle veri analizi ve makine öğrenimi gibi alanlarda önemli bir popülerlik kazanan, geniş çapta beğenilen ve çalışmanın gereksinimlerine uygunluğu nedeniyle Python programlama dili kullanıldı. Python ile ilişkili makine öğrenimi kütüphanelerinin çokluğu, bu alanda çalışma yürüten araştırmacılar için avantajlı olduğunu kanıtlanmıştır. Python dilinin modüler yapısı ve çeşitli işletim sistemlerinde sorunsuz bir şekilde çalışabilme yeteneği, bu çalışma için seçilmesine katkıda bulunan temel faktörlerdir. Piyasada özellikle Python programlama dili için tasarlanmış çeşitli entegre geliştirme ortamları (IDE'ler) bulunmaktadır.

Python için tercih edilen geliştirme ortamları arasında Google Colab veri analistleri, mühendisler ve bilim insanları tarafından oldukça beğenilen, belli bir kapasiteye kadar ücretsiz olan bir seçenek olarak öne çıkmaktadır. Çalışmada, çalışma ortamı Şekil 3.6'da gösterildiği gibi Google Colab kullanılmıştır.



Şekil 3.6. Google colab çalışma ortamı

Çalışma süreci boyunca Sklearn, NumPy, Pandas ve Matplotlib dâhil olmak üzere çeşitli Python kütüphanelerinden yararlanıldı.

3.8.1. Sklearn Kütüphanesi

Yaygın olarak "scikit-learn" veya "Sklearn" olarak bilinen Python açık kaynak kütüphanesi, makine öğrenimi ve veri bilimi alanlarında yaygın olarak kullanılmaktadır. Bu kütüphane, kümeleme, karar ağaçları ve regresyon gibi çeşitli yöntemler sağlayarak veri analizini basitleştirir. Bu yöntemler, veri analizi sürecini geliştiren bir dizi işlev sunmaktadır.

3.8.2. NumPy Kütüphanesi

NumPy kütüphanesinin arka planı iyi belgelenmiştir. "Sayısal Python" yani NunPy, amacı doğru bir şekilde yansıtmak ve hızlı bilimsel hesaplamaları kolaylaştırmaktır. Diziler NunPy kütüphanesinin temelini oluşturarak verimli dizi işlemlerine olanak tanımaktadır. Veri bilimi ve makine öğreniminde büyük veri kümelerinin yaygınlığı göz önüne alındığında, NumPy kütüphanesi bu alanlarda değerli bir araçtır.

3.8.3. Pandas Kütüphanesi

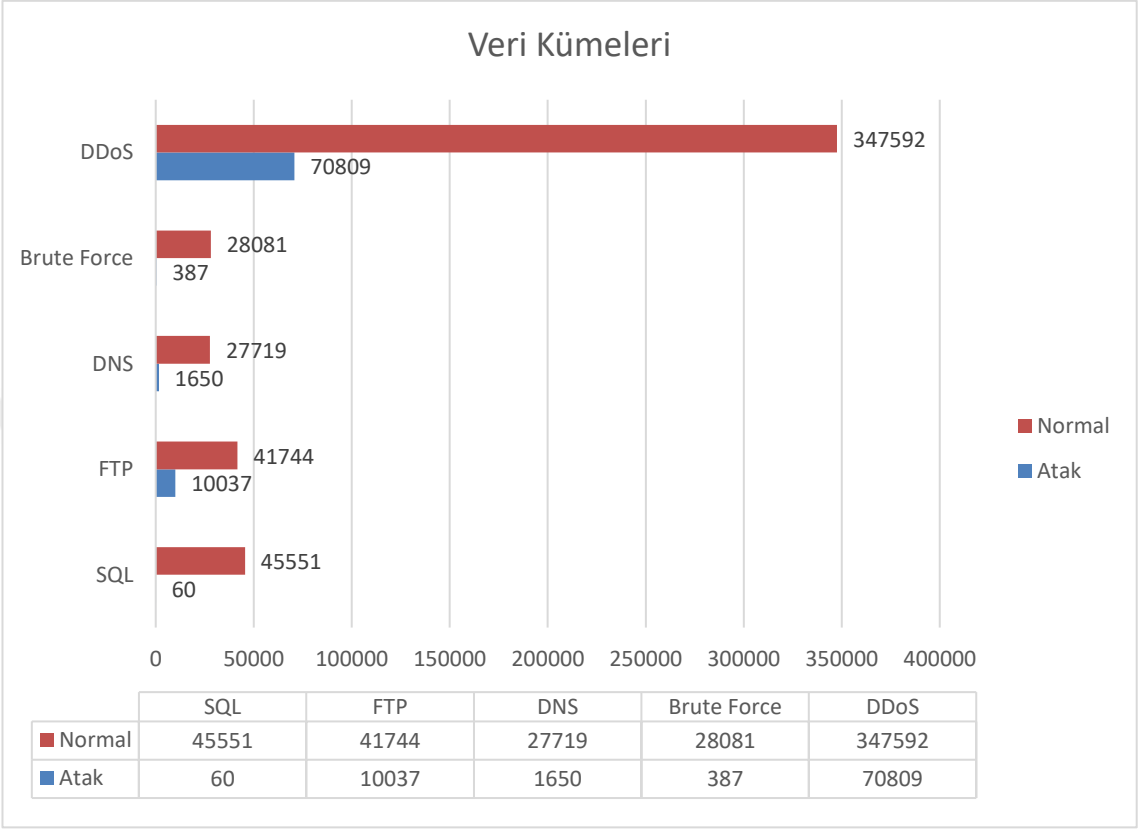
NumPy kütüphanesinin çerçevesi üzerine inşa edilen bu Python paketi, verimli ve uyarlanabilir veri yapıları sunarak etiketli ve ilişkisel verilerin manipülasyonunu basitleştirmeyi amaçlamaktadır. Veri temizleme, analiz ve model oluşturma dâhil olmak üzere veriyle ilgili çeşitli görevler için paha biçilmez bir kaynak görevi görmektedir. Pandas kütüphanesi, kapsamlı veri kümeleri içindeki eksik verileri tanımlamanın yanı sıra veri dönüştürmeyi, yeniden şekillendirmeyi ve boyutluluk değişikliğini kolaylaştırma yeteneği nedeniyle de büyük saygı ve ilgi görmektedir.

3.8.4. Matplotlib Kütüphanesi

Görsel temsil, veri bilimi alanında çok önemli bir rol oynamaktadır. Verileri analiz etme süreci en etkili şekilde görselleştirme yoluyla gerçekleştirilir. Matplotlib'in çok yönlülüğü, esnekliği, geniş kütüphane desteği ve güçlü topluluk desteği, onu veri görselleştirme için en iyi araçlardan biri yapmaktadır. Bu çalışmadaki grafikler, geniş bir fonksiyon yelpazesi sunan ve kolay uygulamaya olanak sağlayan 2 boyutlu çizim kütüphanelerinden biri olan Matplotlib kütüphanesi kullanılarak oluşturulmuştur.

3.9. Veri Kümesi

Tasarlanan topoloji ile emülatör ortamında oluşturulan veri kümesi Şekil 3.7'de verilmiştir.



	SQL	FTP	DNS	Brute Force	DDoS
Normal	45551	41744	27719	28081	347592
Atak	60	10037	1650	387	70809

Şekil 3.7. Veri kümesi değerleri

Şekil 3.7'den de anlaşılacağı üzere, 60 adet SQL, 10.037 adet FTP, 1.650 adet DNS, 387 adet kaba kuvvet ve 70.809 adet DDoS saldırısı bulunmaktadır. Ayrıca toplamda 1.120.750 adet normal kullanım bulunmaktadır. Veri kümesi oluşturulurken ayrı ayrı ve farklı zamanlarda oluşturulmuştur. Her bir saldırının ve normal trafiğin adetsel olarak çizelgede gösterilmiştir.

Oluşturulan veri kümesinde 79 adet öznitelik bulunmaktadır. Bu öznitelikler Çizelge 3.1'de verilmiştir. Özniteliklerin açıklamaları EK 1- Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Veri kümesi öznitelikleri

1-Dst Port	28-Bwd IAT Mean	55-Fwd Seg Size Avg
2-Protocol	29-Bwd IAT Std	56-Bwd Seg Size Avg
3-Flow Duration	30-Bwd IAT Max	57-Fwd Byts/b Avg
4-Tot Fwd Pkts	31-Bwd IAT Min	58-Fwd Pkts/b Avg
5-Tot Bwd Pkts	32-Fwd PSH Flags	59-Fwd Blk Rate Avg
6-TotLen Fwd Pkts	33-Bwd PSH Flags	60-Bwd Byts/b Avg
7-TotLen Bwd Pkts	34-Fwd URG Flags	61-Bwd Pkts/b Avg
8-Fwd Pkt Len Max	35-Bwd URG Flags	62-Bwd Blk Rate Avg
9-Fwd Pkt Len Min	36-Fwd Header Len	63-Subflow Fwd Pkts
10-Fwd Pkt Len Mean	37-Bwd Header Len	64-Subflow Fwd Byts
11-Fwd Pkt Len Std	38-Fwd Pkts/s	65-Subflow Bwd Pkts
12-Bwd Pkt Len Max	39-Bwd Pkts/s	66-Subflow Bwd Byts
13-Bwd Pkt Len Min	40-Pkt Len Min	67-Init Fwd Win Byts
14-Bwd Pkt Len Mean	41-Pkt Len Max	68-Init Bwd Win Byts
15-Bwd Pkt Len Std	42-Pkt Len Mean	69-Fwd Act Data Pkts
16-Flow Byts/s	43-Pkt Len Std	70-Fwd Seg Size Min
17-Flow Pkts/s	44-Pkt Len Var	71-Active Mean
18-Flow IAT Mean	45-FIN Flag Cnt	72-Active Std
19-Flow IAT Std	46-SYN Flag Cnt	73-Active Max
20-Flow IAT Max	47-RST Flag Cnt	74-Active Min
21-Flow IAT Min	48-PSH Flag Cnt	75-Idle Mean
22-Fwd IAT Tot	49-ACK Flag Cnt	76-Idle Std
23-Fwd IAT Mean	50-URG Flag Cnt	77-Idle Max
24-Fwd IAT Std	51-CWE Flag Count	78-Idle Min
25-Fwd IAT Max	52-ECE Flag Cnt	79-Label
26-Fwd IAT Min	53-Down/Up Ratio	
27-Bwd IAT Tot	54-Pkt Size Avg	

Veri kümesinden örnek veriler Şekil 3.8'de verilmiştir.

	A	B	C	D	E	F	G	H	I	J
1	Dst Port	Protocol	Flow Dura	Tot Fwd P	Tot Bwd P	TotLen Fw	TotLen Bw	Fwd Pkt Le	Fwd Pkt Le	Fwd Pkt Len Mean
2	51368	6	0	1	1	0.0	0.0	0.0	0.0	0.0
3	61960	6	41896	5	8	248.0	974.0	124.0	0.0	49.6
4	443	6	40812	0	2	0.0	0.0	0.0	0.0	0.0
5	59918	6	106185	13	9	908.0	60.0	89.0	58.0	6.984.615.384.615.380
6	0	0	85086	5	1	0.0	0.0	0.0	0.0	0.0
7	443	6	0	1	1	0.0	0.0	0.0	0.0	0.0
8	58606	6	60524	23	13	33466.0	1460.0	1460.0	1417.0	14.550.434.782.608.600
9	51009	6	536	11	7	10894.0	1412.0	1412.0	97.0	9.903.636.363.636.360
10	5001	17	54366	11	3	12031.0	1489.0	1439.0	99.0	10.937.272.727.272.700
11	61964	6	119985	75	41	103954.0	1424.0	1424.0	1.0	13.860.533.333.333.300
12	61981	6	0	1	1	1424.0	1424.0	1424.0	1424.0	1424.0
13	51875	6	2726	0	3	0.0	1666.0	0.0	0.0	0.0
14	51875	6	9890	1	2	107.0	0.0	107.0	107.0	107.0

Şekil 3.8. Veri kümesi görünümü

3.10. Veri Ön İşleme

İşlemenin ilk aşamasında kodlama ortamı, orijinal veri kümesinden ayrı ayrı *csv* dosyalarını okumak için Pandas kitaplığını kullanır. Bu *csv* dosyaları içerisinde sonsuz veya anlamsız değerler içeren tüm kayıtlar tanımlanır ve ardından kaldırılır. Çalışma boyunca tutarlılığın sağlanması amacıyla herhangi bir değer içermeyen kayıtlar da belirlenmekte ve silinmektedir.

Çalışmanın ilerlemesini kolaylaştırmak için sayısal değerler kayan nokta ve tamsayı veri türlerine dönüştürüldü. Bu dönüşümün ardından çalışma, türü ne olursa olsun bir saldırının oluşumunu tespit etmeyi amaçlamaktadır. Veri kümesini ikili sınıflandırmaya hazırlamak için "Normal" etiketi "0", saldırı türleri ise "1" olarak ayarlandı. Bu ayarlama, verilerin uygun şekilde kategorize edilmesini ve analize hazır olmasını sağlamaktadır. Ardından verilerin %80 eğitim, %20'si test olarak ayrılmıştır.

3.11. Öznitelik Seçimi ve SMOTE Kullanımı

Öznitelik seçimi, Python programlama dilinde scikit-learn kütüphanesi aracılığıyla gerçekleştirilmiştir. Özyinelemeli öznitelik seçimi için Recursive Feature Elimination (RFE) yöntemi kullanılmış ve bu yöntemle 79 öznitelik içeren veri kümesi, en bilgilendirici 20 özniteliğe indirgenmiştir. İleri yönelimli öznitelik seçimi amacıyla Sequential Feature Selector sınıfı tercih edilmiş ve bu yöntemle aynı veri kümesinden 26 öznitelik seçilmiştir. Manuel öznitelik seçimi ise Çizelge 3.2'de belirtilen öznitelikler üzerinde yapılan denemeler sonucunda, en üstün performansı sergileyen 5 özniteliğin kullanılmasıyla sonuçlandırılmıştır.

Veri kümelerindeki sınıf dengesizliklerini gidermek için SMOTE tekniği, Python programlama dilinde geliştirilmiş olan imbalanced-learn kütüphanesi kullanılarak uygulanmıştır. Bu kütüphane, scikit-learn ile uyumluluğu sayesinde makine öğrenimi iş akışlarına sorunsuz bir şekilde entegre edilebilir. SMOTE sınıfı, eğitim veri kümesindeki azınlık sınıfı örneklerini sentetik olarak çoğaltarak, modelin daha dengeli bir veri üzerinde eğitilmesini sağlamaktadır. Veri kümesi daha dengeli olunca modelin genel performansının ve tahmin doğruluğunun artmasına katkıda bulunmaktadır.

3.12. Python ile Saldırı Tespiti

Wireshark ağ trafiğini izleyerek, bu trafiği *pcap* formatında kaydeder. Bu dosyalar, ağ üzerinde iletilen ham paket verilerini içerir ve her paket için zaman damgaları, kaynak ve hedef ip adresleri, protokol bilgileri ve diğer başlık bilgilerini barındırır.

CICFlowMeter, bu *pcap* dosyalarını işleyerek ağ akışlarına ilişkin öznitelikler barındıran *csv* formatında dosyalar üretmektedir. Bu işlem sırasında, her bir ağ akışına dair çeşitli istatistiksel veriler hesaplanır. Bu istatistik veriler arasında toplam paket sayısı, bayt cinsinden veri akışı miktarı ve akış süresi gibi bilgiler yer almaktadır. CICFlowMeter, bu öznitelikleri çıkararak, her bir ağ akışı için tek bir satır oluşturacak şekilde *csv* dosyalarına kaydederler.

Python programlama dili ve pandas kütüphanesi kullanılarak elde edilen *csv* dosyası yüklenir. Veri ön işleme süreçleri, eksik verilerin doldurulması veya hatalı verilerin düzeltilmesi gibi işlemleri içermektedir. Modelin doğruluğunu artırmak amacıyla, etkili özniteliklerin seçimi önemlidir. Bu öznitelik seçimi, istatistiksel yöntemler ve öznitelik önemi analizleri ile desteklenebilir. Saldırı tespiti için çeşitli sınıflandırma algoritmaları uygulanabilir ve model seçimi, veri setinin niteliklerine ve belirlenen performans kriterlerine göre yapılır. Eğitimden geçirilen model, test veri seti üzerinde değerlendirilir ve modelin saldırı tespit performansı, çeşitli performans metrikleri ile ölçülür. Başarılı bir şekilde test edilen model, gerçek zamanlı sistemlere entegre edilebilir. Model, sürekli izlenen ağ trafiğinde potansiyel tehditleri tespit etmek üzere kullanılır. Oluşturulan bu süreç, ağ güvenliğinin sağlanması ve siber saldırıların erken evrede tespit edilmesinde hayati öneme sahiptir. CICFlowMeter ve Python, bu tür güvenlik çözümlerinin geliştirilmesinde etkili araçlar olarak öne çıkmaktadır.

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Yapılmış olan çalışmada Eve-NG ile oluşturulan emülatör ortamından elde edilen veri kümesi üzerinden saldırı tespiti hedeflenmiştir. Verilerimiz %80 eğitim, %20 test olacak şekilde bölünmüştür. Makine öğrenme yöntemlerinden olan rastgele orman, K en yakın komşu, Karar Ağacı ve Destek Vektör Makinesi modelleri kullanılmış ve her bir modelde 5 kat çapraz doğrulama yapılmıştır. Özyinelemeli, ileri yönelimli ve manuel öznitelik seçini yapılmıştır. Öznitelik seçimi yapıldıktan sonra kalan öznitelikler sırası ile Çizelge 4.1, 4.2 ve 4.3'de verilmiştir.

Çizelge 4.1. Özyinelemeli özellik seçimi

1-TotLen Fwd Pkts	6-Fwd Pkt Len Std	11-Flow Pkts/s	16-Bwd Pkts/s
2-TotLen Bwd Pkts	7-Bwd Pkt Len Max	12-Flow IAT Mean	17-Pkt Len Min
3-Fwd Pkt Len Max	8-Bwd Pkt Len Min	13-Fwd IAT Max	18-Pkt Len Max
4-Fwd Pkt Len Min	9-Bwd Pkt Len Std	14-Fwd IAT Min	19-Fwd Byts/b Avg
5-Fwd Pkt Len Mean	10-Flow Byts/s	15-Fwd Pkts/s	20-Fwd Pkts/b Avg

Çizelge 4.1'de özyinelemeli özellik seçimi sonrasında kalan 20 öznitelik verilmiştir.

Çizelge 4.2. İleri yönelimli özellik seçimi

1-Fwd Pkt Len Max	8-Flow IAT Max	15-Pkt Len Min	22-Bwd Pkts/b Avg
2-Fwd Pkt Len Min	9-Flow IAT Min	16-Pkt Len Max	23-Active Mean
3-Fwd Pkt Len Mean	10-Fwd IAT Max	17-Fwd Byts/b Avg	24-Active Std
4-Fwd Pkt Len Std	11-Fwd IAT Min	18-Fwd Pkts/b Avg	25-Active Max
5-Flow Byts/s	12-Bwd IAT Min	19-Fwd Blk Rate Avg	26-Active Min
6-Flow Pkts/s	13-Fwd Pkts/s	20-Bwd Byts/b Avg	
7-Flow IAT Mean	14-Bwd Pkts/s	21-Bwd Blk Rate Avg	

Çizelge 4.2'de ileri yönelimli özellik seçimi sonrasında kalan 26 öznitelik verilmiştir.

Çizelge 4.3. Manuel özellik seçimi

1-Flow Byts/s
2-Flow Pkts/s
3-Flow IAT Mean
4-Fwd Pkts/s
5-Bwd Pkts/s

Çizelge 4.3'de manuel özellik seçimi sonrasında kalan 5 öznitelik verilmiştir

Oluşturulan bütün veri kümelerinde SMOTE tekniği kullanarak veri dengelenmesi de sağlanmıştır. Toplanan veriler hem orijinal hali ile makine öğrenme algoritmaları ile eğitilmiş, hem de SMOTE veri artırımı yapılarak makine öğrenme algoritmaları ile işlenmiştir. Her iki durumda da elde edilen doğruluk oranları 4.4, 4.5, 4.6, 4.7 ve 4.8 çizelgelerinde verilmiştir.

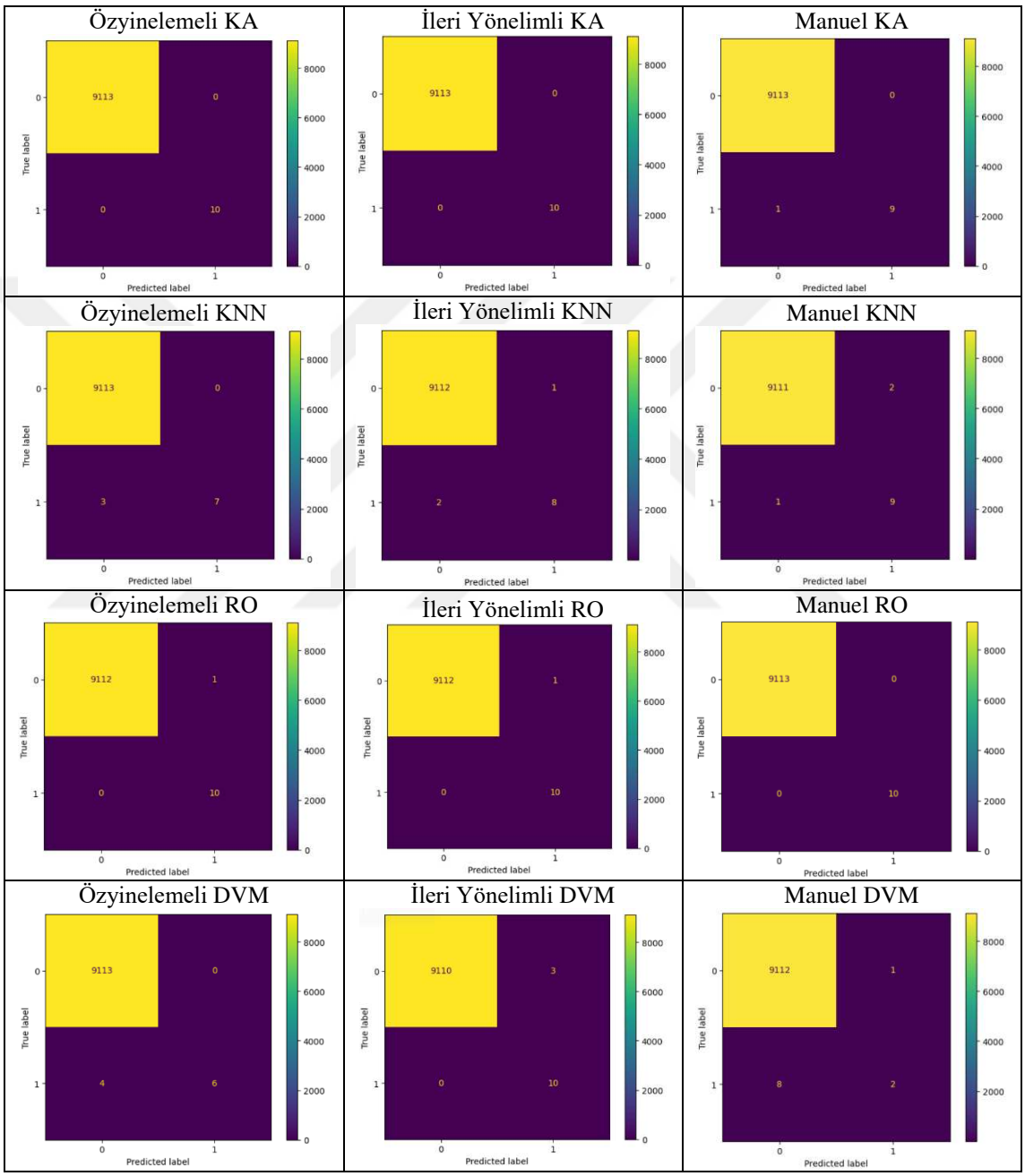
Tasarlanan emülatör ortamında server 2016 kurulmuş sunucu üzerine SQL server Express ve FTP uygulamaları yapılandırılmıştır. Saldırgan rolünde olan kali-linuxler ile saldırılar yapılmıştır. 30 dakikalık trafik toplanmış, bu süre içerisinde 5 dakika saldırı yapılmıştır. Saldırı sırasında wireshark ile toplanan paketler *pcap* dosyası olarak kayıt edilmiştir. Kayıt edilen *pcap* dosyasının boyutu 4.78 GB'tır. Bu oluşturulan *pcap* dosyası CIC-Flowmeter tarafından işlenerek flow kayıtları olarak *csv* haline dönüştürülmüştür. Dönüştürülen *csv* dosyasının boyutu 22 MB'tır. Bu *csv* dosyalarındaki verilerin, pyhthon da makine öğrenme algoritmaları ve 3 farklı öznitelik seçim yöntemi ile sınıflandırılması yapılmıştır. Yapılan sınıflandırma Çizelge 4.4'de ve karmaşıklık matrisleri Şekil 4.1'de gösteriliştir.

Çizelge 4.4. SQL atak veri kümesi için öznitelik seçimi doğruluk oranları

Öznitelik Seçimi	SMOTE Kullanımı	KNN	KA	RO	DVM
Özyinelemeli	Hayır	99.96	1.0	99.98	99.95
	Evet	99.60	99.95	99.96	98.82
İleri Yönelimli	Hayır	99.96	1.0	99.98	99.95
	Evet	99.73	99.97	99.96	99.22
Manuel	Hayır	99.96	99.98	1.0	99.90
	Evet	97.20	98.37	99.61	97.03

Çizelge 4.4'de SQL saldırısı yapmakta kullanılan öznitelikler ve makine öğrenme algoritmalarının doğruluk oranları verilmiştir. Ayrıca, özyinelemeli, ileri yönelimli ve manuel olarak adlandırılan üç farklı veri öznitelik seçimi ile bunların SMOTE uygulanmış halleri de yer almaktadır. Algoritmaların performansı genellikle çok yüksektir, ancak bazı varyasyonlarda bulunmaktadır. KNN algoritması SQL veri kümesinin tüm durumlarında %97,20 ile %99,96 arasında bir doğruluk oranına sahip olmuştur. SMOTE uygulandığında, doğruluk biraz düşse de hala yüksek bir doğruluk oranı korunmaktadır. KA algoritmasının doğruluk oranı %98,37 ile %100 arasındadır. Bu durum karar ağaçlarının genellikle bu tür veri kümelerinde mükemmel bir sonuç ve uyum sağladığını göstermektedir. RO algoritması da yüksek doğruluk oranlarını

göstermektedir. Bu oranlar %99,61 ile %100 arasındadır. Böylelikle RO algoritması da güçlü bir performans göstermiş olmuştur. Son olarak da DVM algoritmasının doğruluk oranı %97,03 ile %99,95 arasında kalmıştır. SMOTE ile kullanıldığında doğruluk oranı düşüşlerinin diğer algoritmalara göre biraz fazla olduğu görülmektedir. Sonuç olarak SQL veri kümesinde en başarılı algoritmayı RO ve KA olarak değerlendirilmektedir.



Şekil 4.1. SQL atakları için karmaşıklık matrisleri

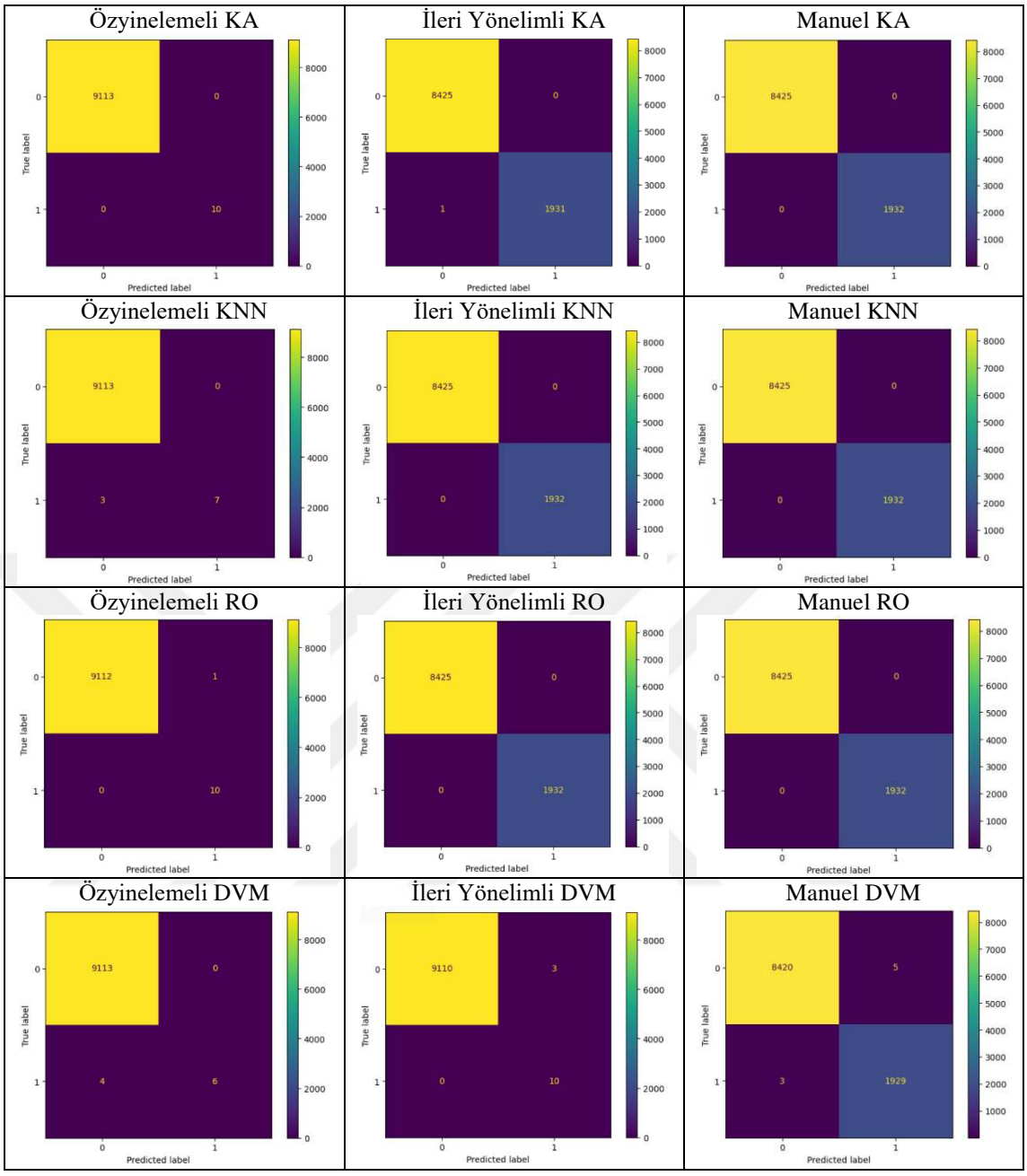
Server 2016 sunucuya yüklediğimiz FTP rolüne, Kali-Linux bilgisayarlardan sunucuya yapılan saldırılarla birlikte 28 dakika boyunca trafik toplanmıştır. Bu trafiğin 4 dakika 48 saniyesini FTP atakları oluşturmaktadır. Oluşturulan tüm trafik wireshark

kurulu bilgisayar sayesinde toplanmıştır. Toplanan bu trafik 4,21 GB'lık *pcap* dosyası olarak CIC-Flowmeter ile *csv* formatında FTP veri kümesi oluşturulmuştur. Bu *csv* dosyasının boyutu 23,6 MB'tır. Python kullanılarak, oluşturulan FTP veri kümesindeki veriler, makine öğrenimi algoritmaları ve üç farklı özellik seçme yöntemi kullanılarak sınıflandırılmıştır. Çizelge 4.5'de ortaya çıkan sınıflandırma çizelgesi, sınıflandırma çizelgesinin oluşturulmasında temelindeki karmaşıklık matrisleri de Şekil 4.2'de görülebilir.

Çizelge 4.5. FTP atak veri kümesi için öznitelik seçimi doğruluk oranları

Öznitelik Seçimi	SMOTE Kullanımı	KNN	KA	RO	DVM
Özyinelemeli	Hayır	1.0	1.0	1.0	99.96
	Evet	1.0	99.99	1.0	99.92
İleri Yönelimli	Hayır	1.0	99.99	1.0	99.95
	Evet	1.0	99.99	1.0	99.94
Manuel	Hayır	1.0	1.0	1.0	99.92
	Evet	1.0	99.99	1.0	99.90

Çizelge 4.5'de FTP saldırısı yapmakta kullanılan öznitelikler ve makine öğrenme algoritmalarının doğruluk oranları verilmiştir. Buna ilaveten SMOTE tekniği ile de veri artırımı yapılarak alınan sonuçların doğruluk oranları verilmiştir. KNN ve RO algoritmaları tam başarı sağlayarak %100 olarak doğruluk oranın yakalamıştır. KA algoritması da bu iki algoritmaya çok yakın bir performans sergilemiş olup %99.99 ile %100 arasında doğruluk oranına ulaşmıştır. DVM algoritması %99.90 ile %99.96 arasında bir başarı yakalamıştır. Sonuç olarak FTP veri kümesi için, makine öğrenimi algoritmalarının FTP saldırılarını tespit etmede oldukça etkili olduğunu görülmüştür. KNN ve RO, bu senaryo için en yüksek performansı gösterirken, KA ve DVM de oldukça rekabetçi bir performans sergilemiştir.



Şekil 4.2. FTP atakları için karmaşıklık matrisleri.

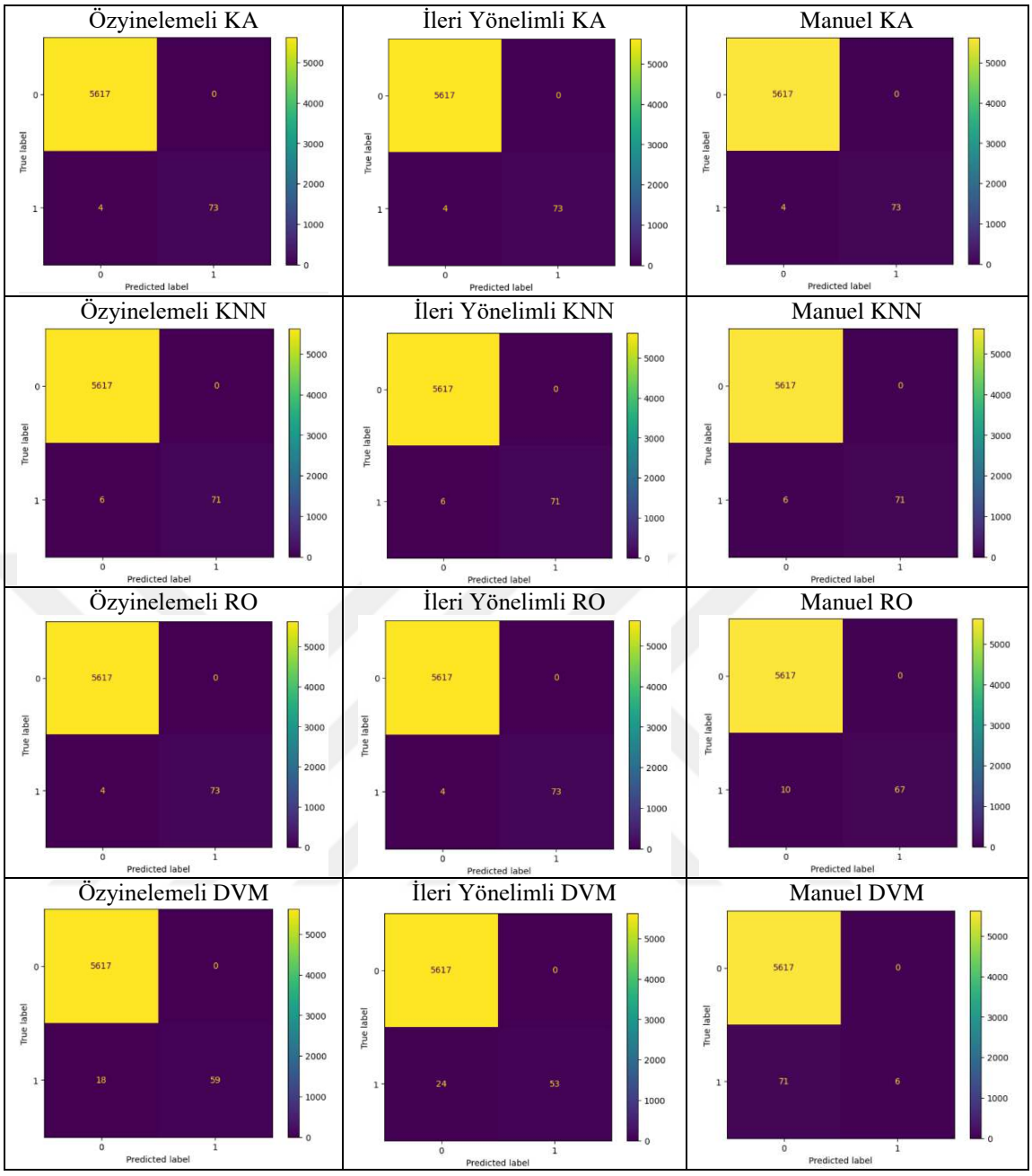
Emulatör ortamında bulunan clientlara kaba kuvvet saldırıları Kali-Linux bilgisayarlar tarafından tek seferde 1 dakika 15 saniye olarak, toplamda 3 dakika 45 saniye olacak şekilde 3 kez gerçekleştirilmiştir. Normal trafik ile birlikte toplamda oluşan trafik 29 dakika sürmüştür. Oluşan *pcap* dosyasının ve CIC-Flowmeter ile dönüştürülen *csv* dosyasının büyüklüğü sırası ile 3,89 GB ve 14.5 MB'tır. Kaba Kuvvet veri kümesindeki veriler, makine öğrenimi algoritmaları ve üç farklı özellik seçme yöntemi kullanılarak sınıflandırıldı. Çizelge 4.6'da ortaya çıkan sınıflandırma çizelgesi,

sınıflandırma çizelgesinin oluşturulmasında temelindeki karmaşıklık matrisleride Şekil 4.3'de verilmiştir.

Çizelge 4.6. Kaba Kuvvet atak veri kümesi için öznitelik seçimi doğruluk oranları

Öznitelik Seçimi	SMOTE Kullanımı	KNN	KA	RO	DVM
Özyinelemeli	Hayır	99.89	99.92	99.92	99.68
	Evet	99.75	99.89	99.92	99.61
İleri Yönelimli	Hayır	99.89	99.92	99.92	99.57
	Evet	99.75	99.91	99.92	99.54
Manuel	Hayır	99.89	99.92	99.82	98.75
	Evet	99.52	99.87	97.85	95.85

Çizelge 4.6'da çeşitli makine öğrenimi algoritmalarının kaba kuvvet saldırılarının tespiti için eğitilmiş veri kümelerindeki doğruluk oranlarını göstermektedir. Ayrıca, öznitelik seçimi yöntemleri olarak özyinelemeli, ileri yönelimli ve manuel yaklaşımlar değerlendirilmiş ve SMOTE ile dengelenmiş veri kümeleri de analiz edilmiştir. Bu analizler sonucunda KA ve RO en başarılı algoritma olarak ön plana çıkmaktadır. Bu iki algoritmanın doğruluk oranı %99.92 oranında olmuştur. Bu iki algoritmayı %99.89 ile KNN, %99.68 doğruluk oranı ile de DVM gelmektedir. Genel olarak, bu algoritmaların yüksek doğruluk oranları, kaba kuvvet saldırılarının tespit edilmesinde makine öğreniminin etkili olduğunu ve bu tür saldırıların genellikle iyi ayrılabilir olduğunu göstermektedir. Bu sonuçlar, siber güvenlik alanında makine öğreniminin önemli bir rol oynayabileceğini vurgulamaktadır.



Şekil 4.3. Kaba Kuvvet atakları için karmaşıklık matrisleri.

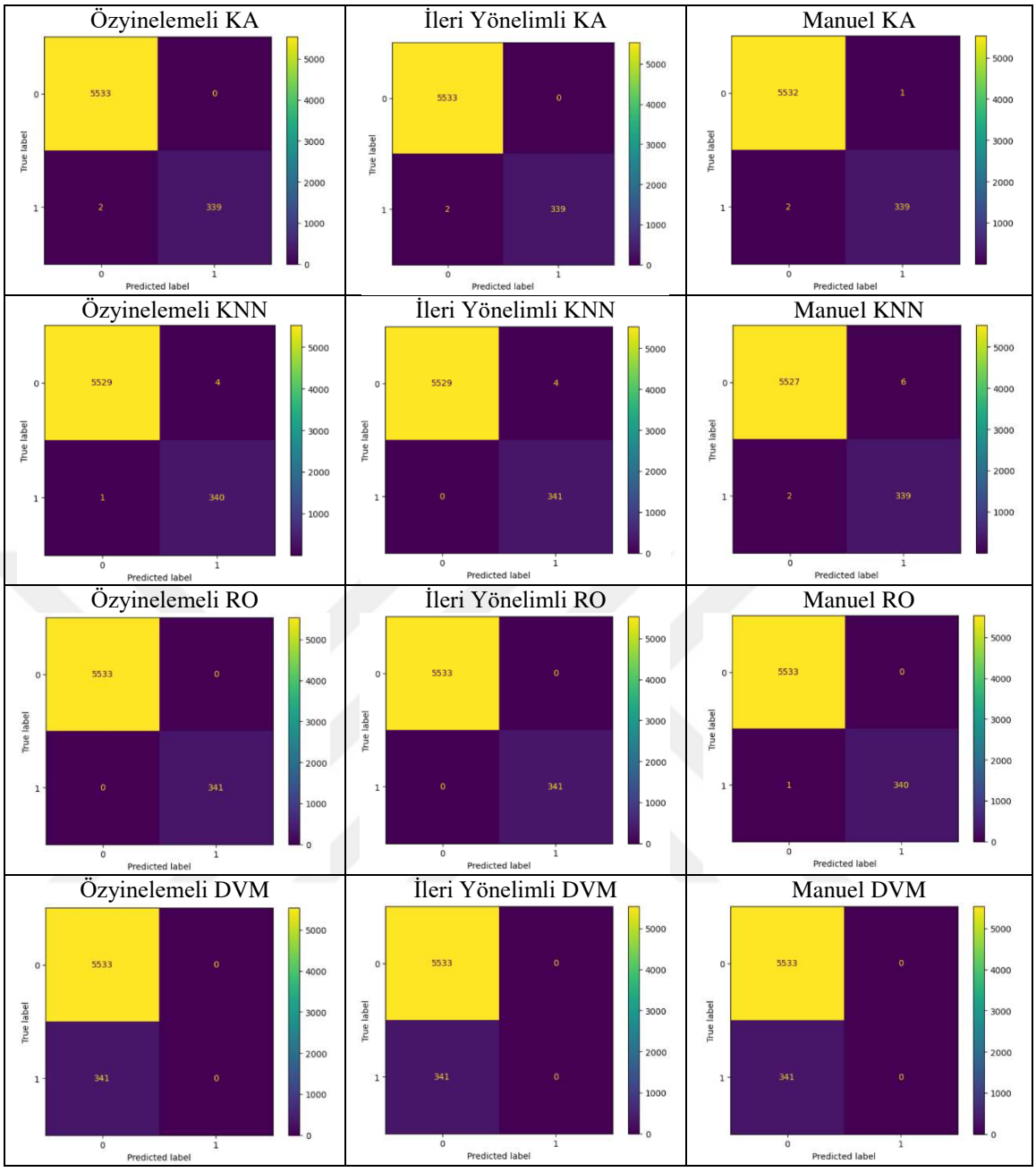
Server 2012 sunucu olan bilgisayara DNS rolü yüklenmiş ve DNS rolü için gerekli tüm ayarlamalar yapılmıştır. Kali-linux sunuculardan DNS sunucuya 2 dakika 10 saniye boyunca DNS amplification saldırısı yapılmıştır. Toplam trafiğin süresi ise 27 dakika ölçülmüştür. Wireshark ile toplanan trafiğin *pcap* dosyası boyutu 3.97 GB'tır. CIC-Flowmeter uygulaması ile *csv* formatında oluşturulan veri kümesinin boyutu 14.5 MB olmuştur. DNS veri kümesindeki veriler, makine öğrenimi algoritmaları ve üç farklı özellik seçme yöntemi kullanılarak sınıflandırılmıştır. Çizelge 4.7'de ortaya çıkan

sınıflandırma çizelgesi, sınıflandırma çizelgesinin oluşturulmasında temelindeki karmaşıklık matrisleride Şekil 4.4'de görülebilir.

Çizelge 4.7. DNS atak veri kümesi için öznitelik seçimi doğruluk oranları

Öznitelik Seçimi	SMOTE Kullanımı	KNN	KA	RO	DVM
Özyinelemeli	Hayır	99.91	99.96	1.0	94.19
	Evet	99.91	99.96	1.0	93.63
İleri Yönelimli	Hayır	99.93	99.96	1.0	94.19
	Evet	99.91	99.98	1.0	95.79
Manuel	Hayır	99.86	99.94	99.98	94.19
	Evet	99.82	99.96	99.98	94.62

Çizelge 4.7'de, DNS saldırılarını tespit etmek için kullanılan DNS veri kümesindeki çeşitli makine öğrenimi algoritmalarının doğruluğunu göstermektedir. Çizelgede KNN, KA, RO, DVM algoritmaları yer almaktadır. Öznitelik seçme yöntemleri olarak özyinelemeli, ileri ve manuel yöntemler değerlendirilmiş ve SMOTE tekniği de kullanılarak dengeli veri kümeleri de analiz edilmiştir. RO algoritması diğer algoritmalardan daha başarılı olarak %99.98 ile %100 arasında doğruluk oranlarına sahip olmuştur. RO algoritmasının ardında gelen KA algoritması %99.94 ile %99.98 oranları arasında bir doğruluk başarışı gerçekleştirmiştir. Bu iki algoritmadan sonra gelen KNN algoritması %99.82 ile %99.93 arasında bir başarı elde etmiştir. Son olarak da DVM algoritması %93.63 ile %95.79 arsında bir doğruluk oranında kalmıştır. RO algoritması, bu senaryoda en yüksek performansı göstermektedir. RO ve KA yüksek doğruluk oranlarına sahip olmuştur ve yorumlana bilirliği sayesinde saldırı tespiti için yararlı olabilmektedir. Kullanılan algoritmaların DNS trafiğindeki anormal davranışları belirlemek için etkili birer araç olduğunu göstermektedir. DVM'nin doğruluk oranı biraz daha düşük olsa da, bazı durumlarda etkili olabileceği düşünülmektedir.



Şekil 4.4. DNS atakları için karmaşıklık matrisleri.

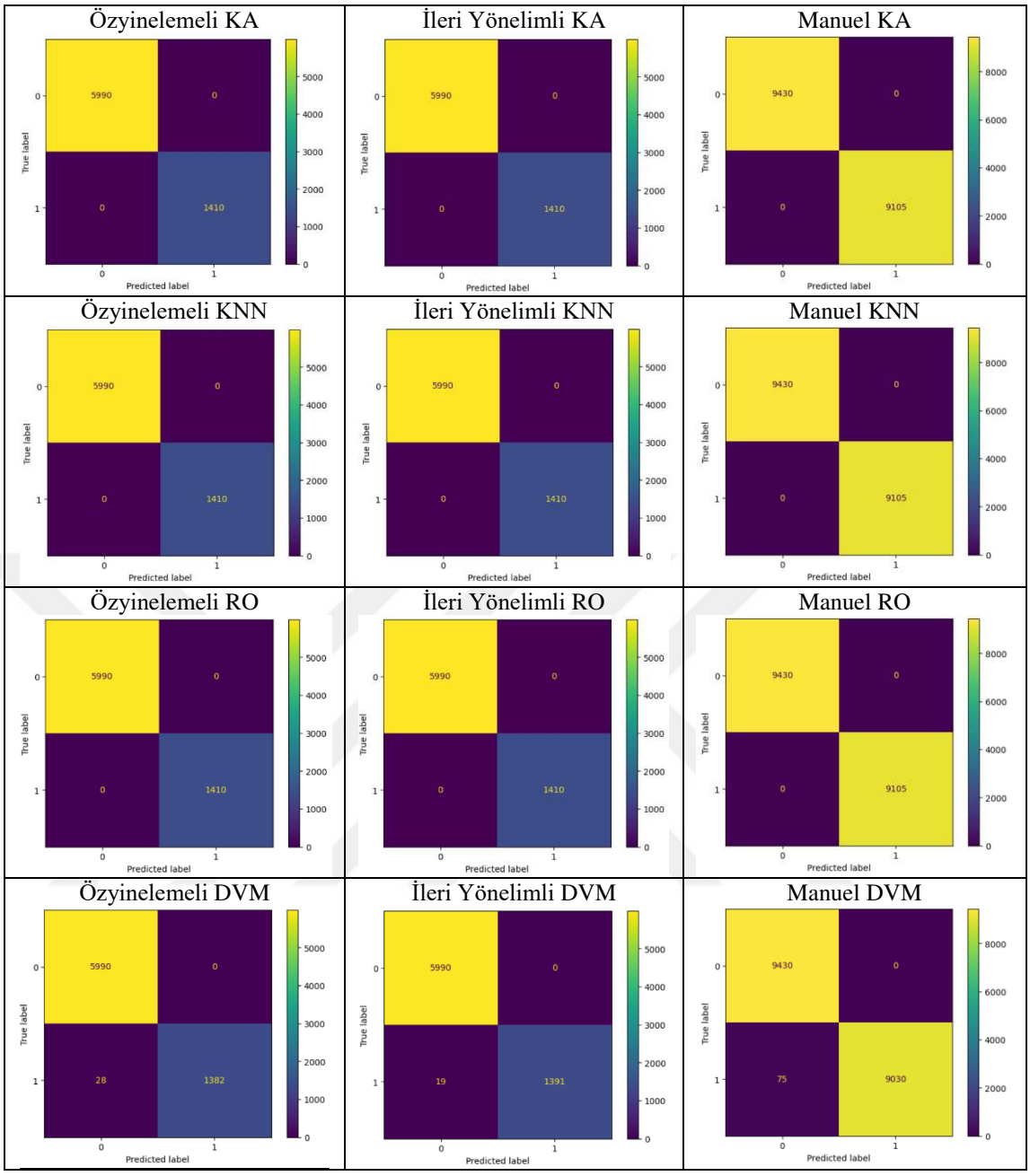
Server 2012 sunucu olan bilgisayara web sunucu hizmeti yüklenmiş ve gerekli tüm ayarlamalar yapılmıştır. Kali-linux sunuculardan web sunucuya toplamda 5 dakika 15 saniye boyunca DDoS saldırısı yapılmıştır. Toplam trafiğin süresi ise 33 dakika olarak ölçülmüştür. Wireshark ile toplanan trafiğin *pcap* dosyası boyutu 4.9 GB'tır. CIC-Flowmeter uygulaması ile *csv* formatında oluşturulan veri kümesinin boyutu 114 MB olmuştur. DDoS veri kümesindeki veriler, makine öğrenimi algoritmaları ve üç farklı özellik seçme yöntemi kullanılarak sınıflandırılmıştır. Çizelge 4.8'de ortaya çıkan

sınıflandırma çizelgesi, sınıflandırma çizelgesinin oluşturulmasında temelindeki karmaşıklık matrisleri de Şekil 4.5'de görülebilir.

Çizelge 4.8. DDoS atak veri kümesi için öznitelik seçimi doğruluk oranları

Öznitelik Seçimi	SMOTE Kullanımı	KNN	KA	RO	DVM
Özyinelemeli	Hayır	1.0	1.0	1.0	99.62
	Evet	1.0	1.0	1.0	99.77
İleri Yönelimli	Hayır	1.0	1.0	1.0	99.74
	Evet	1.0	1.0	1.0	99.83
Manuel	Hayır	1.0	1.0	1.0	99.59
	Evet	1.0	1.0	1.0	99.49

Çizelge 4.8'de çeşitli makine öğrenimi algoritmalarının DDoS saldırılarının tespiti için eğitilmiş veri kümelerindeki doğruluk oranlarını göstermektedir. Ayrıca, öznitelik seçimi yöntemleri olarak özyinelemeli, ileri yönelimli ve manuel yaklaşımlar değerlendirilmiş ve SMOTE ile dengelenmiş veri kümeleri de analiz edilmiştir. Bu analizler sonucunda KNN, KA ve RO en başarılı algoritma olarak ön plana çıkmaktadır. Bu üç algoritmanın doğruluk oranları %100 olmuştur. Bu üç algoritmayı %99.83 ile DVM algoritması takip etmektedir. Genel olarak, bu algoritmaların yüksek doğruluk oranları, DDoS saldırılarının tespit edilmesinde makine öğreniminin etkili olduğunu ve bu tür saldırıların genellikle iyi ayrılabilir olduğunu göstermektedir. DDOS saldırılarının tespit edilmesinde kullanılan veri kümesi oldukça iyi bir ayrılabilirliğe sahip görünmekte ve bu nedenle doğruluk oranları yüksek çıkmaktadır.



Şekil 4.5. DDoS atakları için karmaşıklık matrisleri.

Araştırmada kullanılan veri kümeleri, emülatör ortamında oluşturulan siber saldırı trafiğinden toplanmıştır. Bu durum, çalışmanın gerçek dünya koşullarına yakın bir ortamda gerçekleştirildiğini göstermektedir. Farklı saldırı türleri, emülatör ortamında simüle edilerek, gerçekçi bir saldırı ortamı yaratılmıştır. Bu saldırı ortamından elde edilen bulguların siber güvenlik uygulamalarında pratik bir öneme sahip olmasını sağlamaktadır.

Her bir saldırı türü için makine öğrenimi algoritmalarının etkinliği karşılaştırılmıştır. KNN, KA ve RO algoritmalarının genellikle en iyi performansı

sergilediği, DVM algoritmasının ise biraz daha düşük doğruluk oranlarına sahip olduğu gözlemlenmiştir. Yapılan bu gözlemler sonucunda, ağaç tabanlı algoritmaların ve KNN'nin siber saldırıların tespitinde daha etkili olabileceğini göstermektedir. Ancak bu durum, DVM algoritmasının da bazı senaryolarda etkili olabileceği gerçeğini değiştirmemektedir.

Çizelge 4.9'da literatürde yer alan çalışmaların doğruluk oranlarının karşılaştırılmalı bir çizelgesi verilmiştir. Çizelgede verilmiş olan çalışmalar hazır veri setleri kullanılmış olup, çalışmamızda kullanılan veri kümelerimizin tamamı emülatör ortamımızda elde ettiğimiz veri kümeleridir. Makine öğrenme teknikleri, öznitelik eleme ve SMOTE tekniği kullanılarak genel olarak %99.92 ile %1.0 arasında değişen doğruluk oranları elde edilmiştir. Çizelge 4.9'da verilmiş olan doğruluk oranlarına göre aldığımız sonuçlar diğer yapılan çalışmalara göre daha başarılı olmuştur.

Çizelge 4.9. Literatür karşılaştırması

Çalışma	**Metot**	**Doğruluk Oranı**
Sarkar, Sharma, and Singh 2023	MLP	89,32
Tang, Luktarhan, and Zhao 2020	LightGBM	89,82
Han, Kim, and Kim 2022	YSA	96,55
Mazini, Shirazi, and Mahdavi 2019	AdaBoost.M2	98,90
Kanimozhi and Prem Jacob 2019	YSA	99,97
Panigrahi et al. 2021	KA	97,28
Türkoğlu et al. 2022	KA	99,35
Emanet et al. 2021	EA	98,79
Ekici and Takcı 2022	RO	94.00
Jumabek, Yang, and Noh 2021	CatBoost	92,41
Sanchez et al. 2021	RO	99,98
Bıçakcı and Toklu 2022	RO	99,67
Fitni and Ramli 2020	RO	98,80
Emhan and Akın 2019	RO	93,40
Pehlivanoğlu, Atay, and Odabaş 2019	CNN+ RO	98,00
(Hamid, Sugumaran, and Journaux 2016	KA	99,96
Dey and Rahman 2019	GRU-LSTM	87,91
Nanda et al. 2017	Bayes	91,68
Siva Shankar et al. 2024	GEO based SMPIF	99,97
Ye et al. 2018	DVM	95,24
Haider et al. 2020	CNN	99,45
Bu Çalışma	KNN, RO, KA	99,92 -1.0 Aralığında

5. SONUÇ VE ÖNERİLER

Günümüz toplumunda siber güvenlik konusu önemli bir endişe kaynağı olarak ön plana çıkmıştır. Devlet kurumları ve kurumsal işletmelerin elinde bulunan hassas bilgilerin güvenli bir şekilde korunmasını sağlamak büyük önem taşımaktadır. Kötü niyetli saldırganların yetkisiz erişimi ve güvenliği ihlal etmesi, hafife alınamayacak ciddi bir tehdit oluşturmaktadır. Bu değerli bilgilerin korunması, sürekli olarak yetkisiz erişim tehdidi nedeniyle zorluk teşkil etmektedir. Bu riski azaltmak için araştırmacılar kapsamlı çalışmalar yürütmekte, kendilerini aktif olarak siber saldırganlar gibi düşünerek siber saldırılara karşı önleyici tedbirler geliştirmektedir. Devam eden çabaları, siber güvenliği destekleyen üç temel ilkenin titizlikle uygulanması etrafında dönmektedir. Bu üç temel ilke gizlilik, bütünlük ve erişilebilirliktir.

Bu çalışmanın bir diğer amacı ağ trafiğindeki anormallikleri tespit edip raporlayabilen bilgisayarlı sistemlerin emülasyon ortamında test edilmesini ve yeni test havuzlarının oluşturulabilmesini sağlamaktır. Emülatör ortamında gerçek ağ cihazlarına ait işletim sistemleri ve gerçek bilgisayar işletim sistemleri kullanılmış ve böylece elde edilen veri kümesinin gerçek sistemlere oldukça benzer olması sağlanmıştır. Böylelikle elde edilen başarılı sonuçlardan yola çıkarak canlı sistemlerde de bu tespit yöntemlerinin kullanılabileceği düşünülmektedir. Tasarlanan sistem emülatör ortamında olduğu için gerektiğinde farklı topolojiler oluşturarak farklı durumlar rahatlıkla test edilebilir. Bu çalışma bu açıdan da önemli bir katkı sağlayacaktır.

Anormallik tespit sistemleri, ağ davranışının normal kalıplarını analiz ederek olağandışı veya beklenmedik etkinlikleri tespit edebilir ve işaretleyebilir. Bu sistemlerin uygulanması ağdaki riskleri etkili bir şekilde azaltabilir ve verileri koruyabilir. Bu hedefe ulaşmak için, literatürün kapsamlı bir incelemesi yoluyla ağ anormallik tespiti için temel bileşenleri belirleyen önceki araştırmaların kapsamlı bir analizi yapıldı. Yapılan literatür araştırmasından makine öğrenimi algoritmalarının son zamanlarda dikkate değer düzeyde yüksek doğruluk oranlarına ulaştığı açıkça görülmektedir.

Yapılan çalışmanın bulgularına göre, siber saldırıların tespiti için farklı makine öğrenimi algoritmaları kullanılarak önemli sonuçlar elde edilmiştir. Çalışmanın, SQL Injection, FTP, DDoS ve DNS saldırıları gibi çeşitli saldırı türlerine odaklandığı görülmektedir. Tüm bu saldırı türlerinin tespit edilmesinde kullanılan algoritmalar arasında genel olarak yüksek doğruluk oranları elde edilmiştir. Özellikle KNN, KA ve

RO algoritmaları, birçok durumda %100 doğruluk oranına ulaşarak, bu tür saldırıların tespitinde son derece etkili olduklarını kanıtlamıştır.

Araştırmada kullanılan öznitelik seçimi yöntemleri, özyinelemeli, ileri yönelimli ve manuel yaklaşımları kapsamaktadır. SMOTE tekniği, veri dengesizliğini gidermek için etkili bir yöntem olarak kullanılmıştır. Öznitelik seçimi, veri kümesinin karmaşıklığını azaltarak daha etkili makine öğrenimi modelleri oluşturulmasına olanak sağlamıştır. Araştırmanın bulgularına göre, en başarılı algoritmalar genellikle KA ve RO olmuştur. Bu algoritmalar, yüksek doğruluk oranlarına sahip olmanın yanı sıra, kullanıcıların saldırı tespiti sürecini anlamalarını kolaylaştıran yorumlanabilir modellerdir.

Çalışmanın bir diğer önemli bulgusu, siber saldırıların tespitinde makine öğrenimi algoritmalarının genellikle etkili olduğudur. Kaba kuvvet saldırıları, DNS saldırıları, DDoS saldırıları ve SQL saldırılarının tümü için yüksek doğruluk oranlarına ulaşılmıştır. Alınan bu yüksek doğruluk oranları, siber güvenlik alanında makine öğreniminin önemli bir rol oynayabileceğini göstermektedir. Yapılan çalışmanın, çeşitli saldırı türlerini kapsayarak, siber saldırıların tespiti için kapsamlı bir yaklaşım sunması, gelecekteki araştırmalar için de faydalı olabilecek bir model ortaya koymaktadır.

KAYNAKLAR

Acito, F. (2023). Ordinary Least Squares Regression. *Predictive Analytics with KNIME*, 105–124. https://doi.org/10.1007/978-3-031-45630-5_6

Adiwal, S., Rajendran, B., D., P. S., ve Sudarsan, S. D. (2023). DNS Intrusion Detection (DID) — A SNORT-based solution to detect DNS Amplification and DNS Tunneling attacks. *Franklin Open*, *2*, 100010. https://doi.org/10.1016/J.FRAOPE.2023.100010

Akar, Ö., ve Güngör, O. (2012). Rastgele Orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması. *Jeodezi ve Jeoinformasyon Dergisi*, *1*(106), 139–146. https://doi.org/10.9733/JGG.241212.1T

Alfa. (2024). *SPSS'te İlişki ve Anlamlılık*. https://alfaistatistik.com/blog/16/spss-te-iliski-ve-anlamlilik

Ali, B. H., Sulaiman, N., Al-Haddad, S. A. R., Atan, R., ve Hassan, S. L. M. (2022). DDoS Detection Using Active and Idle Features of Revised CICFlowMeter and Statistical Approaches. *ICOASE 2022 - 4th International Conference on Advanced Science and Engineering*, 148–153. https://doi.org/10.1109/ICOASE56293.2022.10075591

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., ve Aljaaf, A. J. (2020). *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science*. 3–21. https://doi.org/10.1007/978-3-030-22475-2_1

Altuntas, Y., Kocamaz, A. F., ve Ulkgun, A. M. (2020). Determination of Individual Investors' Financial Risk Tolerance by Machine Learning Methods. *2020 28th Signal Processing and Communications Applications Conference, SIU 2020 - Proceedings*. https://doi.org/10.1109/SIU49456.2020.9302294

Aung, H. L. (2020). Implementation of traffic engineering with segment routing and opendaylight controller on emulated virtual environment next generation (EVE-NG). *Chulalongkorn University Theses and Dissertations (Chula ETD)*. https://doi.org/10.58837/CHULA.THE.2020.161

Aydın, A., Doğru, İ. A., ve Dörterler, M. (2018). Makine Öğrenmesi Algoritmalarıyla Android Kötücül Yazılım Uygulamalarının Tespiti. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *22*(2), 1087. https://doi.org/10.19113/SDUFBED.20066

Aydın, H. (2023). Yönetim Bilgi Sistemlerinde (YBS) Siber Güvenliğin Önemi. *Bilgisayar Bilimleri ve Teknolojileri Dergisi*, *3*(2), 38–45. https://doi.org/10.54047/BIBTED.1138252

Baykara, M., ve Daş, R. (2019). Saldırı tespit ve engelleme araçlarının incelenmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *10*(1), 57–75. https://doi.org/10.24012/DUMF.449059

Bıçakcı, M. S., ve Toklu, S. (2022). Bilgisayar Ağı Güvenliği için Hibrit Öznitelik Azaltma ile Makine Öğrenmesine Dayalı Bir Saldırı Tespit Sistemi Tasarımı. *Journal of Gaziosmanpasa Scientific Research*, *11*(3), 203–220. https://dergipark.org.tr/en/pub/gbad/issue/74308/1200540

Bro, R., Sidiropoulos, N. D., ve Smilde, A. K. (2002). Maximum likelihood fitting using ordinary least squares algorithms. *Journal of Chemometrics*, *16*(8–10), 387–400. https://doi.org/10.1002/CEM.734

Çelik, S., Çetinkaya Bozkurt, Ö., ve Ekşili, N. (2022). Çalışan Performansı Ölçeğindeki İfadelerin Karar Ağacı Algoritması İle Belirlenmesi. *Journal of Mehmet Akif Ersoy University Economics and Administrative Sciences Faculty*, *9*(1), 561–584. https://doi.org/10.30798/MAKUIIBF.913478

Dey, S. K., ve Rahman, M. M. (2019). Effects of Machine Learning Approach in Flow-Based Anomaly Detection on Software-Defined Networking. *Symmetry 2020, Vol. 12, Page 7*, *12*(1), 7. https://doi.org/10.3390/SYM12010007

Dina, A. S., ve Manivannan, D. (2021). Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things*, *16*, 100462. https://doi.org/10.1016/J.IOT.2021.100462

Dung, N. Q. (2023). Detecting Cyber-Attacks on Internet of Things Devices: An Effective Preprocessing Method. *Lecture Notes in Electrical Engineering*, *984 LNEE*, 531–546. https://doi.org/10.1007/978-981-19-8493-8_40/TABLES/7

Ekici, B., ve Takcı, H. (2022). Bilgisayar Ağlarında Anomali Tespiti Yaklaşımı ile Saldırı Tespiti. *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, *22*(5), 1016–1027. https://doi.org/10.35414/AKUFEMUBID.1114906

Emanet, S., Karatas Baydogmus, G., Demir, O., Üniversitesi, M., Bölümü, M., ve Üniversitesi, B. (2021). Öznitelik Seçme Yöntemlerinin Makine Öğrenmesi Tabanlı Saldırı Tespit Sistemi Performansına Etkileri. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *12*(5), 743–755. https://doi.org/10.24012/DUMF.1051340

Emhan, Ö., ve Akın, M. (2019). Filtreleme Tabanlı Öznitelik Seçme Yöntemlerinin Anomali Tabanlı Ağ Saldırısı Tespit Sistemlerine Etkisi. *DÜMF Mühendislik Dergisi*, *10*(2), 549–559. https://doi.org/10.24012/dumf.565842

Fitni, Q. R. S., ve Ramli, K. (2020). Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. *Proceedings - 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2020*, 118–124. https://doi.org/10.1109/IAICT50021.2020.9172014

Gönen, S., Ulus, H. İ., ve Nurcan Yılmaz, E. (2016). Bilişim Alanında İşlenen Suçlar Üzerine Bir İnceleme. *Bilişim Teknolojileri Dergisi*, *9*(3), 229. https://doi.org/10.17671/BTD.90710

Güler, H., Şahinkayasi, Y., ve Şahinkayası, H. (2017). *İnternet Ve Mobil Teknolojilerin Yaygınlaşması: Fırsatlar Ve SınırlılıklaR The Penetration of Internet and Mobile Information Technologies: Opportunities and Limitations*.

Haider, S., Akhunzada, A., Mustafa, I., Patel, T. B., Fernandez, A., Choo, K. K. R., ve Iqbal, J. (2020). A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks. *IEEE Access*, *8*, 53972–53983. https://doi.org/10.1109/ACCESS.2020.2976908

Hamid, Y., Sugumaran, M., ve Journaux, L. (2016). Machine learning techniques for intrusion detection: A comparative analysis. *ACM International Conference Proceeding Series*, *25-26-August-2016*. https://doi.org/10.1145/2980258.2980378

Han, H., Kim, H., ve Kim, Y. (2022). An Efficient Hyperparameter Control Method for a Network Intrusion Detection System Based on Proximal Policy Optimization. *Symmetry 2022, Vol. 14, Page 161*, *14*(1), 161. https://doi.org/10.3390/SYM14010161

Han, S., Kim, H., ve Lee, Y. S. (2020). Double random forest. *Machine Learning*, *109*(8), 1569–1586. https://doi.org/10.1007/S10994-020-05889-1/TABLES/8

Harahus, M., Čavojský, M., Bugár, G., ve Pleva, M. (2023). Interactive Network Learning: An Assessment of EVE-NG Platform in Educational Settings. *Acta Electrotechnica et Informatica*, *23*(3), 3–9. https://doi.org/10.2478/AEI-2023-0011

Hasan, M. F., Hadi, H. S., ve Hayder Jasim, N. A. (2021). The Validity of Altman's Models in Predicting Iraqi Private-Banks Soundness. *Journal of Management and Accounting Studies*, *9*(01), 79–89. https://doi.org/10.24200/JMAS.VOL9ISS01PP79-89

Janiesch, C., Zschech, P., ve Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695. https://doi.org/10.1007/S12525-021-00475-2/TABLES/2

Jumabek, A., Yang, S., ve Noh, Y. (2021). *CatBoost-Based Network Intrusion Detection on Imbalanced CIC-IDS-2018 Dataset*. *46*(12), 2191–2197. https://doi.org/10.7840/KICS.2021.46.12.2191

Kanimozhi, V., ve Prem Jacob, T. (2019). Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. *Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019*, 33–36. https://doi.org/10.1109/ICCSP.2019.8698029

Kara, Ş., Zengin, A., ve Hizal, S. (2023). *Ağ Sistemlerinin Güvenliği İçin Siber Saldırıların Ayrık Olaylı Sistem Tanımlama Tabanlı Modellenmesi ve Simülasyonu Discrete Event System Identification Based Modeling and Simulation of Cyber Attacks for the Security of Network Systems*. *8*, 6–202.

Karaman, M. S., Turan, M., ve Aydın, M. A. (2021). Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespit Modeli Uygulaması. *Avrupa Bilim ve Teknoloji Dergisi*, *Ejosat Ek Özel Sayı (HORA)*, 10–17. https://doi.org/10.31590/EJOSAT.1115825

Kaur, R., ve Gupta, N. (2023). Comprehending SMOTE Adaptations to Alleviate Imbalance in Intrusion Detection Systems. *2023 4th International Conference on Electronics and Sustainable Communication Systems, ICESC 2023 - Proceedings*, 976–982. https://doi.org/10.1109/ICESC57686.2023.10193257

Kemalbay, G., ve Alkış, B. N. (2021). Borsa endeks hareket yönünün çoklu lojistik regresyon ve k-en yakın komşu algoritması ile tahmini. *Pamukkale University Journal of Engineering Sciences*, *27*(4), 556–569. https://doi.org/10.5505/pajes.2020.57383

Kim, M. (2019). Supervised learning-based DDoS attacks detection: Tuning hyperparameters. *ETRI Journal*, *41*(5), 560–573. https://doi.org/10.4218/ETRIJ.2019-0156

Küçüksille, E. U., ve Ateş, N. (2016). Destek Vektör Makineleri ile Yaramaz Elektronik Postaların Filtrelenmesi. *TBV Journal of Computer Science and Engineering*, *6*(1). https://dergipark.org.tr/en/pub/tbbmd/issue/22246/238807

Li, Z.-H., Sun, J.-C., ve Jain, G. (2021). *Application of SNORT and Wireshark in Network Traffic Analysis*. https://doi.org/10.1088/1757-899X/1119/1/012007

Masdari, M., ve Khezri, H. (2020). A survey and taxonomy of the fuzzy signature-based Intrusion Detection Systems. *Applied Soft Computing*, *92*, 106301. https://doi.org/10.1016/J.ASOC.2020.106301

Mazini, M., Shirazi, B., ve Mahdavi, I. (2019). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University - Computer and Information Sciences*, *31*(4), 541–553. https://doi.org/10.1016/J.JKSUCI.2018.03.011

Meng, T., Jing, X., Yan, Z., ve Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, *57*, 115–129. https://doi.org/10.1016/J.INFFUS.2019.12.001

Musa, U. S., Chhabra, M., Ali, A., ve Kaur, M. (2020). Intrusion Detection System using Machine Learning Techniques: A Review. *Proceedings - International Conference on Smart Electronics and Communication, ICOSEC 2020*, 149–155. https://doi.org/10.1109/ICOSEC49089.2020.9215333

Nanda, S., Zafari, F., Decusatis, C., Wedaa, E., ve Yang, B. (2017). Predicting network attack patterns in SDN using machine learning approach. *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2016*, 167–172. https://doi.org/10.1109/NFV-SDN.2016.7919493

Okay, M. O., Samet, R., Aslan, O., ve Gupta, D. (2021). A Comprehensive Systematic Literature Review on Intrusion Detection Systems. *IEEE Access*, *9*, 157727–157760. https://doi.org/10.1109/ACCESS.2021.3129336

Özger, F. (2023). *Makine öğrenmesi algoritmalarının hibrit yaklaşımı ile ağ anomalisi tespiti*. http://acikerisim.subu.edu.tr/xmlui/handle/20.500.14002/2440

Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Jhaveri, R. H., ve Chowdhary, C. L. (2021). Performance Assessment of Supervised Classifiers for Designing Intrusion Detection Systems: A Comprehensive Review and Recommendations for Future Research. *Mathematics 2021, Vol. 9, Page 690*, *9*(6), 690. https://doi.org/10.3390/MATH9060690

Patro, S. G. K., Mishra, B. K., Panda, S. K., Kumar, R., Long, H. V., Taniar, D., ve Priyadarshini, I. (2020). A Hybrid Action-Related K-Nearest Neighbour (HAR-KNN) Approach for Recommendation Systems. *IEEE Access*, *8*, 90978–90991. https://doi.org/10.1109/ACCESS.2020.2994056

Pehlivanoğlu, K. M., Atay, R., ve Odabaş, D. E. (2019). İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi ile Saldırı Tespiti. *Gazi Mühendislik Bilimleri Dergisi*, *5*(3), 258–272. https://doi.org/10.30855/GMBD.2019.03.07

Sanchez, O. R., Repello, M., Carrega, A., ve Bolla, R. (2021). Evaluating ML-based DDoS Detection with Grid Search Hyperparameter Optimization. *Proceedings of the 2021 IEEE Conference on Network Softwarization: Accelerating Network Softwarization in the Cognitive Age, NetSoft 2021*, 402–408. https://doi.org/10.1109/NETSOFT51509.2021.9492633

Sanjay, B. R., ve D, P. S. (2020). DNS Amplification DNS Tunneling Attacks Simulation, Detection and Mitigation Approaches. *Proceedings of the 5th International Conference on Inventive Computation Technologies, ICICT 2020*, 230–236. https://doi.org/10.1109/ICICT48043.2020.9112413

Sarkar, A., Sharma, H. S., ve Singh, M. M. (2023). A supervised machine learning-based solution for efficient network intrusion detection using ensemble learning based on hyperparameter optimization. *International Journal of Information Technology (Singapore)*, *15*(1), 423–434. https://doi.org/10.1007/S41870-022-01115-4/TABLES/7

Sasi, G., Thanapal, P., Balaji, V. S., Venkat Babu, G., ve Elamaran, V. (2020). A Handy Approach for Teaching and Learning Computer Networks using Wireshark. *Proceedings of the 4th International Conference on Inventive Systems and Control, ICISC 2020*, 456–461. https://doi.org/10.1109/ICISC47916.2020.9171197

Savaş, T., ve Savaş, S. (2022). Tekdüzen Kaynak Bulucu Yoluyla Kimlik Avı Tespiti için Makine Öğrenmesi Algoritmalarının Özellik Tabanlı Performans Karşılaştırması. *Politeknik Dergisi*, *25*(3), 1261–1270. https://doi.org/10.2339/politeknik.1035286

Shams, M. (2020). *Ağ Anomalisi Tespitinde Makine Öğrenmesi Algoritmalarının Kullanımı ve Karşılaştırmalı Analizi*. https://acikerisim.sakarya.edu.tr/handle/20.500.12619/97091

Shurrab, S., ve Duwairi, R. (2022). Self-supervised learning methods and applications in medical imaging analysis: a survey. *PeerJ Computer Science*, *8*, e1045. https://doi.org/10.7717/PEERJ-CS.1045

Siva Shankar, S., Hung, B. T., Chakrabarti, P., Chakrabarti, T., ve Parasa, G. (2024). A novel optimization based deep learning with artificial intelligence approach to detect intrusion attack in network system. *Education and Information Technologies*, *29*(4), 3859–3883. https://doi.org/10.1007/S10639-023-11885-4/TABLES/3

Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., ve Sakurai, K. (2020). Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture. *Sensors 2020, Vol. 20, Page 4372*, *20*(16), 4372. https://doi.org/10.3390/S20164372

Taha Jijo, B., ve Mohsin Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165

Tang, C., Luktarhan, N., ve Zhao, Y. (2020). An Efficient Intrusion Detection Method Based on LightGBM and Autoencoder. *Symmetry 2020, Vol. 12, Page 1458*, *12*(9), 1458. https://doi.org/10.3390/SYM12091458

Tüfekci, P., ve Önal, Ç. M. (2024). Kötü Amaçlı Yazılım Tespiti için Makine Öğrenmesi Algoritmalarının Kullanımı. *Duzce University Journal of Science and Technology*, *12*(1), 307–319. https://doi.org/10.29130/DUBITED.1287453

Türkoğlu, M., Polat, H., Koçak, C., ve Polat, O. (2022). Recognition of DDoS attacks on SD-VANET based on combination of hyperparameter optimization and feature selection. *Expert Systems with Applications*, *203*, 117500. https://doi.org/10.1016/J.ESWA.2022.117500

Van Engelen, J. E., ve Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/S10994-019-05855-6/FIGURES/5

Wang, H. (2020). Garbage recognition and classification system based on convolutional neural network vgg16. *Proceedings - 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2020*, 252–255. https://doi.org/10.1109/AEMCSE50948.2020.00061

Yazar, B. K., Akleylek, S., ve Kılıç, E. (2022). Bulut Bilişim Güvenliği İçin Kullanılan Makine Öğrenimi Yöntemleri Üzerine Bir Derleme. *Duzce University Journal of Science and Technology*, *10*(2), 893–913. https://doi.org/10.29130/DUBITED.979040

Ye, J., Cheng, X., Zhu, J., Feng, L., ve Song, L. (2018). A DDoS Attack Detection Method Based on SVM in Software Defined Network. *Security and Communication Networks*, *2018*. https://doi.org/10.1155/2018/9804061

Zhang, S., Xie, X., ve Xu, Y. (2020). A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity. *IEEE Access*, *8*, 128250–128263. https://doi.org/10.1109/ACCESS.2020.3008433

EKLER

EK 1 - **Çizelge 3.1.** Veri kümesi öznitelikleri ve açıklamaları

Öznitelik	Açıklama
1-Dst Port	Hedef port
2-Protocol	Protokol
3-Flow Duration	Akış süresi
4-Tot Fwd Pkts	İleri yöndeki toplam paket sayısı
5-Tot Bwd Pkts	Geriye doğru yöndeki toplam paket sayısı
6-TotLen Fwd Pkts	İleri yöndeki toplam paket boyutu
7-TotLen Bwd Pkts	Geri yöndeki toplam paket boyutu
8-Fwd Pkt Len Max	İleri yöndeki maksimum paket boyutu
9-Fwd Pkt Len Min	İleri yöndeki minimum paket boyutu
10-Fwd Pkt Len Mean	İleri yönde paketin ortalama boyutu
11-Fwd Pkt Len Std	Paketin ileri yöndeki standart sapma boyutu
12-Bwd Pkt Len Max	Geriye doğru yönde maksimum paket boyutu
13-Bwd Pkt Len Min	Geriye doğru yönde minimum paket boyutu
14-Bwd Pkt Len Mean	Geri yönde paketin ortalama boyutu
15-Bwd Pkt Len Std	Paketin geriye yöndeki standart sapma boyutu
16-Flow Byts/s	Saniyede aktarılan paket sayısı olan akış bayt hızı
17-Flow Pkts/s	Saniyede aktarılan paket sayısı olan akış paketleri oranı
18-Flow IAT Mean	İki akış arasında geçen ortalama süre
19-Flow IAT Std	İki akışın standart sapma süresi
20-Flow IAT Max	İki akış arasındaki maksimum süre
21-Flow IAT Min	İki akış arasındaki minimum süre
22-Fwd IAT Tot	İleri yönde gönderilen iki paket arasındaki toplam süre
23-Fwd IAT Mean	İleri yönde gönderilen iki paket arasındaki ortalama süre
24-Fwd IAT Std	İleri yönde gönderilen iki paket arasındaki standart sapma süresi
25-Fwd IAT Max	İleri yönde gönderilen iki paket arasındaki maksimum süre
26-Fwd IAT Min	İleri yönde gönderilen iki paket arasındaki minimum süre
27-Bwd IAT Tot	Geri yönde gönderilen iki paket arasındaki toplam süre
28-Bwd IAT Mean	Geri yönde gönderilen iki paket arasındaki ortalama süre
29-Bwd IAT Std	Geri yönde gönderilen iki paket arasındaki standart sapma süresi
30-Bwd IAT Max	Geri yönde gönderilen iki paket arasındaki maksimum süre
31-Bwd IAT Min	Geri yönde gönderilen iki paket arasındaki minimum süre
32-Fwd PSH Flags	İleri yönde hareket eden paketlerde PSH bayrağının ayarlanma sayısı
33-Bwd PSH Flags	Geri yönde hareket eden paketlerde PSH bayrağının ayarlanma sayısı
34-Fwd URG Flags	İleri yönde hareket eden paketlerde URG bayrağının ayarlanma sayısı
35-Bwd URG Flags	Geri yönde hareket eden paketlerde URG bayrağının ayarlanma sayısı
36-Fwd Header Len	İleri yöndeki başlıklar için kullanılan toplam bayt sayısı
37-Bwd Header Len	İleri yöndeki başlıklar için kullanılan toplam bayt sayısı
38-Fwd Pkts/s	Saniyedeki ileri paket sayısı

39-Bwd Pkts/s	Saniyedeki geri paket sayısı
40-Pkt Len Min	Saniyedeki geri paket sayısı
41-Pkt Len Max	Bir akışın maksimum uzunluğu
42-Pkt Len Mean	Bir akışın ortalama uzunluğu
43-Pkt Len Std	Bir akışın standart sapma uzunluğu
44-Pkt Len Var	Paketin minimum varışlar arası süresi
45-FIN Flag Cnt	FIN'li paket sayısı
46-SYN Flag Cnt	SYN'li paket sayısı
47-RST Flag Cnt	RST'li paket sayısı
48-PSH Flag Cnt	PUSH'lı paket sayısı
49-ACK Flag Cnt	ACK'li paket sayısı
50-URG Flag Cnt	URG'li paket sayısı
51-CWE Flag Count	CWE'li paket sayısı
52-ECE Flag Cnt	ECE'li paket sayısı
53-Down/Up Ratio	İndirme ve yükleme oranı
54-Pkt Size Avg	Ortalama paket boyutu
55-Fwd Seg Size Avg	İleri yönde gözlemlenen ortalama boyut
56-Bwd Seg Size Avg	Geri yönde gözlemlenen ortalama boyut
57-Fwd Byts/b Avg	İleri yöndeki ortalama bayt sayısı toplu hızı
58-Fwd Pkts/b Avg	İleri yönde ortalama paket sayısı toplu hızı
59-Fwd Blk Rate Avg	İleri yönde ortalama toplu oran sayısı
60-Bwd Byts/b Avg	Geri yöndeki ortalama bayt sayısı toplu oranı
61-Bwd Pkts/b Avg	Geri yönde ortalama paket sayısı toplu oranı
62-Bwd Blk Rate Avg	Geri yöndeki ortalama toplu oran sayısı
63-Subflow Fwd Pkts	İleri yönde bir alt akıştaki ortalama paket sayısı
64-Subflow Fwd Byts	İleri yönde bir alt akıştaki ortalama bayt sayısı
65-Subflow Bwd Pkts	Bir alt akışta geriye doğru yöndeki ortalama paket sayısı
66-Subflow Bwd Byts	Geri yönde bir alt akıştaki ortalama bayt sayısı
67-Init Fwd Win Byts	Başlangıç penceresinde ileri yönde gönderilen bayt sayısı
68-Init Bwd Win Byts	Başlangıç penceresinde geriye doğru gönderilen bayt sayısı
69-Fwd Act Data Pkts	İleri yönde en az 1 baytlık TCP veri yüküne sahip paketlerin sayısı
70-Fwd Seg Size Min	İleri yönde gözlenen minimum segment boyutu
71-Active Mean	Bir akışın boşta kalmadan önce aktif olduğu ortalama süre
72-Active Std	Bir akışın boşta kalmadan önce aktif olduğu standart sapma süresi
73-Active Max	Bir akışın boşta kalmadan önce etkin olduğu maksimum süre
74-Active Min	Bir akışın boşta kalmadan önce aktif olduğu minimum süre
75-Idle Mean	Bir akışın aktif olmadan önce boşta kaldığı ortalama süre
76-Idle Std	Bir akışın aktif hale gelmeden önce boşta kaldığı standart sapma süresi
77-Idle Max	Bir akışın aktif olmadan önce boşta kaldığı maksimum süre
78-Idle Min	Bir akışın aktif hale gelmeden önce boşta kaldığı minimum süre
79-Label	Etiket