MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE BOTNET SALDIRI TESPİTİ: BOYUT İNDİRGEME YAKLAŞIMLARI

YÜKSEK LİSANS TEZİ

Ayşegül SAĞLAM GÜLBAĞÇA

Danışman
Doç. Dr. Gür Emre GÜRAKSIN

BİLGİSAYAR ANABİLİM DALI

Ocak 2025

AFYON KOCATEPE ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE BOTNET SALDIRI TESPİTİ: BOYUT İNDİRGEME YAKLAŞIMLARI

Ayşegül SAĞLAM GÜLBAĞÇA

Danışman
Doç. Dr. Gür Emre GÜRAKSIN

BİLGİSAYAR ANABİLİM DALI

Ocak 2025

TEZ ONAY SAYFASI

Ayşegül SAĞLAM GÜLBAĞÇA tarafından hazırlanan "Makine Öğrenmesi Yöntemleri İle Botnet Saldırı Tespiti: Boyut İndirgeme Yaklaşımları" adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 21 / 01 / 2025 tarihinde aşağıdaki jüri tarafından **oybirliği** ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Anabilim Dalı'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman **:** Doç. Dr. Gür Emre GÜRAKSIN

İmza

Başkan **:** Doç. Dr. Mevlüt ERSOY
Süleyman Demirel Üniversitesi, Mühendislik Fakültesi ……...

Üye **:** Doç. Dr. Gür Emre GÜRAKSIN
Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi ……...

Üye **:** Dr. Öğr. Üyesi Kerem GENCER
Afyon Kocatepe Üniversitesi, Mühendislik Fakültesi ……...

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

…/…/2025 tarih ve

…….sayılı kararıyla onaylanmıştır.

…….

Prof. Dr. Bekir YALÇIN

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİM SAYFASI

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

21 / 01 / 2025

Ayşegül SAĞLAM GÜLBAĞÇA

ÖZET

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE BOTNET SALDIRI TESPİTİ: BOYUT İNDİRGEME YAKLAŞIMLARI

Ayşegül SAĞLAM GÜLBAĞÇA
Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Anabilim Dalı
Danışman: Doç. Dr. Gür Emre GÜRAKSIN

Bu çalışmada, siber güvenlikte önemli bir tehdit oluşturan botnet saldırılarının tespiti için bir saldırı tespit sistemi geliştirilmiştir. CSE-CIC-IDS2018 veri seti üzerinde gerçekleştirilen çalışmada, makine öğrenmesi algoritmalarından k-en yakın komşu, lojistik regresyon, karar ağacı ve rastgele orman modelleri kullanılarak anomali tabanlı bir sınıflandırma yöntemi oluşturulmuştur. Veri setindeki yüksek boyutlu özelliklerin işlenmesi için temel bileşen analizi, korelasyon tabanlı ve karşılıklı bilgi boyut indirgeme yöntemleri uygulanmış; ayrıca bu yöntemlerin güçlü yönlerini birleştiren hibrit yöntemler geliştirilmiştir.

Boyut indirgeme yöntemleriyle, veri setindeki gereksiz özellikler elimine edilerek sınıflandırma süresi optimize edilmiş ve model performansında iyileştirme sağlanmıştır. Hibrit yöntemler, temel bileşen analizi, korelasyon tabanlı ve karşılıklı bilgi yöntemlerinin avantajlarını bir araya getirerek hem işlem süresini azaltmış hem de doğruluk ve f1 skoru açısından yüksek performans göstermiştir. Bu yöntem, karmaşık veri yapılarında önemli özellikleri etkin bir şekilde seçerken modelin eğitim süresini kısaltarak gerçek zamanlı saldırı tespit sistemleri için uygun bir çözüm sunmaktadır.

Çalışmada elde edilen bulgular, hibrit yöntemlerin diğer boyut indirgeme tekniklerine kıyasla yüksek performansı korurken işlem süresini önemli ölçüde azalttığını göstermiştir. Elde edilen sonuçlar, doğruluk, kesinlik, duyarlılık, f1 skoru ve eğitim

süresi açısından kapsamlı bir değerlendirme sunmuş, özellikle hibrit yöntemler, hız ve doğruluk oranını dengede tutarak siber güvenlik alanında etkili bir araç olduğunu kanıtlamıştır. Bu çalışma, doğruluk oranını koruyarak sınıflandırma süresini azaltmayı amaçlayan bir yaklaşım geliştirmiş ve hibrit yöntemlerin saldırı tespit sistemlerinin performansını artırmada etkili bir çözüm olduğunu ortaya koymuştur. Çalışma, siber güvenlikte hızlı, verimli ve yüksek doğruluk oranına sahip saldırı tespit sistemlerinin geliştirilmesine önemli katkılar sağlamaktadır.

2025, xii + 68 sayfa

Anahtar Kelimeler: Saldırı tespit sistemi, Makine öğrenmesi, Boyut indirgeme, Hibrit yöntem.

ABSTRACT

M.Sc. Thesis

BOTNET ATTACK DETECTION WITH MACHINE LEARNING METHODS: DIMENSION REDUCTION APPROACHES

Ayşegül SAĞLAM GÜLBAĞÇA

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Computer

Supervisor: Assoc. Prof. Gür Emre GÜRAKSIN

In this study, an attack detection system was developed to detect botnet attacks, a significant threat in cybersecurity. The study, conducted on the CSE-CIC-IDS2018 dataset, utilized machine learning algorithms such as k-nearest neighbors, logistic regression, decision tree, and random forest models to create an anomaly-based classification method. Principal component analysis, correlation-based, and mutual information dimensionality reduction methods were applied to process the high-dimensional features in the dataset, and hybrid methods combining the strengths of these techniques were developed.

By using dimensionality reduction methods, unnecessary features in the dataset were eliminated, optimizing the classification time and improving model performance. Hybrid methods, combining the advantages of principal component analysis, correlation-based, and mutual information techniques, reduced both the processing time and provided high performance in terms of accuracy and F1 score. This approach effectively selects important features in complex data structures while shortening the model training time, making it suitable for real-time attack detection systems.

The findings of the study show that hybrid methods significantly reduced processing time while maintaining high performance compared to other dimensionality reduction techniques. The results offered a comprehensive evaluation in terms of accuracy,

precision, sensitivity, F1 score, and training time. In particular, hybrid methods proved to be an effective tool in cybersecurity by balancing speed and accuracy. This study developed an approach aimed at reducing classification time while maintaining accuracy and demonstrated that hybrid methods are an effective solution to improve the performance of attack detection systems. The study contributes significantly to the development of fast, efficient, and high-accuracy attack detection systems in cybersecurity.

2025, xii + 68 pages

Keywords: Intrusion detection system, Machine learning, Dimensionality reduction, Hybrid method.

TEŞEKKÜR

Yüksek Lisans yolculuğumda deneyimleri ve bilgi birikimleriyle bana her zaman destek olan, yönlendirmeleriyle akademik hayatıma değerli katkılar sunan tez danışmanım Sayın Doç.Dr. Gür Emre GÜRAKSIN'a;

Hayatım boyunca ellerimi hiç bırakmayan, bana iyiliği ve vazgeçmemeyi öğreten canım annem Fadime ve canım babam Ümmet SAĞLAM'a;

Bana her daim sonsuz güvenip sonsuz cesaretlendiren canım kardeşim Gonca Gül SAĞLAM'a;

Her zaman olduğu gibi yüksek lisans yolculuğumun da her anında gösterdiği sabır ve fedakarlıklarıyla yanımda olan ve benimle birlikte aşk ile koşan sevgili eşim Ferhat'a;

Ve tek bir gülüşüyle bütün zorlukları kolaylaştıran, içimde çiçekler açtıran güzeller güzeli kızım Miray'a; sonsuz minnet ve teşekkürlerimi sunuyorum.

Ayşegül SAĞLAM GÜLBAĞÇA
AFYONKARAHİSAR, 2025

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET i

ABSTRACT iii

TEŞEKKÜR v

İÇİNDEKİLER DİZİNİ vi

SİMGELER ve KISALTMALAR ix

ŞEKİLLER DİZİNİ x

ÇİZELGELER DİZİNİ xii

1. GİRİŞ 1

2. GENEL BİLGİLER ve LİTERATÜR İNCELEMESİ 5

2.1 Makine Öğrenmesi 5

2.2 Makine Öğrenmesi Türleri 6

2.2.1 Denetimli Öğrenme 6

2.2.2 Denetimsiz Öğrenme 7

2.2.3 Yarı Denetimli Öğrenme 8

2.2.4 Pekiştirmeli Öğrenme 9

2.3 Sınıflandırma Algoritmaları 9

2.3.1 Lojistik Regresyon 9

2.3.2 K-En Yakın Komşu 10

2.3.3 Karar Ağaçları 11

2.3.4 Rastgele Orman 12

2.4 Model Performansının Değerlendirilmesi 13

2.4.1 Karmaşıklık Matrisi 13

2.4.2 Doğruluk (Accuracy) 13

2.4.3 Kesinlik (Precision) 14
2.4.4 Duyarlılık (Recall) 14
2.4.5 F1 Skor 14
2.5 Boyut İndirgeme 14
2.5.1 Özellik Çıkarımı 15
2.5.1.1 Temel Bileşen Analizi 15
2.5.2 Özellik Seçimi 15
2.5.2.1 Korelasyon Tabanlı Özellik Seçimi 16
2.5.2.2 Karşılıklı Bilgi Özellik Seçimi 16
2.6 Literatür İncelemesi 16
3. MATERYAL ve METOT 21
3.1 Çalışma Ortamının Hazırlanması 21
3.2 Veri Setinin Tanımlanması 22
3.3 Veriye İlk Bakış 23
3.4 Veri Ön İşleme 27
3.4.1 Veri Temizleme 28
3.4.2 Kategorik Değerlerin Kodlanması 29
3.4.3 Veri Ölçeklendirme 30
3.4.4 Boyut İndirgeme 31
3.4.4.1 Temel Bileşen Analizi 31
3.4.4.2 Korelasyon Tabanlı Özellik Seçimi 32
3.4.4.3 Karşılıklı Bilgi Özellik Seçimi 36
3.5 Hibrit Yöntemler 38
3.5.1 Hibrit1 38
3.5.2 Hibrit2 40

3.6 Hedef Değişken ve Bağımsız Değişkenin Belirlenmesi 44
4. BULGULAR 45
4.1 Makine Öğrenmesi Algoritmalarının Performans Karşılaştırması 46
4.1.1 Lojistik Regresyon 46
4.1.2 K En Yakın Komşu Algoritması 48
4.1.3 Karar Ağacı 51
4.1.4 Rastgele Orman 54
5. TARTIŞMA ve SONUÇ 57
6. KAYNAKLAR 61
ÖZGEÇMİŞ 68

SİMGELER ve KISALTMALAR

Simgeler

sn	Saniye

Kısaltmalar

ANN	Yapay Sinir Ağları (Artificial Neural Network)
CB	Korelasyon Tabanlı (Correlation Based)
DT	Karar Ağacı (Decision Tree)
FN	Yanlış Negatif (False Negative)
FP	Yanlış Pozitif (False Positive)
KNN	K-En Yakın Komşu (K-Nearest Neighbor)
LR	Lojistik Regresyon (Logistic Regression)
MI	Karşılıklı Bilgi (Mutual Information)
NB	Navie Bayes
PCA	Temel Bileşen Analizi (Principal Component Analysis)
RF	Rastgele Orman (Random Forest)
SMOTE	Sentetik Azınlık Aşırı Örnekleme Tekniği (Synthetic Minority Oversampling Technique)
STS	Saldırı Tespit Sistemi
SVM	Destek Vektör Makinesi (Support Vector Machine)
TN	Doğru Negatif (True Negative)
TP	Doğru Pozitif (True Positive)

ŞEKİLLER DİZİNİ

Sayfa

Şekil 2.1 Denetimli öğrenme süreci 6

Şekil 2.2 Denetimli öğrenme yöntemleri 7

Şekil 2.3 Denetimsiz öğrenme süreci 7

Şekil 2.4 Denetimsiz öğrenme yöntemleri 8

Şekil 2.5 Yarı denetimli öğrenme süreci 8

Şekil 2.6 Pekiştirmeli öğrenme süreci 9

Şekil 2.7 K-en yakın komşu algoritması çalışma mantığı (Kalaycı 2018). 10

Şekil 2.8 Karar ağacı yapısı 11

Şekil 2.9 Rastgele orman algoritması örneği (Zengin 2024) 12

Şekil 3.1 Label değerine göre veri sayıları dağılımı 27

Şekil 3.2 Veri ön işleme adımları 27

Şekil 3.3 Veri setine uygulanan veri ön işlem adımları 28

Şekil 3.4 Veri seti içerisindeki tekrar eden ve benzersiz gözlem sayıları dağılımı 29

Şekil 3.5 Açıklanan varyans değerleri 31

Şekil 3.6 Bileşen sayısına göre açıklanan varyans değerleri grafiği 31

Şekil 3.7 Isı haritası 33

Şekil 3.8 Label ile diğer değişkenler arası korelasyon 34

Şekil 3.9 0,2'den büyük korelasyona sahip değişkenlerin listesi 35

Şekil 3.10 Karşılıklı bilgi katsayılarına göre sıralanmış özellikler 36

Şekil 3.11 0,42'den büyük karşılıklı bilgi değerine sahip değişkenlerin listesi 37

Şekil 3.12 "Korelasyon tabanlı + karşılıklı bilgi" yöntemleri ile seçilen özelliklerin birleşimi 38

Şekil 3.13 "Korelasyon tabanlı + karşılıklı bilgi" birleşimi sonucu oluşan özelliklerin listesi 39

Şekil 3.14 Açıklanan varyans değerleri 39

Şekil 3.15 Bileşen sayısına göre açıklanan varyans değerleri grafiği 40

Şekil 3.16 Isı haritası 41

Şekil 3.17 Label ile diğer değişkenler arası korelasyon 41

Şekil 3.18 0,2'den büyük korelasyona sahip değişkenlerin listesi 42

Şekil 3.19 Karşılıklı bilgi katsayılarına göre sıralanmış özellikler 42

Şekil 3.20 0,45'den büyük karşılıklı bilgi katsayısına sahip değişkenlerin listesi 43

Şekil 3.21 "Korelasyon tabanlı + karşılıklı bilgi" yöntemleri ile seçilen özelliklerin birleşimi 43

Şekil 3.22 "Korelasyon tabanlı + karşılıklı bilgi" birleşimi sonucu oluşan özelliklerin listesi 44

Şekil 4.1 Veri setlerinin eğitim süreci 45

Şekil 4.2 Model başarısı ve eğitim süresi karşılaştırması 48

Şekil 4.3 Model başarısı ve eğitim süresi karşılaştırması 51

Şekil 4.4 Model başarısı ve eğitim süresi karşılaştırması 53

Şekil 4.5 Model başarısı ve eğitim süresi karşılaştırması 56

Şekil 5.1 Genel sonuçlar 57

ÇİZELGELER DİZİNİ

Sayfa

Çizelge 2.1 Karmaşıklık matrisi 13

Çizelge 3.1 CIC-IDS2018 veri setinden elde edilen bilgiler 22

Çizelge 3.2 Veri setine ait öznitelikler ve açıklamaları 23

Çizelge 3.2 (Devam) Veri setine ait öznitelikler ve açıklamaları 24

Çizelge 3.2 (Devam) Veri setine ait öznitelikler ve açıklamaları 25

Çizelge 3.2 (Devam) Veri setine ait öznitelikler ve açıklamaları 26

Çizelge 3.3 Label değerine göre veri sayıları 26

Çizelge 3.4 Temizlik sonrası Label değerine göre veri sayıları 29

Çizelge 3.5 Tekrar eden kayıtlar silindikten sonra Label değerine göre veri sayıları 29

Çizelge 3.6 Label değerine göre veri sayıları 30

Çizelge 4.1 Hiper parametre optimizasyon işlemi konfigürasyonu ve sonuçları 46

Çizelge 4.2 LR modeli sonuçları 47

Çizelge 4.3 Hiper parametre optimizasyon işlemi konfigürasyonu ve sonuçları 49

Çizelge 4.4 KNN modeli sonuçları 49

Çizelge 4.5 Hiper parametre optimizasyon işlemin konfigürasyonu ve sonuçları 52

Çizelge 4.6 DT modeli sonuçları 52

Çizelge 4.7 Hiper parametre optimizasyon işlemin konfigürasyonu ve sonuçları 54

Çizelge 4.8 RF modeli sonuçları 55

Çizelge 5.1 Literatür karşılaştırması 58

Çizelge 5.1 (Devam) Literatür karşılaştırması 59

1. GİRİŞ

Güvenlik, insanlık tarihinin en temel ihtiyaçlarından biri olarak, bireylerin, toplumların ve devletlerin varlığını koruyabilmesi için vazgeçilmez bir öneme sahiptir. Tarih boyunca dönemin koşullarına ve ihtiyaçlarına göre yeniden şekillenen güvenlik anlayışı, günümüzde bilişim teknolojilerinin hızla gelişmesi ve bu teknolojilerin günlük yaşamın ayrılmaz bir parçası haline gelmesiyle birlikte köklü bir dönüşüm geçirmiştir. Teknolojinin ilerlemesi ve internetin hızla yaygınlaşmasıyla birlikte güvenlik, yalnızca fiziksel alanları değil, siber uzayı kapsayan geniş bir anlam kazanmıştır.

Teknolojinin her geçen gün gelişmesi, birçok fayda sağlamakla birlikte, çeşitli saldırı türlerini de beraberinde getirmektedir. Siber saldırılar çok farklı yöntemlerle yapılabilmekte ve gün geçtikçe yeni saldırı türleri ortaya çıkmaktadır. Saldırı türlerinin en önemlilerinden biri de botnet saldırılarıdır.

Botnet saldırısı, bir kişi ya da bir grup saldırgan tarafından yönlendirilen ve internet aracılığıyla birbiriyle bağlantılı cihazların kullanılmasıyla gerçekleştirilen bir saldırı türüdür (Kılınç ve Eyüpoğlu 2023). Botnet saldırıları, internete bağlı cihazlar arasında yayılan çok ciddi saldırılardır. Bu tür saldırılar, zararlı medya dosyalarının çalıştırılması, spam e-posta eklerinin indirilmesi veya tehlikeli web sitelerinin ziyaret edilmesiyle enfekte olan cihazların, saldırganlar tarafından kontrol edilerek çeşitli kötü amaçlı faaliyetlerde kullanılmasına yol açar (Kılınç ve Eyüpoğlu 2023). Bu saldırılardan korunmak için siber güvenliğe yönelik önlem alınması her geçen gün daha önemli ve gerekli hale gelmektedir.

Siber saldırılarla mücadelede çözüm olarak sunulan saldırı tespit sistemleri (STS), ağ trafiğini analiz ederek, ağdaki kötü niyetli faaliyetleri tespit etmek için kullanılan bir siber güvenlik teknolojisidir (Gürmen 2020). STS kullandığı tespit yöntemlerine göre imza tabanlı ve anomali tabanlı STS olmak üzere ikiye ayrılmaktadır (Kaynar vd. 2018). İmza tabanlı STS'ler, daha önceden kaydedilmiş saldırıların özelliklerine dayalı kurallar kullanarak bilinen saldırı türlerini tespit etmek için tasarlanmıştır (Emekli 2024). Anomali tabanlı STS'ler, sistemi izleyerek tanımlanmış normal davranış modeline

dayalı olarak sistemdeki anormal aktiviteleri tespit eder (Serdaroğlu 2021). İmza tabanlı yöntemler, daha önce tanımlanmamış saldırı türlerine karşı etkisiz kalırken, anomali tabanlı yöntemler, yeni saldırı türlerini tespit etme ve sınıflandırmada etkili bir çözüm sunar (Ölmez ve İnce 2022).

Anomali tabanlı STS'lerde, yeni saldırı türlerini tespit etme ve sınıflandırma sürecinde makine öğrenimi teknikleri kullanılmaktadır (Gürmen 2020). Siber saldırıları tespit etmek için kullanılan ağ verileri, genellikle büyük boyutlu verilerdir ve bu verilerin makine öğrenmesi algoritmalarıyla işlenmesi, anomali tespiti sürecini zorlaştırmaktadır (Emhan ve Akın 2019).Veri setleri, birçok öznitelik içerir ve sınıflandırma işlemi için her bir öznitelik gerekli değildir. Gereksiz öznitelikler, işlem yükünü artırırken tespit doğruluğunu da olumsuz etkiler (Emhan ve Akın 2019). Bu bağlamda, ağ verilerinin verimli bir şekilde işlenebilmesi için boyut indirgeme önemli bir adımdır. Gereksiz özniteliklerin elimine edilmesi, yalnızca işlemin hızını artırmakla kalmaz, aynı zamanda daha doğru ve etkili bir anomali tespiti sağlanmasına da katkı sunar. Bu süreç, makine öğrenmesi algoritmalarının daha verimli çalışmasını destekler, çünkü doğru öznitelikler üzerinde odaklanmak, modelin öğrenme sürecini hızlandırır ve gereksiz veri gürültüsünü ortadan kaldırır. Ayrıca, öznitelik seçimiyle ilgili doğru stratejiler, hem işlem kaynaklarını daha verimli kullanmayı sağlar hem de siber saldırıların daha hızlı bir şekilde tespit edilmesini mümkün kılar.

Siber güvenlikte artan veri boyutları ve karmaşıklık, STS'lerde hızlı ve doğru analiz yöntemlerine olan ihtiyacı artırmıştır. Bu çalışmada, botnet saldırının tespiti için makine öğrenmesi algoritmalarından k-en yakın komşu (KNN), lojistik regresyon (LR), karar ağacı (DT) ve rastgele orman (RF) algoritmaları kullanılarak anomali tabanlı STS geliştirilmesi hedeflenmiştir. CSE-CIC-IDS2018 veri setindeki yüksek boyutlu özellikler, temel bileşen analizi (PCA), korelasyon tabanlı (CB) ve karşılıklı bilgi (MI) boyut indirgeme yöntemleriyle azaltılmış ve bu yöntemlerin saldırı tespitindeki etkileri detaylı bir şekilde analiz edilmiştir. Ayrıca, PCA, MI ve CB yöntemlerinin hibrit şekilde birleştirildiği iki farklı boyut indirgeme yöntemi geliştirilmiştir. Hibrit yöntemler, her bir boyut indirgeme metodunun güçlü yönlerini birleştirerek, daha verimli ve doğru bir özellik seti elde etmeyi amaçlamaktadır. Hibrit yöntemlerin

performansı, diğer tekli yöntemlerle karşılaştırılmış ve daha yüksek doğruluk oranları ile daha hızlı işlem süreleri elde edilmiştir. Çalışma, başarı oranlarının yanı sıra sınıflandırma sürelerine de odaklanarak, gerçek dünyada verimli bir sınıflandırma süresiyle yüksek başarı oranlarına ulaşmayı amaçlamaktadır. Bu çalışma, STS'lerde yüksek doğruluk oranını koruyarak işlem süresini ve maliyetini azaltmayı hedefleyen etkili bir yaklaşım sunmaktadır.

Literatürde boyut indirgeme yöntemlerinin çeşitli varyasyonları uygulanmış olsa da, bu yöntemlerin boyut indirgeme işlemine tabi tutulmamış veri setleri üzerinde yaratacağı etkilere yeterince yer verilmemiştir. Ayrıca, bu yöntemlerin sınıflandırma süreleri üzerinde nasıl bir etkisi olduğu da genellikle göz ardı edilmiştir. Ancak gerçek dünyada işletilebilir bir sistem kurmak için modelin doğruluğu kadar işlem hızı da kritik öneme sahiptir. Bu çalışmada, CSE-CIC-IDS2018 veri seti üzerinde PCA, CB ve MI boyut indirgeme yöntemleri ve geliştirdiğimiz hibrit yöntemler kullanılarak yapılan ön işlemler detaylı bir şekilde ele alınmıştır. Kullanılan boyut indirgeme yöntemlerinin yanı sıra, boyut indirgeme yöntemi uygulanmayan veri setiyle elde edilen sonuçlar kapsamlı bir şekilde karşılaştırılmıştır. Bu karşılaştırma, boyut indirgeme yöntemi uygulanmayan veri setlerinin sınıflandırma performansı ve süresi üzerinde nasıl bir etkisi olduğunu ortaya koymuştur. Elde edilen sonuçlar, boyut indirgeme yöntemlerinin yalnızca doğruluk oranını değil, aynı zamanda sınıflandırma sürelerini de önemli ölçüde etkileyebileceğini ve gerçek dünya uygulamaları için modelin işlem hızının da kritik bir faktör olduğunu göstermektedir.

Hibrit yöntemler, PCA, CB ve MI gibi klasik boyut indirgeme yöntemlerinin güçlü yönlerini birleştirerek hem doğruluk oranını yüksek tutmayı hem de işlem süresini optimize etmeyi hedeflemiştir. Bu yöntemler, özellikle karmaşık veri yapılarında önemli özellikleri etkin bir şekilde seçerken gereksiz verileri eleyerek modelin eğitim süresini kısaltmış ve gerçek zamanlı sistemler için daha uygun bir yapı sunmuştur. Çalışmada elde edilen bulgular, hibrit yöntemlerin diğer boyut indirgeme tekniklerine kıyasla doğruluk ve f1 skoru açısından yüksek performans sergilediğini ve işlem süresini önemli ölçüde azalttığını ortaya koymuştur. Hibrit yöntemlerden özellikle "Hibrit2" yöntemi, hem doğruluk oranı hem de işlem süresi açısından dengeli bir performans

sergileyerek gerçek zamanlı saldırı tespit sistemlerinin geliştirilmesinde etkili bir çözüm sunduğunu ortaya koymaktadır. Bu çalışma, siber güvenlikte verimli, hızlı ve yüksek doğruluğa sahip saldırı tespit sistemlerinin tasarımına önemli katkılar sağlamaktadır.

2. GENEL BİLGİLER ve LİTERATÜR İNCELEMESİ

2.1 Makine Öğrenmesi

Makine öğrenmesi, bilgisayar programlarının deneyimlerden öğrenerek insan müdahalesi olmaksızın sınıflandırma, ses ve görüntü tanıma, doğal dil işleme, siber saldırı tespiti gibi çeşitli görevlerin gerçekleştirebilmesini sağlar (Aydın 2023). Bu görevler, algoritmaların veri üzerinden model inşa ederek tahmin ve karar mekanizmaları oluşturması sayesinde mümkün hale gelir ve böylece statik talimatlar yerine dinamik ve adaptif bir yaklaşım benimsenir.

Makine öğrenmesi geçmişteki verilerden öğrenerek büyük veri tabanlarından düzenlilikler çıkarır ve güvenilir, tekrarlanabilir kararlar üretir (Janiesch vd. 2021). İnsan müdahalesi olmaksızın sürekli iyileşen bir öğrenme süreci sunar.

Makine öğrenmesi yöntemleri, büyük veri setleri üzerinde çalışırken karmaşık ilişkileri analiz etme ve isabetli tahminler yapma yeteneği ile öne çıkar. Özellikle geleneksel yöntemlerin yetersiz kaldığı doğrusal olmayan desenlerin modellenmesinde, makine öğrenmesi yüksek esneklik ve güçlü genelleştirme kapasitesi sunar (Jordan ve Mitchell 2015). Bunun yanı sıra, artan veri miktarı ve çeşitliliğiyle birlikte performansını sürekli geliştirebilme özelliği, farklı alanlarda etkili ve uyarlanabilir çözümler üretilmesini sağlamaktadır (LeCun vd. 2015). Bu avantajlar, tezimin hedeflerine ulaşmada makine öğrenmesini ideal bir yöntem haline getirmiştir.

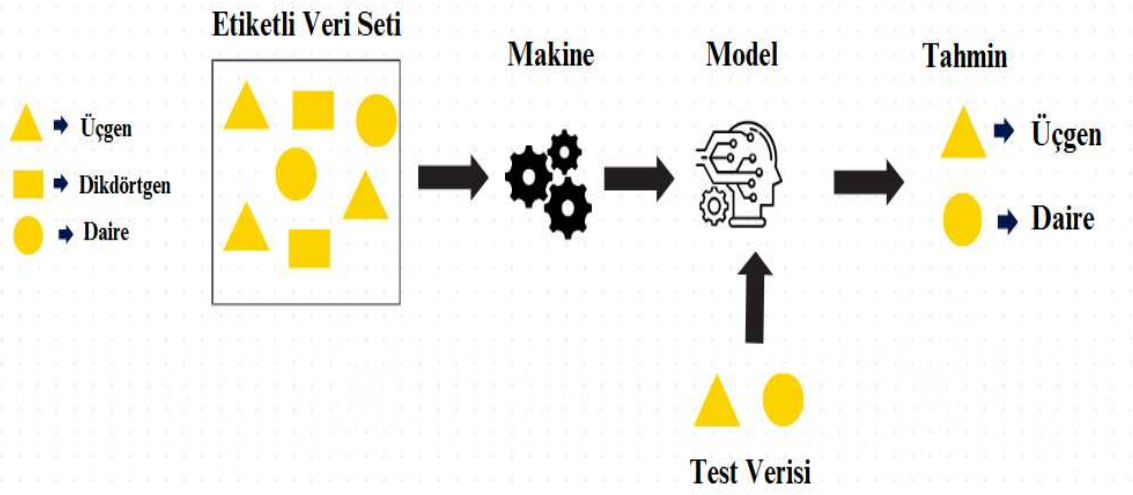
Makine öğrenmesi sistemleri eğitim sırasında aldıkları denetim miktarına ve türüne göre dört ana kategoriye ayrılır (Geron 2021). Bunlar :

- Denetimli öğrenme,
- Denetimsiz öğrenme,
- Yarı denetimli öğrenme,
- Pekiştirmeli öğrenme.

2.2 Makine Öğrenmesi Türleri

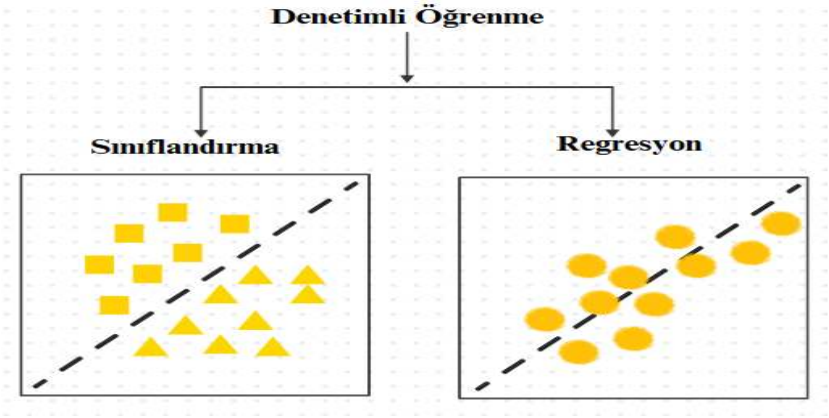
2.2.1 Denetimli Öğrenme

Bir veri setinde girdi değerlerine karşılık gelen bir çıktı değeri varsa, bu veri seti etiketli, yoksa etiketsiz olarak adlandırılmaktadır (Mirtagioğlu ve Demir 2022). Denetimli öğrenme, etiketli veri setlerinin kullanımıyla tanımlanan bir yöntemdir; bu yöntem, girdilere karşılık gelen çıktı değerlerini üreten bir öğrenme modeli oluşturmayı amaçlar (Bayram Arlı vd. 2022). Denetimli öğrenme süreci şekil 2.1'de gösterilmiştir.



Şekil 2.1 Denetimli öğrenme süreci

Denetimli öğrenme yöntemleri şekil 2.2'de gösterildiği gibi sınıflandırma ve regresyon problemlerinde kullanılır. Sınıflandırma, kategoriler halinde hedef değişkenlerin olduğu veri setinde verilerin giriş özelliklerine göre kategorik sınıfın tahmin edilmesiyken; regresyon, hedef değişkenin sürekli değerlerden oluştuğu verilerin giriş özelliklerine göre bu sürekli değişkenlerin tahmin edilmesidir (Çelik ve Altunaydın 2018).

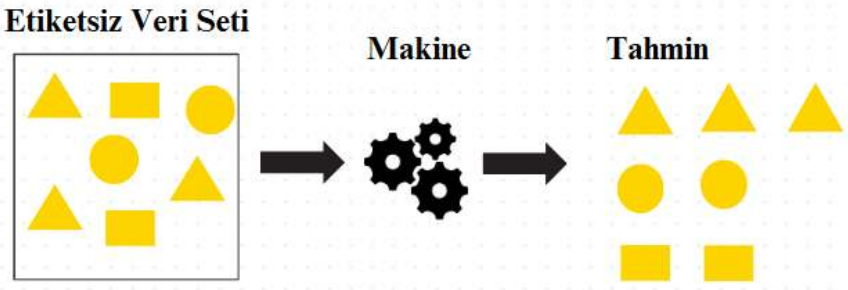


Şekil 2.2 Denetimli öğrenme yöntemleri

Tez kapsamında kullandığımız veri seti etiketli verilerden oluşmaktadır ve hedef değişken kategorik değişkendir. Bu yüzden modellenecek olan makine öğrenmesi algoritmaları denetimli öğrenme algoritmalarıdır.

2.2.2 Denetimsiz Öğrenme

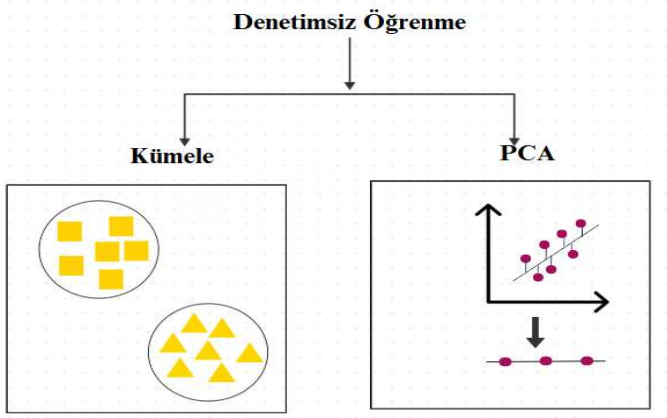
Denetimsiz öğrenmede veriler etiketsizdir ve sistem bir öğreten olmadan öğrenmeye çalışır (Geron 2021). Denetimsiz öğrenme, etiketsiz veriler arasındaki ilişkiyi bulmaya çalışan bir makine öğrenmesi yöntemidir (Çelik 2023). Amaç, veriler arasındaki ortak noktaları belirleyerek gruplama yapmaktır. Denetimsiz öğrenme süreci şekil 2.3'te gösterilmiştir.



Şekil 2.3 Denetimsiz öğrenme süreci

Kümeleme ve ilişkilendirme, iki temel denetimsiz öğrenme yöntemidir; kümeleme, benzer veri gruplarını oluştururken, ilişkilendirme ise veriler arasındaki bağlantıları ortaya çıkarmaktadır (Çelik ve Altunaydın 2018). Ayrıca denetimsiz öğrenme

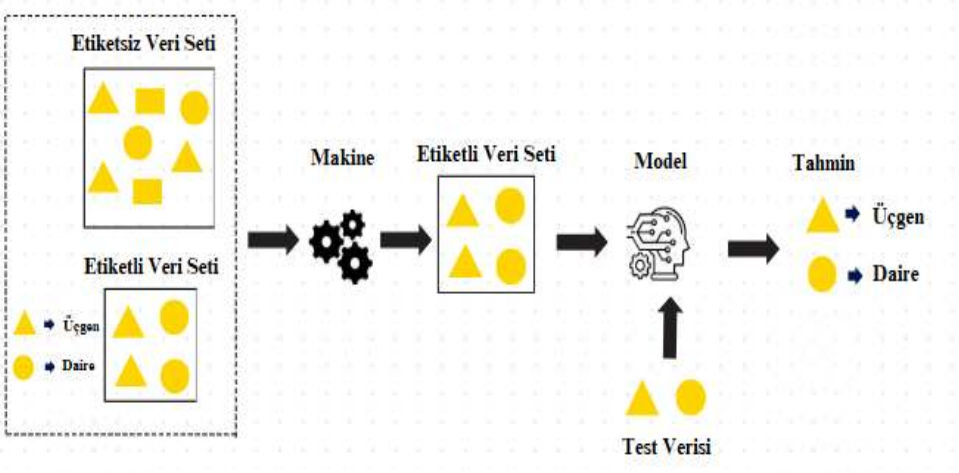
yöntemleri, veri setinin boyutunu azaltmak için de kullanılmakta olup, bu bağlamda PCA yöntemi örnek teşkil etmektedir. Denetimsiz öğrenme yöntemleri şekil 2.4'te gösterilmiştir.



Şekil 2.4 Denetimsiz öğrenme yöntemleri

2.2.3 Yarı Denetimli Öğrenme

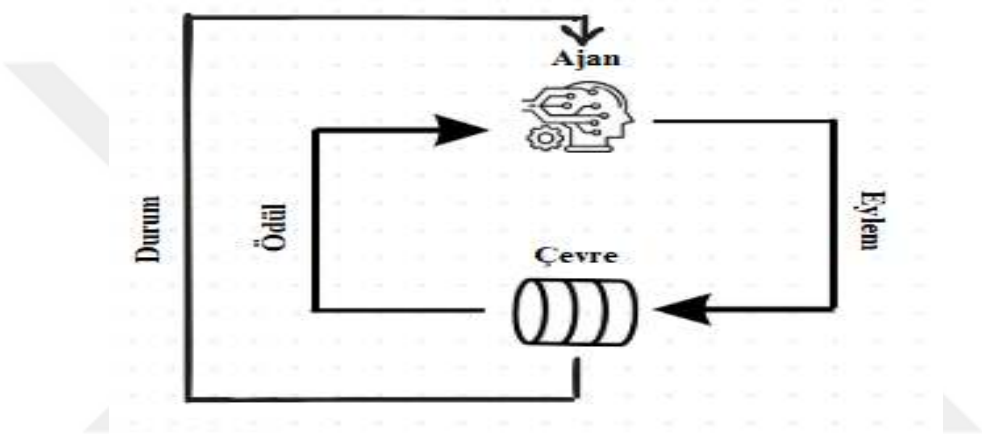
Yarı denetimli öğrenme, hem denetimli hem de denetimsiz öğrenme yaklaşımlarını bir araya getirir. Veri etiketleme genellikle zaman alıcı ve maliyetli olduğundan, genellikle çok sayıda etiketlenmemiş veri olduğunda ve bu etiketlenmemiş veriler hakkında bilgi sahibi olmaya çalışıldığı zaman tercih edilir (Geron 2021). Yarı denetimli öğrenme süreci şekil 2.5'te gösterilmiştir.



Şekil 2.5 Yarı denetimli öğrenme süreci

2.2.4 Pekiştirmeli Öğrenme

Pekiştirmeli öğrenmede, çevresiyle etkileşimleri sonucunda oluşan duruma dayanan bir öğrenme süreci söz konusudur. Pekiştirmeli öğrenmede; çevre, ajan, eylem, durum ve ödül gibi kendine özgü kavramlar mevcuttur (Bayram Arlı vd. 2022). Ajan olarak adlandırılan öğrenme sistemi, çevreyi gözlemleyebilir, eylemleri seçip uygulayabilir ve bunların sonucunda ödüller kazanabilir (Geron 2021). Pekiştirmeli öğrenme süreci şekil 2.6'da gösterilmiştir.



Şekil 2.6 Pekiştirmeli öğrenme süreci

Pekiştirmeli öğrenme, deneme yanılma mantığıyla çalışır (Saygın 2021). Amaç, her eylemin sonucunda kazanılacak ödül puanına göre verilen görevi en yüksek ödül puanıyla tamamlamaktır ve bir dizi eylem arasından en etkili eylemi öğrenmeye çalışılmaktır (Bayram Arlı vd. 2022).

2.3 Sınıflandırma Algoritmaları

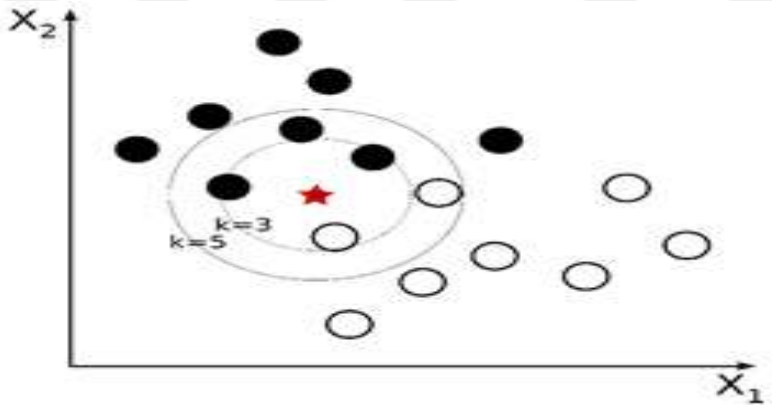
2.3.1 Lojistik Regresyon

Bağımlı değişkenin iki veya çok düzeyli kategorik değişken olduğu durumlarda özellikle 0 ve 1 gibi kesikli değerler içeren sonuçlar için kullanılan istatistiksel bir yöntemdir. (Gök ve Özdemir 2011). Örneğin, bir mailin spam olup olmadığı(1 veya 0) ya da bir siber saldırının olup olmadığı(1 veya 0) gibi durumlarda lojistik regresyon

tercih edilmektedir. Lojistik regresyonda amaç bağımlı ve bağımsız değişkenler arasındaki ilişkiyi tanımlayarak hedef değişkenin hangi kategoriye ait olduğunu bulmaktır. Sınıflandırma yapmak için sigmoid fonksiyonunu kullanır (Gülşen vd. 2015). Sigmoid fonksiyonu, çıktıyı 0 ile 1 arasında sınırlandırarak modelin tahmin edilen değerlerini genellikle 0.5 olan bir eşik değerine göre sınıflara ayırır; 0.5'ten büyükse 1, küçükse 0 olarak tahmin edilir (Arslan 2021).

2.3.2 K-En Yakın Komşu

K-en yakın komşu algoritması, sınıflandırma ve regresyon problemleri için kullanılabilen denetimli makine öğrenmesi algoritmasıdır (Karakaya 2024). Bir noktanın sınıfını tahmin etmek için, önceden etiketlenmiş veriler arasından en yakın k komşusu seçilir ve bu k komşu içinde en fazla bulunan sınıf, tahmin edilmek istenen noktanın sınıfı olarak belirlenir (Arslan 2021). K-en yakın komşu algoritmasının çalışma mantığı şekil 2.7'de gösterilmektedir.

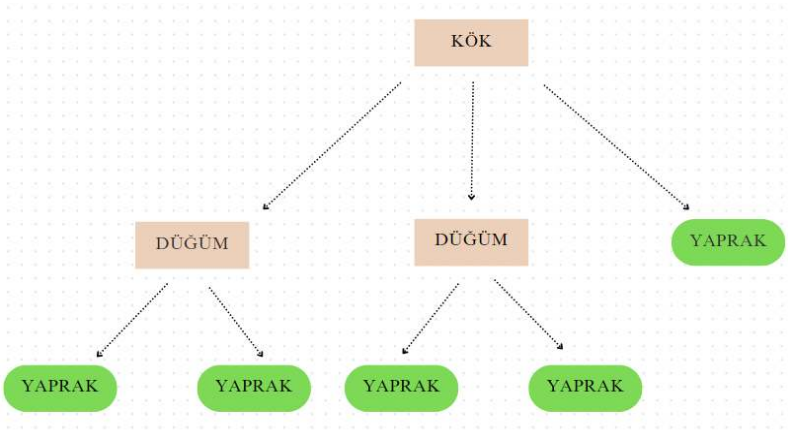


Şekil 2.7 K-en yakın komşu algoritması çalışma mantığı (Kalaycı 2018).

Şekil incelendiğinde, yıldız ile gösterilen veri noktasının hem en yakın 3 komşusu hem de en yakın 5 komşusu dikkate alındığında sınıfının siyah daire olduğu görülmektedir. Yakın komşu seçimi yapılırken noktaların uzaklıkları hesaplanmaktadır. Kullanılmakta olan üç farklı uzaklık ölçüsü bulunmaktadır. Bunlar öklid, manhattan ve minkowski uzaklık ölçüleridir.

2.3.3 Karar Ağaçları

Karar ağaçları sınıflama ve regresyon problemlerinde kullanılabilen denetimli makine öğrenmesi algoritmasıdır (Demiel 2019). Karar ağaçları, gerçek bir ağaç yapısını andıran kök, düğümler, dallar ve yapraklardan oluşmaktadır (Çelik vd. 2022). Karar ağacının yapısı şekil 2.8'de gösterilmiştir.



Şekil 2.8 Karar ağacı yapısı

Kök, ağaç yapısında bir düğüm olup, düğümler dallanmanın gerçekleştiği yerleri temsil ederken, dallanmayı ifade etmek için "dal" terimi kullanılır ve ağaç yapısının en son elemanları ise yapraklardır (Çelik vd. 2022).

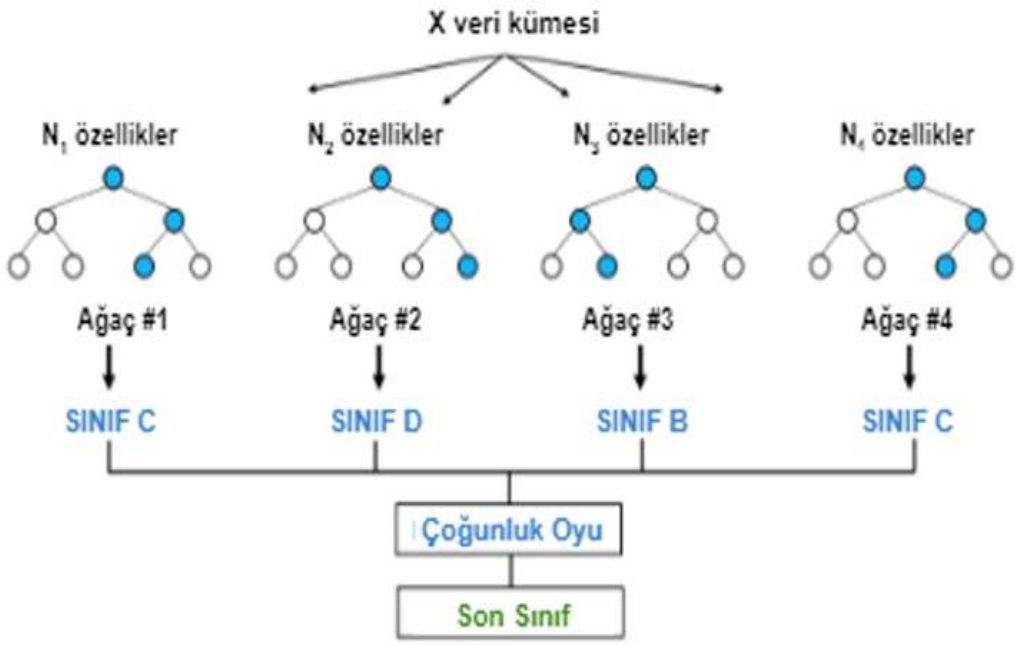
Karar ağaçları, eğitim verilerine dayalı özniteliklerle oluşturulup düğümlerde ilerleyerek örnekleri sınıflandırmak için kullanılır (Beşli ve Tenekeci 2020). Karar ağaçları, eğitim verilerine dayalı sorular sorarak ve elde edilen cevaplara göre en kısa sürede sonuca ulaşmayı hedefleyerek karar kuralları oluşturur (Köktürk 2012). Bu kurallar, kökten yaprağa doğru yazılır ve bir olayın sonuçlandırılmasında sorunun cevabına göre hareket edilir (Daş ve Türkoğlu 2014). Karar ağaçları oluşturulurken, dallanmalar için gerekli kriterlerin belirlenmesi en önemli adımdır ve bu kriterlerin belirlenmesindeki yöntemler; bilgi kazancı, bilgi kazancı oranı, Gini indeksi, Towing kuralı ve Ki-Kare istatistiğidir (Çelik vd. 2022).

Karar ağaçları, yorumlanmasının kolay olması, veri tabanı sistemleriyle entegrasyonunun basitliği ve güvenilirlikleri nedeniyle sınıflama modelleri arasında en yaygın kullanılan yöntemlerdir (Köktürk 2012).

2.3.4 Rastgele Orman

Bir grup tahmincinin tahminlerini birleştirirseniz genellikle en iyi bireysel tahmincinin tahmininden daha iyi bir tahmin alırsınız. Tahminci grubuna topluluk dendiğinde bu teknik topluluk öğrenme olarak adlandırılır (Geron 2021). Rastgele orman, sınıflandırma ve regresyon problemleri için topluluk öğrenme yöntemini temel alan denetimli makine öğrenmesi algoritmasıdır (Nacar ve Erdebilli 2021).

Bu yöntem, eğitim aşamasında birçok karar ağacı oluşturur ve her ağaç sınıf tahmini yapar, tahmin aşamasında bu karar ağaçlarının sonuçlarına göre girdinin sınıfını çoğunluk oyu ile belirler (Kalaycı 2018). RF yapısı şekil 2.9'da gösterilmiştir.



Şekil 2.9 Rastgele orman algoritması örneği (Zengin 2024).

2.4 Model Performansının Değerlendirilmesi

Botnet saldırılarının tespitine yönelik oluşturulan modellerin performansı, doğruluk, kesinlik, duyarlılık ve f1 skor metrikleri kullanılarak değerlendirilmiştir.

2.4.1 Karmaşıklık Matrisi

Karmaşıklık matrisi gerçek ve tahmin edilen sonuçları görselleştiren bir tablodur.

Çizelge 2.1 Karmaşıklık matrisi

		Tahmin Edilen Sınıf	
	Sınıf	**Pozitif**	**Negatif**
Gerçek Sınıf	Pozitif	Doğru Pozitif(TP)	Yanlış Negatif(FN)
	Negatif	Yanlış Pozitif(FP)	Doğru Negatif(TN)

TP: Pozitif sınıflandırılan ve modelin pozitif tahmin ettiği değerlerin sayısını ifade eder.

FP: Negatif sınıflandırılan ve modelin pozitif tahmin ettiği değerlerin sayısını ifade eder.

FN: Pozitif sınıflandırılan ve modelin negatif tahmin ettiği değerlerin sayısını ifade eder.

TN: Negatif sınıflandırılan ve modelin negatif tahmin ettiği değerlerin sayısını ifade eder.

2.4.2 Doğruluk (Accuracy)

Modelde doğru tahmin edilen alanların toplam veri kümesine oranı ile hesaplanmaktadır.

$$Doğruluk = \frac{TP + TN}{n} \tag{2.1}$$

2.4.3 Kesinlik (Precision)

Pozitif olarak tahmin ettiğimiz değerlerin gerçekten kaç tanesinin pozitif olduğunu göstermektedir. Doğru tahmin edilen örneklerin toplam pozitif tahmin edilen örneklere oranıdır.

$$Kesinlik = \frac{TP}{TP + FP} \tag{2.2}$$

2.4.4 Duyarlılık (Recall)

Pozitif olarak tahmin etmemiz gereken örneklerin ne kadarını pozitif olarak tahmin ettiğimizi göstermektedir. Doğru tahmin edilen örneklerin toplam pozitif örneklere oranıdır.

$$Duyarlılık = \frac{TP}{TP + FN} \tag{2.3}$$

2.4.5 F1 Skor

Kesinlik ve duyarlılık değerlerinin harmonik ortalamasını göstermektedir.

$$F1\ skor = 2 * \frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \tag{2.4}$$

2.5 Boyut İndirgeme

Boyut indirgeme, özellikle büyük ve karmaşık veri kümeleriyle çalışırken önemli bir veri ön işleme tekniğidir. Boyut indirgeme, veri setindeki özelliklerin yeniden yapılandırılması yoluyla veri kümelerindeki boyut sayısını azaltarak anlamlı özelliklere

dönüştürme işlemidir (Emeç ve Özcanhan 2023). Boyut indirgemenin amacı hesaplama süresini azaltmak, gereksiz özelliklerden kurtulmak ve verileri daha anlaşılır kılmaktır (Arslan 2021). Özellik seçimi ve özellik çıkarımı yöntemlerine göre gerçekleştirilir (Erbayram ve Erisoglu 2021).

2.5.1 Özellik Çıkarımı

Boyut indirgeme yöntemlerinden bir diğeri olan özellik çıkarımı, orijinal veri setindeki mevcut özelliklerden yeni, daha anlamlı ve genellikle daha düşük boyutlu özellikler türetme işlemidir. Bu süreç, verinin boyutunu azaltarak önemli bilgilerin korunmasını sağlar ve genellikle daha verimli analiz veya modelleme için kullanılır. İstatistiksel veya matematiksel teknikler kullanarak, orijinal özelliklerin kombinasyonlarından veya dönüşümlerinden yeni özellikler oluşturur.

2.5.1.1 Temel Bileşen Analizi

Özellik çıkarımı yöntemlerinden biri olan temel bileşen analizi, veri kümesini daha az sayıda değişkenle açıklamak için kullanılır (Arslan 2021). PCA, verileri yeni bir koordinat sistemine doğrusal olarak dönüştürerek en büyük varyasyonu yakalayan yönleri tanımlar ve değişken sayısını azaltırken maksimum varyasyonu korumayı amaçlar (Karakaya 2024). PCA'da, ana bileşenlerin sayısını belirlemek çok önemlidir. Tüm ana bileşenler arasından seçilecek p sayıda ana bileşen, veriyi en iyi şekilde temsil edecek bileşenler olmalıdır.

2.5.2 Özellik Seçimi

Özellik seçimi, veri kümesini en iyi temsil edecek bir alt özellik kümesinin seçimi olarak tanımlanmaktadır (Budak 2018). Verinin temsilinde en önemli özelliklerin belirlenerek seçilmesi ile boyut indirgeme işlemini gerçekleştirir.

Özellik seçme yöntemlerinden biri olan filtreleme yöntemleri, özellikleri seçmek yerine, mesafe, bilgi, doğruluk, korelasyon veya tutarlılık gibi temellere dayanan bir

değerlendirme fonksiyonu kullanarak tüm özellik kümesini sıralar ve bu sıralamaya göre özellik seçimi kullanıcı tarafından gerçekleştirilir (Büyükkeçeci ve Okur 2023). Bu tez çalışmasında, hem doğrusal hem de doğrusal olmayan ilişkileri değerlendirebilmek, hesaplama verimliliği sağlamak ve model performansını optimize etmek amacıyla filtreleme yöntemlerinden korelasyon tabanlı yöntem ve karşılıklı bilgi yöntemi kullanılmıştır.

2.5.2.1 Korelasyon Tabanlı Özellik Seçimi

Korelasyon tabanlı özellik seçimi, hedef değişkenle yüksek korelasyon gösteren alt kümeleri bulmaya çalışan korelasyon temelli filtreleme algoritmasıdır (Emhan ve Akın 2019). Bu yöntem, seçilen özelliklerin hedef değişkenle doğrusal ilişki kurmasına odaklanır. Böylece, hedef değişkenin tahmin edilmesine en fazla katkıda bulunan özellikler belirlenebilir. Korelasyon tabanlı yöntemler, hesaplama açısından düşük maliyetlidir ve büyük veri setlerinde hızlı bir şekilde uygulanabilir.

2.5.2.2 Karşılıklı Bilgi Özellik Seçimi

Bir diğer filtreleme yöntemi olan karşılıklı bilgi, bir değişkenin başka bir değişken hakkında sahip olduğu bilgi miktarını ve değişkenler arasındaki, doğrusal olmayan ilişkiler de dahil, her türlü ilişkiyi ölçebilen bir yöntemdir (Vergara ve Estévez 2014). Karşılıklı bilgi yöntemi, özelliklerin hedef değişkenle olan bilgi paylaşımını ölçer. Hedef değişkene en fazla bilgi sağlayan özelliklerin seçimi, modelin performansını artırabilir. Yüksek karşılıklı bilgi değeri, ilgili özelliğin hedef değişkene daha fazla bilgi taşıdığı anlamına gelir.

2.6 Literatür İncelemesi

Güncel çalışmalardaki birçok araştırmacı botnet saldırılarının tespiti için farklı yöntemler üzerinde çalışmaktadır. Makine öğrenmesi yöntemleriyle yapılan bu tespit işlemleri oldukça etkili bir yöntemdir.

Alejandre vd. (2017) çalışmalarında, botnet saldırı tespiti için ISOT ve ISCX veri setlerini ayrı ayrı değerlendirip, aynı zamanda bu veri setlerinin birleşiminden oluşan veri seti kombinasyonunu kullanarak özellik seçimi yapma adına yeni bir yöntem geliştirmişlerdir. Özellik seçimi için genetik algoritması ve makine öğrenme algoritması olarak C4.5 yöntemi kullanılmıştır. En iyi sonuç ISOT veri setinde 10-11 özellik kullanılarak %99,46 doğruluk oranı ile elde edilmiştir.

Bahşi vd. (2018), botnet saldırı tespitinde bir laboratuvar ortamında elde edilen özel bir veri seti kullanmışladır. Özellik sayısını en aza indirmek için fishers yöntemi ve makine öğrenmesi algoritmalarından DT ve KNN algoritmaları kullanılmıştır. 10 kat çapraz doğrulama yönteminde 10 özellik ile DT algoritmasında %98,9 başarı oranı elde edilmiştir.

Kanimozhi ve Jacob (2019), çalışmalarında veri seti CSE-CIC-IDS2018 üzerinde botnet saldırılarını tespit etmek için yapay sinir ağları (ANN) algoritmasını kullanmışlardır. Modelin aşırı uyum durumuna düşmesi sebebiyle hiper parametre optimizasyonu yapılmış ve %99,97 doğruluk oranına ulaşılmıştır.

McKay vd. (2019), botnet saldırı tespitinde CICIDS2017 veri kümesini kullanarak çeşitli makine öğrenme algoritmalarını değerlendirmiştir. Çalışmalarında; RF, OneR, KNN, J48, çok katmanlı algılayıcı ve Naive Bayes (NB) gibi yöntemler kullanılmış ve dengeli veri kümesiyle daha başarılı sonuçlar elde edilmiştir. Özellikle J48 algoritması, %98,73 doğruluk oranı ile en yüksek performansı sergilemiştir.

Muhammad vd. (2020), çalışmalarında Cyber Clean Center veri seti üzerinde botnet saldırılarını tespit etmek amacıyla LR, Destek Vektör Makinesi (SVM), RF ve Çok Katmanlı Algılayıcı algoritmalarını kullanmışlardır. Veri seti üzerinde boyut indirgeme yöntemlerinden PCA ve bilgi kazanımı tekniklerini uygulayarak özellik sayısını azaltmışlardır. 55 olan özellik sayısını PCA yöntemi ile 35 bileşene, bilgi kazanımı yöntemi ile ise 5 özelliğe düşürerek toplamda 40 özellik elde etmişlerdir. RF algoritması, %99 doğruluk oranı ile en yüksek performansı sergilemiştir.

Karaman vd. (2020), çalışmalarında CSE-CIC-IDS2018 veri seti üzerinde çeşitli saldırı türlerini ele almışlardır. Bunlardan birisi de botnet saldırılarıdır. Saldırıları tespit etmek için ANN algoritması kullanılmıştır. Veri ön işleme adımı olarak eksik değerler veri setinden atılmış ve özellikler normalize edilmiştir. Veri seti %80 eğitim ve %20 test olarak ayrılmıştır. Özelliklerin en aza indirilmesi için BruteForce yöntemi kullanılmıştır. 4 özellik ile %93,23 doğruluk oranı elde edilmiştir.

Kanimozhi ve Jacob (2021), çalışmalarında CSE-CIC-IDS2018 veri seti üzerinde botnet saldırılarını tespit etmek için ANN, KNN, SVM, Adaboost, NB ve RF algoritmalarını uygulamışlardır. Bu algoritmaların başarı durumları karşılaştırılmıştır. Veri ön işleme adımı olarak eksik değerler ortalama ile doldurulmuş ve özellikler StandardScaler kullanılarak ölçeklendirilmiştir. Veri seti %80 eğitim ve %20 test olarak ayrılmıştır. Performanslar, kalibrasyon eğrileri aracılığıyla değerlendirilmiştir. Kalibrasyon eğrisine göre ANN algoritması botnet saldırısı tespitinde (%99,9) en başarılı algoritma olarak tespit edilmiştir.

Altunay ve Albayrak (2021), çalışmalarında CSE-CIC-IDS2018 veri seti üzerinde çeşitli saldırı türlerini ele almışlar bunlardan birisi botnet saldırılarıdır. Bu saldırı türlerini tespit etmek için CNN mimarisi ile derin öğrenme modeli oluşturulmuştur. Veri seti içerisindeki saldırı türleri sayısı birbirinden farklıdır. Dengesiz sınıf dağılımını dengeli hale getirmek için Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE) ile sentetik veri üretimi yapılmıştır. Veri seti %80 eğitim ve %20 test olarak ayrılmıştır. Sınıflandırma başarımı botnet saldırıları için %98,9 oranında tespit edilmiştir.

Odeyemi vd. (2023), çalışmalarında CSE-CIC-IDS2018 veri setinde botnet saldırılarını içeren dosya üzerinde yedi farklı makine öğrenmesi algoritmalarının yanı sıra topluluk modelleri algoritmalarının başarılarını karşılaştırmışlardır. Kullanılan makine öğrenmesi algoritmaları : NB, DT, KNN, LR, ANN, Quadratic Discriminant Analysis ve SVM. Kullanılan topluluk modelleri : Adaboost, Gradient Boosting, RF ve Max Voting. Kullanılan yöntemler arasından en başarılı algoritma %99,89 doğruluk oranı ile topluluk modellerinden biri olan RF algoritması olarak tespit edilmiştir.

Mishra ve Sharma (2024), çalışmalarında Net Flow veri seti üzerinde botnet saldırı tespiti için yeni bir özellik kombinasyonu tabanlı yaklaşım önermektedirler. Özellik çıkarımı sırasında, genel özellikler, istatistiksel ve alt ağ özellikleri gibi çeşitli özellikler çıkarılmıştır. Ardından, özellik setlerinin kombinasyonu üzerinde makine öğrenmesi yöntemleri uygulanmıştır. Kullanılan makine öğrenmesi algoritmaları LR, KNN ve RF'dir. Önerilen yöntem, en iyi sonucu RF algoritmasında yaklaşık %99,8 doğruluk oranı ile elde etmiştir.

Wijaya vd. (2024), botnet saldırı tespitinde NCC-2 veri seti üzerinde KNN algoritmasıyla sınıf dengeleme tekniklerinin performansını incelemiş ve Adaptif Sentetik Örnekleme, Rastgele Alt Örnekleme ve SMOTE gibi yöntemlerin etkinliğini değerlendirmişlerdir. Veri ön işleme aşamasında kategorik değişkenler sayısal forma dönüştürülmüş ve özellikler manuel olarak azaltılmıştır. %70 eğitim, %30 test veri ayrımıyla yapılan deneylerde, dengeleme yapılmadan elde edilen sınıflandırma sonuçlarının en iyi doğruluğa (%98,45) ulaştığı görülmüştür. Ancak, KNN ve Rastgele Aşırı Örnekleme kombinasyonu, azınlık sınıfı için en yüksek recall değerini sağlamıştır.

Jahbel vd. (2024), botnet saldırı tespitinde NCC-2 veri seti üzerinde iki aşamalı makine öğrenmesi algoritmaları kullanmışlardır. Kullanılan algoritmalar arasında DT, RO, KNN ve XGBoost yer alırken, özellik seçiminde ki-kare testi kullanılarak en ilgili 15 özellik seçilmiştir. Veri setindeki dengesizlik, SMOTE aşırı örnekleme tekniği ile giderilmiştir. Ki-kare testi ve SMOTE kombinasyonu, özellikle RO modelinde doğru pozitif oranını arttırmış ve %98,58 doğruluk oranı elde edilmiştir. Çalışma, bu kombinasyonun botnet tespiti performansını artırdığını göstermektedir.

Ertuğrul (2024), CSE-CIC-IDS2018 veri seti üzerinde çeşitli saldırı türlerini ele almışlar bunlardan birisi botnet saldırılarıdır. Saldırıları tespit etmek amacıyla makine öğrenimi tabanlı bir saldırı tespit sistemi geliştirmiş ve kapsamlı bir performans analizi gerçekleştirmiştir. Çalışmada üç farklı öznitelik seçme yöntemi kullanılmıştır: SelectKBest, MI ve Ki-Kare. Her bir öznitelik seçme algoritmasından ilk 10 öznitelik belirlenmiş ve toplam 30 öznitelik üzerinden yapılan analiz sonucunda 5 öznitelik seçilmiştir. Veri dengesizliğini gidermek için SMOTE yöntemi uygulanmıştır. Sınıflandırma işlemlerinde, denetimli makine öğrenimi algoritmalarından KNN, DT,

NB, RF, Gradyan Artırma, AdaBoost, LR ve Doğrusal Ayrımcılık Analizi ile derin öğrenme alanından Çok Katmanlı Algılayıcı kullanılmıştır. Çalışmada en yüksek başarı, %99,83 doğruluk oranıyla RF algoritmasıyla elde edilmiştir.

3. MATERYAL ve METOT

3.1 Çalışma Ortamının Hazırlanması

Makine öğrenmesi projelerini geliştirmek için güçlü bir donanıma ve birçok paket kurulumuna ihtiyacınız vardır. Programlama dillerinin localde kullanılabilmesi için tümleşik geliştirme ortamlarının kurulumunun gerçekleştirilmesi ve bu işlemler için bilgisayarın işlemcisinin oldukça güçlü olması gerekir (İnt.Kyn.1). Bu ihtiyaçları çevrimiçi bir platform olan Google Colab, kullanıcılara hazır bir ortam olarak sunmaktadır.

Google Colab; internet erişimi ve google hesabı olan herkesin tarayıcıları aracılığıyla Python kodu yazıp çalıştırmasına olanak tanıyan, çok kullanıcılı, işbirliğine dayalı bir ortamdır. (Werth vd. 2022) Google Colab'ın önemli özelliklerinden biri güçlü bir işlemciye ücretsiz erişim sağlamasıdır (Carneiro vd. 2018).

Ücretsiz olarak GPU ve TPU gibi güçlü donanım kaynaklarına erişim sunan, bu sayede büyük veri setleriyle çalışmayı ve makine öğrenmesi modellerini hızlandırmayı kolaylaştırdığı için çalışma ortamı olarak Google Colab kullanılmıştır (İnt.Kyn.2).

Çalışmada Google Colab Pro versiyonu tercih edilmiştir. Standart sürüme kıyasla daha yüksek hız ve performans sunan Pro versiyonu, araştırma süreçlerinde önemli avantajlar sağlamaktadır. NVIDIA T4 GPU desteği sayesinde, makine öğrenimi modellerinin eğitimi ve veri işleme süreçlerinde yüksek hız ve verimlilik elde edilmiştir. Ayrıca, Pro versiyonunun sunduğu yüksek RAM seçeneği, büyük veri setleriyle çalışmayı mümkün kılarak analiz süreçlerini kolaylaştırmıştır. Standart sürümde karşılaşılan kısa bağlantı süreleri, Pro versiyonunda yerini daha uzun ve kesintisiz oturumlara bırakmıştır. Bu özellik, özellikle uzun süren hesaplama işlemlerinde büyük bir avantaj sağlamaktadır. Sonuç olarak, Google Colab Pro, sunduğu güçlü donanım ve kullanıcı dostu özellikleriyle, tez çalışmasının gereksinimlerini karşılayan ve araştırma süreçlerini optimize eden ideal bir çalışma ortamı sunmaktadır.

Bilimsel hesaplar için Numpy, veri analizi ve manipülasyonu için Pandas, veri görselleştirme işlemleri için Seaborn ve Matplotlib, makine öğrenimi işlemleri için Scikit-learn kütüphaneleri sisteme dahil edilmiştir.

3.2 Veri Setinin Tanımlanması

Siber saldırganların yeni saldırılarını tespit etmek, birçok araştırmacının ilgisini çekerken, sistemin karmaşıklığı gerçek dünya uygulamalarını zorlaştırmakta ve bu nedenle normal ve anormal davranışları içeren gerçek etiketli ağ izleri üzerinde kapsamlı test ve değerlendirme süreçleri en iyi yöntem olmaktadır (İnt.Kyn.3). Bu çerçevede veri kümeleri oluşturmak üzere Kanada İletişim Güvenliği Kuruluşu (Canada Communications Security Establishment-CSE) ve Kanada Siber Güvenlik Enstitüsü (Canadian Institute for Cybersecurity-CIC) arasındaki ortak bir proje ile CIC-IDS2018 veri seti geliştirilmiştir. Bu veri seti, siber güvenlik araştırmalarında saldırı tespiti sistemlerini eğitmek, test etmek ve değerlendirmek için yaygın olarak kullanılmaktadır.

Yapmış olduğumuz çalışmada CIC-IDS2018 veri seti kullanılmıştır. Veri seti Brute-force, Heartbleed, Botnet, DoS, DDoS, web saldırıları ve ağa içeriden sızma şeklinde 7 farklı saldırı senaryosu içermektedir. Saldırı altyapısı 50 makineden ve kurban altyapısı 420 bilgisayar ve 30 sunucudan oluşmaktadır. CICFlowMeter-V3 kullanılarak elde edilen paketler ağ trafik akışlarına dönüştürülmüş ve 80 öznitelik sunulmuştur (İnt.Kyn.3).

CIC-IDS2018 veri seti 10 farklı günde kaydedilen csv uzantılı dosyalardan oluşmaktadır. Bu dosyalar, farklı saldırı türlerini ve normal ağ trafiğini temsil eder. Çalışmada botnet saldırıları inceleneceğinden bu saldırı türünü barındıran "Friday-02-03-2018.csv" dosyası kullanılmıştır. Dosyanın içeriği çizelge 3.1'de gösterilmiştir.

Çizelge 3.1 CIC-IDS2018 veri setinden elde edilen bilgiler

Dosya Adı	**Size**	**Label**
Friday-02-03-2018.csv	(1048575, 80)	Benign: 762384 Bot: 286191

3.3 Veriye İlk Bakış

Veriye ilk bakış, bir veri setinin temel özelliklerini hızla değerlendirmek amacıyla gerçekleştirilen başlangıç analizidir. Bu aşamada, Pandas kütüphanesi içerisindeki info() metodu ile veri seti hakkında bilgi ediniriz. Bu metot ile veri çerçevesinin yapısı, sütun adları, veri türleri, eksik değerler ve bazı temel istatistikler gözlemlenir.

Çalışmada kullanılan botnet saldırı dosyası içerisinde 1048575 gözlem değeri 80 değişken bulunmaktadır. Çizelge 3.2'de bu dosya içerisindeki veri setine ait bilgiler yer almaktadır.

Çizelge 3.2 Veri setine ait öznitelikler ve açıklamaları

#	Öznitelik	Açıklama
1	Dst Port	Hedef port bilgisi
2	Protocol	Protokol bilgisi
3	Timestamp	Zaman bilgisi
4	Flow Duration	Akış süresi
5	Tot Fwd Pkts	İleri yönde toplam paket sayısı
6	Tot Bwd Pkts	Geri yönde toplam paket sayısı
7	TotLen Fwd Pkts	İleri yönde paketlerin toplam boyutu
8	TotLen Bwd Pkts	Geri yönde paketlerin toplam boyutu
9	Fwd Pkt Len Max	İleri yönde maksimum paket boyutu
10	Fwd Pkt Len Min	İleri yönde minimum paket boyutu
11	Fwd Pkt Len Mean	İleri yönde ortalama paket boyutu
12	Fwd Pkt Len Std	İleri yönde paket standart sapma boyutu
13	Bwd Pkt Len Max	Geri yönde maksimum paket boyutu
14	Bwd Pkt Len Min	Geri yönde minimum paket boyutu
15	Bwd Pkt Len Mean	Geri yönde ortalama paket boyutu
16	Bwd Pkt Len Std	Geri yönde paket standart sapma boyutu
17	Flow Byts/s	Saniyede aktarılan byte sayısı
18	Flow Pkts/s	Saniyede aktarılan paket sayısı
19	Flow IAT Mean	İki akış arasındaki ortalama süre

Çizelge 3.2 (Devam) Veri setine ait öznitelikler ve açıklamaları

#	Öznitelik	Açıklama
20	Flow IAT Std	İki akış arasındaki standart sapma süre
21	Flow IAT Max	İki akış arasındaki maksimum süre
22	Flow IAT Min	İki akış arasındaki minimum süre
23	Fwd IAT Tot	İleri yönde gönderilen iki paket arasındaki toplam süre
24	Fwd IAT Mean	İleri yönde gönderilen iki paket arasındaki ortalama süre
25	Fwd IAT Std	İleri yönde gönderilen iki paket arasındaki standart sapma süresi
26	Fwd IAT Max	İleri yönde gönderilen iki paket arasındaki maksimum süre
27	Fwd IAT Min	İleri yönde gönderilen iki paket arasındaki minimum süre
28	Bwd IAT Tot	Geri yönde gönderilen iki paket arasındaki toplam süre
29	Bwd IAT Mean	Geri yönde gönderilen iki paket arasındaki ortalama süre
30	Bwd IAT Std	Geri yönde gönderilen iki paket arasındaki standart sapma süresi
31	Bwd IAT Max	Geri yönde gönderilen iki paket arasındaki maksimum süre
32	Bwd IAT Min	Geri yönde gönderilen iki paket arasındaki minimum süre
33	Fwd PSH Flags	İleri yönde hareket eden paketlerde PSH bayrağının ayarlanma sayısı
34	Bwd PSH Flags	Geri yönde hareket eden paketlerde PSH bayrağının ayarlanma sayısı
35	Fwd URG Flags	İleri yönde hareket eden paketlerde URG bayrağının ayarlanma sayısı
36	Bwd URG Flags	Geri yönde hareket eden paketlerde PSH bayrağının ayarlanma sayısı
37	Fwd Header Len	İleri yöndeki başlıklar için kullanılan bayt sayısı
38	Bwd Header Len	Geri yöndeki başlıklar için kullanılan bayt sayısı
39	Fwd Pkts/s	Saniye gönderilen ileri yöndeki paket sayısı
40	Bwd Pkts/s	Saniye gönderilen geri yöndeki paket sayısı
41	Pkt Len Min	Bir akışın minimum uzunluğu
42	Pkt Len Max	Bir akışın maksimum uzunluğu
43	Pkt Len Mean	Bir akışın ortalama uzunluğu
44	Pkt Len Std	Bir akışın standart sapma uzunluğu
45	Pkt Len Var	Ardışık olarak iletilen paket arasındaki minimum süre

Çizelge 3.2 (Devam) Veri setine ait öznitelikler ve açıklamaları

#	Öznitelik	Açıklama
46	FIN Flag Cnt	FIN içeren paketlerin sayısı
47	SYN Flag Cnt	SYN içeren paketlerin sayısı
48	RST Flag Cnt	RST içeren paketlerin sayısı
49	PSH Flag Cnt	PSH içeren paketlerin sayısı
50	ACK Flag Cnt	ACK içeren paketlerin sayısı
51	URG Flag Cnt	URG içeren paketlerin sayısı
52	CWE Flag Count	CWE içeren paketlerin sayısı
53	ECE Flag Cnt	ECE içeren paketlerin sayısı
54	Down/Up Ratio	İndirme ve yükleme oranı
55	Pkt Size Avg	Ortalama paket boyutu
56	Fwd Seg Size Avg	İleri yönde gözlemlenen ortalama boyut
57	Bwd Seg Size Avg	Geri yönde gözlemlenen ortalama boyut
58	Fwd Byts/b Avg	İleri yönde iletilen kütle oranı ortalama byte sayısı
59	Fwd Pkts/b Avg	İleri yönde iletilen kütle oranı ortalama paket sayısı
60	Fwd Blk Rate Avg	İleri yönde iletilen kütle oranının ortalama sayısı
61	Bwd Byts/b Avg	Geri yönde iletilen kütle oranı ortalama byte sayısı
62	Bwd Pkts/b Avg	Geri yönde iletilen kütle oranı ortalama paket sayısı
63	Bwd Blk Rate Avg	Geri yönde iletilen kütle oranının ortalama sayısı
64	Subflow Fwd Pkts	İleri yönde bir alt akıştaki ortalama paket sayısı
65	Subflow Fwd Byts	İleri yönde bir alt akıştaki ortalama bayt sayısı
66	Subflow Bwd Pkts	Geri yönde bir alt akıştaki ortalama paket sayısı
67	Subflow Bwd Byts	Geri yönde bir alt akıştaki ortalama bayt sayısı
68	Init Fwd Win Byts	İleri yönde ilk pencerede gönderilen bayt sayısı
69	Init Bwd Win Byts	Geri yönde ilk pencerede gönderilen bayt sayısı
70	Fwd Act Data Pkts	İleri yönde iletilen en az 1 byte TCP veri yüküne sahip paket sayısı
71	Fwd Seg Size Min	İleri yönde gözlemlenen minimum segment boyuytu
72	Active Mean	Bir akışın boşta kalmadan önce aktif olduğu ortalama süre
73	Active Std	Bir akışın boşta kalmadan önce aktif olduğu standart sapma süresi
74	Active Max	Bir akışın boşta kalmadan önce aktif olduğu maksimum süre

Çizelge 3.2 (Devam) Veri setine ait öznitelikler ve açıklamaları

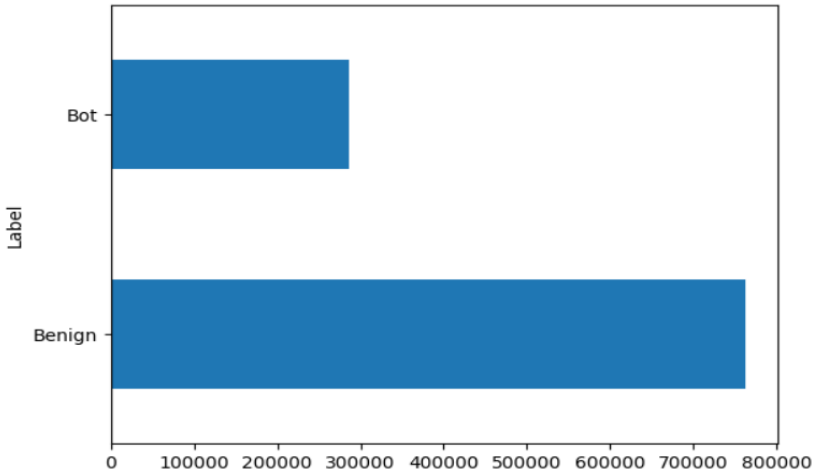
#	Öznitelik	Açıklama
75	Active Min	Bir akışın boşta kalmadan önce aktif olduğu minimum süre
76	Idle Mean	Bir akışın aktif hale gelmeden önce boşta kaldığı ortalama süre
77	Idle Std	Bir akışın aktif hale gelmeden önce boşta kaldığı standart sapma süresi
78	Idle Max	Bir akışın aktif hale gelmeden önce boşta kaldığı maksimum süre
79	Idle Min	Bir akışın aktif hale gelmeden önce boşta kaldığı minimum süre
80	Label	Etiket

Veri setinde yer alan "Label" değeri, sınıflandırma işlemi için hedef değişken olarak kullanılmaktadır. Bu değişken, iki farklı sınıf içermektedir: "Bening" ve "Bot". Bening, veri setindeki örneklerin zararsız veya normal davranış sergileyen durumları temsil ettiğini ifade eder. Genellikle, sistemin doğal ve beklenen şekilde çalışmasını yansıtan veriler bu sınıfa dahildir. Bot**,** veri setindeki örneklerin kötü amaçlı, zararlı veya anormal bir davranış sergileyen durumlarını ifade eder. Bu tür veriler genellikle tehdit oluşturan ya da sistemde istenmeyen davranışları temsil eder.

Veri setinde yer alan "Label" değerine ait veri sayıları çizelge 3.3'te detaylı bir şekilde sunulmuştur. Şekil 3.1'de Label değerlerinin görsel dağılımı sunmaktadır. Bu görsel sınıf dağılımının hangi kategoride yoğunlaştığını anlaşılır bir şekilde görselleştirir.

Çizelge 3.3 Label değerine göre veri sayıları

Label	Veri Sayısı
Benign	768423
Bot	286191



Şekil 3.1 Label değerine göre veri sayıları dağılımı

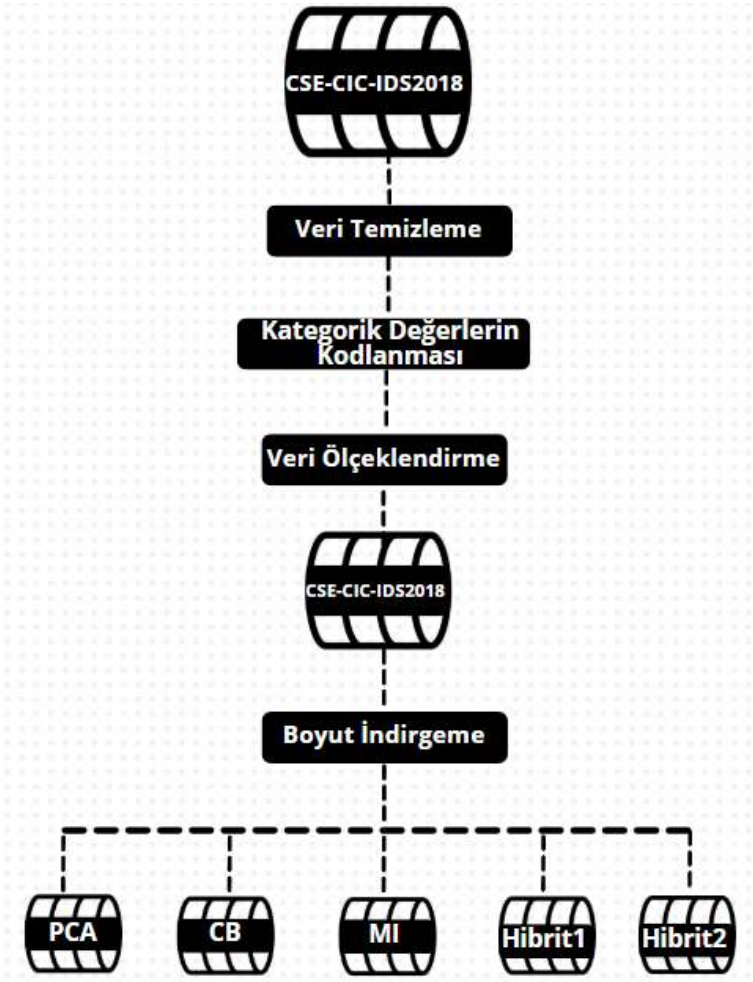
3.4 Veri Ön İşleme

Veri setinin makine öğrenmesi modellerine girdi olarak verilmeden önce, modelin en iyi sonuçları elde etmesi için belirli işlemlerden geçirilmesi gerekmektedir (Asan vd. 2024). Veri ön işleme teknikleri verilerin kalitesini artırmaktadır (Oğuzlar 2003). Kaliteli veriler, kullanılmak istenilen makine öğrenmesi modellerinin doğru sonuçlar vermesini sağlayacaktır (Asan vd. 2024).

Farklı birçok veri ön işleme teknikleri bulunmaktadır. Veri ön işleme aşamaları, çeşitli yöntemler kullanılarak gerçekleştirilebilir. Bu aşamalar şekil 3.2'de gösterilmiştir.

Şekil 3.2 Veri ön işleme adımları

Veri ön işleme adımları olarak veri setine uygulanan işlemler şekil 3.3'te gösterilmiştir.



Şekil 3.3 Veri setine uygulanan veri ön işlem adımları

3.4.1 Veri Temizleme

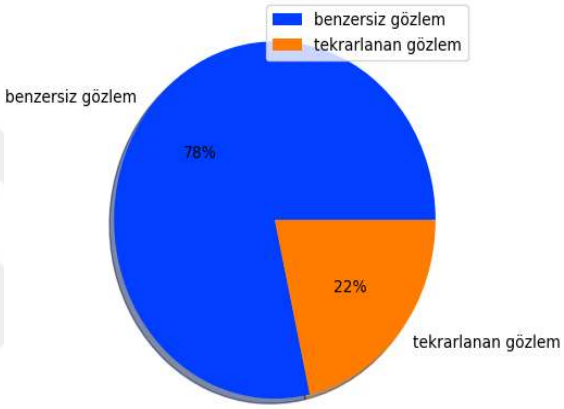
Veri temizleme, veri setindeki hataları, eksik veya tekrar eden verilerin tespiti ve kaldırılması gibi işlemleri gerektirmektedir (Oğuzlar 2023).

Veri seti içerisinde infinitive değerler bulunmaktadır. Bu yüzden veri seti içerisindeki infinive değerler (np.inf, -np.inf) NaN değerlere dönüştürülerek diğer NaN olan tüm gözlem değerleriyle birlikte temizlenmiştir. Temizleme işlemi sonrası Label değerine göre veri sayıları çizelge 3.4'te gösterilmiştir.

Çizelge 3.4 Temizlik sonrası Label değerine göre veri sayıları

Label	Veri Sayısı
Benign	758334
Bot	286191

Veri seti içerisinde tekrar eden kayıtlar tespit edilmiştir. Bu tekrar eden kayıt sayısı 224621'dir. Veri seti içerisindeki tekrar eden ve benzersiz olan gözlem değerleri dağılımı şekil 3.4'te gösterilmiştir.



Şekil 3.4 Veri seti içerisindeki tekrar eden ve benzersiz gözlem sayıları dağılımı

Tekrar eden kayıtlar veri setinden silinmiştir. Tekrar eden kayıtlar silindikten sonra veri setindeki Label değerine göre veri sayıları çizelge 3.5'te verilmiştir. Son durumda veri seti boyutu (819904,80) olmuştur.

Çizelge 3.5 Tekrar eden kayıtlar silindikten sonra Label değerine göre veri sayıları

Label	Veri Sayısı
Benign	675369
Bot	144535

3.4.2 Kategorik Değerlerin Kodlanması

Makine öğrenmesi modellerinin kurulabilmesi için değişken tiplerinin numerik olması gerekmektedir (Büyükçapar 2023). Veri setinde object türünde iki değişken bulunmaktadır: "Timestamp" ve "Label". Saldırı tespitinde zamanın etkisinin olmadığı

değerlendirilerek “Timestamp” değişkeni veri setinden doğrudan çıkarılmıştır. Diğer object türündeki “Label” değişkeni ise saldırı türlerini ifade eden bir kategorik değişken olarak kullanılmış ve bu değişken üzerinde veri dönüşümü işlemi gerçekleştirilmiştir. Veri dönüşümü gerçekleştirilirken "Benign":0, “Bot":1 olarak etiketlenmiştir. Böylelikle, makine öğrenmesi algoritmaları için uygun bir veri formatı oluşturulmuş ve modelleme sürecine geçiş sağlanmıştır. Etiketleme ve veri dönüşümü sonucunda elde edilen bilgiler ve ilgili değişiklikler çizelge 3.6’da gösterilmiştir.

Çizelge 3.6 Label değerine göre veri sayıları

Label	Veri Sayısı
0	675369
1	144535

3.4.3 Veri Ölçeklendirme

Verilerin farklı boyutlarda olması bazı sınıflandırma yöntemlerinin öğrenme doğruluğunu olumsuz yönde etkileyebilir (Saygın 2021). Genellikle veri setlerinde çok farklı ölçeklere sahip özellikler bulunduğunda uygulanması gereken bir adım olan ölçeklendirme, bir özelliği belirli aralıklara getirme sürecidir (Çetin ve Yıldız 2022).

Veri seti içerisindeki gözlem değerleri farklı boyutlarda olduğu için ölçeklendirme işlemi uygulanmıştır. Ölçeklendirme yöntemlerinden standardizasyon yöntemi kullanılmıştır. Standardizasyon, veri setindeki özelliklerin aritmetik ortalaması 0 standart sapması 1 olacak şekilde veri setinin dönüştürülmesini sağlayan bir özellik ölçekleme yöntemidir (Karakaya 2024).

Projeye “from sklearn.preprocessing import StandardScaler” kütüphanesi eklenmiş ve StandardScaler() fonksiyonu kullanılarak veri setindeki özelliklerin ölçeklendirme işlemi gerçekleştirilmiştir.

3.4.4 Boyut İndirgeme

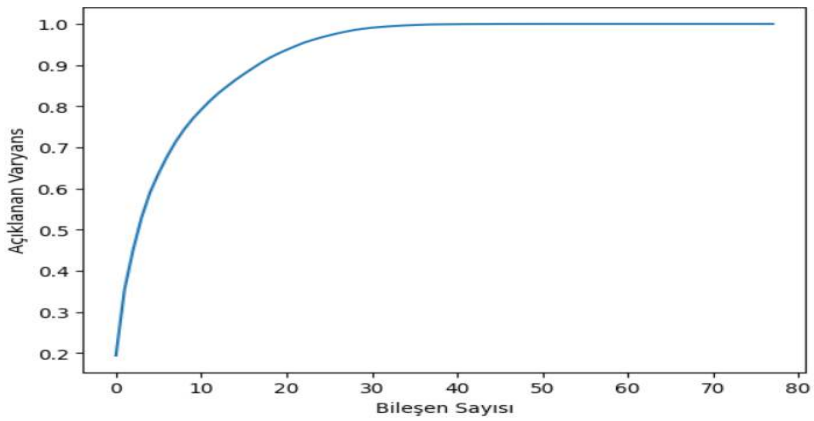
Boyut indirgeme işlemleri veri ön işleme adımlarından geçen veri seti üzerinde gerçekleştirilmiştir.

3.4.4.1 Temel Bileşen Analizi

PCA yöntemini uygulamak için "from sklearn.decomposition import PCA" kütüphanesi kullanılmıştır. PCA'da optimum bileşen sayısını belirlemek için açıklanan varyans değerleri şekil 3.5'te, bileşen sayısına göre açıklanan varyans değerleri grafiği şekil 3.6'da gösterilmiştir.

```
array([0.19355463, 0.35314469, 0.44960047, 0.52903155, 0.59042728,
       0.63654178, 0.67776052, 0.7136203 , 0.743608  , 0.76944756,
       0.79164698, 0.81252603, 0.83145342, 0.84774677, 0.86363864,
       0.87796167, 0.89182039, 0.90526517, 0.91731241, 0.92767529,
       0.93694492, 0.94558804, 0.95409806, 0.96067746, 0.96682082,
       0.97204041, 0.97717245, 0.9814434 , 0.98535102, 0.98819025,
       0.9907361 , 0.9924492 , 0.99408919, 0.99528042, 0.9964099 ,
       0.99720398, 0.99797062, 0.9986418 , 0.99888702, 0.99910207,
       0.99930227, 0.99946329, 0.99957904, 0.99967115, 0.99975333,
       0.9998293 , 0.99988034, 0.99992046, 0.99994411, 0.99996043,
       0.99997336, 0.99998032, 0.99998501, 0.99998903, 0.99999261,
       0.99999538, 0.99999781, 0.99999919, 0.99999995, 0.99999999,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        , 1.        ])
```

Şekil 3.5 Açıklanan varyans değerleri



Şekil 3.6 Bileşen sayısına göre açıklanan varyans değerleri grafiği

Veri setindeki gereksiz veya düşük bilgi taşıyan özellikleri eleyerek veri boyutunu azaltmak ve modelin genelleme yeteneğini artırmak amacıyla açıklanan varyans değerleri ve grafik incelenmiş, toplam varyansın %80'ini temsil eden bileşen sayısı seçilmiştir. Bu oran, verinin büyük bir kısmını açıklarken modelin karmaşıklığını artırmadan performansını dengede tutmayı hedeflemiştir. Daha yüksek varyans oranlarının hesaplama maliyetini artırabileceği ve modelin gereksiz ayrıntılara odaklanmasına yol açabileceği göz önüne alınarak, doğruluk ve verimlilik açısından %80 varyans oranı tercih edilmiştir. Analiz sonucunda, %80 varyans oranını karşılayan bileşen sayısının 12 olduğu belirlenmiştir.

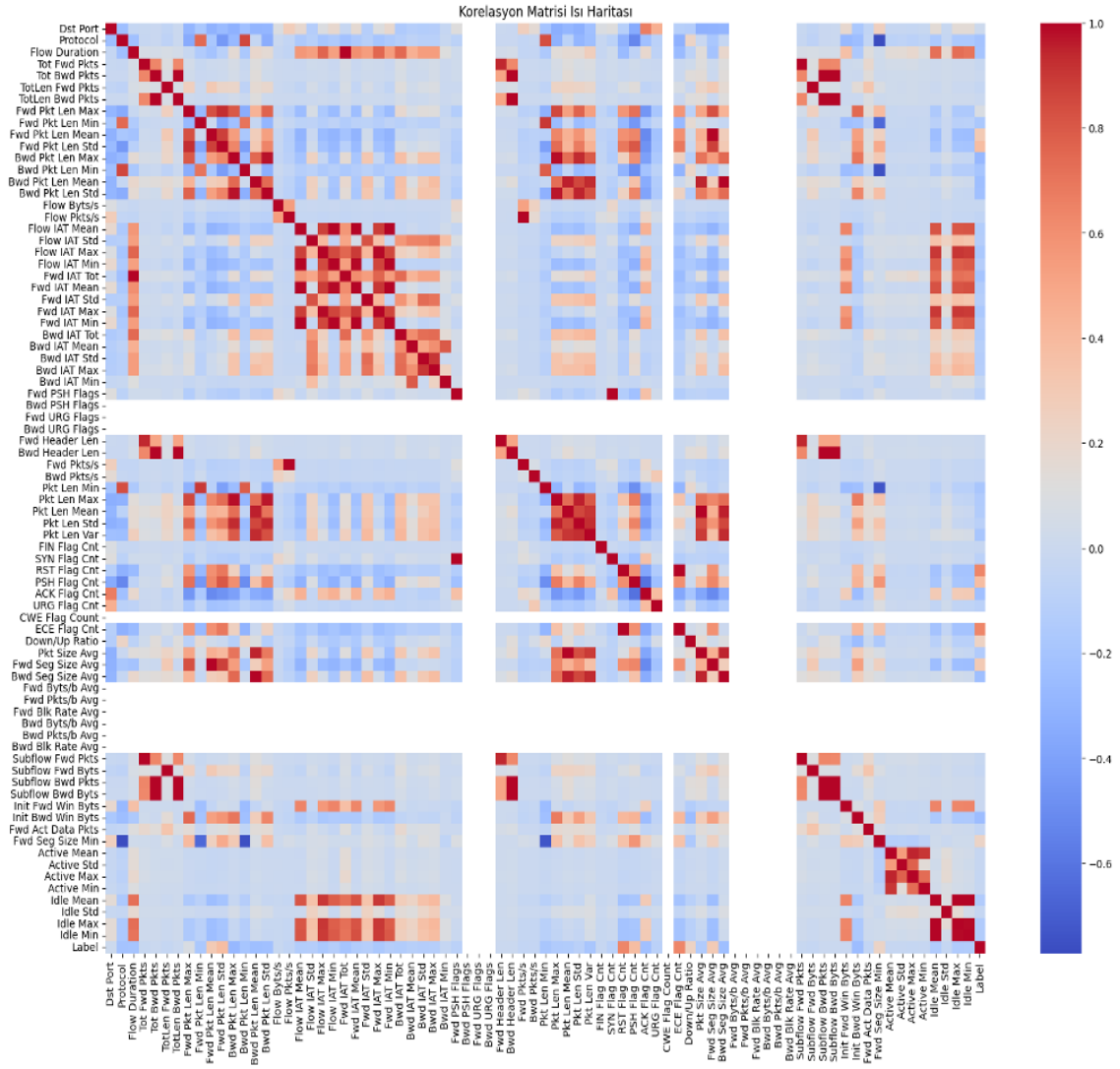
PCA yöntemiyle yapılan dönüşüm sonucunda elde edilen 12 bileşen, veri setindeki orijinal özelliklerle birebir eşleşmeyen soyutlanmış özelliklerdir. Bu bileşenler, orijinal özelliklerin doğrusal kombinasyonlarından oluşur ve her biri verideki varyansın belirli bir kısmını açıklar. Yani, her bir bileşen, orijinal tüm özelliklerin belirli ağırlıklarla birleştirilmesiyle oluşturulmuş bir vektördür. Bu süreç, verinin daha anlamlı ve anlaşılabilir bir şekilde temsil edilmesini sağlar, ancak her bir bileşen, orijinal özelliklerin doğrudan karşılığı değildir.

PCA sonucunda elde edilen 12 bileşen kullanılarak yeni bir veri seti oluşturulmuştur. Oluşturulan bu veri seti, çalışmada kullanılan yöntemle aynı adı taşıyan "PCA" ismiyle kaydedilmiştir.

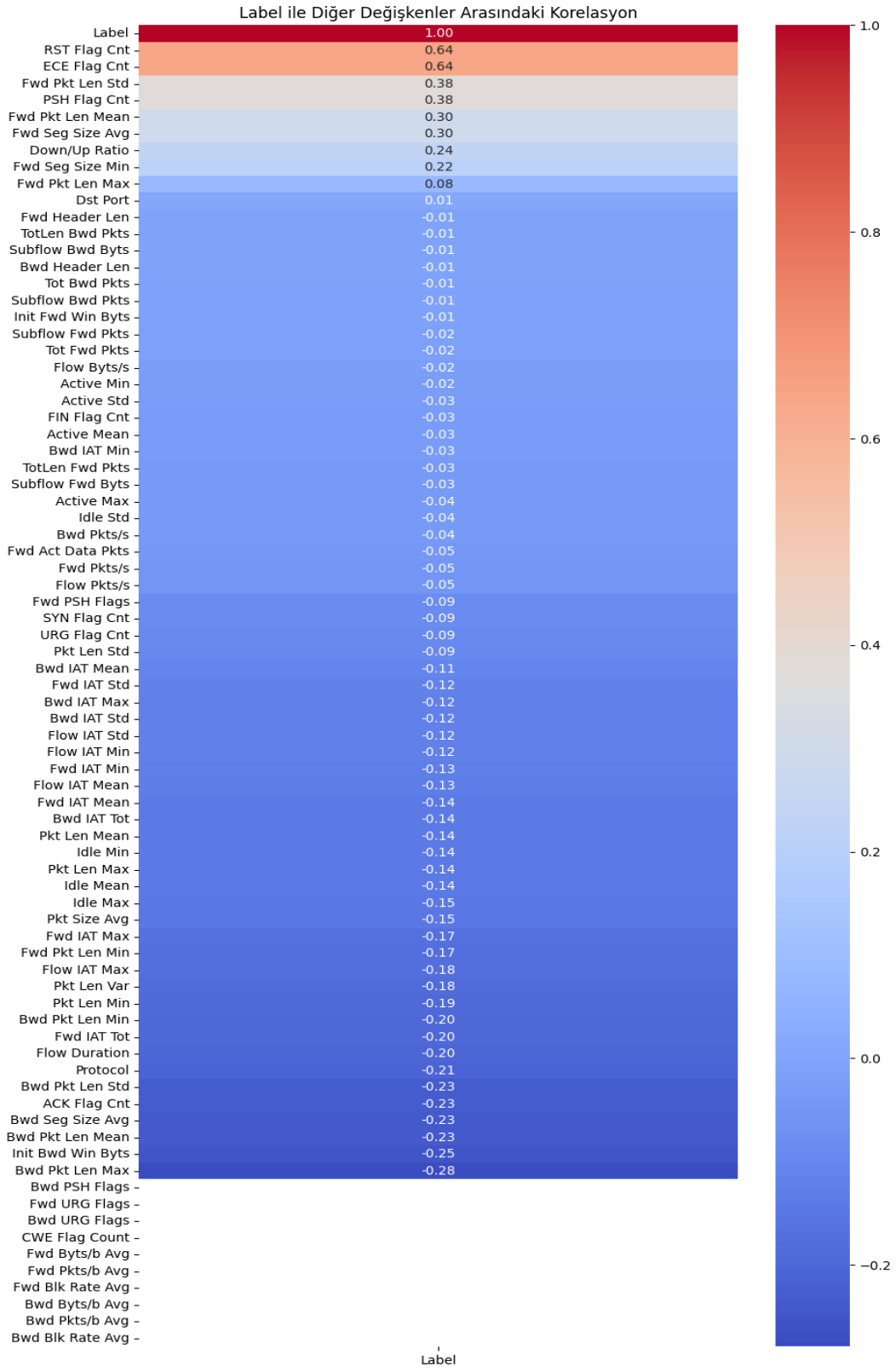
3.4.4.2 Korelasyon Tabanlı Özellik Seçimi

Korelasyon tabanlı özellik seçimi, hedef değişken ile bağımsız değişkenler arasındaki ilişkinin gücüne odaklanarak en açıklayıcı değişkenlerin seçilmesi amacıyla yapılmıştır. Bu işlem için, veri setindeki ilişkileri daha iyi anlayabilmek amacıyla korelasyon matrisi hesaplanmış ve bu matrisin görsel bir temsili olan ısı haritası oluşturulmuştur. Oluşturulan ısı haritası, şekil 3.7'de gösterilmektedir ve değişkenler arasındaki ilişkiyi açıkça ortaya koymaktadır. Ayrıca, veri setindeki tüm bağımsız değişkenlerin hedef değişken ile korelasyon katsayıları hesaplanmış ve bu katsayılar şekil 3.8'de görsel olarak sunulmuştur. Bu sayede, hedef değişkenle en güçlü ilişkiye sahip bağımsız

değişkenler belirlenmiş ve modelin başarısını artırmak için hangi özelliklere daha fazla odaklanılması gerektiği konusunda önemli bilgiler elde edilmiştir.



Şekil 3.7 Isı haritası



Şekil 3.8 Label ile diğer değişkenler arası korelasyon

Isı haritası, veriler arasındaki korelasyonları görsel olarak sunarak, değişkenler arasındaki ilişkinin gücünü ve yönünü hızlı bir şekilde anlamamıza yardımcı olur. Renk yoğunluğu, korelasyonun gücünü belirtir; koyu renkler güçlü korelasyonu, açık renkler ise zayıf korelasyonu ifade eder. Bu sayede, veri setindeki bağımsız değişkenlerin hedef değişkenle veya diğer değişkenlerle olan ilişkileri kolayca gözlemlenebilir.

Isı haritası ve hedef değişkenle diğer değişkenler arasındaki korelasyon incelendiğinde, özellik seçimi için 0,2'lik bir eşik değeri belirlenmiştir. Bu eşik değeri, modelin doğruluğunu artıracak güçlü korelasyona sahip değişkenleri seçerken, gereksiz karmaşıklığı önleyerek performansı optimize etmeyi amaçlamaktadır. Daha yüksek bir eşik değeri, önemli bilgilerin kaybolmasına yol açabilirken, daha düşük bir eşik değeri ise aşırı öğrenmeye (overfitting) neden olabilir. Bu denge, modelin anlamlı ilişkileri koruyarak gereksiz bilgileri dışlamasını sağlar. 0,2'lik eşik değerinin altında kalan değişkenler analizden çıkarılmış ve hedef değişkenle güçlü korelasyon gösteren bağımsız değişkenlerin listesi Şekil 3.9'da gösterilmiştir.

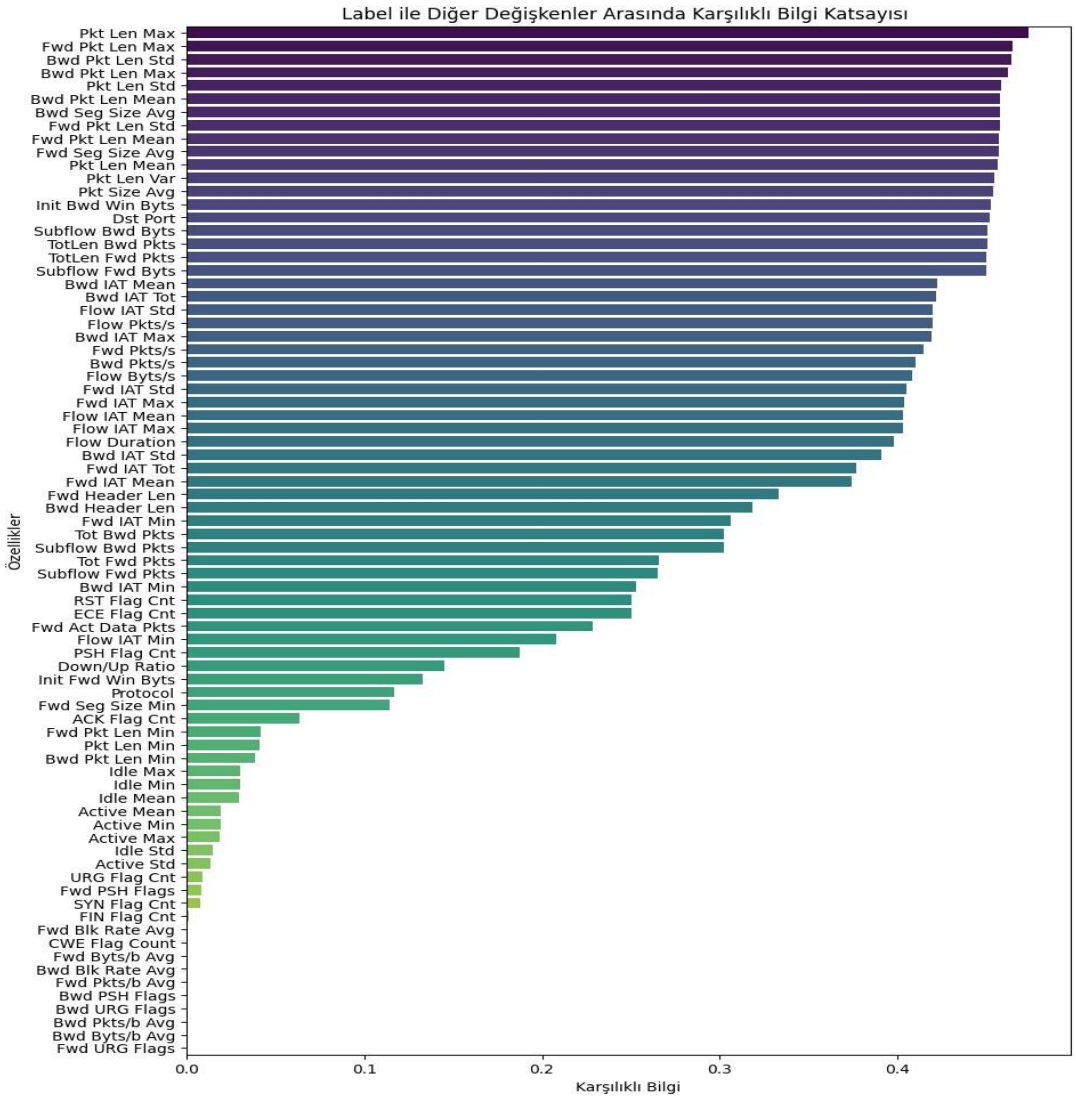
RST Flag Cnt	0.636889
ECE Flag Cnt	0.636887
Fwd Pkt Len Std	0.380461
PSH Flag Cnt	0.380049
Fwd Pkt Len Mean	0.297489
Fwd Seg Size Avg	0.297489
Bwd Pkt Len Max	0.278847
Init Bwd Win Byts	0.251478
Down/Up Ratio	0.238548
Bwd Pkt Len Mean	0.234383
Bwd Seg Size Avg	0.234383
ACK Flag Cnt	0.230189
Bwd Pkt Len Std	0.229077
Fwd Seg Size Min	0.218601
Protocol	0.205685
Flow Duration	0.201995

Şekil 3.9 0,2'den büyük korelasyona sahip değişkenlerin listesi

Çalışmada kullanılmak üzere, yukarıda listelenen yüksek korelasyona sahip 16 özellik seçilerek yeni bir veri seti oluşturulmuştur. Bu veri seti, çalışmada kullanılan yöntemle aynı adı taşıyan "CB" ismiyle kaydedilmiştir.

3.4.4.3 Karşılıklı Bilgi Özellik Seçimi

Karşılıklı bilgi özelik seçimi, bağımsız değişkenlerin hedef değişkene taşıdıkları bilginin gücüne odaklanarak en açıklayıcı değişkenlerin seçilmesi amacıyla yapılmıştır. Bu işlem için "from sklearn.feature_selection import mutual_info_classif" kütüphanesi projeye dahil edilmiştir. Veri setindeki her bir özellik ile hedef değişken arasındaki ilişkiyi ölçmek için karşılıklı bilgi katsayıları hesaplanmıştır. Bu katsayı, her bir özelliğin hedef değişken hakkında ne kadar bilgi sağladığını gösterir. Şekil 3.10'da her bir özelliğin hedef değişkenle olan karşılıklı bilgi katsayısı görsel olarak sunulmuştur.



Şekil 3.10 Karşılıklı bilgi katsayılarına göre sıralanmış özellikler

Şekil 3.10'da yüksek karşılıklı bilgi değerlerine sahip özellikler, hedef değişkenin tahminine katkı sağlama potansiyeli yüksek olan özellikler olarak öne çıkmaktadır. Bu şekil, hangi özelliklerin daha anlamlı ve bilgi taşıyan değişkenler olduğunu kolayca görmemizi sağlar.

Şekil 3.10 incelendiğinde, yüksek karşılıklı bilgi değerlerine sahip olan ve modelin karmaşıklığını gereksiz yere artırmayacak özelliklerin belirlenmesi amacıyla, eşik değerinin 0,4 ve üzerinde olması gerektiği görülmektedir. Ancak, 0,4'ün seçilmesi daha fazla özellik ile çalışmayı gerektireceğinden, bu durum modelin doğruluğunu olumsuz etkileyebilecek gereksiz bilgilerin modele dahil olmasına neden olabilir. Bu nedenle, daha dengeli bir seçim yapmak adına 0,4'ten biraz daha yüksek olan 0,42 değeri tercih edilmiştir. Bu değer, veri setindeki anlamlı ilişkileri korurken, gereksiz değişkenleri dışarıda bırakmak için uygun bir eşik değeri sunmaktadır. Bu sayede, modelin performansını artıracak yüksek karşılıklı bilgi değerlerine sahip özellikler seçilirken, veri setinin karmaşıklığı ve hesaplama maliyeti de optimal seviyede tutulmuştur. Belirlenen eşik değerinin altındaki özellikler analizden çıkarılmıştır. Seçilen özelliklerin listesi şekil 3.11'de gösterilmiştir.

Pkt Len Max	0.473742	Pkt Len Var	0.454727
Fwd Pkt Len Max	0.464602	Pkt Size Avg	0.453849
Bwd Pkt Len Std	0.464281	Init Bwd Win Byts	0.452391
Bwd Pkt Len Max	0.462319	Dst Port	0.452235
Pkt Len Std	0.458513	Subflow Bwd Byts	0.450595
Bwd Pkt Len Mean	0.457714	TotLen Bwd Pkts	0.450494
Bwd Seg Size Avg	0.457652	TotLen Fwd Pkts	0.449949
Fwd Pkt Len Std	0.457468	Subflow Fwd Byts	0.449941
Fwd Pkt Len Mean	0.457304	Bwd IAT Mean	0.422574
Fwd Seg Size Avg	0.457152	Bwd IAT Tot	0.421745
Pkt Len Mean	0.456348		

Şekil 3.11 0,42'den büyük karşılıklı bilgi değerine sahip değişkenlerin listesi

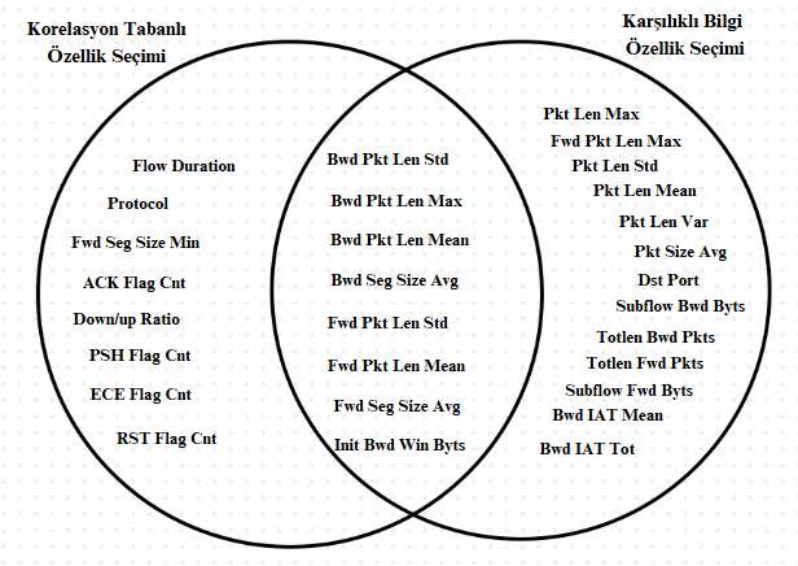
Çalışmada kullanılmak üzere listelenen yüksek karşılıklı bilgi katsayısına sahip 21 özelliğin seçilmesiyle yeni bir veri seti oluşturulmuştur. Bu veri seti, çalışmada kullanılan yöntemle aynı adı taşıyan "MI" ismiyle kaydedilmiştir.

3.5 Hibrit Yöntemler

Bu çalışma kapsamında iki farklı hibrit yöntem geliştirilmiştir. Geliştirilen hibrit yöntemler, üç farklı boyut indirgeme tekniğinin farklı sıralamalarla birleştirilmesiyle oluşturulmuştur. Bu kombinasyonlar, boyut indirgeme süreçlerinin performansını ve etkinliğini artırmayı hedeflemektedir. Her bir hibrit yöntem, tekniklerin farklı sıralama ve entegrasyon stratejilerine dayanarak oluşturulmuştur.

3.5.1 Hibrit1

Hibrit1 yönteminde, veri ön işleme sürecinde, korelasyon tabanlı yöntem ile seçilen 16 özellik ve karşılıklı bilgi yöntemiyle seçilen 21 özellik birleştirilerek, yeni bir özellik kümesi oluşturulmuştur. Özelliklerin birleşim kümesi şekil 3.12'de gösterilmiştir.



Şekil 3.12 "Korelasyon tabanlı + karşılıklı bilgi" yöntemleri ile seçilen özelliklerin birleşimi

Her iki yöntemde de ortak olarak 8 özellik bulunmaktadır. Yöntemlerin birleşimiyle toplamda 29 özellik elde edilmiştir. Elde edilen özelliklerin listesi şekil 3.13'te gösterilmiştir.

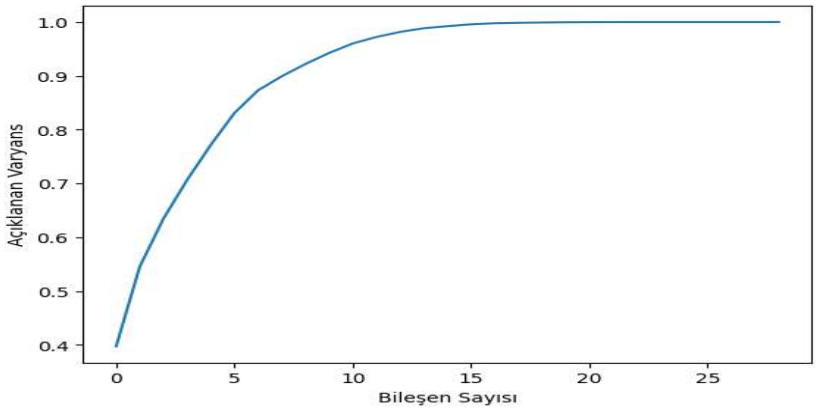
1	Flow Duration	16	RST Flag Cnt
2	Protocol	17	Pkt Len Max
3	Fwd Seg Size Min	18	Fwd Pkt Len Max
4	Bwd Pkt Len Std	19	Pkt Len Std
5	ACK Flag Cnt	20	Pkt Len Mean
6	Bwd Seg Size Avg	21	Pkt Len Var
7	Bwd Pkt Len Mean	22	Pkt Size Avg
8	Down/Up Ratio	23	Dst Port
9	Init Bwd Win Byts	24	Subflow Bwd Byts
10	Bwd Pkt Len Max	25	TotLen Bwd Pkts
11	Fwd Pkt Len Mean	26	TotLen Fwd Pkts
12	Fwd Seg Size Avg	27	Subflow Fwd Byts
13	PSH Flag Cnt	28	Bwd IAT Mean
14	Fwd Pkt Len Std	29	Bwd IAT Tot
15	ECE Flag Cnt		

Şekil 3.13 "Korelasyon tabanlı + karşılıklı bilgi" birleşimi sonucu oluşan özelliklerin listesi

Sonraki aşamada, PCA kullanılarak, birleştirilen bu yeni veri kümesinin boyutu daha da küçültülmüş ve en fazla varyansı açıklayan bileşenler seçilmiştir. PCA'da optimum bileşen sayısını belirlemek için açıklanan varyans değerleri şekil 3.14'te ve bileşen sayısına göre açıklanan varyans değerleri grafiği şekil 3.15'te verilmiştir.

```
array([0.39711036, 0.54537443, 0.63458821, 0.70641255, 0.7714983 ,
       0.83080047, 0.8730909 , 0.89905377, 0.92192636, 0.94246647,
       0.96009681, 0.97216779, 0.98157497, 0.98839739, 0.99230727,
       0.99570841, 0.99769349, 0.99853098, 0.9990848 , 0.99949184,
       0.99975297, 0.9998988 , 0.9999607 , 0.99999986, 0.99999996,
       1.        , 1.        , 1.        , 1.        ])
```

Şekil 3.14 Açıklanan varyans değerleri



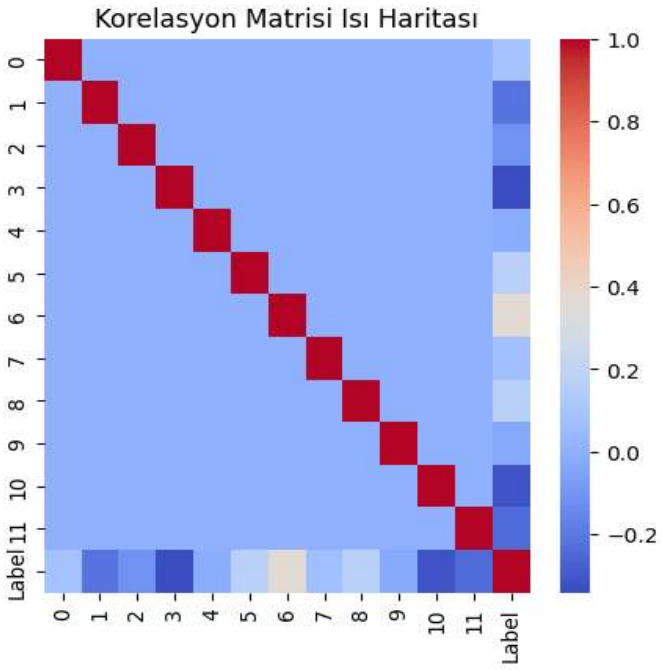
Şekil 3.15 Bileşen sayısına göre açıklanan varyans değerleri grafiği

Özellik seçimi uygulanmış veri setinde yalnızca bilgi yoğunluğu yüksek ve hedef değişkenle anlamlı ilişkiye sahip özellikler kaldığı için, bu veri setinde %99 varyans oranı tercih edilmiştir. Özellik seçimi aşamasında gereksiz veya tekrarlı bilgiler önceden elendiği için, PCA uygulandığında daha yüksek bir varyans oranının açıklanması mümkündür. Bu, anlamlı bilgiyi tamamen korurken modelin performansını artırmak için uygun bir seçim olmuştur. %99 varyans oranı, verinin büyük bir kısmını açıklarken, modelin doğruluğunu yükseltmek ve daha iyi genelleme sağlamak adına optimum bir dengeyi sunmaktadır. Açıklanan varyans değerleri ve grafik incelendiğinde, özelliklerin toplam %99'unu temsil edecek bileşen sayısının 15 olduğu belirlenmiştir. Bu 15 bileşen ile yeni bir veri seti oluşturulmuş ve bu veri seti "Hibrit1" adıyla kaydedilmiştir.

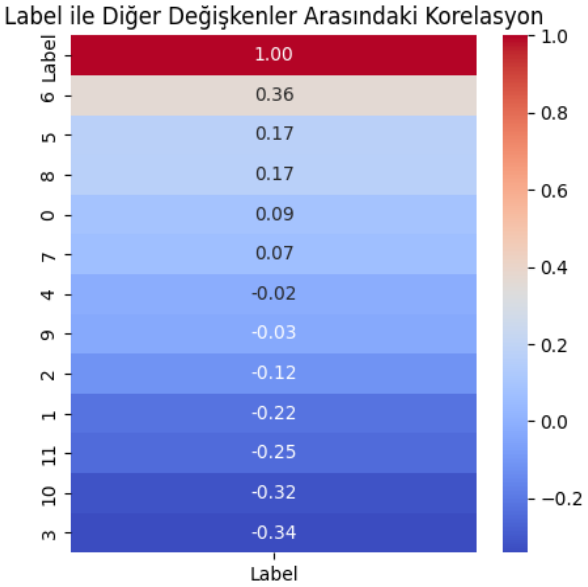
3.5.2 Hibrit2

Hibrit2 yönteminde, veri ön işleme aşamasında PCA ile seçilen 12 özellik üzerinde korelasyon tabanlı ve karşılıklı bilgi yöntemleri ayrı ayrı uygulanmış ve ardından bu yöntemlerin birleşimiyle yeni bir özellik kümesi oluşturulmuştur.

Korelasyon tabanlı özellik seçimi için korelasyon matrisi hesaplanmış ve ısı haritası oluşturulmuştur. Oluşturulan ısı haritası şekil 3.16'da gösterilmiştir. Veri setindeki tüm bağımsız değişkenlerin hedef değişken ile korelasyon katsayıları hesaplanmış ve şekil 3.17'de gösterilmiştir.



Şekil 3.16 Isı haritası



Şekil 3.17 Label ile diğer değişkenler arası korelasyon

Isı haritası ve hedef değişkenle diğer değişkenler arasındaki korelasyon incelendiğinde, özellik seçimi için 0,2'lik bir eşik değeri belirlenmiştir. Bu eşik değeri, modelin doğruluğunu artıracak güçlü korelasyona sahip değişkenleri seçerken, gereksiz karmaşıklığı önleyerek performansı optimize etmeyi amaçlamaktadır. 0,2'lik eşik

değerinin altında kalan değişkenler analizden çıkarılmış ve hedef değişkenle güçlü korelasyon gösteren bağımsız değişkenlerin listesi Şekil 3.18'de gösterilmiştir. Korelasyon tabanlı özellik seçimi sonucunda 5 özellik seçilmiştir.

```
6     0.361931
3     0.340725
10    0.315429
11    0.246297
1     0.219934
```

Şekil 3.18 0,2'den büyük korelasyona sahip değişkenlerin listesi

Karşılıklı bilgi özellik seçimi için veri setindeki özellikler ile hedef değişken arasındaki karşılıklı bilgi katsayıları hesaplanmıştır. Hesaplanan bu değerlere göre özellikler şekil 3.19'da sıralanmıştır.

Şekil 3.19 Karşılıklı bilgi katsayılarına göre sıralanmış özellikler

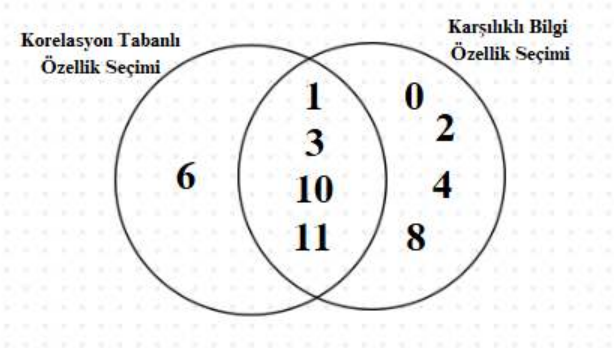
Şekil 3.19 incelendiğinde, tüm özelliklerin karşılıklı bilgi katsayısı 0,4'ün üzerindedir. Bu yüzden eşik değerinin 0,4'ün üzerinde belirlenmesi gerekmektedir. Bu nedenle, daha dengeli bir seçim yapmak adına 0,45 olarak belirlenmiştir. Bu değer, veri setindeki anlamlı ilişkileri korurken, gereksiz değişkenleri dışarıda bırakmak için uygun bir eşik değeri sunmaktadır. Bu sayede, modelin performansını artıracak yüksek karşılıklı bilgi değerlerine sahip özellikler seçilirken, veri setinin karmaşıklığı ve hesaplama maliyeti

de optimal seviyede tutulmuştur. Belirlenen eşik değerinin altındaki özellikler analizden çıkarılmıştır. Seçilen özelliklerin listesi şekil 3.20'de gösterilmiştir. Karşılıklı bilgi özellik seçimi sonucunda 8 özellik seçilmiştir.

0	0.460592
11	0.456607
2	0.455368
4	0.454758
8	0.452559
10	0.451885
3	0.451102
1	0.451069

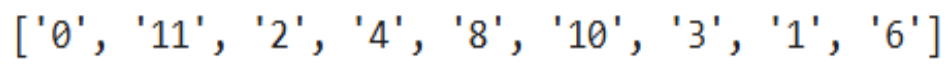
Şekil 3.20 0,45'den büyük karşılıklı bilgi katsayısına sahip değişkenlerin listesi

Korelasyon tabanlı yöntemden elde edilen 5 özellik ve karşılıklı bilgi yönteminden elde edilen 8 özellik bir araya getirilerek, yeni ve daha kapsamlı bir özellik kümesi oluşturulmuştur. Bu birleşime ilişkin detaylar Şekil 3.21'de gösterilmektedir.



Şekil 3.21 "Korelasyon tabanlı + karşılıklı bilgi" yöntemleri ile seçilen özelliklerin birleşimi

Her iki yöntemde de ortak olarak 4 özellik bulunmaktadır. Yöntemlerin birleşimiyle toplamda 9 özellik elde edilmiştir. Elde edilen özelliklerin listesi şekil 3.22'de gösterilmiştir. Bu 9 özellik ile yeni bir veri seti oluşturulmuş ve bu veri seti "Hibrit2" adıyla kaydedilmiştir.



Şekil 3.22 "Korelasyon tabanlı + karşılıklı bilgi" birleşimi sonucu oluşan özelliklerin listesi

3.6 Hedef Değişken ve Bağımsız Değişkenin Belirlenmesi

Saldırı durumunun olup olmaması bizim hedef değişkenimizdir. Bu yüzden Label değişkeni hedef değişken olarak belirlenmiştir. Geriye kalan değişkenler de bağımsız değişkendir.

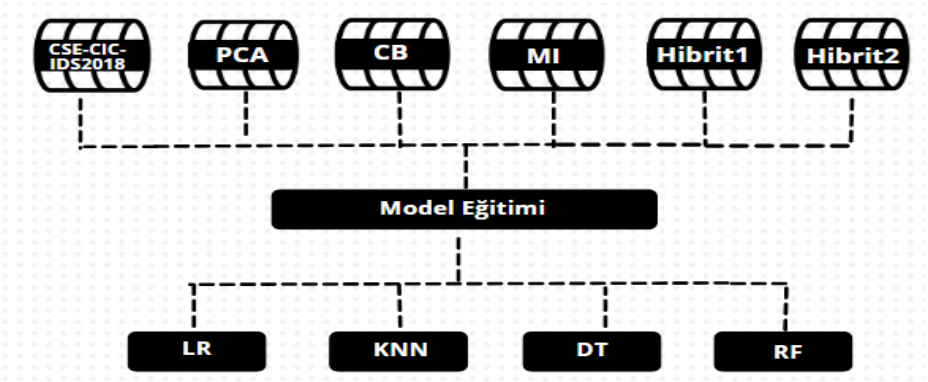
4. BULGULAR

Bu bölümde, boyut indirgeme yöntemleri uygulanmadan oluşturulan veri seti ile boyut indirgeme yöntemleri uygulanarak oluşturulan "PCA", "CB", "MI", "Hibrit1" ve "Hibrit2" veri setleri, farklı makine öğrenmesi algoritmaları ile kapsamlı bir şekilde test edilmiştir. Çalışmada, model eğitimi için Şekil 4.1'de gösterilen LR, KNN, DT ve RO algoritmaları kullanılmış; her bir modelin sınıflandırma performansı doğruluk, kesinlik, duyarlılık, f1 skor ve eğitim süreleri açısından analiz edilmiştir.

Elde edilen sonuçlar, aşağıdaki başlıklar altında değerlendirilmiştir:

- Boyut indirgeme yöntemleri ve hibrit yaklaşımların başarı oranlarını ve eğitim sürelerini nasıl etkilediği,
- Boyut indirgeme uygulanmayan veri seti ile boyut indirgeme sonrası oluşturulan veri setlerinin doğruluk, hız ve genel performans bakımından karşılaştırmalı analizleri.

Bulgular, her bir veri seti ve algoritma kombinasyonu için detaylı tablolar ve grafikler eşliğinde sunulmuştur. Tablolar, kullanılan özellik sayıları, doğruluk, kesinlik, duyarlılık, f1 skor ve eğitim süresi ölçütlerini kapsamış; grafiklerle bu sonuçların görsel olarak yorumlanması sağlanmıştır. Bu analizler, hem boyut indirgeme yöntemlerinin hem de kullanılan algoritmaların sistem performansı üzerindeki etkisini daha iyi anlamayı mümkün kılmıştır.



Şekil 4.1 Veri setlerinin eğitim süreci

4.1 Makine Öğrenmesi Algoritmalarının Performans Karşılaştırması

Kullanılan her bir makine öğrenmesi yöntemi için performans değerlendirme ölçütleri detaylı bir şekilde analiz edilmiştir. Daha tutarlı ve güvenilir bir model oluşturabilmek amacıyla k-kat çapraz doğrulama yöntemi uygulanmıştır. Geliştirilen sistemlerde, hibrit yöntem hariç, "scikit-learn" kütüphanesi içerisindeki sınıflandırıcılar varsayılan parametrelerle kullanılmıştır. Ancak, hibrit yöntemle elde edilen veri setlerinde model performansını artırmak için hiper parametre ayarlamaları gerçekleştirilmiştir. Bu süreçte, grid search tekniği kullanılarak en uygun hiperparametre değerleri belirlenmiştir. Tüm modellerde 10 kat çapraz doğrulama tekniği kullanılarak modelin başarısı; doğruluk, kesinlik, duyarlılık, f1 skor ve eğitim süresi ölçütleri üzerinden değerlendirilmiştir. Eğitim süreleri sn cinsinden hesaplanmıştır.

4.1.1 Lojistik Regresyon

LR algoritması ile sınıflandırma işleminin gerçekleştirilebilmesi için "from sklearn.linear_model import LogisticRegression" kütüphanesi projeye dahil edilmiştir. LogisticRegression sınıfı kullanılarak model oluşturulmuş ve eğitilmiştir. LR modelinin, boyut indirgeme yöntemleri uygulanmadan önceki performansıyla boyut indirgeme yöntemleri uygulandıktan sonraki performanslarının karşılaştırmalı sonuçları analiz edilmiştir. Hibrit yöntemler için gerçekleştirilen hiper parametre ayarlamalarının sonuçları çizelge 4.1'de gösterilmektedir.

Çizelge 4.1 Hiper parametre optimizasyon işlemi konfigürasyonu ve sonuçları

Hiper parametre	Varsayılan Değer	Optimizasyon Aralığı	En İyi Değer	
			Hibrit1	Hibrit2
C	1.0	[0.1,1,10,100]	100	10

Çizelge 4.2'de LR modeli için boyut indirgeme yöntemleri uygulanmadan ve uygulandıktan sonra elde edilen sonuçlar yer almaktadır. Bu bulgular, modelin sınıflandırma performansını ve eğitim süresini detaylı bir şekilde göstermektedir.

Çizelge 4.2 LR modeli sonuçları

Yöntem	Özellik Sayısı	Doğruluk	Kesinlik	Duyarlılık	F1	Eğitim Süresi (sn)
Yok	78	0,9981	0,9986	0,9906	0,9945	37
PCA	12	0,9961	0,9893	0,9889	0,9891	17
CB	16	0,9975	0,9969	0,9890	0,9930	15
MI	22	0,9967	0,9980	0,9832	0,9906	23
Hibrit1	15	0,9978	0,9984	0,9891	0,9937	20
Hibrit2	9	0,9903	0,9840	0,9609	0,9723	17

Boyut indirgeme yöntemi uygulanmadığında, model %99,81 doğruluk ve %99,45 f1 skoru ile en yüksek performansı göstermiştir. Ancak, bu yöntemle eğitim süresi 37 sn olarak ölçülmüş ve diğer yöntemlere kıyasla oldukça yüksek kalmıştır. Bu durum, fazla sayıda özelliğin eğitim süresini artırdığına işaret etmektedir.

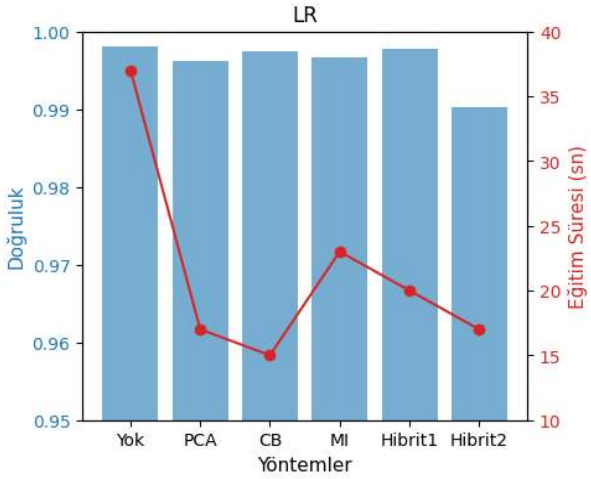
PCA yönteminde, modelin doğruluk oranı %99,61'e düşerken, eğitim süresi 17 sn ile önemli ölçüde kısalmıştır. Bu, daha az sayıda özelliğin eğitim süresini kısalttığını göstermektedir.

CB yöntemi, %99,75 doğruluk ve %99,30 f1 skoru ile hem yüksek bir performans hem de 15 sn'lik eğitim süresi ile en düşük işlem süresini sunmuştur. CB yöntemi, doğruluk ve işlem süresi arasında etkili bir denge kurmaktadır.

MI yöntemi, %99,67 doğruluk ve %99,06 f1 skoru elde etmiştir. Eğitim süresi 23 sn olarak ölçülmüş olup, bu diğer yöntemlere kıyasla daha uzun bir süreyi ifade etmektedir. Ancak MI yöntemi, doğruluk oranını yüksek tutarak işlem süresini optimize etmiştir.

Hibrit1 yöntemi, %99,78 doğruluk ve %99,37 f1 skoru ile yüksek bir performans sunmuştur. Eğitim süresi 20 sn olarak ölçülmüştür. Hibrit1 yöntemi, doğruluk oranını koruyarak işlem süresini optimize etmesiyle dikkat çekmektedir.

Hibrit2 yöntemi, %99,03 doğruluk ve %97,23 f1 skoru ile diğer yöntemlere kıyasla daha düşük bir performans sergilemiştir. Ancak, yalnızca 9 özellik ile 17 sn'lik eğitim süresi sunarak işlem süresini önemli ölçüde azaltmıştır. Ancak bu durum, özelliklerin azalmasıyla performansın sınırlanabileceğini göstermektedir.



Şekil 4.2 Model başarısı ve eğitim süresi karşılaştırması

Şekil 4.2'deki grafik incelendiğinde, boyut indirgeme yöntemleri uygulanmadığında doğruluk oranı en yüksek seviyeye ulaşsa da eğitim süresi en uzun olarak ölçülmüştür. Gerçek zamanlı sistemlerde işlem hızının kritik öneme sahip olduğu düşünüldüğünde, bu durum bazı boyut indirgeme yöntemlerini daha tercih edilebilir hale getirmektedir. LR modelinde boyut indirgeme yöntemlerinin işlem süresini önemli ölçüde azalttığı ve modelin performansını büyük oranda koruduğu görülmüştür.

Sonuç olarak Hibrit1 yöntemi, yüksek doğruluk oranı ve düşük işlem süresi ile dengeli bir çözüm sunarak, gerçek zamanlı sistemler için daha uygun bir alternatif olduğunu ortaya koymaktadır.

4.1.2 K En Yakın Komşu Algoritması

KNN algoritması ile sınıflandırma işleminin gerçekleştirilebilmesi için "from sklearn.neighbors import KNeighborsClassifier" kütüphanesi pojeye dahil edilmiştir. KNeighborsClassifier sınıfı kullanılarak model oluşturulmuş ve eğitilmiştir. KNN

modelinin, boyut indirgeme yöntemleri uygulanmadan önceki performansıyla boyut indirgeme yöntemleri uygulandıktan sonraki performanslarının karşılaştırmalı sonuçları analiz edilmiştir. Hibrit yöntemler için gerçekleştirilen hiper parametre ayarlamalarının sonuçları çizelge 4.3'te gösterilmektedir.

Çizelge 4.3 Hiper parametre optimizasyon işlemi konfigürasyonu ve sonuçları

Hiper Parametre	**Varsayılan Değer**	**Optimizasyon Aralığı**	**En İyi Değer**	
			Hibrit1	**Hibrit2**
n_neighbors	5	[3, 5, 7, 9, 11]	11	3
metric	minkowski	['euclidean','manhattan', 'minkowski']	euclidean	manhattan
weights	uniform	['uniform', 'distance']	uniform	distance
p	2	[1, 2]	1	1

Çizelge 4.4'te KNN modeli için boyut indirgeme yöntemleri uygulanmadan ve uygulandıktan sonra elde edilen sonuçlar yer almaktadır. Bu bulgular, modelin sınıflandırma performansını ve eğitim süresini detaylı bir şekilde göstermektedir.

Çizelge 4.4 KNN modeli sonuçları

Yöntem	**Özellik Sayısı**	**Doğruluk**	**Kesinlik**	**Duyarlılık**	**F1**	**Eğitim Süresi (sn)**
Yok	78	0,9999	0,9998	0,9999	0,9998	1246
PCA	12	0,9999	0,9997	0,9999	0,9998	71
CB	16	0,9980	0,9924	0,9967	0,9945	805
MI	22	0,9996	0,9998	0,9981	0,9990	807
Hibrit1	15	0,9997	0,9993	0,9993	0,9993	78
Hibrit2	9	0,9999	0,9998	0,9999	0,9999	31

Boyut indirgeme yöntemi uygulanmadığında, model %99,99 doğruluk ve %99,98 f1 skoru ile yüksek performans elde etmiş olsa da, eğitim süresi 1246 sn ile açık ara en uzun süre olarak ölçülmüştür. Bu durum, fazla sayıda özelliğin eğitim süresini önemli

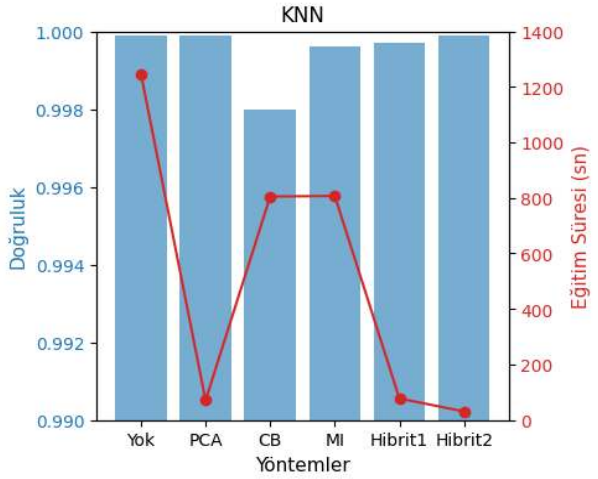
ölçüde artırdığını göstermektedir ve gerçek zamanlı sistemler için dezavantaj oluşturabilir.

PCA yöntemi, özellik sayısını 12'ye indirerek eğitim süresini 71 sn'ye düşürmüştür. Doğruluk (%99,99) ve f1 skoru (%99,98) üzerinde hiçbir performans kaybı yaşanmadığı gözlemlenmiştir. Bu durum, PCA yönteminin doğruluk ve eğitim süresi açısından dengeli ve verimli bir çözüm sunduğunu göstermektedir.

CB yöntemi ile eğitim süresi 805 sn'ye düşürülmüş, ancak doğruluk oranı %99,80'e ve f1 skoru %99,45'e düşmüştür. Bu yöntem, eğitim süresini azaltmakla birlikte doğruluk ve f1 skorunda düşüşe neden olmuştur.

MI yöntemi, %99,96 doğruluk ve %99,90 f1 skoru ile CB yöntemine göre yüksek bir performans sergilemiş ve eğitim süresini 807 sn'ye düşürmüştür. CB yöntemine kıyasla daha iyi doğruluk oranı sunmakla birlikte, işlem süresi bakımından benzer sonuç elde etmiştir.
Hibrit1 yöntemi, %99,97 doğruluk ve %99,93 f1 skoru elde etmiştir. Eğitim süresi ise 78 sn ile oldukça kısalmıştır. Bu durum, hibrit yöntemin doğruluk oranında minimal bir düşüşle eğitim süresi açısından dengeli bir çözüm sunduğunu göstermektedir.

Hibrit2 yöntemi, yalnızca 9 özellik ile %99,99 doğruluk ve %99,99 f1 skoru elde etmiş, performansı korurken 31 sn ile en düşük eğitim süresini sunmuştur. Bu sonuçlar, Hibrit2 yönteminin hem doğruluk hem de hız açısından en etkili yöntem olduğunu göstermektedir.



Şekil 4.3 Model başarısı ve eğitim süresi karşılaştırması

Şekil 4.3'teki grafik incelendiğinde, boyut indirgeme yöntemleri uygulanmadığında doğruluk oranı yüksek seviyeye ulaşsa da eğitim süresi en uzun olarak ölçülmüştür. Bu durum, fazla sayıda özelliğin işlem süresini artırarak gerçek zamanlı sistemlerde kullanılabilirliğini sınırladığı ve işlem hızının kritik öneme sahip olduğu durumlarda boyut indirgeme yöntemlerini daha tercih edilebilir hale getirdiğini göstermektedir. KNN modelinde boyut indirgeme yöntemlerinin işlem süresini önemli ölçüde azalttığı ve model performansını koruduğu gözlemlenmiştir.

Sonuç olarak, Hibrit2 yöntemi, hem yüksek doğruluk oranını koruması hem de işlem süresini minimize etmesiyle gerçek zamanlı uygulamalar için en uygun yöntem olarak öne çıkmaktadır.

4.1.3 Karar Ağacı

DT algoritması ile sınıflandırma işleminin gerçekleştirilebilmesi için "from sklearn.tree import DecisionTreeClassifier" kütüphanesi pojeye dahil edilmiştir. DecisionTreeClassifier sınıfı kullanılarak model oluşturulmuş ve eğitilmiştir. DT modelinin, boyut indirgeme yöntemleri uygulanmadan önceki performansıyla boyut indirgeme yöntemleri uygulandıktan sonraki performanslarının karşılaştırmalı sonuçları analiz edilmiştir. Hibrit yöntemler için gerçekleştirilen hiper parametre ayarlamalarının sonuçları çizelge 4.5'te gösterilmektedir.

Çizelge 4.5 Hiper parametre optimizasyon işlemin konfigürasyonu ve sonuçları

Hiper Parametre	Varsayılan Değer	Optimizasyon Aralığı	En İyi Değer	
			Hibrit1	Hibrit2
max_depth	None	[10,15]	15	15
min_samples_split	2	[5,10]	10	5
min_samples_leaf	1	[2,5]	2	2
max_features	None	['sqrt', 'log2']	log2	sqrt

Çizelge 4.6'da DT modeli için boyut indirgeme yöntemleri uygulanmadan ve uygulandıktan sonra elde edilen sonuçlar yer almaktadır. Bu bulgular, modelin sınıflandırma performansını ve eğitim süresini detaylı bir şekilde göstermektedir.

Çizelge 4.6 DT modeli sonuçları

Yöntem	Özellik Sayısı	Doğruluk	Kesinlik	Duyarlılık	F1	Eğitim Süresi (sn)
Yok	78	0,9999	0,9998	0,9999	0,9999	64
PCA	12	0,9999	0,9997	0,9998	0,9998	91
CB	16	0,9981	0,9990	0,9904	0,9947	17
MI	22	0,9996	0,9999	0,9982	0,9990	20
Hibrit1	15	0,9997	0,9992	0,9992	0,9992	24
Hibrit2	9	0,9999	0,9998	0,9997	0,9997	25

Boyut indirgeme yöntemi uygulanmadığında model, %99,99 doğruluk ve %99,99 f1 skoru ile yüksek performans göstermiştir. Ancak eğitim süresi 64 sn ile diğer yöntemlere kıyasla daha uzundur. Bu durum, boyut indirgeme yapılmadığında daha fazla özelliğin işlem süresini artırabileceğini göstermektedir.

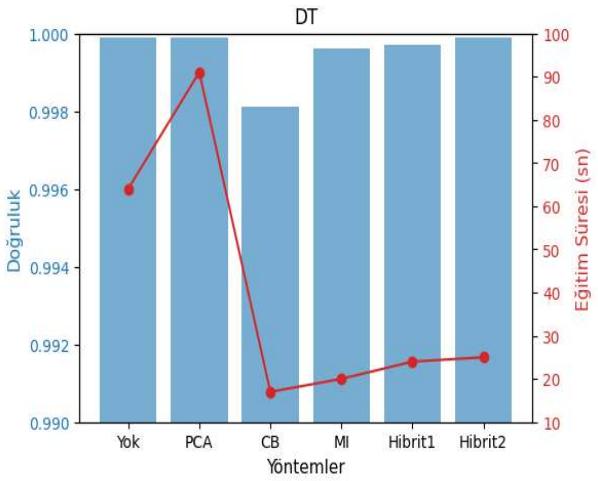
PCA yönteminde, doğruluk ve f1 skorunda (%99,99 ve %99,98) neredeyse hiçbir kayıp yaşanmamıştır. Ancak, eğitim süresi 91 sn'ye çıkarak boyut indirgeme yöntemlerinden beklenen süreden daha uzun bir işlem süresi göstermiştir.

CB yöntemi, doğruluk oranında hafif bir düşüşe (%99,81) ve f1 skorunda azalmaya (%99,47) neden olmuş, ancak eğitim süresini 17 sn ile en düşük seviyeye indirmiştir. Eğitim süresindeki azalma, CB yönteminin avantajını ortaya koymaktadır.

MI yöntemi, %99,96 doğruluk ve %99,90 f1 skoru ile yüksek bir performans sergilemiş, eğitim süresini ise 20 sn'ye indirerek işlem süresini optimize etmiştir. Bu yöntem, işlem süresi ve doğruluk arasında dengeli bir sonuç sağlamıştır.

Hibrit1 yöntemi, model %99,97 doğruluk ve %99,92 f1 skoru ile iyi bir performans sunmuş, eğitim süresini 24 sn'ye indirerek işlem süresi ve doğruluk arasında dengeli bir sonuç sağlayarak etkili bir alternatif olarak öne çıkmaktadır.

Hibrit2 yöntemi, yalnızca 9 özellik ile %99,99 doğruluk ve %99,97 f1 skoru elde etmiş, performansı koruduğu gibi eğitim süresini 25 sn'ye indirmiştir. Bu sonuçlar, Hibrit2 yönteminin hem doğruluk hem de hız açısından en etkili yöntem olduğunu göstermektedir.



Şekil 4.4 Model başarısı ve eğitim süresi karşılaştırması

Çizelge 4.4'daki sonuçlar incelendiğinde, boyut indirgeme yöntemleri uygulanmadığında modelin doğruluk oranı en yüksek seviyeye ulaşmış olsa da eğitim süresi en uzun olarak ölçülmüştür. Gerçek zamanlı sistemlerde işlem hızının kritik öneme sahip olduğu düşünüldüğünde, bu durum boyut indirgeme yöntemlerini daha

tercih edilebilir hale getirmektedir. DT modelinde boyut indirgeme, eğitim süresini düşürmüş ve model performansını büyük ölçüde korumuştur.

Sonuç olarak Hibrit2 yöntemi, hem yüksek doğruluk oranını koruması hem de işlem süresini minimize etmesiyle gerçek zamanlı uygulamalar için en uygun yöntem olarak öne çıkmaktadır.

4.1.4 Rastgele Orman

RF algoritması ile sınıflandırma işleminin gerçekleştirilebilmesi için "from sklearn.ensemble import RandomForestClassifier" kütüphanesi pojeye dahil edilmiştir. RandomForestClassifier sınıfı kullanılarak model oluşturulmuş ve eğitilmiştir. RF modelinin, boyut indirgeme yöntemleri uygulanmadan önceki performansıyla boyut indirgeme yöntemleri uygulandıktan sonraki performanslarının karşılaştırmalı sonuçları analiz edilmiştir. Hibrit yöntemler için gerçekleştirilen hiper parametre ayarlamalarının sonuçları çizelge 4.5'de gösterilmektedir.

Çizelge 4.7 Hiper parametre optimizasyon işlemin konfigürasyonu ve sonuçları

Hiper Parametre	**Varsayılan Değer**	**Optimizasyon Aralığı**	**En İyi Değer**	
			Hibirt1	**Hibrit2**
max_depth	None	[10,20]	10	20
min_samples_split	2	[5,10]	10	5
min_samples_leaf	1	[2,4]	2	2
n_estimators	100	[50,70]	70	50

Çizelge 4.8'de RF modeli için boyut indirgeme yöntemleri uygulanmadan ve uygulandıktan sonra elde edilen sonuçlar yer almaktadır. Bu bulgular, modelin sınıflandırma performansını ve eğitim süresini detaylı bir şekilde göstermektedir.

Çizelge 4.8 RF modeli sonuçları

Yöntem	Özellik Sayısı	Doğruluk	Kesinlik	Duyarlılık	F1	Eğitim Süresi (sn)
Yok	78	0,9999	0,9999	0,9996	0,9998	864
PCA	12	0,9999	0,9999	0,9998	0,9999	1376
CB	16	0,9981	0,9990	0,9905	0,9947	366
MI	22	0,9996	0,9999	0,9982	0,9990	431
Hibrit1	15	0,9997	0,9994	0,9992	0,9993	247
Hibrit2	9	0,9999	0,9999	0,9998	0,9998	176

Boyut indirgeme yöntemi uygulanmadığında model, %99,99 doğruluk ve %99,98 f1 skoru ile yüksek performans göstermiştir. Ancak, eğitim süresi 864 sn ile diğer yöntemlere kıyasla oldukça uzundur. Bu durum, fazla sayıda özelliğin işlem süresini artırarak gerçek zamanlı sistemlerde kullanılabilirliği sınırladığını göstermektedir.

PCA yönteminde, modelin doğruluk ve f1 skoru değerleri korunmuştur. Ancak eğitim süresi beklenenden daha uzun, 1376 sn olarak ölçülmüştür. Bu durum, PCA'nın RF modeli ile uyumunda işlem süresini artırabileceğini göstermektedir.

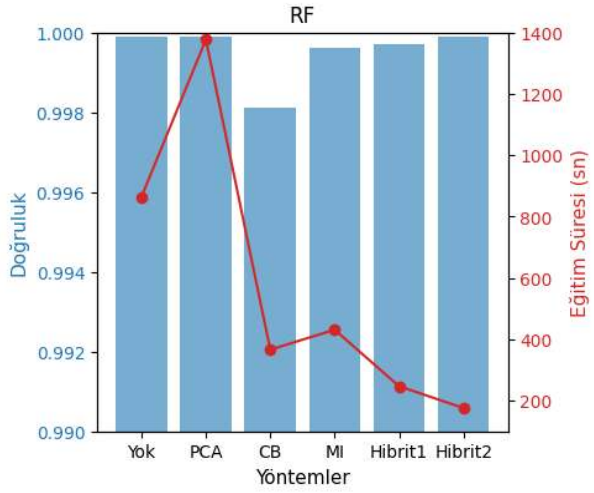
CB yöntemi, %99,81 doğruluk ve %99,47 f1 skoru ile daha düşük bir performans sergilemiş, ancak eğitim süresini 366 sn'ye düşürerek hız açısından önemli bir avantaj sağlamıştır.

MI yöntemi, %99,96 doğruluk ve %99,90 f1 skoru ile yüksek performansını korurken eğitim süresini 431 sn'ye indirmiştir. Bu yöntem, doğruluk ve işlem süresi açısından dengeli bir çözüm sunmaktadır.

Hibrit1 yöntemi, %99,97 doğruluk ve %99,93 f1 skoru elde etmiş, eğitim süresini 247 sn'ye düşürerek hem performans hem de işlem süresi açısından etkili bir alternatif olmuştur.

Hibrit2 yöntemi, sadece 9 özellik ile %99,99 doğruluk ve %99,98 f1 skoru elde etmiş, performansı koruduğu gibi eğitim süresini 176 sn ile en düşük seviyeye indirmiştir. Bu

sonuçlar, Hibrit2 yönteminin hem doğruluk hem de hız açısından en etkili yöntem olduğunu göstermektedir.



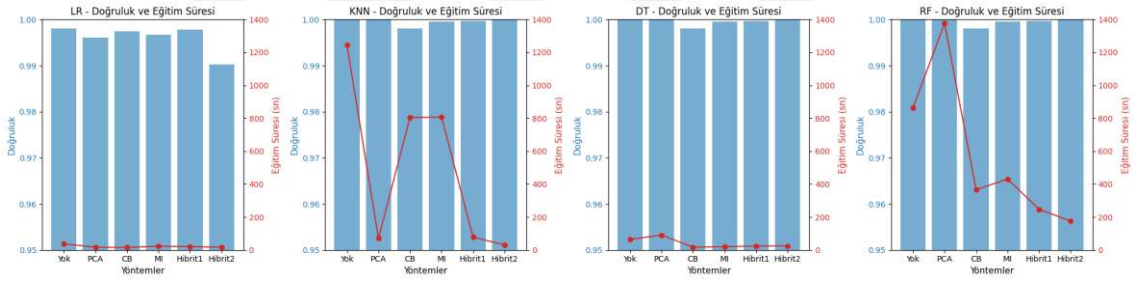
Şekil 4.5 Model başarısı ve eğitim süresi karşılaştırması

Şekil 4.5'teki grafik incelendiğinde, boyut indirgeme yöntemleri uygulanmadığında doğruluk oranı en yüksek seviyeye ulaşmış olsa da eğitim süresi diğer yöntemlere kıyasla oldukça uzun ölçülmüştür. Gerçek zamanlı sistemlerde işlem hızının kritik bir gereklilik olduğu düşünüldüğünde, bu durum bazı boyut indirgeme yöntemlerini daha avantajlı hale getirmektedir. RF modelinde boyut indirgeme yöntemleri, eğitim süresini belirgin bir şekilde azaltarak işlem hızını artırmış ve model performansını büyük ölçüde korumuştur.

Sonuç olarak Hibrit2 yöntemi, hem yüksek doğruluk oranını koruması hem de işlem süresini minimize etmesiyle gerçek zamanlı uygulamalar için en uygun yöntem olarak öne çıkmaktadır.

5. TARTIŞMA ve SONUÇ

Bu çalışmada, botnet saldırılarının tespiti için boyut indirgeme yöntemleri ve makine öğrenmesi algoritmaları kullanılarak hem doğruluk oranını hem de sınıflandırma hızını optimize etmeyi amaçlayan anomali tabanlı saldırı tespit sistemi geliştirilmiştir. Çalışma kapsamında, PCA, CB ve MI gibi geleneksel boyut indirgeme yöntemleri ve bu yöntemlerin güçlü yönlerini birleştiren hibrit yöntemler geliştirilerek boyut indirgemenin model üzerindeki etkileri detaylı bir şekilde analiz edilmiştir. Elde edilen bulgular, farklı boyut indirgeme yöntemlerinin ve hibrit yaklaşımların performans üzerindeki etkilerini ortaya koymuştur. Bu bölümde, sonuçlar literatürle karşılaştırılmış, yöntemlerin güçlü ve zayıf yönleri tartışılmış ve gelecekte yapılacak çalışmalara yönelik önerilerde bulunulmuştur. Genel sonuçlar şekil 5.1'de gösterilmektedir.



Şekil 5.1 Genel sonuçlar

Sonuçlar, boyut indirgeme yöntemlerinin model performansını hem doğruluk hem de işlem süresi açısından önemli ölçüde etkileyebileceğini ortaya koymaktadır. Boyut indirgeme yapılmadan kullanılan veri seti, doğruluk oranlarında genellikle yüksek performans sağlamış olsa da, işlem süreleri oldukça uzundur. Yapılan çalışmalar, boyut indirgeme yöntemlerinin model performansını koruyarak işlem süresini kısalttığını göstermektedir.

Hibrit yöntemler, diğer boyut indirgeme yöntemlerine kıyasla daha yüksek doğruluk oranı ve daha düşük eğitim süresi sağlamaktadır. Bu; PCA, MI ve CB yöntemlerinin birlikte kullanılmasıyla elde edilen avantajların bir kanıtıdır. Hibrit yöntemlerin başarısı, PCA, CB ve MI yöntemlerinin güçlü yönlerini birleştirerek önemli özelliklerin daha etkili bir şekilde seçilmesi ve gereksiz özelliklerin elenmesine dayanmaktadır. Özellikle

PCA'nın doğruluk oranını yüksek tutma avantajı ve CB'nin işlem süresini optimize etme kabiliyeti, hibrit yöntemlerle birleştirilmiştir. Bu yöntemler, doğruluk oranını korurken işlem süresini önemli ölçüde azaltarak, gerçek zamanlı saldırı tespit sistemleri için ideal bir çözüm sunmaktadır. Özellikle Hibrit2 yaklaşımı model performansını koruyarak ve işlem süresini optimize etmesinden dolayı, gerçek dünya uygulamalarında hızlı ve doğru botnet saldırı tespiti için etkili bir çözüm sunmaktadır.

Çalışmanın etkinliği, literatürde aynı veri seti üzerinde yapılmış olan önceki çalışmalarla karşılaştırılarak değerlendirilmiştir. Literatürdeki diğer çalışmalarla yapılan bu karşılaştırma, çalışmanın güçlü ve zayıf yönlerini ortaya koymak amacıyla çizelge 5.1'de sunulmuştur. Bu çizelge, doğruluk oranları ve işlem sürelerinin yanı sıra, her bir çalışmanın kullandığı yöntemler ve veri setleri hakkında da bilgi vermektedir, böylece yapılan çalışmanın literatürdeki yeri daha net bir şekilde anlaşılmaktadır.

Çizelge 5.1 Literatür karşılaştırması

Yazarlar	**Veri Seti**	**Algoritmalar**	**Boyut İndirgeme**	**Doğruluk**	**Zaman**
Kanimozhi ve Jacob (2019)	CSE-CIC-IDS2018	**ANN**	-	%99,97	-
Karaman vd. (2020)	CSE-CIC-IDS2018	**ANN**	BruteForce	%93,23	-
Kanimozhi ve Jacob (2021)	CSE-CIC-IDS2018	**ANN**, KNN, SVM, Adaboost, NB, RF	-	%99,9	-
Altunay ve Albayrak (2021)	CSE-CIC-IDS2018	CNN	-	%98,9	-

Çizelge 5.1 (Devam) Literatür karşılaştırması

Yazarlar	Veri Seti	Algoritmalar	Boyut İndirgeme	Doğruluk	Zaman
Odeyemi vd. (2023)	CSE-CIC-IDS2018	NB, DT, KNN, LR, ANN, Quadratic Discriminant Analysis ve SVM, Adaboost, Gradient Boosting, **RF** ve Max Voting	-	%99,89	-
Ertuğrul (2024)	CSE-CIC-IDS2018	KNN, DT, NB, **RF**, Gradyan Artırma, AdaBoost, LR ve Doğrusal Ayrımcılık Analizi, Çok Katmanlı Algılayıcı	SelectKBest, MI, Ki-Kare	%99,83	-
Bu çalışma	CSE- CIC-IDS2018	LR, KNN, **DT**, RF	PCA, CB, MI, Hibrit1,**Hibrit2**	%99,99	24 sn

Algoritmalar sütununda **kalın yazı tipi** ile belirtilen yöntem, her bir algoritma için en başarılı yöntemdir.

Literatürdeki çalışmalar, genellikle model doğruluk oranlarını ve veri setleri üzerindeki başarımlarını karşılaştırmaya odaklanmışken, bu tez hem doğruluk hem de zaman bazlı bir inceleme yaparak daha kapsamlı bir analiz sunmaktadır. Bu çalışmada zaman performansının dahil edilmesi, modellerin gerçek zamanlı sistemlere uygulanabilirliği açısından kritik bir veri sunmaktadır. Literatürdeki çalışmalar incelendiğinde, en yüksek başarı oranının Kanimozhi ve Jacob (2019) tarafından yapılan çalışmada elde edildiği görülmektedir. Bu çalışmada, tüm özellikler kullanılarak %99,97 doğruluk oranı sağlanmıştır. Bununla birlikte, önerilen Hibrit2 yöntemi ile sadece 9 özellik kullanılarak, DT algoritması ile %99,99 doğruluk oranına ulaşılmış ve eğitim süresi 24 sn olarak belirlenmiştir. Bu sonuç, literatürdeki en yüksek başarı olma özelliğine

sahiptir ve daha az sayıda özellik kullanılarak yüksek doğruluk oranlarına ulaşılabileceğini ve işlem sürelerinin önemli ölçüde kısaltılabileceğini göstermektedir.

Çalışma ayrıca, literatürdeki boyut indirgeme yaklaşımlarına kıyasla farklı ve daha güçlü bir yöntem sunmuştur. Hibrit yöntemlerin kullanımı, hem doğruluk oranını artırmış hem de botnet tespiti için daha kapsamlı bir yaklaşım sağlamıştır. Bu, çalışmayı sadece teorik olarak güçlü bir model değil, aynı zamanda gerçek dünyada uygulanabilirliği yüksek bir çözüm haline getirmiştir. Bu, yalnızca akademik başarı değil, aynı zamanda pratik uygulamalar için de uygun bir çözüm olarak değerlendirilmektedir. Bu bağlamda, botnet saldırılarının tespitinde literatürde eksik olan zaman bazlı performans değerlendirmesi bu çalışmanın özgün katkılarından biri olarak öne çıkmakta ve uygulamalı güvenlik sistemlerinde pratik bir rehber niteliği taşımaktadır.

Sonuç olarak, bu çalışmada önerilen hibrit yöntemler, işlem sürelerini optimize ederken yüksek performansı koruyarak dengeli bir sonuç elde edilmesini sağlamıştır. Özellikle, gerçek zamanlı saldırı tespit sistemlerinde işlem hızının kritik bir gereklilik olduğu düşünüldüğünde, hibrit yöntemler bu gereksinimleri karşılayan etkili bir çözüm olarak öne çıkmaktadır. Gelecek çalışmalarda, hibrit özellik seçimi tekniklerinin daha geniş veri setleri üzerinde değerlendirilmesi önerilmektedir.

6. KAYNAKLAR

Alejandre F V, Cortés N C, Anaya E A, 2017, Feature Selection To Detect Botnets Using Machine Learning Algorithms, 2017 İnternational Conference On Electronics, Communications And Computers, February 22-24, Mexico, 1-7.

Altunay H C, Albayrak Z, 2021, Network Intrusion Detection Approach Based On Convolutional Neural Network, Avrupa Bilim Ve Teknoloji Dergisi, 26, 22-29.

Aremu B L, Aro A T, Saka K K, Seriki A A, Raji R O, 2024, Botnet Attack Detection in Internet of Things Using Selected Learning Algorithms, University of Ibadan Journal of Science and Logics in ICT Research, 12(2), 18-26.

Arslan İ, 2021, Python ile Veri Bilimi, Pusula Yayıncılık, 420s, İstanbul.

Asan M E, Taşkın H, Alemdar M, Capoglu R, 2024, Tiroit kanseri hastalık tanısında lojistik regresyon kullanımı, Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 39(3), 1509-1524.

Aydın A, 2023, Topluluk Öğrenmesi Kullanılarak Zararlı Alan Adlarının Sınıflandırılması, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 67s, İstanbul.

Bahşi H, Nõmm S, La Torre F B, 2018, Dimensionality Reduction For Machine Learning Based Iot Botnet Detection, 2018 15th International Conference on Control, Automation, Robotics and Vision, November 18-21, Singapore, 1857-1862.

Bayram Arlı N, Güraksal S, Engin M, 2022, Denetimli Makine Öğrenmesi Algoritmalarına Giriş, Arlı N B, Güraksal S, Engin M (Ed.), Denetimli Makine Öğrenmesi Algoritmaları - R ve Python Uygulamaları (1-9), Nobel Akademik Yayıncılık, 310s, Ankara.

Beşli N, Tenekeci E, 2020, Uydu Verilerinden Karar Ağaçları Kullanarak Orman Yangını Tahmini, Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 11(3), 899-906.

Budak H, 2018, Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 22, 21-31.

Bulut S, Günlü A, 2016, Arazi Kullanım Sınıfları İçin Farklı Kontrollü Sınıflandırma Algoritmalarının Karşılaştırılması, Kastamonu University Journal of Forestry Faculty, 16(2), 528-535.

Büyükçapar S E, 2023, Makine Öğrenmesi Kullanılarak Türkiye'de Yıllara Göre Tarım Ürünleri Üretici Fiyat Endeksi Analizi, İzmir Katip Çelebi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 46s, İzmir.

Büyükkeçeci M, Okur M C, 2023, A Comprehensive Review Of Feature Selection And Feature Selection Stability İn Machine Learning, Gazi University Journal of Science, 36(4), 1506-1520.

Carneiro T, Da Nóbrega R V M, Nepomuceno T, Bian G B, De Albuquerque V H C, Reboucas Filho P P, 2018, Performance analysis of google colaboratory as a tool for accelerating deep learning applications, IEEE Access, 6, 61677-61685.

Çelik Ö, Altunaydın S S, 2018, A Research on Machine Learning Methods and Its Applications, Journal of Educational Technology and Online Learning, 1(3), 25-40.

Çelik S, Bozkurt Ö Ç, Ekşili N, 2022, Çalışan Performansı Ölçeğindeki İfadelerin Karar Ağacı Algoritması İle Belirlenmesi, Mehmet Akif Ersoy Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi, 9(1), 561-584.

Çelik S, 2023, Makine Öğrenmesi Farkında Büyük Veri Odaklı Üst Veri Havuzunun Gerçekleştirilmesi, Ege Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 124s, İzmir.

Çetin V, Yıldız O, 2022, A comprehensive review on data preprocessing techniques in data analysis, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 28(2), 299-312.

Daş B, Türkoğlu İ, 2014, DNA Dizilimlerinin Sınıflandırılmasında Karar Ağacı Algoritmalarının Karşılaştırılması, Eleco 2014 Elektrik, Elektronik, Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu, 27-29 Kasım 2019, Bursa.

Demirel Ş, 2019, Karar Ağacı Algoritmaları Ve Çocuk İşçiliği Üzerine Bir Uygulama, Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, 126s, İstanbul.

Emeç M, Özcanhan M H, 2023, Veri Ön İşleme ve Öznitelik Mühendisliğinin Yapay Zekâ Yöntemlerine Uygulanması, Mühendislikte Öncü ve Çağdaş Çalışmalar, 33-54.

Emekli B, 2024, Yazılım Tanımlı Ağlarda Makine Öğrenme Temelli Saldırı Tespit Sistemi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 83s, Sakarya.

Emhan Ö, Akın M, 2019, Filtreleme Tabanlı Öznitelik Seçme Yöntemlerinin Anomali Tabanlı Ağ Saldırısı Tespit Sistemlerine Etkisi, Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 10(2), 549-559.

Erbayram T, Erisoglu M, 2021, Yeni Bir Özellik Seçim Yöntemi Ve Özellik Seçim Yöntemlerinin Sınıflama Performanslarının Karşılaştırılması, Nicel Bilimler Dergisi, 3(1), 72-90.

Ertuğrul B, 2024, Saldırı Tespit Sistemlerinde Makine Öğrenimi Tekniklerinin Kullanımı ve Analizi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 106s, Edirne.

Geron A, 2021, Scikit - Learn Keras ve TensorFlow ile Uygulamalı Makine Öğrenmesi, Çev.: Aksoy B, Kaya Ö, Buzdağı Yayınevi, 896s, Ankara.

Gök A C, Özdemir A, 2011, Lojistik Regresyon Analizi İle Banka Sektör Paylarının Tahminlenmesi, Dokuz Eylül Üniversitesi İşletme Fakültesi Dergisi, 12(1), 43-51.

Gülşen E, Gündüz H, Cataltepe Z, Serinol L, 2015, Big Data Feature Selection And Projection For Gender Prediction Based On User Web Behaviour, In 2015 23nd Signal Processing and Communications Applications Conference (SIU), 16-19 May 2015, Malatya, 1545-1548.

Gürmen C, 2020, Saldırı Tespit Sistemleri İçin Makine Öğrenme Yöntemlerinin Performans Karşılaştırması, Harran Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 107s, Şanlıurfa.

Jahbel A, Ahmad T, Putra M A R, 2024, Improving Spam Botnet Detection with Chi Square Feature Selection and Multiclass Machine Learning Classification, 2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering, August 29-30, Indonesia, 115-120.

Janiesch C, Zschech P, Heinrich K, 2021, Machine Learning And Deep Learning, Electronic Markets, 31(3), 685-695.

Jordan M I, Mitchell T M, 2015, Machine Learning: Trends, Perspectives, And Prospects, Science, 349(6245), 255-260.

Kalaycı T E, 2018, Kimlik hırsızı web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması, Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi, 24(5), 870-878.

Kanimozhi V, Jacob T P, 2019, Artificial İntelligence Based Network İntrusion Detection With Hyper-Parameter Optimization Tuning On The Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing, 2019 İnternational Conference On Communication And Signal Processing, April 04-06, India, 33-36.

Kanimozhi V, Jacob, T P, 2021, Artificial Intelligence Outflanks All Other Machine Learning Classifiers İn Network Intrusion Detection System On The Realistic Cyber Dataset CSE-CIC-IDS2018 Using Cloud Computing, ICT Express, *7*(3), 366-370.

Karakaya A, 2024, Meme Kanseri Tahmininde Makine Öğrenmesi Algoritmaları Ve Automl, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 108s, Denizli.

Karaman M S, Turan M, Aydın M A, 2020, Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespit Modeli Uygulaması, Avrupa Bilim ve Teknoloji Dergisi, 10-17.

Kaynar O, Arslan H, Görmez Y, Işık Y E, 2018, Makine Öğrenmesi Ve Öznitelik Seçim Yöntemleriyle Saldırı Tespiti, Bilişim Teknolojileri Dergisi, 11(2), 175-185.

Kersting H B K, Zelezny S N F, 2013, Machine Learning And Knowledge Discovery İn Databases, Springer, 760p, Germany.

Kılınç F, Eyüpoğlu C, 2023, Ağ Ortamındaki Saldırı Türleri: Saldırı Senaryo Örnekleri, İstanbul Ticaret Üniversitesi Teknoloji ve Uygulamalı Bilimler Dergisi, 6(1), 99-109.

Köktürk F, 2012, K-En Yakın Komşuluk, Yapay Sinir Ağları Ve Karar Ağaçları Yöntemlerinin Sınıflandırma Başarılarının Karşılaştırılması, Bülent Ecevit Üniversitesi, Sağlık Bilimleri Enstitüsü, Doktora Tezi, 88s, Zonguldak.

LeCun Y, Bengio Y, Hinton G, 2015, Deep Learning, Nature, 521(7553), 436-444.

McKay R, Pendleton B, Britt J, Nakhavanit B, 2019, Machine Learning Algorithms On Botnet Traffic: Ensemble And Simple Algorithms, Proceedings Of The 2019 3rd İnternational Conference On Compute And Data Analysis, March 14-17, New York, 31-35.

Mirtagioğlu H, Demir Y, 2022, Makine Öğrenmesi Eğitim Ve Test Verisi Kullanılarak Ridge Regresyon Uygulaması, Çelik Ş, Köleoğlu N, Çemrek F (Ed.), Veri Madenciliği Ve Makine Öğrenmesi İle Farklı Alanlarda Uygulamalar (183-202), Holistence Publication, 214s, Çanakkale.

Mishra B, Sharma R S, 2024, Feature Extraction Based IoT Botnet Detection Using Machine Learning Technique, Library Progress International, 44(3), 26203-26209.

Muhammad A, Asad M, Javed A R, 2020, Robust Early Stage Botnet Detection Using Machine Learning, International Conference On Cyber Warfare And Security, October 20-21, Pakistan,1-6.

Nacar E N, Erdebilli B, 2021, Makine Öğrenmesi Algoritmaları İle Satış Tahmini, Endüstri Mühendisliği, 32(2), 307-320.

Odeyemi T O, Isiekwene C C, Abidoye A P, 2023, Cyber Attack Detection in A Global Network Using Machine Learning Approach, FUOYE Journal Of Engineering And Technology, 8(4), 448-455.

Oğuzlar A, 2003, Veri ön işleme, Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 21.

Ölmez E G, İnce K, 2022, IoT Botnet Verisetlerinin Karşılaştırmalı Analizi, Computer Science, 151-164.

Özlen T, 2022, Servikal Kanserlerin Teşhisinde Kullanılan Makine Öğrenmesi Algoritmalarının Karşılaştırmalı Analizi, İstanbul Aydın Üniversitesi, Lisansüstü Eğitim Enstitüsü, Yüksek Lisans Tezi, 100s, İstanbul.

Saygın E, 2021, Karaciğer Hastalığı Teşhisinde Makine Öğrenmesi Yöntemlerinin Başarılarının Ölçülmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 82s, Elazığ.

Serdaroğlu F, 2021, Makine Öğrenmesi Algoritmaları ile Ddos Saldırı Tespiti Ve Sınıflandırması, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 89s, İstanbul.

Topbaş Z, 2024, Bir Boyutlu Evrişimli Sinir Ağları Kullanılarak Ağ Saldırı Tespiti, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 112s, Konya.

Vergara J R, Estévez P A, 2014, A Review Of Feature Selection Methods Based On Mutual İnformation, Neural Computing And Applications, 24, 175-186.

Zengin H F, 2024, Makine Öğrenmesi Yöntemlerinde Hiper-Parametre Seçimi, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 70s, Eskişehir.

Werth A, Oliver K A, West C G, Lewandowski H J, 2022, Engagement in collaboration and teamwork using Google Colaboratory, PERC Proceedings, 481-487.

Wijaya A R, Ahmad T, Hostiadi D P, Putra M A R, Alzamzami M N, 2024, Comparative Analysis of Data Balancing Methods for the Optimization of Botnet Attack Detection Models, 2024 10th International Conference on Smart Computing and Communication, July 25-27, Indonesia, 608-613.

İnternet Kaynakları

1-https://ruveydakardelcetin.medium.com/yapay-sinir-a%C4%9Flar%C4%B1-google-colab-%C3%A7al%C4%B1%C5%9Fma-ortam%C4%B1-3e029af37095, 10.07.2024

2-https://colab.research.google.com, 10.07.2024

3-https://www.unb.ca/cic/datasets/ids-2018.html, 10.07.2024