T.C.

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

ANABİLİM DALI

AĞ SALDIRILARININ SINIFLANDIRILMASINDA KARAR AĞAÇLARINA DAYALI ARTTIRMA (BOOSTING) ALGORİTMALARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

KORAY ÇOŞKUN

TEMMUZ 2020

MUĞLA

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

KORAY ÇOŞKUN tarafından hazırlanan **AĞ SALDIRILARININ SINIFLANDIRILMASINDA KARAR AĞAÇLARINA DAYALI ARTTIRMA (BOOSTING) ALGORİTMALARININ KARŞILAŞTIRILMASI** başlıklı tezinin, 09/07/2020 tarihinde aşağıdaki jüri tarafından Bilişim Sistemleri Mühendisliği Anabilim Dalı'nda yüksek lisans derecesi için gerekli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JURİSİ

Doç. Dr. Ali Hakan IŞIK **(Jüri Başkanı)** İmza:

Bilgisayar Mühendisliği Anabilim Dalı,
Mehmet Akif Ersoy Üniversitesi, Burdur _____

Doktor Öğretim Üyesi Gürcan ÇETİN **(Danışman)** İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Doç. Dr. Hüseyin GÜRÜLER **(Üye)** İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

ANA BİLİM DALI BAŞKANLIĞI ONAYI

Doç. Dr. Hüseyin Gürüler İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı Başkanı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Doktor Öğretim Üyesi Gürcan ÇETİN İmza:

Danışman, Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla _____

Savunma Tarihi: 09/07/2020

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Koray ÇOŞKUN

09/07/2020

ÖZET

AĞ SALDIRILARININ SINIFLANDIRILMASINDA KARAR AĞAÇLARINA DAYALI ARTTIRMA (BOOSTING) ALGORİTMALARININ KARŞILAŞTIRILMASI

Koray ÇOŞKUN

Yüksek Lisans Tezi
Fen Bilimleri Enstitüsü
Bilişim Sistemleri Mühendisliği Anabilim Dalı
Danışman: Dr. Öğr. Üyesi Gürcan ÇETİN
Temmuz 2020, 106 Sayfa

Son yıllarda bilişim kaynaklarının güvenliğini sağlama oldukça önemli bir konu haline gelmiştir. Özellikle internet erişimi sağlayan ağ yapısı ulusal ve uluslarası birçok konumdan gelen saldırıların hedefi olabilmekte ve bu ağ yapısını kullanan kurumlarda kayıplara neden olabilmektedir. Birçok kaynakta saldırı tespit sistemleri olarak da anılan anomali tespit sistemleri de bu tür ağlarda oluşan düzensizlikleri tespit etmeyi hedeflerler. Ancak saldırı tespit sistemlerinin başarısı bu yazılımların arka planda kullandıkları algoritmalar ve öğrenme kapasiteleri ile sınırlıdır. Bu tez çalışmasında, topluluk öğrenmesi algoritmalarından arttırma (boosting) yönteminin zararlı ağ trafiğini belirlemedeki başarım oranı araştırılmıştır. Bu bağlamda AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost algoritmaları Google Colab üzerinde modellenmiş ve CICID2017 veri kümesi üzerinde uygulanmıştır. Ayrıca, tez çalışmasında, CICIDS2017 veri kümesindeki etiketlenmiş ağ saldırılarının tespit edilmesinde Karar Ağaçları yöntemi, arttırma algoritmalarının başarımını karşılaştırmak amacıyla kullanılmıştır. Eğitilen ve test edilen sınıflandırıcıların performansları doğruluk, kesinlik, duyarlılık, F1 değeri, Kappa değeri ve ROC eğrisi gibi performans değerlendirme ölçütleri ile incelenmiştir. Bunlar haricinde başka bir değerlendirme ölçütü olarak da algoritmaların öğrenme süreleri de kullanılmıştır. Çalışma sonucunda F1 değeri açısından en iyi sonucu %99.89 ile XGBoost algoritmasının verdiği ve ayrıca AUC değerlerinin de 0.9989 ile 1'e çok yakın olduğu belirlenmiştir. Ancak bunun yanında yüksek başarım elde

edilen diğer arttırma algoritmalarının yanında XGBoost algoritmasının işlem süresinin daha uzun sürdüğü görülmüştür. Hesaplamalarda 0.9773 değerini veren LightGBM algoritması eğitim süresi bakımından incelendiğinde 4.5 dakikalık bir süreyle Karar Ağaçları algoritmasının ardından en hızlı ikinci algoritma olarak tespit edilmiştir.

Anahtar Kelimeler: Saldırı Tespit Sistemi, Ağ Saldırıları, Makine Öğrenmesi
Arttırma Algoritmaları

ABSTRACT

COMPARISON OF THE DECISION TREE BASED BOOSTING ALGORITHMS IN CLASSIFICATION OF NETWORK ATTACKS

Koray ÇOŞKUN

Master Thesis

Institute of Science and Technology

Information System Engineering

Supervisor: Asst. Prof. Dr. Gürcan ÇETİN

July 2020, 106 pages

In recent years, securing IT resources has become a very important issue. Especially, the network structure providing internet access can be the target of attacks coming from many national and international locations and it may cause losses in the institutions using this network structure. Anomaly detection systems, also known as intrusion detection systems in many sources, aim to detect anomalies in such networks. However, the success of intrusion detection systems is limited by the algorithms and learning capacities that this software use in the background. In this thesis, the success rate of the boosting method, which is one of the community learning algorithms, in determining the harmful network traffic has been investigated. In this context, AdaBoost, CatBoost, Gradient Boosting, LightGBM and XGBoost algorithms are modeled on Google Colab and implemented on CICID2017 dataset. In addition, in the thesis, the Decision Trees method was used to compare the performance of boosting algorithms in detecting labeled network attacks in the CICIDS2017 dataset. The performances of the trained and tested classifiers were examined with performance evaluation criteria such as accuracy, precision, sensitivity, F1 value, Kappa value and ROC curve. Apart from these, the learning times of the algorithms were also used as another evaluation criterion. As a result of the study, it was determined that the best result in terms of F1 value was given by XGBoost algorithm with 99.89% and also AUC values were very close to 1.0 with 0.9989. However, it has been observed that the processing time of the XGBoost algorithm takes longer, in addition to other boosting algorithms with high performance. When the LightGBM algorithm, which gives the value of 0.9773 in the

calculations, is examined in terms of training time, it was determined as the second fastest algorithm after the Decision Trees algorithm for a period of 4.5 minutes.

Keywords: Intrusion Detection System, Network Attacks, Machine Learning Boosting Algorithms

Ülkeme, Aileme, Sevgili Eşime ve Gelecek Nesillere,

ÖNSÖZ

Yüksek lisans eğitimim boyunca ve bu çalışmanın gerçekleştirilmesinde hiçbir desteği esirgemeyen, bilgilerini benimle paylaşan, çalışmamda konu, kaynak ve yöntem açısından yol gösteren, bundan sonraki meslek hayatımda da bana verdiği bilgilerden faydalanacak olduğum, danışmanlığımı yapan değerli hocam Doktor Öğretim Üyesi Gürcan ÇETİN'e, Bilişim Sistemleri Mühendisliği Anabilim Dalı'nda görev yapan değerli hocalarıma ve her zaman desteğini yanımda hissettiğim eşim Damla Nur Çoşkun'a en içten duygularımla teşekkür ederim.

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

KISALTMALAR DİZİNİ

Kısaltmalar	Açıklama
AB	Avrupa Birliği
AdaBoost	Adaptive Boosting
ANN	Artificial Neural Network (Yapay Sinir Ağı)
AUC	Area Under Curve (Eğri Altında Kalan Alan)
Bagging	Bootstrap Aggregating
BENIGN	Normal ağ hareketi
Boosting	Arttırma
BZ2	BZip2 Compressed File Extension
CART	Classification and Regression Trees (Sınıflandırma ve Regresyon Ağaçları)
CIC DoS	CIC DoS Dataset (2017)
CICFlowMeter	Network Traffic Flow Generator
CICIDS2017	Intrusion Detection Evaluation Dataset
CISC-2010v2	Dataset name
CNN	Convolutional Neural Networks (Evrimsel Sinir Ağları)
Colab	Google Colaboratory
CPU	Central Processing Unit
CSE-CIC-IDS2018	CSE-CIC-IDS2018 Dataset
CSV	Comma Separated Values
DDoS	Distributed Denial of Service
DMTK	Distributed Machine Learning Toolkit

DoS	Denial of Service
DQN-IDS	Deep Q Ağ Saldırı Tespit Sistemi
EFS	Ensemble Feature Selection (Topluluk Öznitelik Seçimi)
FN	False Negatives (Yanlış Negatif)
FP	False Positives (Yanlış Pozitif)
FP7	7th Framework programme
FPR	False Positive Rate (Yanlış Pozitif Oran)
GBM	Gradient Boosting Machine
GBS	Gradient Boosting Sınıflandırıcı
GBT	Gradient Boosted Trees
GPU	Graphics Processing Unit
ID3	Iterative Dichotomiser 3 (Tekrarlı İkilikçi Ağacı)
IDS	Intrusion Detection System (Saldırı Tespit Sistemi)
IoT	Internet of Things (Nesnelerin İnterneti)
ISEAR	International Survey on Emotion Antecedents and Reactions Dataset
J48	Decision Trees (Karar Ağaçları)
KNN	K-Nearest Neighbourhood (K-En Yakın Komşu)
LGBM	Light Gradient Boosting Machine
LightGBM	Light Gradient Boosting Machine
MAE	Mean Absolute Error (Ortalama Mutlak Hatası)
MLP	Multi Layer Perceptron (Çok Katmanlı Algılayıcı)
MNB	Multinomial Naive Bayes
NaN	Not a Number
NB	Naive Bayes
NIPS	Neural Information Processing Systems Conference
NSL-KDD	Dataset name
NULL	Boş/Kayıp veri

OneR	One Rule Algorithm
PCA	Principal Component Analysis (Temel Bileşen Analizi)
PCAP	Packet Capture
PM10	Partikül Madde
QDA	Quadratic Discriminant Analysis (Karesel Diskriminant Analizi)
R2L	Remote to Local
RF	Random Forest (Rastgele Orman)
RFE	Recursive Feature Elimination(Özyinelemeli Unsur Giderme)
RMSE	Root-Mean-Square Error (Kök-Ortalama-Kare Hatası)
ROC	Receiver Operating Characteristics (Alıcı İşlem Karakteristiği)
SMOTE	Synthetic Minority Oversampling Technique
Stacking	Stacked Generalization
STS	Saldırı Tespit Sistemleri
SVM	Support Vector Machine (Destek Vektör Makinesi)
TCP/IP	Transmission Control Protocol / Internet Protocol
TN	True Negatives (Doğru Negatif)
TP	True Positives (Doğru Pozitif)
U2R	User to Root
UNSW-NB15	Dataset name
VITAL	Virtualized programmable InTerfAces for smart, secure and cost-effective IoT depLoyments in smart cities
Voting	Oylama
XGBoost	Extreme Gradient Boosting

1. GİRİŞ

Saldırı Tespit Sistemleri (STS), ağda meydana gelen aykırı davranışları tespit ederek bu aykırı durumlar hakkında yetkilileri ve/veya güvenlik yazılımlarını bilgilendiren sistemlerdir (Budak, ve diğerleri, 2013). STS'ler incelendiğinde, temel olarak imza tabanlı ve anomali tabanlı olmak üzere ikiye ayrıldıkları belirtilebilir. İmza tabanlı sistemler, daha önce tespit edilmiş ve öğrenilmiş olan saldırı çeşitlerinin veri tabanında saklanması yoluyla ağdaki yeni karşılaşılan aykırı davranışların tespiti için bu verilerle karşılaştırma yapılarak gerçekleşmektedir. Diğer taraftan, anomali tabanlı sistemler ise, ağda gerçekleşmekte olan paket trafiğini izleyerek gerçek zamanlı olarak değerlendirme yapmaktadır (Atay, ve diğerleri, 2019). Bu konuda Çalışır ve arkadaşları 2019 yılında yaptıkları çalışmalarında (Çalışır, ve diğerleri, 2019); *"STS makine öğrenmesi algoritmalarıyla eğitilerek, gerçek zamanlı saldırıların tespiti için gerçek zamanlı önlemlerin alınmasını amaçlar." ş*eklinde ifade ederek STS'nin amacını tanımlamıştır.

Bilişim sistemlerinde STS kullanılması sayesinde, güvenlik yetkilileri sistemlerine yapılan saldırıların çeşitleri hakkında detaylı bilgi alabilmekte ve ayrıca sistemde varsa zayıf noktaların tespit edilmesi mümkün olabilmektedir (Baykara, 2016).

Gerçek zamanlı STS tasarımlarının yapılabilmesi amacıyla 1998-2016 yılları arasında çeşitli veri kümeleri kullanılarak akademik çalışmalar yapılmıştır. Kullanılan veri kümelerinin kapsamlı ve güncel saldırı tekniklerinden eksik olması sebebiyle bu veri kümelerini kullanan çalışmalar günümüzdeki ağ saldırılarına dayalı problemleri işlemekte yetersizlik gösterebilmektedir. Bu tez çalışmasında, Kanada Siber Güvenlik Enstitüsü tarafından oluşturulmuş olan CICIDS2017 veri kümesi (Sharafaldin, ve diğerleri, 2018) kullanılmıştır. Bu veri kümesi, literatürde kullanılan diğer veri kümelerindeki eksiklikler dikkate alınarak oluşturulmuştur. Ayrıca gerçek

dünyadaki ağ sistemleri gibi normal ve saldırı ağ trafiği içeren veri kümesi elde etmek amaçlanmıştır (Özekes, ve diğerleri, 2019).

Bu çalışmada, CICIDS2017 veri kümesindeki etiketlenmiş ağ saldırılarının tespit edilmesinde Karar Ağaçları yöntemi ve bu yöntemi karşılaştırmak amacıyla Arttırma (Boosting) yöntemleri kullanılmıştır. Karar ağaçları, yorumlanmasının kolay olması ve öznitelik ölçeklendirmeye gerek duymaması gibi avantajları sayesinde sınıflandırma işlemlerinde yaygın olarak kullanılmaktadır (Yılmaz, 2019).

Arttırma algoritmaları ise makine öğrenmesi yöntemlerinden Topluluk Öğrenmesi yapısına dahildir. Topluluk öğrenmesi algoritmaları ile yüksek oranda başarım ve yüksek düzeyde performanslar elde edilebilmektedir. Topluluk öğrenmesi algoritmaları içerisinde torbalama (bagging), arttırma (boosting), oylama (oylama) ve yığınlama (stacking) teknikleri bulunmaktadır. Arttırma algoritmaları, yavaş öğrenmeye dayalı, sıralı bir yöntem olup hatadan öğrenmeyi amaçlar. Arttırma algoritmalarının genel çalışma prensibi, her biri bir önceki sonucu düzeltmeye çalışan modelleri sırayla eğiterek daha iyi sonucu elde etmektir (Yılmaz, 2019). Bu tez çalışmasında, topluluk öğrenmesi algoritmalarından arttırma yönteminin zararlı ağ trafiğini belirlemedeki başarım oranı araştırılmıştır. Bu bağlamda AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost algoritmaları Google Colab üzerinde modellenmiş ve CICID2017 veri kümesi üzerinde uygulanmıştır.

Tez çalışmasının geri kalanı şu şekilde düzenlenmiştir. Gelecek bölümde sırasıyla; literatür çalışmalarına yer verilerek, ağ saldırıları hakkında temel bilgilendirme, saldırı tipleri ve saldırı tespit sistemlerine dair detaylı açıklamalar Bölüm 2'de sunulmuştur. 3. Bölüm; veri madenciliği, makine öğrenmesi ve topluluk öğrenmesini ayrıntılarıyla açıklar. Bölüm 4; CICIDS2017 veri kümesi üzerinde yapılan işlemleri ayrıntılı olarak anlatır. Modellerin değerlendirme ölçütleri, çalışmanın deneysel sonuçları yine bu bölümde sunulmaktadır. Gerçekleştirilen uygulama sonucu elde edilen bulgular ve modellerin karşılaştırılması ise Bölüm 5'te verilmiştir. Son bölüm olan Bölüm 6'da uygulama sonucunda elde edilen sonuçlar aktarılmaktadır.

1.1. Literatür Araştırması

Çalışır ve arkadaşları 2019 yılında yaptıkları çalışmalarında; 2017 yılında geliştirilen DoS saldırılarının yer aldığı CIC DoS veri kümesi üzerinde Rastgele Orman, XGBoost, LGBM, Gradient Boosting, MLP, CNN ve SVM algoritmalarını kullanarak DoS saldırı tespitini amaçlamıştır. Rastgele Orman, XGBoost, LGBM, Gradyan Arttırma, algoritmalarının başarımını doğrulamak için algoritmaların her birini 10 kez çalıştırmışlardır. Sonuçlar incelendiğinde, bu veri kümesi için, LGBM karar ağacı algoritması F1 değeri bakımından en doğru sınıflandırmayı yaptığı gözlenmiştir (Çalışır, ve diğerleri, 2019).

Atay ve arkadaşları 2019 yılında yaptıkları çalışmalarında; Tek ve İki Seviyeli olmak üzere farklı hibrit yöntemleri CSECIC-IDS2018 isimli veri kümesinde kullanmışlardır. Sonuçlar karşılaştırıldığında, iki seviyeli hibrit yöntemin başarı oranının daha yüksek olduğu tespit edilmiştir. İki seviyeli yöntem, %98 doğruluk oranı ve F1 değerinin makro ortalaması ise 0.86 değeriyle en yüksek değerlere sahip modeldir (Atay, ve diğerleri, 2019).

Sharafaldin ve arkadaşları 2018 yılındaki çalışmalarında, ağ trafiği dinlemesi ve saldırıların tespit edilmesinde kullanılan veri kümelerinin yetersizliklerinden ötürü CICIDS2017 veri kümesini oluşturmuşlardır. Çalışmalarında, KNN, Rastgele Orman, ID3, Adaboost, MLP, NB ve QDA yöntemlerini kullanmışlar. Sonuçlar incelendiğinde ID3 algoritmasının 0.98 F1 değeri ile en iyi sonuç veren algoritma olduğu belirtilmiştir (Sharafaldin, ve diğerleri, 2018).

Aydın 2018 yılındaki çalışmasında, NB, SVM, J48, KNN, Rastgele Orman, GBT sınıflandırma algoritmaları kullanılmıştır. Kullanılan veri kümesi, İzmir Büyükşehir Belediyesi İtfaiye Daire Başkanlığı bünyesindeki yangın raporlarından elde edilmiştir. 2015-2017 tarihleri arasındaki yangın kayıtları analiz edilerek bölgelerin sınıflandırılmasında %93.84 ile en başarılı algoritmanın Random Forest algoritması olduğu tespit edilmiştir. En başarılı algoritma tespit edilirken doğruluk, MAE, RMSE ve Kappa değerleri göz önüne alınmıştır (Aydın, 2018).

Özekes ve Karakoç 2019 yılındaki çalışmaları kapsamında, zararlı ağ trafiğinin tespit edilmesi için karar ağacı ve rastgele orman algoritmaları kullanılmıştır. Veri kümesi olarak CICIDS2017 kullanılmıştır. Bir milyondan fazla kayıt kullanılarak eğitilen ve test edilen sınıflandırıcıların performanslarını değerlendirmek için kullanılan ölçütler; doğruluk, hassasiyet, geri çekilme, F1 değeri, ROC eğrileri ve eğri altında kalan alandır. Değerlendirme sonucunda, rastgele orman algoritmasının performansı, karar ağacı algoritması performansına oranla az bir farkla bile olsa daha iyi olduğu ölçülmüştür (Özekes, ve diğerleri, 2019).

Yangın 2019 yılındaki Yüksek Lisans Tez çalışmasında, karar ağaçları, rastgele orman, gradyan arttırma ve XGBoost algoritmalarını kullanmışlardır. Bu algoritmalar iki veri kümesine uygulanmış ve performans değerlendirme ölçütlerinden doğru sınıflandırma oranı, kesinlik, duyarlılık, Kappa değeri ve F1 değeri sonuçları elde edilmiştir. Doğru sınıflandırma oranı bakımından Karar Ağaçları, Rastgele Orman, gradyan arttırma ve XGBoost algoritmalarının sırasıyla birinci veri kümesinde %56.10, %60.98, %65.85 ve %70.73 iken; 'PIMA INDIANS' veri kümesinde ise %75.82, %81.05, %81.70 ve %82.35 sonuçlarını verdiği ölçülmüştür. Bu sonuçların ışığında her iki veri kümesinde de XGBoost algoritmasının daha iyi sonuç verdiği görülmüştür (Yangın, 2019).

Yılmaz 2019 yılındaki Yüksek Lisans Tez çalışmasında, İngilizce bir metin içerisindeki neşe, korku, öfke, üzüntü, tiksinti ve şaşkınlık duygularının otomatik olarak sınıflandırılmasında Topluluk Öğrenmesi yöntemlerinin etkileri ölçülmüştür. Bu amaçla Twitter'dan elde edilen veri kümesi yanında ISEAR veri kümesi üzerinde de deneysel çalışmalar yapılmıştır. ISEAR veri kümesinde Torbalama, Arttırma ve Oylama algoritmaları kullanılmıştır. Topluluk Öğrenmesi yöntemleri kullanımının temel sınıflandırıcıların başarı oranlarını arttırdığı, denenenler içerisinde en başarılı yöntemin de oylama algoritması olduğu görülmüştür (Yılmaz, 2019).

Kaya 2019 yılındaki Yüksek Lisans Tezi kapsamında, İstanbul için PM10 kirleticisi aracılığıyla hava kirliliği yoğunluğu tahmini yapılmıştır. Avrupa ve Anadolu yakasında hava kirliliğine oldukça maruz kalan toplamda dokuz bölgeye ait Ağustos

2011 - Şubat 2018 tarih aralığını kapsayan gerçek dünya verileri tahminlemede kullanılmıştır. Topluluk modelleri ile EAS, GAS, ROS; çekirdek tabanlı poli-DVMS, rbf-DVMS ve DRS performansları test kümeleri üzerinde karşılaştırıldığında ROS, dokuz bölge için genişletilmiş testlerde de öndedir (Kaya, 2019).

Gülaçar 2018 yılındaki Yüksek Lisans Tezi kapsamında, VITAL üzerinde kayıtlı sensörler için, kısa ve uzun süreli veri tahmini yapan modül geliştirilmiştir. Sınıflandırma işlemi için ise Karar Ağacı, Gradyan Arttırma, KNN, Rastgele Orman, SVM algoritmaları uygulanmıştır. Ortalamalara bakıldığında SVM algoritmasının diğer sınıflandırma algoritmalarından daha iyi öngörülerde bulunduğu görülmüştür (Gülaçar, 2018).

Güven 2018 yılındaki Yüksek Lisans Tez çalışmasında, makine öğrenmesi yöntemlerinden Karar Ağacı, SVM, KNN, Derin Öğrenme ve NB algoritmaları kullanılmıştır. Çalışmada, IoT uygulamaları için makine öğrenmesi yöntemleri karşılaştırılmıştır. KNN algoritmasının daha doğru sonuç verdiği görülmüştür (Güven, 2018).

Yiğidim 2012 yılındaki Yüksek Lisans Tez çalışmasında, veri kümesi olarak 10 Haziran 2001 tarihli 24 saatlik AucklandIV-20010610 kayıtları kullanılmıştır. Sınıflandırma algoritmalarının performanslarını yükseltmek için kullanılan Adaboost'un, C4.5 ve NB algoritmalarının öğrenme yeteneklerini geliştirdiği gözlemlenmiştir (Yiğidim, 2012).

Hossen ve Janagam 2018 yılındaki Yüksek Lisans Tez çalışmasında, geçmiş deneyimlere bağlı kalmayıp ilk denemede farklı saldırı türlerini tespit etmeyi amaçlamışlardır. İzinsiz giriş tespiti için bir referans veri kümesi olan CICIDS2017 veri kümesi kullanılmıştır. Ayrıca izinsiz giriş parametrelerinin sınıflandırılması için Softmax sınıflandırıcısı kullanılmıştır. Saldırıları analiz etmek için DQN-IDS CICIDS2017 veri kümesiyle entegre edilmiştir (Hossen, ve diğerleri, 2018).

Abdulkareem 2019 yılındaki Yüksek Lisans Tez çalışmasında, makine öğrenimi ve sinir ağı modellerini kullanarak anormal trafik akışını tespit etmeyi amaçlamıştır. Kullanılan algoritmalar, Logistic Regression, Rastgele Orman, QDA, Stochastic

Gradient Descent ve ANN. Kullanılan veri kümesi ise, 2018'de güncellenen CICIDS2017 veri kümesidir. Çalışma sonucunda Rastgele Orman modeli diğer modeller arasında en iyi performansı elde etmiştir (Abdulkareem, 2019).

Gülmez, 2019 yılındaki Yüksek Lisans Tez çalışmasında, uzun-kısa dönem hafızalı yinelemeli sinir ağları ve derin pekiştirmeli öğrenme gibi farklı makine öğrenmesi algoritmalarını karşılaştırma amacıyla kullanmıştır. NSL-KDD, UNSW-NB15, CICIDS2017 gibi farklı veri kümeleriyle deneyler yapılmıştır. Deneyler sonucunda, derin pekiştirmeli öğrenme algoritmasının daha iyi olduğu gözlenmiştir (Gülmez, 2019).

Faker, 2019 yılındaki Yüksek Lisans Tez çalışmasında, ağ trafiği veri kümelerini sınıflandırmak için üç sınıflandırıcı kullanılmıştır. Bunlar; Derin İleri Beslemeli Yapay Sinir Ağı, Rastgele Orman ve Gradyan Arttırma Ağacı'dır. Veri kümelerinden öznitelikleri seçmek ve değerlendirmek için homojenlik ölçümü kullanılmıştır. UNSW-NB15 ve CICIDS2017 veri kümeleri bu çalışmada kullanılmıştır (Faker, 2019).

Tama ve arkadaşları 2020 yılındaki çalışmalarında, web trafiğindeki aykırılık tespiti için özel olarak tasarlanmış iki veri kümesi olan, CISC-2010v2 ve CICIDS2017 veri kümelerini seçmişlerdir. Rastgele Orman, GBM ve XGBoost algoritmaları kullanılmıştır. Önerilen modelin genelleştirilebilirliğini kanıtlamak için literatürde bulunan CSIC-2010v2, CICIDS2017, NSL-KDD ve UNSWNB15 veri kümelerinde testler gerçekleştirilmiştir. Çalışma kapsamında önerilen yaklaşım, doğruluk ve FPR değeriyle neredeyse mükemmel bir algılama performansı sağlamıştır (Tama, ve diğerleri, 2020).

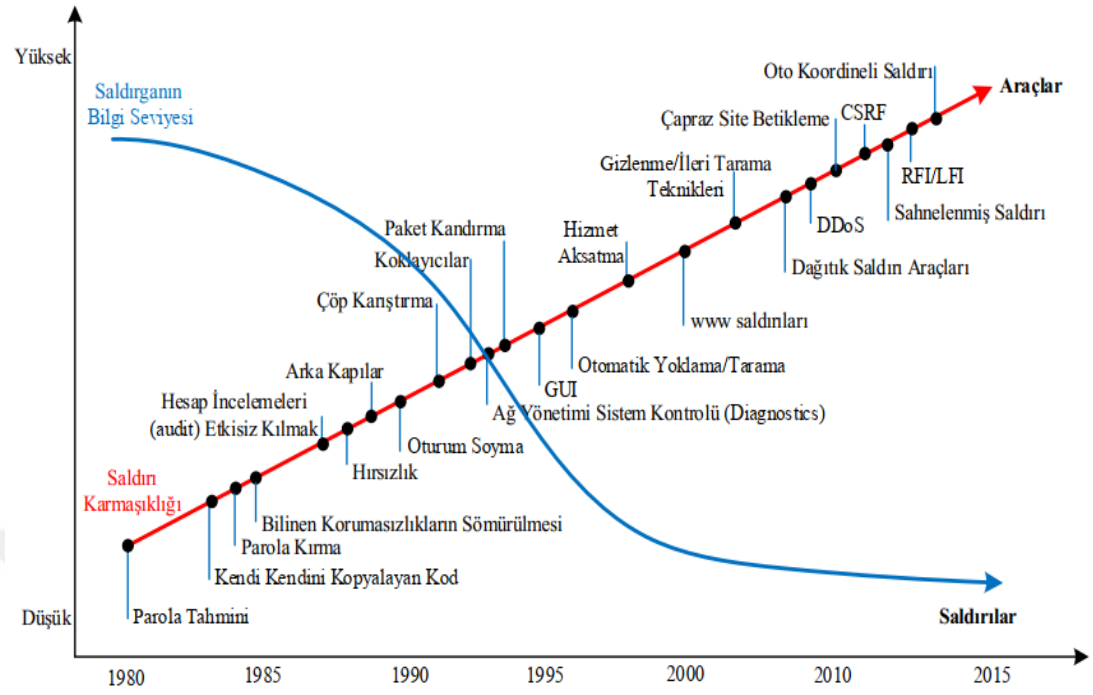
Abdulrahman ve Ibrahem 2018 yılındaki çalışmalarında, CICIDS2017 isimli güvenilir bir veri kümesini analiz etmişlerdir. C5.0, NB, SVM ve RF algoritmalarını kullanarak 4 makine öğrenimi modeli gerçekleştirilmiş ve modeller birbirleriyle karşılaştırılmıştır. Karşılaştırılan 4 sınıflandırma algoritmasından, RF ve C5.0 sınıflandırıcılar sırasıyla %86.80, %86.45 ortalama doğruluk değerleriyle diğerlerini geçmiştir (Abdulrahman, ve diğerleri, 2018).

Hosseini ve Seilani 2019 yılındaki çalışmalarının temel amacı, eğitim süresini azaltırken sistemdeki doğruluğu artırmaktır. Bu çalışmada, CICIDS2017 ve NSL-KDD veri kümesi kullanılmıştır. Lojistik Regresyon, RF, Karar Ağacı ve KNN makine öğrenme algoritmaları kullanılmıştır. Yapılan deneyler sonucunda, makine öğrenme algoritmaları ve iki veri kümesi kullanılarak, 27.36 sn elde edilen bir çalışma süresi aralığında yaklaşık %99 daha fazla doğruluk elde edilmiştir. (Hosseini, ve diğerleri, 2019).

2. AĞ SALDIRILARI

Bilgi güvenliği, en genel açıklamayla, elektronik ortamda bulunan verilerin bütünlüğünün sağlanması ve izinsiz erişimlerden korunması için, yapılan güvenlik önlemlerinin sağlanması olarak belirtilebilir. Bilgi güvenliğini sağlamak amacıyla oluşturulmuş güvenlik sistemlerine yapılan, zarar vermek amaçlı sistem erişimleri, güvenlik sistemini aşma girişimleri, sistemlerin yavaşlamasını veya durdurulmasını sağlamak gibi her türlü yapılan girişimler saldırı veya atak olarak isimlendirilmektedir. Bu tür girişimleri gerçekleştiren saldırganlar, amaçlarına ulaşmak için farklı çeşitlerde saldırılar gerçekleştirmektedirler. Bilgi güvenliği için; sistemlere yapılması muhtemel saldırı çeşitlerini bilmek, saldırıları doğru bir şekilde analiz etmek ve bunlara karşı alınabilecek önlemleri belirlemek büyük bir önem arz etmektedir (Canbek, ve diğerleri, 2006).

Günümüzde, e-ticaret, dosya ve fotoğraf paylaşımı, video konferanslar gibi işlemler internet üzerinde gerçekleşmektedir. Bu işlemleri yapanlar devlet veya özel şirket çalışanı olabilmekte, hatta üst düzey yönetici bile olabilir. İnternet üzerinde yapılan işlemler içerisinde, kişilerin kişisel bilgileri, banka hesap bilgileri vb. güvenliği sağlanması gereken bilgiler bulunmaktadır. İnternet kullanıcılarının zamanla sayısı artmakta ve saldırılar da benzer oranda artmasından dolayı özellikle kişisel verilerin güvenliği tehdit altındadır. Teknolojinin gelişmesiyle saldırganların saldırı çeşitleri artmış ve saldırı yapması kolaylaşmış durumdadır (Atalay, ve diğerleri, 2019). Şekil 2.1.'de de gösterildiği gibi, saldırıların çeşitleri zamanla farklılıklar göstermektedir (Baykara, 2016). Ayrıca yıllara göre saldırganların bu saldırıları gerçekleştirmek için gerek duydukları bilgi seviyesi oranında da azalma gözlenmektedir.



Şekil 2.1. Yıllara göre saldırı çeşitliliği

Saldırıların yapısı ve saldırganların tercih ettiği saldırı araçları, teknik açıdan bakıldığında oldukça gelişmekte ve karmaşıklaşmaktadır. Bu yüzden saldırganın ihtiyacı olan bilginin seviyesi de gittikçe azalmaktadır. Bu durum saldırıların çeşitliliğini, sayısını ve verdikleri hasarın seviyesini arttırırken, saldırılara karşı alınacak güvenlik önlemlerinin alınmasını zorlaştırmaktadır (Canbek, ve diğerleri, 2006). Tehditlerin ve risklerin hızla artması sebebiyle, güvenlik alanındaki teknolojilerde de gelişim hızlı şekilde gerçekleşmektedir. Güvenlik duvarları, antivirüs ve STS gibi yazılımsal veya donanımsal araçlar geliştirilmiştir.

2.1. Saldırı Tespit Sistemleri

Özekes ve Karakoç (Özekes, ve diğerleri, 2019) 2019 yılındaki çalışmalarında; *"STS, tüm tedbirlere karşın bilgisayar sistemlerine yapılan saldırılar gerçekleşirken ya da gerçekleştikten sonra tespit etmek, internet veya yerel ağdan gelebilecek, ağdaki*

sistemlere zarar verebilecek, çeşitli paket ve verilerden oluşan bu saldırıları fark etmek üzere tasarlanmış sistemlerdir" ifadesiyle STS'yi tanımlamışlardır. STS'de Kötüye Kullanım ve Anomali Tespiti olmak üzere iki çeşit yaklaşım bulunmaktadır (Baykara, 2016).

2.2. Saldırı Tespit Sistemlerinin Sınıflandırılması

2.2.1. Anomali tespiti

Anomali tespiti, sistemdeki anormal olayları, normal olarak değerlendirilecek olaylardan ayırarak fark etmeyi amaçlayan bir yöntemdir.

Bu yöntemin en önemli avantajı, önceden karşılaşılmamış saldırıların da tespit edilebiliyor olmasıdır. Dezavantajı ise, aslında saldırı olmayan normal ağ trafiğinin saldırı olarak belirlenme oranının yüksek olmasıdır (Özekes, ve diğerleri, 2019).

2.2.2. Kötüye kullanım tespiti

Kötüye kullanım tespitinde amaç, saldırganların davranış modellerinin çıkarılmasıdır. Önceden tespit edilmiş ve kaydedilmiş olan saldırı bilgileri ile eşleştirilen ağ trafiği hareketleri saldırı olarak tanımlanır (Özekes, ve diğerleri, 2019).

Bu yöntemin en önemli avantajı, önceden tespit edilmiş her saldırının tespiti ve YP (Yanlış Pozitif) değeri üretmemesidir. Dezavantajı ise, önceden tespit edilememiş saldırıların tespitin mümkün olmaması ve YN (Yanlış Negatif) oranının yüksek olabilmesidir.

2.3. Saldırı Tespit Sistemlerinde Kullanılan Yöntemler

Saldırıları tespit etmek amacıyla farklı yöntemler (Baykara, 2016) kullanılmaktadır. Literatürde yer bulan, yapılmış güncel uygulamalarda kullanılan yöntemlerin başlıcaları aşağıda sunulmuştur.

- *Metin madenciliği:* Metin madenciliği, metinlerin yarı otomatik veya tam otomatik olarak işlenmesi süreçlerini içeren bir yöntemdir.
- *Topluluk yöntemi:* Topluluk yöntemi, güçsüz olarak tespit edilmiş birçok öğrenme algoritmasının birleştirilerek daha güçlü bir öğrenme algoritmasının oluşturulmasını sağlamaktadır.
- *Bağışıklık sistemleri:* Bağışıklık sistemleri, gerçek dünyadaki canlıların bağışıklık sistemi yapısının matematiksel bir modellemeyle farklı problemlerde kullanılmasını sağlayan yapay zekâ yöntemidir.
- *Veri madenciliği:* Veri madenciliği, veriler ve modeller arasındaki ilişkileri bularak yeni kurallar çıkarmak için kullanılır.
- *Kural tabanlı sistemler:* Kural tabanlı sistemler, ağ trafiğini inceleyip çeşitli kurallar oluşturur ve saldırı analizi aşamasında kural veri tabanında belirlenmiş olan kurallara göre değerlendirirler.
- *Açıklayıcı istatistikler:* Açıklayıcı istatistikler, kullanıcı veya sistemin davranışsal yapısını izleyerek bir model oluştururlar. Modeli oluşturmak için istatistiksel yöntemler kullanılarak çeşitli değişkenlerin yardımından faydalanılmaktadır.
- *Eşik değeri tespiti:* Bu yöntemde, ağ trafiğindeki paket hareketleri belirlenen eşik değerlerinin aşılması durumunda saldırı olarak değerlendirilmektedir. Eşik değer tespiti tek başına yeterli değildir ancak büyük STS yapılarında yardımcı olarak değerlendirilmektedirler.

- *Durum geçiş analizi:* Bu yöntem, saldırının gerçekleştirilmesi için birbirini izleyen ağ hareketlerinin bir sırası olduğu varsayımına dayanır. Bu sıraya göre bir durum değişim serisi oluşturulur.
- *Uzman sistemler:* Uzman sistemler, belirli bir alana yönelik problemlere o alandaki uzman bir kişinin çözümler ürettiği gibi çözüm üretebilen sistemler olarak tanımlanabilir.
- *Örüntü eşleme:* Bu yöntemde, sistem tarafından saldırı olduğunu belirten durum daha önceden tanımlanmıştır. Böylece sistemde gerçekleşen ağ trafiğinde önceden tanımlanan durum gerçekleşirse saldırı uyarısı verilebilmektedir.

Çizelge 2.1.'de STS'lerde kullanılan yöntemler temel özelliklerine göre karşılaştırılmaktadır (Baykara, 2016).

Çizelge 2.1. Saldırı Tespit Sistemlerinde Kullanılan Yöntemlerin Karşılaştırılması

Yöntem	STS Yaklaşımı	Bilgi Kaynağı	Bilinen Saldırı	Bilinmeyen Saldırı	Başarım
Metin Madenciliği	Anomali	Denetim verisi, Bilgi tabanı, Log kayıtları	Evet	Evet	Orta
Topluluk Yöntemi	Anomali	STS Bilgi Tabanı	Evet	Evet	Orta
Bağışıklık Sistemleri	Anomali	Denetim Verisi, Bilgi Tabanı	Evet	Evet	Yüksek
Veri Madenciliği	Anomali	Denetim Verisi, Bilgi Tabanı	Evet	Evet	Orta
Açıklayıcı İstatistikler	Anomali	Denetim Verisi, Kullanıcı Profili	Evet	Evet	Orta
Durum Geçiş Analizi	Kötüye Kullanım	Denetim Kayıtları, Bilinen Saldırı Durum Geçiş Diyagramları	Evet	Hayır	Yüksek
Örüntü Eşleme	Kötüye Kullanım	Denetim Kayıtları, Saldırı İmzaları	Evet	Hayır	Yüksek
Uzman Sistemler / Kural Tabanlı Sistemler	Kötüye Kullanım	Kural Bilgi Tabanı	Evet	Hayır	Yüksek

2.4. Saldırı Tipleri

STS'de saldırılar, bilgisayar sistemine yapılan saldırı çeşitlerinin kullandıkları yöntemlere göre 4 temel kategoride incelenmektedirler (Yıldırım, 2017).

2.4.1. Bilgi tarama saldırısı

Probe ya da *scan* olarak da bilinen bu saldırılar, bir sunucunun veya terminalin, IP adreslerini, kullandıkları portları veya işletim sistemini öğrenmek amacıyla yapılan saldırılardır (Yıldırım, 2017).

- *Ipsweep:* Belirli bir portun sürekli olarak taranma işlemi yapılması.
- *Portsweep:* Bir sunucuda çalışan hizmetleri bulmak için yapılan tüm portların taranması.

2.4.2. Hizmet engelleme saldırısı

DoS (Denial of Service) olarak bilinen bu saldırılar, TCP/IP protokolünde bulunan açıklardan faydalanmakta veya sisteme çok sayıda paket isteği göndererek sunucuya erişimi engellemektedirler (Yıldırım, 2017).

- *Smurf:* ICMP mesajlarının broadcast ile tüm ağa dağıtılması.
- *Selfping:* Kullanıcının kendi makinasına sürekli ping atması.
- *Tcpreset:* Saldırganın, kurbanın kurmaya çalıştığı bağlantıları engellemek amacıyla kurban adına reset paketi göndermesi.
- *Mailbomb:* Saldırganın sunucuya sürekli olarak mail göndermesi.

2.4.3. Yönetici hesabı ile yerel oturum açma

R2L (Remote to Local) olarak da bilinen bu saldırılar, sisteme erişim sağlamak amacıyla misafir kullanıcısıyla ya da sistemde tanımlı başka bir kullanıcı gibi giriş yapılmasıdır (Yıldırım, 2017).

- *Sshtrojan:* Unix'de çalışan bir trojan saldırısı.
- *Guest:* Tahmin yöntemiyle kolay şifreleri bularak sisteme erişim sağlanması.

2.4.4. Kullanıcı hesabının yönetici hesabına yükseltilmesi

U2R (User to Remote) olarak da bilinen bu saldırılar, yönetici olmayan bir kullanıcının sadece yönetici yetkisi ile gerçekleştirilebilecek işlemleri yapmaya çalışmasıdır (Yıldırım, 2017).

- *Eject:* Solaris üzerinde eject programı ile yönetici yetkilerine sahip olunması.
- *Sqlattack:* Linux makinalarda sunucuya bağlanan kullanıcının SQL kodları yardımıyla yönetici yetkilerine sahip olması.

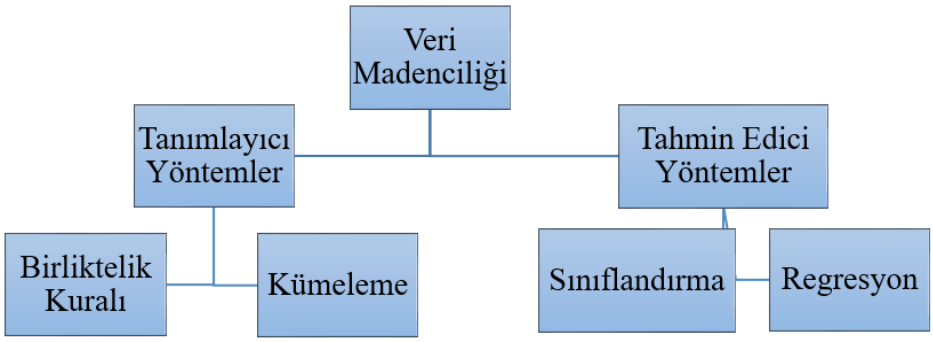
3. VERİ MADENCİLİĞİ, MAKİNE ÖĞRENMESİ VE TOPLULUK ÖĞRENMESİ

3.1. Veri Madenciliği

Veri madenciliği, büyük yığınlar halindeki veriler içerisinde bulunan, geleneksel yöntemler ile çıkarılması mümkün olmayan bilgilerin, ortaya çıkarılmasını sağlayan bir kavramdır. Veri madenciliği temelinde, çeşitli yazılım yöntemleri kullanılarak büyük verilerin analiz edilmesi yer almaktadır (Yiğidim, 2012; Akçay, 2014).

3.2. Veri Madenciliği Yöntemleri

Veri madenciliği yöntemleri Şekil 3.1.'de grafiksel olarak gösterilmektedir. Şekil 3.1'e göre veri madenciliği yöntemleri; tahmin edici ve tanımlayıcı olmak üzere iki ana başlık altında incelenmektedir. Tahmin edici yöntemler, bilinen verilerden yararlanarak, bilinmeyen bir değeri tahmin etmeye çalışırlar. Tanımlayıcı yöntemler ise verilerdeki gizli ortak özellikleri ve ilişkileri araştırırlar.



Şekil 3.1. Veri Madenciliği Yöntemleri

3.2.1. Tahmin edici yöntemler

Sonuçları bilinen veriler incelenerek yeni bir yöntem geliştirilir. Kurulan bu yöntemden yararlanarak sonuçları bilinmeyen veri kümeleri için yeni sonuçlar öngörmeye çalışmaktır. Sınıflandırma ve regresyon olmak üzere iki yönteme ayrılır.

- *Regresyon:* Regresyon analizi, bağımlı bir değişkenin, bir veya birden fazla bağımsız değişkenle oluşturduğu ilişki yapısının bir fonksiyon şeklinde yazılıp ve bu fonksiyon yardımıyla bağımlı değişkenin ulaşabileceği değerlerin tahmin edilebilmesidir (Akçay, 2014).
- *Sınıflandırma:* Sınıflandırma yöntemi için asıl önemli olan her bir sınıfın özelliklerinin önceden net bir şekilde belirlenmiş olmasıdır. Önceden belirlenen sınıflar, verinin sınıflandırılması için, yöntem geliştirmede kullanılır.

Regresyon ve sınıflandırma modelleri karşılaştırıldığında; tahmin edilecek alan eğer sayısal bir değişkense bir regresyon, kategorik bir değişkense sınıflandırma yöntemi kullanılmaktadır (Akçay, 2014).

3.2.2. Karar ağaçları sınıflandırma yöntemi

Karar ağaçları, regresyon ve sınıflandırma işlemlerinde kullanılan makine öğrenme yöntemlerinden biridir. Oluşturulma bakımından ucuz, yorumlama açısından kolay ve yüksek güvenilirliğe sahip olmasından dolayı yaygın olarak kullanılmaktadır. Ana fikir, grubun tüm elemanları aynı sınıf etiketine sahip olana kadar bir kümeleme algoritmasıyla iterasyon şeklinde gruplara bölünmesidir (Akçay, 2014; Aydın, 2018; Yılmaz, 2019).

Dezavantajları olarak sırasıyla; kararsız yapıda olmaları, ağacın hiyerarşik yapısından dolayı sonucun en ufak değişiklikte bile değişmesi, aşırı öğrenme ya da aşırı uyumun yaşanması, ağaç oluşturma ve ağaç budama karmaşıklığının çok fazla olması, sınıf sayısının fazla, öğrenme kümesi sayısının az olması belirtilebilir (Yangın, 2019).

Karar ağacı sınıflandırma yöntemi, ağaç yapısını kullanarak tahminleme yapar. Bu ağaç yapısını açıklamak gerekirse; ağacın düğümlerinde; karar değişkenleri, yapraklarında ise; tahmin edilen hedef değerler vardır. Ayrıca karar ağacının kalitesi ve performansı, ağacın büyüklüğüne ve sınıflandırma doğruluğuna bağlıdır. Bu sebeple düğümlerin belirlenmesi çok önemlidir. Ağacı inşa etmek için hangi özniteliklerin kullanılacağı önceden belirlenmelidir (Gülaçar, 2018; Özekes, ve diğerleri, 2019).

3.2.3. Tanımlayıcı yöntemler

Tanımlayıcı yöntemler, büyük veri kümelerindeki desen ve ilişkileri belirlemeyi, verileri birbirleriyle ilişkilendirerek anlamlandırmayı amaçlamaktadır (Akçay, 2014).

- *Kümeleme:* Heterojen yapıda dağılmış verileri belirledikten sonra benzerlikleri bulunan verileri bir araya toplayarak homojen alt kümeler oluşturmaktadır. Kümeleme analizi, veri kümesinde önceden belirlenmemiş fakat veri kümesinde var olan altsınıfları bulmaktadır (Akçay, 2014).

- *Birliktelik Kuralı:* Veri kümeleri arasında olası ilişkilerin biçimini tanımlamayı amaçlayan bir yöntemdir. Değişkenlerin arasındaki ilişkinin büyüklüğünü ve yakınlığını saptamaya çalışmaktadır (Akçay, 2014).

3.3. Makine Öğrenmesi

Literatüre yeni yayınlar eklendikçe ve yeni çalışmalar yapıldıkça tanımlamalar da değişmekte ve gelişmektedir. Makine öğrenmesi teriminin literatürdeki tanımının zamanla değişimini göstermesi amacıyla farklı tanımlamalar incelendiğinde; Yiğidim (Yiğidim, 2012), "*Makine Öğrenmesi kavramı, makinelerin karşılaştıkları durumlar karşısında kendini eğiterek daha iyi kararlar verebilmesini sağlayan algoritmaların geliştirilmesi olgusudur*." şeklinde belirtmişken; Türkmenoğlu (Türkmenoğlu, 2015) makine öğrenmesini "*Makine Öğrenmesi, istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma gibi alanlarla yakından ilintilidir*." olarak tanımlamaktadır. Teknolojinin gelişmesiyle beraber tanımlamalarda yapay zeka ibaresi yer almaya başlamıştır. Yıldırım (Yıldırım, 2017) tanımlamasını "*Yapay Zekâ'nın bir alt dalı olan makine öğrenmesi belirli algoritma ve teknikler kullanarak bilgisayarların insanlar gibi öğrenmesini sağlayacak bir çalışma alanıdır.*" şeklinde yapmaktayken, Yılmaz (Yılmaz, 2019) "*Makine Öğrenmesi, veriye dayalı öğrenim gerçekleştirerek yeni verilere maruz kaldıklarında yeni veri hakkında tahmin yapabilmeyi amaçlayan Yapay Zekâ'nın alt alanıdır*." ifadesiyle tanımlamaktadır.

Makine öğrenmesi süreci incelendiğinde veri madenciliği süreci ile benzerlikler gösterdiği belirtilebilir. Bu iki süreç de veri kümesi üzerinde çalışarak desenler aramakta ve bu desenleri kullanarak sonuç elde etmeye çalışmaktadırlar. Bu konuda ayrıldıkları kısım olarak öğrenme işlemini programın mı kullanıcının mı yapacağı olarak belirtilebilir. Bu durumu Türkmenoğlu, (Türkmenoğlu, 2015) "*Veri Madenciliği veriyi insanların karşılaştırıp bilgi çıkarabilmeleri için elde ederken, Makine Öğrenmesi elde ettiği bilgiyi programın kendi öğrenme becerisini geliştirmesi için kullanır*." şeklinde açıklamaktadır.

3.4. Makine Öğrenmesi Yöntemleri

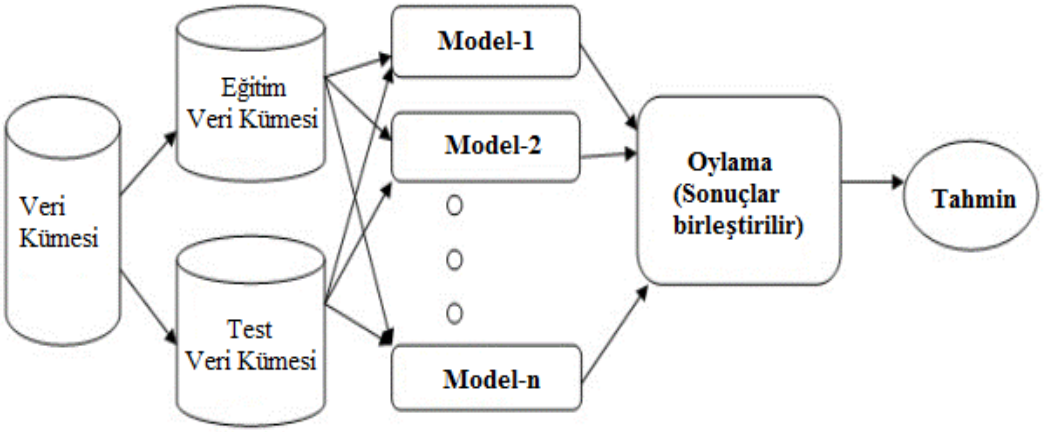
Makine öğrenmesi yöntemleri, denetimli ve denetimsiz olmak üzere iki grup altında incelenmektedir. Günümüzde teknolojinin hızla ilerlemesi ile beraber bunlara, yarı denetimli, takviyeli öğrenme, transdüksiyon ve öğrenmeyi öğrenme yöntemleri de eklenmiştir.

- *Denetimli öğrenme:* Etiketlenmiş eğitim veri kümesi üzerinde öğrenme tekniklerini uygulayan bir makine öğrenmesi yöntemidir. Bu yöntemde, girdiler ile hedef çıktılar arasında ilişki kurabilen bir fonksiyon üretilir (Yılmaz, 2019).
- *Denetimsiz öğrenme:* Girdilere bağlı olmadan çıkarım yapabilen bir yöntemdir. Burada amaç, mevcut verileri arttırarak daha fazla bilgi edinmek ve arttırılmış verilerle genel yapıyı modellemektir. Denetimli öğrenmeden farkı, sonuç kümesi ve eğitim kümesi olmamasıdır.
- *Yarı-denetimli öğrenme:* Etiketlenmiş ve etiketlenmemiş veri kümelerini beraber ele alarak inceleyip uygun fonksiyon ve sınıflandırıcı oluşturur.
- *Pekiştirmeli öğrenme:* Etrafını algılayan ve kararlarını kendisi verebilen bir sistemin, hedefe ulaşabilmesi için gerekli kararları nasıl alacağını öğretir.
- *Transdüksiyon:* Denetimli öğrenmeye benzeyen bu yöntem, girdilere, çıktılara ve üretilen girdilere bakarak yeni çıktılar için tahminleme işlemi yapmaktadır.
- *Öğrenmeyi öğrenme:* Bu yöntem de, önceki deneyimlere dayanarak bir öğrenme gerçekleştirmektedir.

3.5. Topluluk Öğrenmesi

Topluluk Öğrenmesi, gelişmiş sınıflandırıcılar yaratmak amacıyla, düşük performanslı sınıflandırıcıları birleştiren bir modeldir. Topluluk öğrenmesinin ana fikri, tek bir sınıflandırıcı kullanarak elde edilmiş olan doğru tahmin oranını arttırmak için birden fazla sınıflandırıcının birleştirilebileceği düşüncesine dayanmaktadır. Topluluk öğrenmesinin gerçek hayattaki uygulamaları incelendiğinde, problemleri çok yönlü incelemesi ve çözümlemesiyle ön plana çıkmaktadır. Karar verme sürecinde, doğru karara ulaşabilmek için, farklı düşüncelerden de faydalanarak değerlendirmektedir. Topluluk öğrenmesi yöntemleri, biraz daha iyi olan zayıf modellerden kesin tahminler yapabilen güçlü modellere yükseltebilme özelliğine sahiptir (Yangın, 2019; Yılmaz, 2019).

Şekil 3.2.'de yaygın olarak kullanılan topluluk öğrenmesi yapısı görülmektedir. Kullanılan veri kümesi, test ve eğitim veri kümesi olmak üzere ikiye ayrılmıştır. Her iki veri kümesine de birçok model uygulanarak daha güçlü modeller elde edilmiştir. Modeller tarafından elde edilen sonuçlar kümesi birleştirilir. Sürecin sonucunda da en iyi tahmine ulaşmak amaçlanmaktadır (Yangın, 2019).



Şekil 3.2. Topluluk Öğrenme Yapısı

Topluluk öğrenmesi, birçok sınıflandırıcıyı birleştirmek için, Rastgele Orman algoritması, oylama, yığınlama, torbalama ve arttırma yöntemlerini kullanmaktadır (Yılmaz, 2019).

3.5.1. Rastgele orman algoritması

2001 yılında Breiman tarafından bulunmuş olup, en gelişmiş topluluk yöntemlerinden birisi haline gelmiştir. Rastgele Orman algoritması, torbalamanın genişletilmiş hali olup, iki algoritma arasındaki en büyük fark, rastgele hale getirilmiş öznitelik seçiminin birleştirilmesidir.

İşlemsel açıdan bakıldığında, regresyon ve sınıflandırmayı kolaylıkla ele almakta olup, eğitme ve tahminleme işlemlerinde nispeten daha hızlı olması, sadece bir veya iki parametre ayarına bağlı olması, çok boyutlu problemlerde rahatlıkla kullanılması gibi nedenlerden dolayı oldukça fazla tercih edilmektedir. İstatistiksel olarak ise, değişkenlerin önem seviyeleri, sınıf ağırlığı, görselleştirme, aykırı değerlerin tespiti, denetimsiz öğrenme gibi ek özellikler bakımından da tercih edilmektedir. Rastgele Orman algoritması, tek bir veri kümesi içerisindeki en iyi özniteliği aramak yerine rastgele öznitelikler kümesi içinde en iyi özniteliği aramaktadır. Bu da düşük varyans ve daha yüksek yanlılık değerinin oluşmasına ve bunun sonucunda da ağaç çeşitliliğinin daha büyük olmasına olanak sağlamaktadır (Yangın, 2019).

3.5.2. Oylama (voting) algoritması

Oylama algoritması, birleştireceği temel sınıflandırıcılardan her birinin tahminlerini toplayarak en çok oy alan etiketi tahmin etmektedir. Oylama yönteminde sınıflandırıcıların ağırlığı birbirine eşittir. Oylama algoritması, çoğu zaman en yüksek doğruluk oranı sağlayan sınıflandırıcıdan bile daha yüksek bir doğruluk değeri elde edebilmektedir. (Yılmaz, 2019).

3.5.3. Yığınlama (stacking) algoritması

Çoklu sınıflandırma modelleri tahminlerini yeni bir veri kümesinde birleştiren bir topluluk öğrenme yöntemidir. Elde edilen bu yeni veri kümesi, başka bir sınıflandırıcı için giriş verileri olarak tanımlanmaktadır. Yığınlama algoritmasında, sınıflandırma modelleri orijinal eğitim veri kümesi üzerinde eğitilmektedir. Daha sonra, sınıflandırma modellerinin çıktıları temel alınarak bir ana sınıflandırıcı oluşturulmaktadır (Yangın, 2019; Yılmaz, 2019).

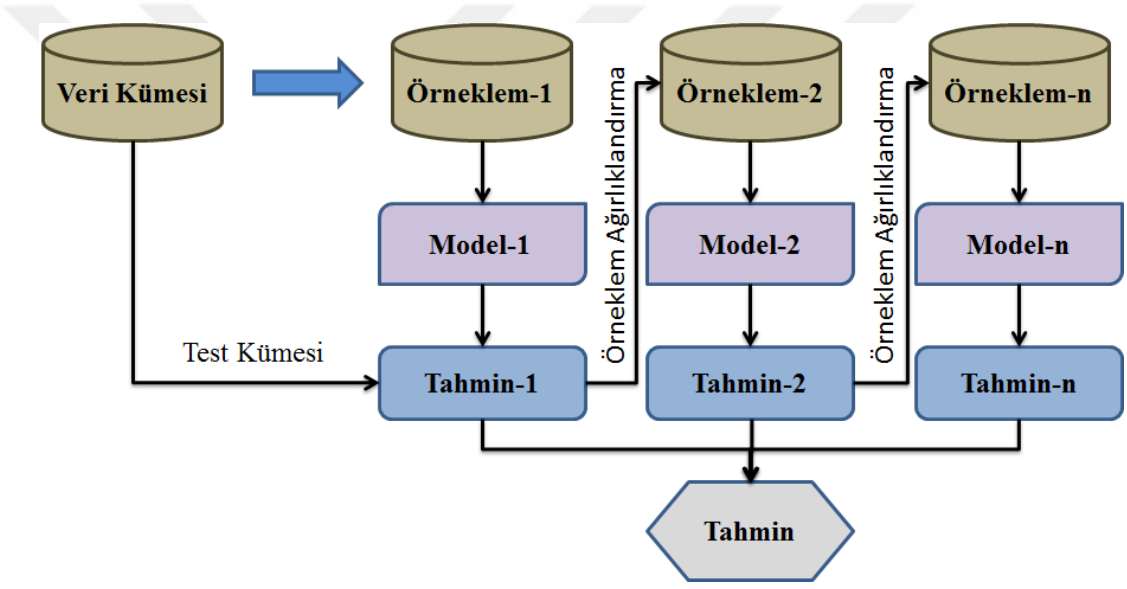
3.5.4. Torbalama (bagging) algoritması

Torbalama algoritması, bootstrap (önyükleme) ve aggregating (toplama) teriminin kısaltılmış hali olup, Breiman tarafından 1996 yılında bulunmuştur (Yangın, 2019).

Torbalama algoritmasının amacı, eğitim veri kümesi kullanarak rastgele yeni veri kümeleri oluşturup farklılıkların oluşmasını ve böylece sınıflandırma başarısını arttırmaktır. Ayrıca, eğitim veri kümesinden, yerine koymalı rastgele seçimle yeni eğitim kümeleri türeterek sınıflandırma modelini yeniden eğitmeyi amaçlamaktadır. Torbalama yönteminde yapılması gereken ilk adım, veri kümesini eğitim kümesi ve test kümesi olarak bölmektir. N tane örneklem içeren eğitim kümesinden, yerine koymalı rastgele seçim yöntemiyle yine n tane örnekleme sahip bir veya birden fazla yeni bir eğitim kümesi üretilir. Torbalama yöntemiyle oluşan topluluktaki her bir sınıflandırıcı, oluşturulmuş farklı örneklemler içeren eğitim kümesi ile eğitilir. Her bir sınıflandırıcının sonucu çoğunluk oylaması ile birleştirilir (Yangın, 2019; Yılmaz, 2019).

3.5.5. Arttırma (boosting) algoritması

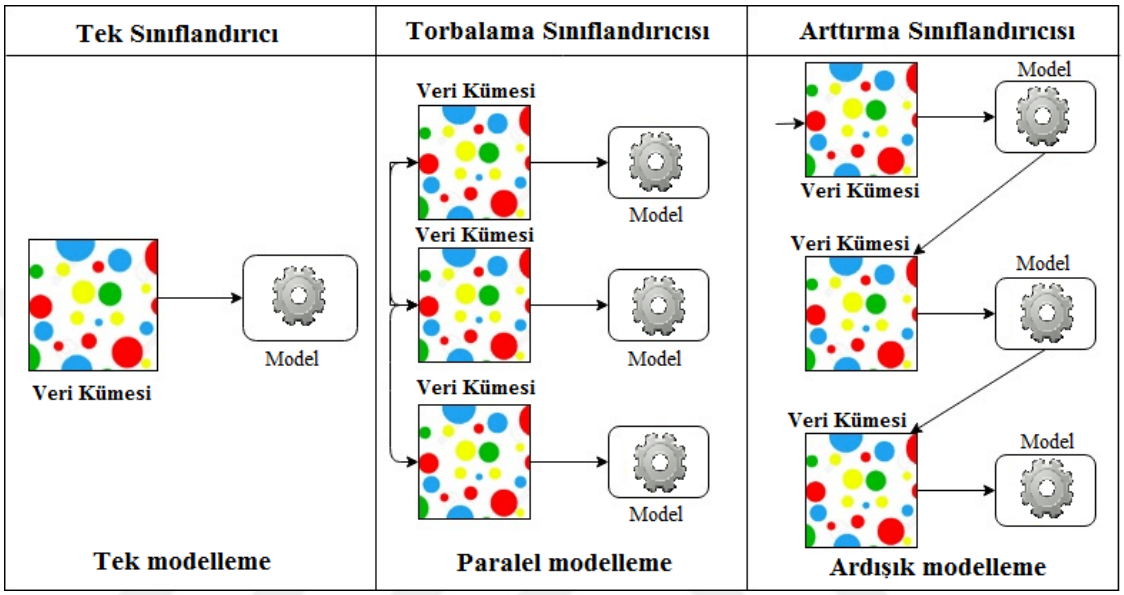
Schapire tarafından 1989 yılında geliştirilmiştir. Freud ve Schapire ve Friedman'ın son çalışmaları ise bu algoritmayı daha da geliştirmiştir. Arttırma algoritması, yavaş öğrenmeye dayalı, sıralı bir yöntemdir ve hatadan öğrenmeyi amaçlamaktadır. Bu algoritmalar, yüksek hassasiyetli modeller oluşturmak için düşük hassasiyetli birkaç modeli birleştirmektedir. Genel prensip olarak, her iterasyonda elde edilen modelleri, belli kurallar çerçevesinde birleştirerek güçlü bir model elde etmeye çalışmaktadır (Yangın, 2019; Yılmaz, 2019).



Şekil 3.3. Arttırma Yöntemi Çalışma Yapısı

Şekil 3.3.'de genel hatları ile gösterilmekte olan arttırma algoritması sürecinde; ilk önce eğitim veri kümesinden rastgele örneklemler oluşturulur (Yangın, 2019). Bu örneklem için bir sınıflandırıcı (örneğin Model-2) eğitilir ve tüm eğitim veri kümesi test edilir. Her bir örneklem tahmini için hata hesaplanır. Eğer örneklem yanlış sınıflandırılmışsa, o örneklem için ağırlık arttırılır ve başka bir örneklem oluşturulur. Sistemden yüksek doğruluk elde edilene kadar bu işlemler tekrarlanır.

Torbalama ve arttırma algoritmalarının karşılaştırılması Şekil 3.4.'de verilmiştir (Yangın, 2019). Eğitim aşamasında torbalama algoritması için model oluşturulması paralel olarak gerçekleşirken, arttırma algoritmasında ise yeni veri kümeleri ardışık bir şekilde oluşturulmaktadır.



Şekil 3.4. Sınıflandırma Algoritmalarının Karşılaştırılması

Arttırma algoritması, torbalama algoritmasından daha düşük hata oranları göstermektedir. Karar ağaçlarının sınıflandırma için etkili olduğu veri kümelerinde arttırma algoritması, sınıflandırma oranlarını biraz daha iyi seviyelere getirmektedir (Yangın, 2019). Arttırma algoritmaları kapsamında, AdaBoost, Gradyan Arttırma, XGBoost, LightGBM ve CatBoost yöntemleri bulunmaktadır.

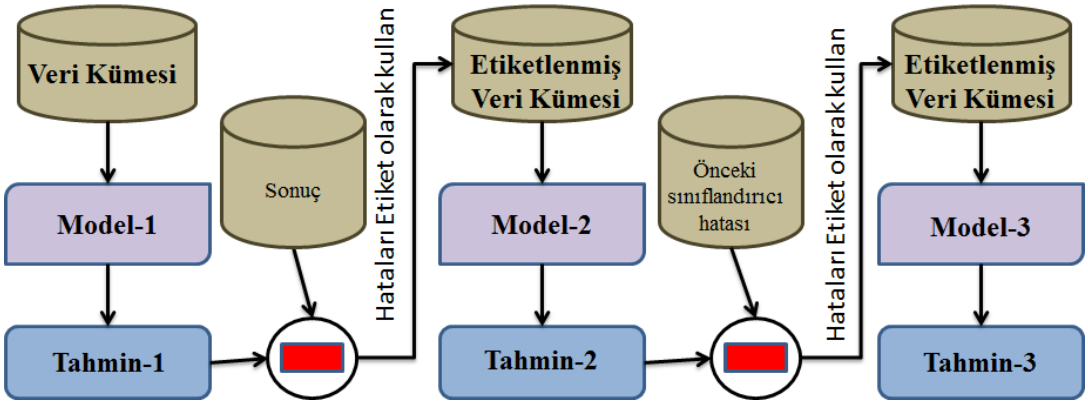
3.5.5.1. *AdaBoost algoritması*

Freund ve Schapire tarafından 1995 yılında geliştirilmiştir. Arttırma yöntemini kullanarak zayıf öğrenme algoritmalarının performanslarının arttırılmasını sağlayan bir sınıflandırma algoritmasıdır. Öznitelik uzayındaki her bir öznitelik üzerinde bir zayıf sınıflandırıcıyı eğitir ve doğrusal olarak birleştirerek güçlü sınıflandırıcıların elde edilmesini sağlar.

Öğrenme kümesi üzerinde her modelin bir ağırlık katsayısı vardır. Her bir öznitelik için zayıf olan sınıflandırıcı eğitildikten sonra yanlış sınıflandırılan modelin ağırlık katsayıları arttırılırken doğru sınıflanan örneklerin ağırlık katsayıları azaltılmaktadır. Adaboost algoritması, optimizasyon işlemlerini içerir. İşlemler sonrasında son oluşan sınıflandırma zayıf sınıflandırma algoritmalarının bir lineer kombinasyonudur. Bu algoritma, diğer arttırma yöntemlerden farklı olarak büyük veri kümesine ihtiyaç duymamaktadır.

3.5.5.2. *Gradyan arttırma algoritması*

Friedman tarafından 2001 yılında tanıtılan güçlü bir makine öğrenme tekniğidir. Gradyan arttırma algoritması, sınıflandırma problemlerinde kullanılmak üzere önerilmiş, zamanla regresyon problemlerinde de tercih edilmeye başlamıştır. İteratif olarak hatayı tahminlemeye çalışıp modeli iyileştirmeye çalışmaktadır. Algoritmaya, rastgeleliği eklemek metodun hızını ve doğru tahmin yapma olasılığını arttırmaktadır. Bu sebeple, alt veri kümeleri, eğitim veri kümesinden rastgele olarak seçilir. Bu seçim sayesinde, aşırı uyum sorununa karşı da önlem alınmış olur (Ekiz, 2019; Kaya, 2019; Yangın, 2019).



Şekil 3.5. Gradyan Arttırma Yöntemi Çalışma Yapısı

Şekil 3.5.'de Gradyan arttırma algoritmasının aşamaları genel hatlarıyla belirtilmektedir. Öncelikle ilk veri kümesinden bir karar ağacı oluşturulur ve tahmin

ile çıktı arasındaki hata bulunur. Veri kümesinin örneklemleri için yeni çıktı değerleri olarak bu hata kullanılır. Veri kümesindeki hatalarla birlikte örneklemin etiketi olarak yeni bir karar ağacı oluşturulur ve önceki ağacın yarattığı hatayı yeniden oluşturmak için ağaç eğitilir. Önceki çıktı ve mevcut tahmin arasındaki hata istenilen seviyeye ulaşıncaya kadar sürece ağaç eklenmeye devam edilir (Yangın, 2019).

3.5.5.3. *XGBoost algoritması*

XGBoost, Extreme Gradient Boosting teriminin kısaltılmış hali olup, temeli gradyan arttırma ve karar ağacı algoritmalarına dayanan bir makine öğrenme yöntemidir. XGBoost algoritmasının orijinal hali Friedman tarafından 2002 yılında (Yangın, 2019) geliştirilmiştir. XGBoost, hem sınıflandırma hem de regresyon modelleri için yüksek doğruluk oranına sahiptir. Ayrıca diğer algoritmalardan 10 kat daha hızlıdır. XGBoost, performansı iyileştiren ve aşırı uyum ya da aşırı öğrenmeyi azaltan bir dizi düzenleme içerir ve bu sayede daha iyi performans elde etmeyi başarmaktadır. Ayrıca, parametre almadan kendi içerisinde çapraz doğrulama yaparak modelin doğruluğunun maksimum olmasını sağlamaktadır (Ekiz, 2019).

Bu algoritma, yüksek esneklik özelliği sayesinde, modele yeni bir boyut katan kriterin değerlendirilmesi ve en uygun amaç fonksiyonun tanımlanmasını mümkün hale getirmektedir. XGBoost, kayıp değerleri işlemek için yerleşik bir rutine sahiptir. Ek olarak XGBoost algoritması ağaç budaması yaparken, belirtilen “max_depth” parametre değerine kadar dallara ayırma işlemini yapar ve sonrasında ağacı geriye doğru budamaya başlar. Ayrıca, herhangi bir olumlu katkısı olmayan dallanmaları kaldırır (Yangın, 2019). Bu algoritma sayesinde karar ağacı çok büyümediği için, bu algoritmanın performansı diğer arttırma yöntemlerine göre çok daha yüksek olmaktadır.

3.5.5.4. *LightGBM algoritması*

LightGBM, Microsoft'un DMTK projesi kapsamında 2017 yılında geliştirilmiş olan bir gradyan arttırma algoritmasıdır. Regresyon, sınıflandırma ve sıralama çözümlemelerinde kullanılmaktadır. Dağıtık sistemlerle çalışan, özel tanımlanmış hata fonksiyonunu destekler. Sürekli değere sahip olan değişkenleri kategorik değere dönüştürerek modelin eğitim süresini kısaltmakta ve kaynak kullanımını azaltmaktadır. LightGBM yaprak tabanlı bir yapıda çalışmaktadır. Böylece ağaç, öncelikli olarak yatayda büyümekte olduğundan ve ağaç derinliği fazla artmadığından dolayı aşırı öğrenmenin önüne geçilebilmektedir (Gümüştaş, 2019).

3.5.5.5. *CatBoost algoritması*

CatBoost, kategorik özellikleri destekleyen, gradyan arttırma algoritmasıdır. Hem GPU hem de CPU uygulamalarında kullanılabilmektedir. CatBoost, kategorik özniteliklere sahip verileri sınıflandırmak için kullanılmaktadır (Tozlu, 2019).

3.6. Model Performans Değerlendirme Ölçütleri

Model değerlendirme ölçütleri, model ve veri arasındaki uyumu değerlendirmek, modelleri birbirleriyle karşılaştırmak ve model seçimi yapılması için elde edilen tahminlerin doğruluğunu tespit etmek amacıyla kullanılır. Bir modelin performansının değerlendirilmesi için, bazı istatistiksel ölçütler hesaplanmaktadır. Bir modelin başarım değerlerini ölçmek için kullanılan ölçütler aşağıda belirtilmiştir. (Tomak, ve diğerleri, 2009)

- *Doğruluk*
- *Duyarlılık*
- *Kesinlik*

- *F1 Değeri*
- *Kohen'in Kappa Değeri*
- *ROC eğrisi ve Eğri Altında Kalan Alan*
- *Karmaşıklık Matrisi*

3.6.1. Doğruluk

Pozitif ve negatif sonuçların toplam gözlem sayısına oranıdır. Başka bir deyişle doğru sınıflandırılan sonuçların tüm sonuçlara bölünmesi ile elde edilir ve (3.1) eşitliği ile gösterilmektedir. Dengesiz dağıtılmış veri kümesinin sonuçları değerlendirilirken doğruluk oranı yanıltıcı olabilir. Çünkü veri kümesinin ağırlıklı olduğu tarafa doğru bir kayma söz konusu olabilmektedir. Az miktarda olan özniteliğin gerçekte doğru sınıflandırıldığından emin olunmaz. Bu nedenle dengesiz dağıtılmış veri kümelerinde kullanılması hatalı sonuçlara neden olabilmektedir (Yangın, 2019).

$$\text{Doğruluk} = \frac{DP + DN}{DP + DN + YP + YN} \tag{3.1}$$

3.6.2. Kesinlik

Gerçek değeri pozitif olan ve pozitif olarak da sınıflandırılan (DP) gözlem sayısının, tahmin değeri pozitif olan tüm gözlemlere oranıdır ve (3.2) eşitliği ile hesaplanmaktadır (Yangın, 2019).

$$\text{Kesinlik} = \frac{DP}{DP + YP} \tag{3.2}$$

3.6.3. Duyarlılık

Gerçek değeri pozitif olan ve pozitif olarak da sınıflandırılan (DP) gözlem sayısının, gerçek değeri pozitif, tahmini değeri de pozitif ve negatif olan gözlem sayılarının toplamlarının oranıdır ve (3.3) eşitliği ile hesaplanmaktadır. Gerçekte pozitif olan ve pozitif olarak da sınıflandırılan (DP) gözlem sayısı arttıkça duyarlılık da artar (Yangın, 2019).

$$\text{Duyarlılık} = \frac{\text{DP}}{\text{DP} + \text{YN}} \tag{3.3}$$

3.6.4. F1 değeri

F1 Score/F-Measure olarak da bilinen F1 değeri, kesinlik ve duyarlılık değerlerinin harmonik ortalaması ile elde edilir ve (3.4) eşitliği ile hesaplanmaktadır. F1 değeri, 0 ile 1 arasında değer alır ve aldığı değer 1'e yaklaşması modelin daha iyi olduğunu göstermektedir (Yangın, 2019).

$$\text{F1} = 2.\frac{\text{Kesinlik} \,.\, \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \tag{3.4}$$

3.6.5. Kappa değeri

Kappa skoru ya da Cohen'in Kappa katsayısı olarak da bilinen Kappa değeri, iki ya da daha fazla değerlendiriciye ait sonuçlar arasındaki uyumun güvenilirliğini ölçer ve (3.5) eşitliği ile hesaplanmaktadır. Burada p_o gerçek değeri, p_e ise tahmin edilen değeri temsil etmektedir.

$$\text{Kappa} = \frac{p_{o} - p_{e}}{1 - p_{e}} \tag{3.5}$$

Kappa değeri, -1 ile +1 arasında değer almaktadır. Kappa değeri +1 değerine yaklaştıkça değerlendiricilerin sonuçlarının birbiri ile uyumlu olduğu, 0 değerine yaklaştıkça değerlendiriciler arasındaki uyumun tamamen rassal olduğu ve son olarak da -1 değerine yaklaştıkça sonuçların birbiri ile ters değerlendirildiği şeklinde yorumlar yapılmaktadır (Yangın, 2019).

3.6.6. ROC eğrisi ve AUC

ROC eğrisi, ikili sınıflandırma probleminde kullanılmaktadır. Eğri altında kalan alan (AUC) değerinin 1'e yakın olması kullanılan modelin başarılı olduğunu gösterir. Kullanılan modellerin performanslarının karşılaştırılmasında ROC eğrisi ve AUC kullanılmaktadır. AUC değerinin de Kappa istatistik değeri gibi 1'e yakın olması istenmektedir. Çünkü AUC değeri ne kadar büyük olursa o kadar iyi bir uyumun olduğu söylenebilir (Göker, 2019).

3.6.7. Karışıklık matrisi

Karışıklık matrisi, sınıflandırma problemlerinde modellerin performansını değerlendirmek için kullanılmaktadır. Sınıflandırma işlemi sonucunda elde edilen değerlerin gerçek değerlerle karşılaştırılması Çizelge 3.1.'de gösterilmekte olan tablo yardımıyla yapılabilmektedir.

Çizelge 3.1. Karışıklık Matrisi

		Tahmin Edilen Veri	
		Pozitif	**Negatif**
Gerçek Veri	**Pozitif**	Doğru Pozitif (DP): Gerçek veride pozitif olan, tahmin edilen veride de pozitif olarak sınıflandırılmıştır.	Yanlış Negatif (YN): Gerçek veride pozitif olan, tahmin edilen veride ise negatif olarak sınıflandırılmıştır.
	Negatif	Yanlış Pozitif (YP): Gerçek veride negatif olan, tahmin edilen veride ise pozitif olarak sınıflandırılmıştır.	Doğru Negatif (DN): Gerçek veride negatif olan, tahmin edilen veride de negatif olarak sınıflandırılmıştır.

Karışıklık matrisi yardımıyla doğruluk, kesinlik, duyarlılık, F1 değeri, kappa değeri ve ROC eğrisi elde edilebilmektedir.

3.7. Bilgi Kazancı Hesaplaması

Karar ağacı yöntemini esas alan modellerde, ağacı dallara ayırma işlemi yapılırken hangi özniteliğin seçildiği oldukça önemlidir. Özniteliği seçmek için entropi, bilgi kazancı ve gini katsayısı kullanılmaktadır (Yangın, 2019).

3.7.1. Entropi

Entropi; rastgele özniteliğin belirsizliğinin ölçülmesinde kullanılan bir ölçüttür ve 0 ile 1 arasında değer almaktadır. Alınan değer 1'e yaklaştıkça belirsizlik artar. Belirsizliğin azaltılması için veri kümesi ile ilgili daha fazla bilgiye ihtiyaç vardır. Entropi değeri (3.6) eşitliği ile hesaplanmaktadır. Burada p_i, D veri kümesindeki 'i' sınıfının olasılığını gösterir ve 'i' sınıfındaki örneklem sayısının tüm veri kümesindeki toplam örneklem sayısına bölünmesiyle elde edilmektedir (Yangın, 2019).

$$\text{Entropi(D)} = -\sum_{i=1}^{n} p_i * \log(p_i) \tag{3.6}$$

3.7.2. Bilgi kazancı

Bilgi kazancını hesaplayabilmek için veri kümesindeki uygun bölünme noktasının belirlenmesi çok önemlidir. Bilgi kazancı değeri, 0 ile 1 arasında değer alır. D veri kümesindeki X özniteliğinin 'n' tane bölünmeye sahip olması durumunda, X özniteliğine ait bilgi kazancı değeri (3.7) eşitliği ile hesaplanmaktadır (Yangın, 2019).

$$\text{Bilgi Kazancı}\,(D, X) = E(D) - \sum_{k=1}^{n} p\,(D_i)E(D_i) \tag{3.7}$$

3.7.3. Gini indeksi

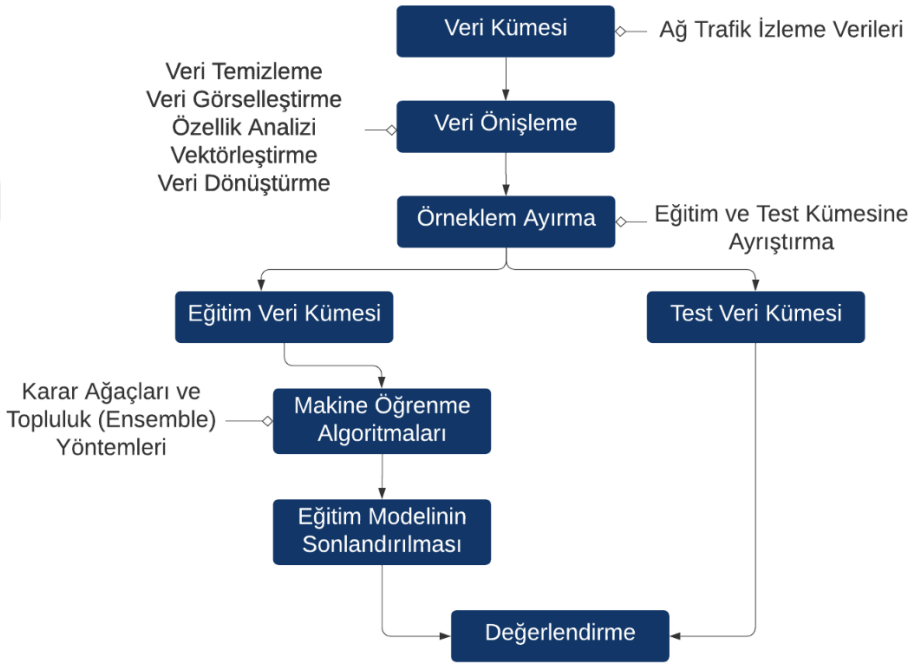
Gini indeksi, hedef özniteliğin değerlerinin olasılık dağılımları arasındaki farkı ölçen karışıklık temelli bir ölçüttür.

D veri kümesinde 'n' sınıf bulunuyorsa ve p_i 'i' sınıfına ait görülme sıklığı olmak üzere gini indeksi (3.8) eşitliği ile hesaplanmaktadır (Yangın, 2019).

$$\text{Gini}(D) = 1 - \sum_{i=1}^{n} p_i^2 \tag{3.8}$$

4. UYGULAMA

Bu bölümde, Bölüm 3'de anlatılan makine öğrenmesi yöntemlerinin CICIDS2017 veri kümesi üzerine uygulanması ve alınan sonuçların analizi anlatılmaktadır. Modelleme aşamasında Karar Ağaçları, AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost algoritmaları kullanılmıştır. Ayrıca, tez çalışmasında, CICIDS2017 veri kümesindeki etiketlenmiş ağ saldırılarının tespit edilmesinde karar ağaçları yöntemi, arttırma algoritmalarının başarımını karşılaştırmak amacıyla kullanılmıştır.



Şekil 4.1. Anomali Tespit Süreci

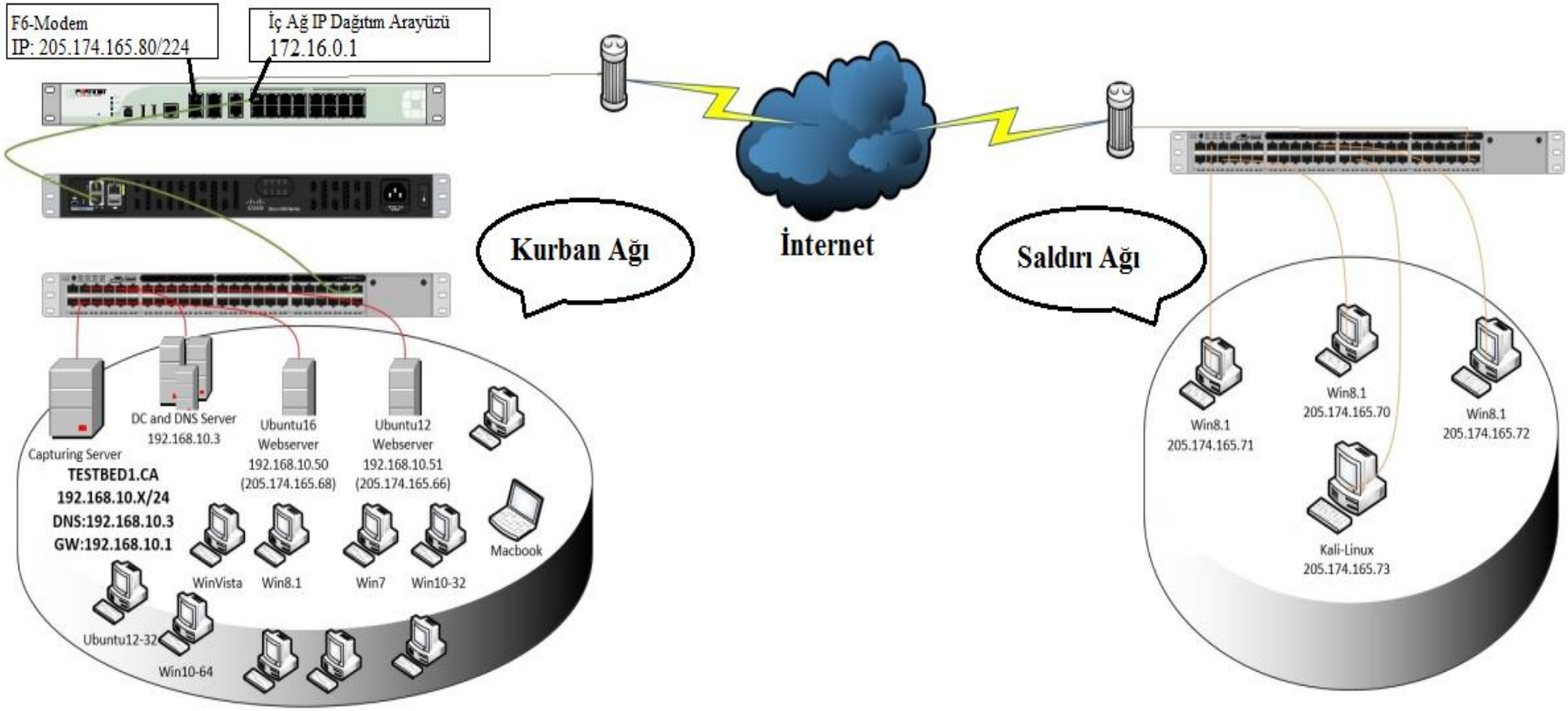
Anomali tespit sürecine dair hazırlanmış genel akışı belirten algoritma şekil 4.1.'de verilmiştir. Çalışma, Colab platformunda Jupyter Notebook sanal sunucusunda Python programlama dili kullanarak gerçekleştirilmiştir.

4.1. Veri Kümesi

Bu çalışmada Kanada Siber Güvenlik Enstitüsü tarafından oluşturulmuş olan CICIDS2017 veri kümesi kullanılmıştır. Bu veri kümesi 1998-2016 yılları arasında literatürde bulunan veri kümelerindeki eksiklikler dikkate alınarak oluşturulmuştur. H. Sharafaldin ve arkadaşları (Sharafaldin, ve diğerleri, 2018) tarafından, CICIDS2017 veri kümesi oluşturulurken, kapsamlı bir test ortamı gerçekleştirebilmek için saldırı ağı ve kurban ağı olmak üzere birbirinden tamamen ayrılmış 2 farklı ağ mimarisi tasarlanmış ve uygulanmıştır. Oluşturulan test ortamına ait görsel Şekil 4.2.'de verilmiştir (Sharafaldin, ve diğerleri, 2018; Özekes, ve diğerleri, 2019).

CICIDS2017 veri kümesi, B-Profil sistemi ile gerçekçi bir ağ trafiği oluşturularak derlenmiştir. Veri kümesi ayrıca kaynak ve hedef IP'leri, kaynak ve hedef bağlantı noktaları, protokoller ve saldırılara dayalı trafik analizi sonuçlarını da içerir. Bu veri kümesi, HTTP, HTTPS, FTP, SSH ve elektronik posta protokollerini temel alan toplam 25 internet kullanıcısının gerçekleştirdiği soyut davranıştan oluşturulmuştur (Hossen, ve diğerleri, 2018; Tama, ve diğerleri, 2020).

Literatürde kullanılan önceki veri kümelerinin aksine, kurban ağında, yönlendirici, güvenlik duvarı, anahtar dâhil olmak üzere tüm yaygın ve gerekli donanımları, Windows, Linux ve Macintosh gibi ortak 3 işletim sisteminin farklı sürümleri ele alınmıştır. Saldırı ağında ise, Kali ve Windows 8.1 işletim sistemlerine sahip toplam 4 bilgisayar, bir yönlendirici ve bir anahtar bulunmaktadır. Kurban ağı, 3 sunucu, 1 güvenlik duvarı, 2 anahtar ve birbirine 1 Etki Alanı Denetleyicisi ve Etkin Dizin ile bağlı toplam 10 bilgisayardan oluşmaktadır. Ayrıca, kurban ağının ana anahtarındaki bir bağlantı noktası ayna bağlantı noktası olarak yapılandırılmış ve böylece tüm ağdaki gönderilen ve alınan ağ trafiği tamamen yakalanabilmiştir (Sharafaldin, ve diğerleri, 2018).



Şekil 4.2. CICIDS2017 Test Ortamı Mimarisi

Çizelge 4.1.'de kurban ağındaki ve saldırı ağındaki bilgisayarların işletim sistemi ve IP adresleri listelenmektedir (Sharafaldin, ve diğerleri, 2018).

Çizelge 4.1. Kurban Ağı ve Saldırgan Ağı IP Listesi

	Cihaz	İşletim Sistemi	IP Adresi
Kurban Ağı	Sunucular	Win Server 2016 (Domain Controller ve DNS) Ubuntu 16 (Web Server) Ubuntu 12	192.168.10.3 192.168.10.50 - 205.174.165.68 192.168.10.51 - 205.174.165.66
	Bilgisayarlar	Ubuntu 14.4 (32,64) Ubuntu 16.4 (32,64) Win 7 Pro Win 8.1 -64 Win Vista Win 10 Pro (32,64) Mac	192.168.10.19 - 192.168.10.17 192.168.10.16 - 192.168.10.12 192.168.10.9 192.168.10.5 192.168.10.8 192.168.10.14 - 192.168.10.15 192.168.10.25
	Firewall	Fortinet	
Saldırgan Ağı	Bilgisayarlar	Kali Win 8.1 Win 8.1 Win 8.1	205.174.165.73 205.174.165.69 205.174.165.70 205.174.165.71

Bu veri kümesinde, yaygın saldırı çeşitleri listesinde yer alan toplam 6 saldırı profili oluşturulmuş, ilgili araçlar ve kodlarla çalıştırılmıştır (Sharafaldin, ve diğerleri, 2018). Oluşturulan 6 saldırı profili şunlardır:

- *Brute Force Attack*
- *Heartbleed Attack*
- *Botnet*
- *DoS Attack*
- *DDoS Attack*
- *Web Attack*
- *Infiltration Attack*

CICIDS2017 veri kümesi için 3 Temmuz 2017'den 7 Temmuz 2017'ye kadar veri toplanmıştır. Çalışma her gün sabah 9:00'da başlayıp 17:00'da sona ermiştir. Pazartesi gününün trafik akışı normal bir gün olarak değerlendirilmiş ve saldırılar diğer 4 günde yapılmıştır. Kanada Siber Güvenlik Enstitüsü tarafından 5 günde oluşturulan normal ve saldırı ağ trafik verileri toplamda 8 farklı dosyaya kaydedilmiş ve yayımlanmıştır. Dosyaların detaylı açıklaması Çizelge 4.2.'de sunulmaktadır.

Çizelge 4.2. CICIDS2017 Veri Kümesini Oluşturan Dosyaların Açıklaması

Kayıt Günü	Zaman Aralığı	Dosya Adı	Dosya Boyutu	Saldırı Çeşitleri	Satır Sayısı
Pazartesi	Tüm Gün	Monday-WorkingHours.pcap_ISCX.csv	172.782 KB	Benign (Hayırrmal human activities)	529.918
Salı	Tüm Gün	Tuesday-WorkingHours.pcap_ISCX.csv	131.914 KB	Brute Force, FTP-Patator, SSH-Patator	445.909
Çarşamba	Tüm Gün	Wednesday-workingHours.pcap_ISCX.csv	219.890 KB	DoS and Hearbleed Attacks, slowloris, Slowhttptest, Hulk and GoldenEye	692.703
Perşembe	Sabah	Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	50.804 KB	Web Attack – Brute Force, Web Attack XSS, Web Attack – SQL Injection	170.366
	Öğle	Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	81.155 KB	Infiltration–Dropbox download and Cool disk	288.602
Cuma	Sabah	Friday-WorkingHours-Morning.pcap_ISCX.csv	56.950 KB	Botnet ARES	191.033
	Öğle	Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	75.317 KB	DDoS LOIC	225.745
	Öğle	Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	75.104 KB	PortScans (sS, sT, sF, sX, sN, sP, sV, sU, sO, sA, sW, sR, sL and B)	286.467

Kanada Siber Güvenlik Enstitüsü web sitesinde herkese açık olarak bulunan CICFlowMeter adlı yazılımı kullanılmıştır. 80'den fazla ağ trafiği özniteliği kullanılarak bu veri kümesi sınıflandırılmıştır. CICFlowMeter kullanılarak elde edilen PCAP dosyasından ise toplam 78 adet özniteliğe indirilerek CICIDS2017 veri kümesi oluşturulmuştur. Saldırı çizelgesine dayanarak üretilen ağ trafiği etiketlenmiştir. Ağ trafiğindeki öznitelikler ve açıklamaları Çizelge 4.3.'de sunulmaktadır (Özekes, ve diğerleri, 2019; Abdulrahman, ve diğerleri, 2018).

Çizelge 4.3. Sınıflandırmada Kullanılan Trafik Özellikleri ve Açıklamaları

Sıra No	Özellik Adı	Açıklama	Sıra No	Özellik Adı	Açıklama
1	**Destination Port**	Hedef port	41	**Packet Length Mean**	Bir akışın ortalama uzunluğu
2	**Flow Duration**	Mikrosaniyedeki akış süresi	42	**Packet Length Std**	Bir akışın standart sapması
3	**Total Fwd Packets**	İleri yönde toplam paket sayısı	43	**Packet Length Variance**	Bir akışın uzunluk varyansı
4	**Total Backward Packets**	Geri yönde toplam paket sayısı	44	**FIN Flag Count**	FIN bayrak sayısı
5	**Total Length of Fwd Packets**	İleri yönde paketlerin toplam uzunluğu	45	**SYN Flag Count**	SYN bayrak sayısı
6	**Total Length of Bwd Packets**	Geri yönde paketlerin toplam uzunluğu	46	**RST Flag Count**	RST bayrak sayısı
7	**Fwd Packet Length Max**	İleri yönde paketlerin maksimum uzunluğu	47	**PSH Flag Count**	PUSH bayrak sayısı
8	**Fwd Packet Length Min**	İleri yönde paketlerin minimum uzunluğu	48	**ACK Flag Count**	ACK bayrak sayısı
9	**Fwd Packet Length Mean**	İleri yönde paketlerin ortalama uzunluğu	49	**URG Flag Count**	URG bayrak sayısı
10	**Fwd Packet Length Std**	İleri yönde paket uzunluklarının standart sapması	50	**CWE Flag Count**	CWE bayrak sayısı
11	**Bwd Packet Length Max**	Geri yönde paketlerin maksimum uzunluğu	51	**ECE Flag Count**	ECE bayrak sayısı
12	**Bwd Packet Length Min**	Geri yönde paketlerin minimum uzunluğu	52	**Down/Up Ratio**	İndirme ve yükleme oranı
13	**Bwd Packet Length Mean**	Geri yönde paketlerin ortalama uzunluğu	53	**Average Packet Size**	Ortalama paket boyutu

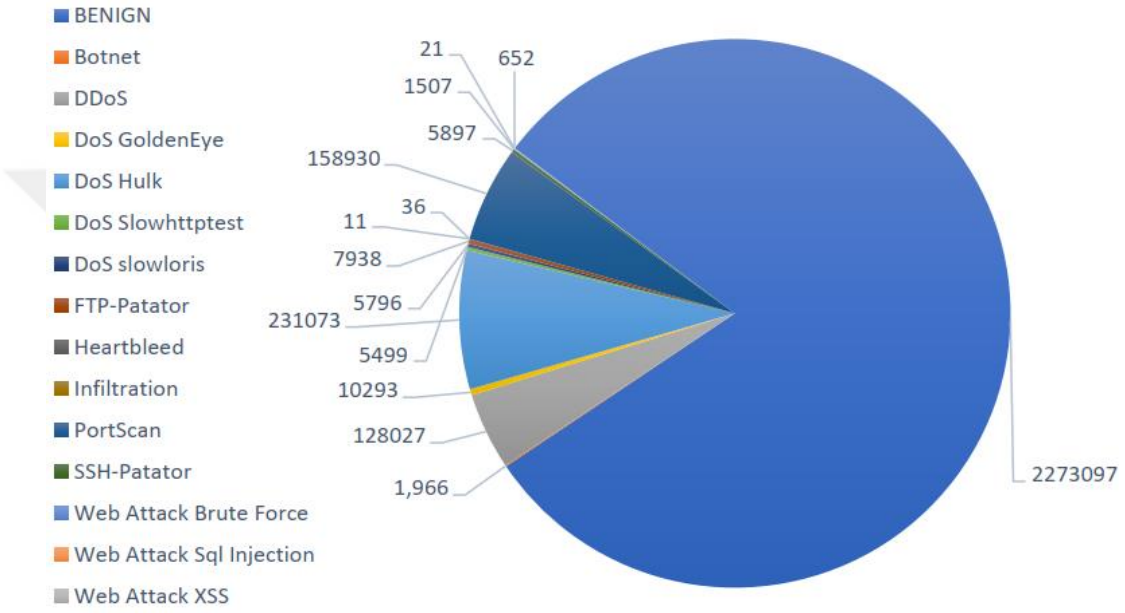
Çizelge 4.4. (devam)

Sıra No	Özellik Adı	Açıklama	Sıra No	Özellik Adı	Açıklama
14	**Bwd Packet Length Std**	Geri yönde paket uzunluklarının standart sapması	54	**Avg Fwd Segment Size**	İleri yönde gözlenen ortalama boyut
15	**Flow Bytes/s**	Saniyedeki byte sayısı	55	**Avg Bwd Segment Size**	Geri yönde gözlenen ortalama boyut
16	**Flow Packets/s**	Saniyedeki paket sayısı	56	**Fwd Header Length_1**	İleri yönde başlıklar için kullanılan toplam byte
17	**Flow IAT Mean**	Paketlerin ortalama varış zamanı	57	**Fwd Avg Bytes/Bulk**	İleri yönde byte/kütle oranının ortalama sayısı
18	**Flow IAT Std**	Paketlerin varış zamanlarının standart sapması	58	**Fwd Avg Packets/Bulk**	İleri yönde paket/kütle oranının ortalama sayısı
19	**Flow IAT Max**	Paketlerin maximum varış zamanı	59	**Fwd Avg Bulk Rate**	İleri yönde kütle oranının ortalama sayısı
20	**Flow IAT Min**	Paketlerin Minimum varış zamanı	60	**Bwd Avg Bytes/Bulk**	Geri yönde byte/kütle oranının ortalama sayısı
21	**Fwd IAT Total**	İleri yönde gönderilen iki paket arasındaki toplam zaman	61	**Bwd Avg Packets/Bulk**	Geri yönde paket/kütle oranının ortalama sayısı
22	**Fwd IAT Mean**	İleri yönde gönderilen iki paket arasındaki ortalama zaman	62	**Bwd Avg Bulk Rate**	Geri yönde kütle oranının ortalama sayısı
23	**Fwd IAT Std**	İleri yönde gönderilen iki paket arasındaki zamanın standart sapması	63	**Subflow Fwd Packets**	İleri yönde bir alt akıştaki ortalama paket sayısı
24	**Fwd IAT Max**	İleri yönde gönderilen iki paket arasındaki maksimum zaman	64	**Subflow Fwd Bytes**	İleri yönde bir alt akıştaki ortalama byte sayısı
25	**Fwd IAT Min**	İleri yönde gönderilen iki paket arasındaki minimum zaman	65	**Subflow Bwd Packets**	Geri yönde bir alt akıştaki ortalama paket sayısı
26	**Bwd IAT Total**	Geri yönde gönderilen iki paket arasındaki toplam zaman	66	**Subflow Bwd Bytes**	Geri yönde bir alt akıştaki ortalama byte sayısı
27	**Bwd IAT Mean**	Geri yönde gönderilen iki paket arasındaki ortalama zaman	67	**Init_Win_bytes_forward**	İleri yönde ilk pencere içinde gönderilen bayt sayısı
28	**Bwd IAT Std**	Geri yönde gönderilen iki paket arasındaki zamanın standart sapması	68	**Init_Win_bytes_backward**	Geri yönde ilk pencere içinde gönderilen bayt sayısı

Çizelge 4.5. (devam)

Sıra No	Özellik Adı	Açıklama	Sıra No	Özellik Adı	Açıklama
29	**Bwd IAT Max**	Geri yönde gönderilen iki paket arasındaki maksimum zaman	69	**act_data_pkt_fwd**	İleri yönde en az 1 bayt TCP veri yüküne sahip paket sayısı
30	**Bwd IAT Min**	Geri yönde gönderilen iki paket arasındaki minimum zaman	70	**min_seg_size_forward**	İleri yönde gözlenen minimum segment boyutu
31	**Fwd PSH Flags**	İleri yönde hareket eden paketlerde PSH bayrağının aktif olma sayısı (UDP için 0)	71	**Active Mean**	Boşta kalmadan önce bir akışın aktif olduğu ortalama zaman
32	**Bwd PSH Flags**	Geri yönde hareket eden paketlerde PSH bayrağının aktif olma sayısı (UDP için 0)	72	**Active Std**	Boşta kalmadan önce bir akışın aktif olduğu zamanın standart sapması
33	**Fwd URG Flags**	İleri yönde hareket eden paketlerde URG bayrağının aktif olma sayısı (UDP için 0)	73	**Active Max**	Boşta kalmadan önce bir akışın aktif olduğu maksimum zaman
34	**Bwd URG Flags**	Geri yönde hareket eden paketlerde URG bayrağının aktif olma sayısı (UDP için 0)	74	**Active Min**	Boşta kalmadan önce bir akışın aktif olduğu minimum zaman
35	**Fwd Header Length**	İleri yöndeki başlıklar için kullanılan toplam byte	75	**Idle Mean**	Aktif hale gelmeden önce bir akışın boşta olduğu ortalama zaman
36	**Bwd Header Length**	Geri yöndeki başlıklar için kullanılan toplam byte	76	**Idle Std**	Aktif hale gelmeden önce bir akışın boşta olduğu zamanın standart sapması
37	**Fwd Packets/s**	Saniyedeki ileri yön paket sayısı	77	**Idle Max**	Aktif hale gelmeden önce bir akışın boşta olduğu maksimum zaman
38	**Bwd Packets/s**	Saniyedeki geri yön paket sayısı	78	**Idle Min**	Aktif hale gelmeden önce bir akışın boşta olduğu minimum zaman
39	**Min Packet Length**	Bir akışın minimum uzunluğu	79	**Label**	Saldırı tiplerine yönelik etiketleme bilgisi
40	**Max Packet Length**	Bir akışın maksimum uzunluğu			

CICIDS2017 veri kümesindeki etiketlenmiş olan paketlerin dağılımı Şekil 4.3.'de gösterilmiştir. Veri kümesi, 14 saldırı çeşidi ve tüm veri kümesinin %80'ini temsil eden (2.273.097 kayıt) normal paket bilgisi içermektedir. Tüm saldırı paketleri ise veri kümesi boyutunun %20'sini (557.646 kayıt) oluşturmaktadır. Veri kümesi, güncel ve yaygın olan saldırıları sağlamaktadır (Faker, 2019).



Şekil 4.3. CICIDS2017 Veri Kümesi Paketlerin Dağılımı

Çizelge 4.4.'de CICIDS2017 veri kümesinde bulunan 14 saldırı çeşidi ve 1 adet normal ağ hareketi, her biri için veri kümesindeki satır sayısı ve bu saldırılar hakkında kısaca açıklama bilgisi yer almaktadır (Faker, 2019).

Çizelge 4.6. CICIDS2017 Veri Kümesinde Bulunan Saldırı Türlerinin Listesi

Saldırı Çeşidi	Satır Sayısı	Tanım
BENIGN	2273097	Normal ağ hareketi
DoS Hulk	231073	Web sunucusunun kaynaklarını engellemek veya durdurmak için web sunucusunda çok sayıda ağ trafiği oluşturur.
PortScan	158930	Amacı, bilgisayar bağlantı noktalarını taramak ve sonrasında sunuculara erişim sağlamak için tespit edilen sistem güvenlik açıklarından yararlanır.
DDoS LOIC	128027	Bu saldırı, yoğun bir trafik akışı oluşturarak sunucunun veya ağın normal trafiğini kesmeye çalışır.
DoS GoldenEye	10293	Amacı,HTTP sunucusundaki tüm kullanılabilir soketleri tüketir.
FTP-Patator	7938	Bir tür Brute Force saldırısı olan bu saldırı çeşidi, şifre üretme yöntemlerini kullanarak sistemdeki şifreleri kırmayı amaçlar.
SSH-Patator	5897	Bir tür Brute Force saldırısı olan bu saldırı çeşidi, şifre üretme yöntemlerini kullanarak sistemdeki şifreleri kırmayı amaçlar.
DoS slowloris	5796	Bu saldırı çeşidi, ağa gelen taleplerin reddedilmesini sağlamak amacıyla sunuculara eksik istekler gönderir ve böylece sunucuları meşgul eder ve sistemi yavaşlatır.
DoS Slowhttptest	5499	HTTP sunucusuna eksik istekler göndererek sunucunun belleğini ve işlemcisini doldurarak sistemi yavaşlatır.
Botnet ARES	1966	Sunuculara erişim için ağa bağlı bilgisayarlara kötü amaçlı yazılım göndererek bu bilgisayarların uzaktan yönetilmesini sağlar.
Web Attack Brute Force	1507	Sistemde kullanılan parolaları elde etmek amacıyla, sistemde ardışık otomatik olarak çalışan programları kullanır.
Web Attack XSS	652	Kurbanı kötü amaçlı sitelere yönlendirmek veya kişisel verilerine erişmek için web sayfasına veya web uygulamasına uygulanmış kodlardır.
Infiltration	36	Bu saldırı çeşidi, sistemdeki güvenlik açıklarını kullanmak için veya sunuculara erişmek için sisteme kötü amaçlı dosyalar gönderir.
Web Attack Sql Injection	21	Sistemdeki gizli tutulan bilgilere erişmeyi veya veri tabanı yapısını bozmayı amaçlar.
Heartbleed	11	OpenSSL şifreleme mimarisinde ciddi bir güvenlik açığını yakalamayı amaçlar.

1998-2016 tarihleri arasında literatürde kullanılan 11 adet veri kümesi ile CICIDS2017 veri kümesinin karşılaştırılması Çizelge 4.5.'de gösterilmiştir (Sharafaldin, ve diğerleri, 2018).

Çizelge 4.7. Literatürde Kullanılan Veri Kümelerinin Karşılaştırılması

						Protokoller					**Saldırı Çeşitliliği**										
	Ağ	Trafik	Etiket	Etkileşim	Yakalama	HTTP	HTTPS	SSH	FTP	EPosta	Browser	Bforce	DoS	Scan	ArkaKapı	DNS	Diğerleri	AHayırnim	Heterojen	Öznitelikler	MetaData
DARPA	Evet	Hayır	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Hayır	Hayır	Evet	Hayır	Hayır	Hayır	Evet
KDD'99	Evet	Hayır	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Hayır	Hayır	Evet	Hayır	Hayır	Evet	Evet
DEFCON	Hayır	Hayır	Hayır	Evet	Evet	Evet	Hayır	Evet	Hayır	Hayır	Hayır	Hayır	Hayır	Evet	Evet	Hayır	Evet	-	Hayır	Hayır	Hayır
CAIDAs	Evet	Evet	Hayır	Hayır	Hayır	-	-	-	-	-	Hayır	Hayır	Evet	Evet	Hayır	Evet	Evet	Evet	Hayır	Hayır	Evet
LBNL	Evet	Evet	Hayır	Hayır	Hayır	Evet	Hayır	Evet	Hayır	Hayır	-	-	-	Evet	-	-	-	Evet	Hayır	Hayır	Hayır
CDX	Hayır	Hayır	Hayır	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Hayır	Hayır	Evet	Evet	Hayır	Evet	-	-	Hayır	Hayır	Hayır
KYOTO	Evet	Hayır	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Hayır	Hayır	Evet	Evet
TWENTE	Evet	Evet	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Hayır	Hayır	Evet	Hayır	Evet	Hayır	Hayır	Evet	-	-	Hayır	Evet
UMASS	Evet	Hayır	Evet	Hayır	Evet	Evet	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Hayır	Evet	-	-	Hayır	Hayır
ISCX2012	Evet	Hayır	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Hayır	Evet	Hayır	Evet	Hayır	Evet
ADFA2013	Evet	Evet	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Evet	Evet	Evet	Hayır	Hayır	Evet	Hayır	Evet	Hayır	-	Hayır	Evet
CICIDS2017	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet

4.2. Geliştirme Ortamı

Bu çalışma için yüksek işlem gücü gerekliliğinden dolayı Google Colaboratory (Colab)'ın sunduğu ücretsiz GPU hizmeti kullanılmıştır. Geliştirmeler ve çalışmalar Python programlama dili ile yapılmıştır. Bu çalışmada donanım aracı olarak, Google Colab platformunda çevrimiçi (online) sunduğu NVIDIA Tesla P100-PCIE-16GB GPU (Grafik İşleme Birimi) işlemcili makine kullanılmıştır. Tesla P100-PCIE-16GB GPU işlemcisi 4 çekirdekli Intel(R) Xeon(R) CPU @ 2.30GHz'dir. GPU kullanan sistemler, CPU (Merkezi İşlem Birimi) kullanan sistemlere göre daha hızlı performans sağlamakta ve daha düşük düzeyde enerji harcamaktadır.

4.3. CICIDS2017 Veri Kümesinin İşleme Hazırlanması

Bu bölümde, makine öğrenme algoritmaları uygulayarak anomali tespiti yapılacak olan CICIDS2017 veri kümesi üzerinde modellerin uygulanabilmesi için gerçekleştirilen işlemler aşağıda verilmiştir.

1. Karar ağaçları ve arttırma algoritmalarını kullanarak anomali tespit etme işlemine başlamadan önce, ağ trafik paketlerinin izlenmesi sonucu elde edilen CICIDS2017 veri kümesine sırasıyla veri önişleme ve normalizasyon işlemleri uygulanmıştır.
2. Öznitelik seçimi işlemi uygulanarak CICIDS2017 veri kümesindeki toplam 78 adet olan öznitelikler 53'e indirgenmiştir. Hesaplamalara etkisi yok denilecek kadar az olan 25 adet öznitelik test, eğitim ve doğrulama veri kümelerinden çıkarılmıştır.

3. Veri önişleme ve normalizasyon işlemleri yapıldıktan sonra ilgili veri kümesi; test, eğitim ve doğrulama olmak üzere üçe ayrılmıştır.
4. Tüm bu yapılan işlemlerden sonra CICIDS2017 veri kümesinde karar ağacı ve arttırma algoritmalarının uygulanması sürecine başlanmıştır.

4.3.1. Veri önişleme

CICIDS2017 veri kümesi üzerinde uygulanan veri önişleme başlıkları şu şekilde sıralanabilir; bozuk verilerin silinmesi ve özel karakterlerin dönüştürülmesi, NAN ve NULL değerinin doldurulması, etiket kodlayıcı kullanarak sayısal olmayan değerlerin sayısal değerlere dönüştürülmesi, sonuçların görselleştirilmesi.

	Destination Port	Flow Duration	...	Idle Min	Label
272642	80	97735843	...	97700000	DoS Hulk
165366	443	118726899	...	5035267	BENIGN
52419	21	9079364	...	0	FTP-Patator
266739	80	5793986	...	0	BENIGN
171279	53	143087	...	0	BENIGN

[5 rows x 79 columns]

Şekil 4.4. Veri Önişleme Öncesi CICIDS2017

	Destination Port	Flow Duration	...	Idle Min	Label
83209	443	1	...	0	0
72103	80	47928	...	0	5
63364	443	231981	...	0	0
99500	80	10007365	...	7948450	2
89738	89	87	...	0	10

[5 rows x 79 columns]

Şekil 4.5. Veri Önişleme Sonrası CICIDS2017

Belirtilen veri önişleme adımları uygulanmadan önce CICIDS2017 veri kümesinin rastgele listelenmiş 5 satırı örnek olarak Şekil 4.4.'de gösterilmiştir. Ayrıca veri

kümesi üzerinde gerçekleştirilen veri önişleme adımları sonucunda elde edilen veri kümesine ait rastgele listelenmiş 5 satır da Şekil 4.5.'de örnek olarak gösterilmiştir.

Veri kümesinde tespit edilen eksik, yanlış ve bozuk ifadeler düzeltilmiştir. Toplamda 1358 adet NAN ve NULL değerli veri düzeltilmiştir. Kategorik özellikteki veriler sayısal değerlere dönüştürülmüştür. Veri kümesinde 79. sırada yer alan; saldırı olup olmadığını belirten 'Label' isimli öznitelik içerisindeki kayıtlı 15 farklı etiketleme bilgisi 0 ile 14 arasında değerler verilerek sayısallaştırılmıştır. Kategorik değerlerin hangi sayısal değerlerle değiştirildiğini gösteren bilgiler Çizelge 4.6.'da listelenmektedir.

Çizelge 4.8. Kategorik Değerlerin Sayısallaştırma Listesi

Kategorik Değeri	Sayısal Değeri
BENIGN	0
Bot	1
DDoS	2
DoS GoldenEye	3
DoS Hulk	4
DoS Slowhttptest	5
DoS slowloris	6
FTP-Patator	7
Heartbleed	8
Infiltration	9
PortScan	10
SSH-Patator	11
Web Attack Brute Force	12
Web Attack Sql Injection	13
Web Attack XSS	14

Veri önişleme sürecinde yazılan Python kodlarına ait örnek kodlar Ek A'da verilmiştir.

4.3.2. Veri normalizasyon

Modelin doğruluğunu arttırmak için modeli standardizasyon ve normalleştirme işlemleri ile ölçeklendirmek; makine öğrenme algoritmaları için çok önemli bir yere sahip olan veri önişleme işleminde önemli bir adımdır. Çünkü makine öğrenme algoritmaları iyi dağıtılmış ve yeniden düzenlenmiş modellerde daha iyi sonuç verme eğilimindedirler.

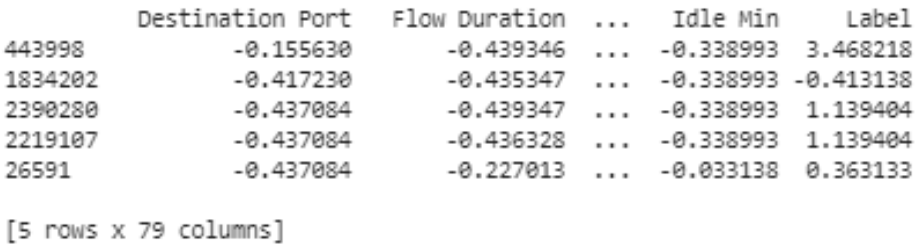
Tez çalışmasında, temizlenen ve hataları düzeltilmiş olan veri kümesinin değerlerine, sınıflandırma algoritmalarının başarımını olumsuz etkilememesi için, Z-Skor Ölçekleme işlemi uygulanmıştır. Z-Skor normalizasyon değişkenin herhangi bir y değeri, değişkenin ortalaması ve standart sapmasına bağlı olarak bilinen Z dönüşümü ile normalleştirilir.

Tüm veri kümesini aynı aralıkta değerlendirmek, olağandışı durumların etkisini azaltmak ve veriyi yumuşatmak için 4.1'de formülü verilen Z-Skor ile normalleştirme işlemi uygulanmıştır (Gülaçar, 2018).

$$Z = \frac{x - \mu}{\sigma} \tag{4.1}$$

Burada; Z Z-Skor değerini, x değer, μ ortalama ve σ standart sapma değerlerini temsil etmektedir.

CICIDS2017 veri kümesine normalizasyon işlemi uygulandıktan sonra elde edilen veri kümesine ait rastgele listelenmiş 5 satır Şekil 4.6.'da örnek olarak gösterilmiştir.



	Destination Port	Flow Duration	...	Idle Min	Label
443998	-0.155630	-0.439346	...	-0.338993	3.468218
1834202	-0.417230	-0.435347	...	-0.338993	-0.413138
2390280	-0.437084	-0.439347	...	-0.338993	1.139404
2219107	-0.437084	-0.436328	...	-0.338993	1.139404
26591	-0.437084	-0.227013	...	-0.033138	0.363133

[5 rows x 79 columns]

Şekil 4.6. Normalizasyon Sonrası CICIDS2017

Veri normalizasyon sürecinde yazılan Python kodlarına ait örnek kodlar Ek B'de verilmiştir.

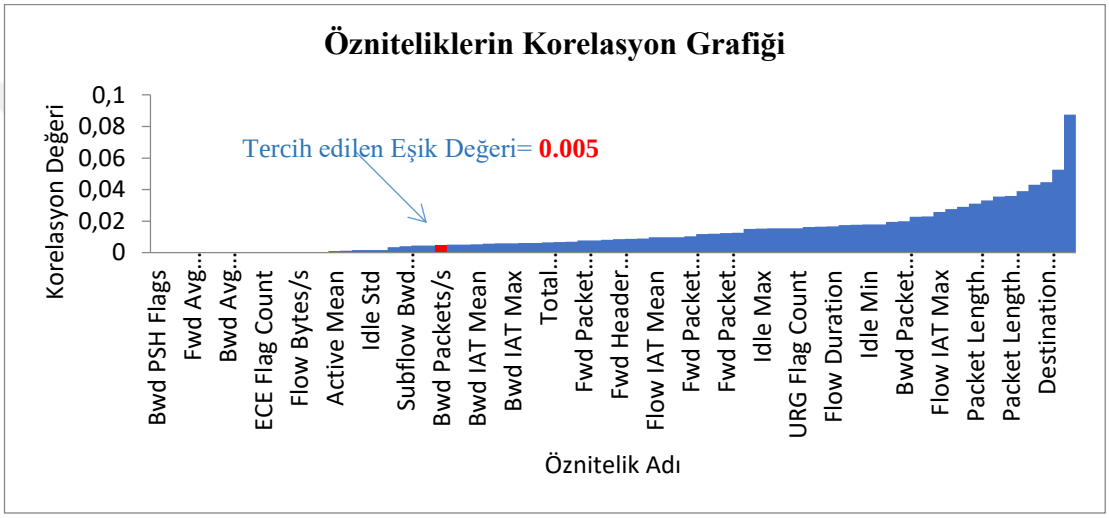
4.3.3. Öznitelik seçimi

CICIDS 2017 veri kümesinde normalizasyon işleminden sonra öznitelik seçimi işlemi gerçekleştirilmiştir. Öznitelik seçim işlemi, Yinelemeli Öznitelik Eleme (RFE) yöntemi kullanılarak gerçekleştirilmiştir.

- *Yinelemeli Öznitelik Eleme (Recursive Feature Elimination – RFE):* Tüm özniteliklerden kademeli olarak bazılarının elenmesi yöntemine dayanmaktadır. Farklı sınıfları ayırt edebilmeye katkısı olmayan öznitelikler elenir. Katkısı olup olmadığını ölçmek için özniteliklere bir sınıflandırma yöntemi uygulanarak ağırlıkları hesaplanır. En yüksek ayırt etme oranına sahip öznitelikler kalana kadar eleme işlemi devam eder. Ayrıca istenirse sınıflandırma işleminin durması için önceden bir öznitelik sayısı belirtilebilir. Böylece öznitelik sayısı belirtilen sayıya ulaştığında hesaplama devam etmeden işlemi durdurur (Turgut, 2017).

Korelasyon değeri 0 olan; *Bwd PSH Flags, Bwd URG Flags, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk ve Bwd Avg Bulk Rate*, öznitelikleri saldırı tespiti için etkisiz olduğundan veri kümesinden çıkarılmışlardır. Korelasyon değerleri 0 olan öznitelikler çıkarıldıktan sonra kalan 70 öznitelik RFE yöntemi uygulanarak

sınıflandırılmış ve 0.005 eşik değerinden düşük değere sahip öznitelikler veri kümesinden çıkarılmıştır. Elenen özniteliklerin sayısı belirlenirken her bir özelliğin eşik değeri önem arz etmektedir. Eşik değeri 0.005 değerinden düşük toplam 17 öznitelik bulunmaktadır. Eşik değeri belirlenirken; kullanılan algoritmalar ve başarım oranları gözlemlendiğinde en verimli sonucun elde edilmesini sağlayan değer tercih edilmiştir. Bu çıkarım sonucunda mevcutta 53 öznitelik elde edilmiştir. Tercih edilen eşik değerinin de gösterildiği, özniteliklerin korelasyon değer grafiği Şekil 4.7.'de verilmiştir.



Şekil 4.7. CICIDS2017 Özniteliklerin Korelasyon Grafiği

CICIDS2017 veri kümesinde bulunan öznitelikler arası korelasyon tablosu Çizelge 4.7.'de listelenmiştir.

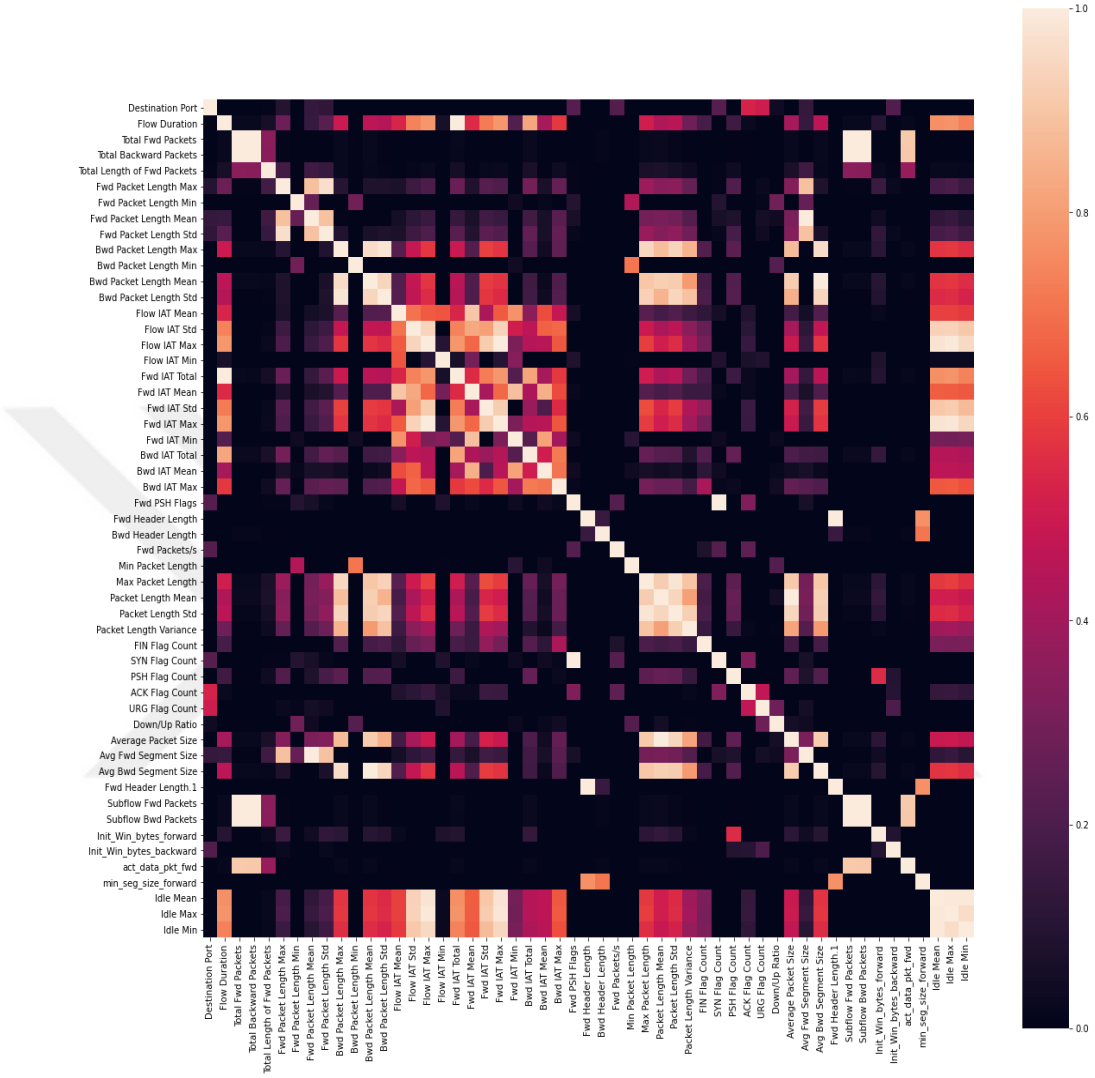
Çizelge 4.9. CICIDS2017 Öznitelikler Arası Korelasyon Tablosu

#	Öznitelik Adı	Korelasyon Değeri	#	Öznitelik Adı	Korelasyon Değeri
1	Bwd PSH Flags	0	40	Fwd Header Length.1	0.00848
2	Bwd URG Flags	0	41	Bwd IAT Total	0.008681
3	Fwd Avg Bytes/Bulk	0	42	Total Fwd Packets	0.008971
4	Fwd Avg Packets/Bulk	0	43	Flow IAT Mean	0.009687
5	Fwd Avg Bulk Rate	0	44	Down/Up Ratio	0.009717

Çizelge 4.10. (devam)

#	Öznitelik Adı	Korelasyon Değeri	#	Öznitelik Adı	Korelasyon Değeri
6	Bwd Avg Bytes/Bulk	0	45	Fwd IAT Mean	0.009803
7	Bwd Avg Packets/Bulk	0	46	Fwd Packet Length Max	0.010415
8	Bwd Avg Bulk Rate	0	47	act_data_pkt_fwd	0.011729
9	RST Flag Count	0.000001	48	Flow IAT Std	0.01198
10	ECE Flag Count	0.000001	49	Fwd Packet Length Mean	0.012369
11	CWE Flag Count	0.000004	50	FIN Flag Count	0.012716
12	Fwd URG Flags	0.000005	51	Avg Fwd Segment Size	0.014999
13	Flow Bytes/s	0.000034	52	Idle Max	0.015179
14	Flow Packets/s	0.000264	53	Max Packet Length	0.015411
15	Active Std	0.000298	54	Fwd Packets/s	0.015426
16	Active Mean	0.001106	55	URG Flag Count	0.015481
17	Active Min	0.001183	56	Fwd Packet Length Min	0.016279
18	Bwd IAT Min	0.001641	57	Fwd IAT Std	0.016406
19	Idle Std	0.001711	58	Flow Duration	0.016651
20	Active Max	0.001731	59	Fwd IAT Total	0.017467
21	Total Length of Bwd Packets	0.003504	60	Idle Mean	0.017736
22	Subflow Bwd Bytes	0.004028	61	Idle Min	0.017841
23	Subflow Fwd Bytes	0.004422	62	Min Packet Length	0.017949
24	Bwd IAT Std	0.004459	63	Bwd Packet Length Std	0.019457
25	Bwd Packets/s	0.004917	64	Bwd Packet Length Min	0.019853
26	**Total Length of Fwd Packets**	**0.005046**	65	Bwd Packet Length Max	0.022677
27	Fwd IAT Min	0.005205	66	Packet Length Variance	0.022987
28	Bwd IAT Mean	0.005267	67	Flow IAT Max	0.02581
29	Subflow Bwd Packets	0.005696	68	Fwd IAT Max	0.027691
30	Bwd Header Length	0.005825	69	Init_Win_bytes_forward	0.029141
31	Bwd IAT Max	0.005904	70	Packet Length Mean	0.031042
32	Init_Win_bytes_backward	0.006145	71	Average Packet Size	0.033021
33	Flow IAT Min	0.006178	72	min_seg_size_forward	0.035632
34	Total Backward Packets	0.006573	73	Packet Length Std	0.035879
35	Fwd PSH Flags	0.006772	74	Bwd Packet Length Mean	0.039089
36	SYN Flag Count	0.007003	75	Avg Bwd Segment Size	0.043064
37	Fwd Packet Length Std	0.007775	76	Destination Port	0.044773
38	Fwd Header Length	0.007776	77	ACK Flag Count	0.052516
39	Subflow Fwd Packets	0.008072	78	PSH Flag Count	0.08745

Şekil 4.8.'de ise seçim yapılan 53 adet özniteliğin korelasyon değerlerine göre hazırlanmış olan karşılaştırma matrisine ait grafik gösterilmektedir.



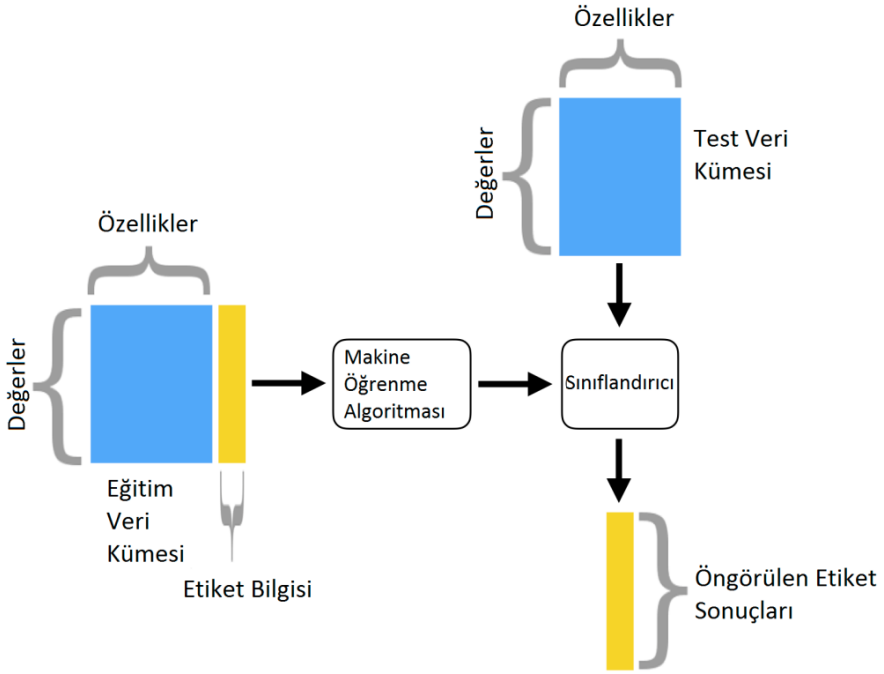
Şekil 4.8. Seçilen 53 Özniteliğin Karşılaştırma Matrisi

Veri kümesindeki öznitelik seçimi sürecinde yazılan Python kodlarına ait örnek kodlar Ek C'de verilmiştir.

4.3.4. Eğitim ve test verisi

Önişleme sürecinden geçirilen CICIDS2017 veri kümesi; makine öğrenmesi algoritmalarının uygulanabilmesi amacıyla, eğitim ve test olmak üzere iki veri kümesine ayrılmıştır. Çalışmada, K-fold değeri 10 olarak belirlenerek Çapraz Doğrulama Değeri işlemi yapılmıştır. Parametrelerde rastgele seçim aktif edildiğinden dolayı ilgili kod her çalıştırıldığında ayrıştırdığı veriler farklılık gösterebilmektedir. Böylece veri kümesi ayrıştırılmasında farklılıklar elde edilerek gerçek dünya verilerine yaklaşılmıştır. CICIDS2017 veri kümesi ayrıştırılırken aynı zamanda etiket bilgisi olan 'Label' isimli öznitelik de ayrı olacak şekilde ayrıştırılmıştır. Şekil 4.9.'da bu ayrıştırma işlemini gösteren örnek çizim bulunmaktadır.

Test ve eğitim veri kümelerinin ayrıştırılma sürecinde yazılan Python kodlarına ait örnek kodlar Ek Ç'de verilmiştir.



Şekil 4.9. CICIDS2017 Etiket Bilgisi

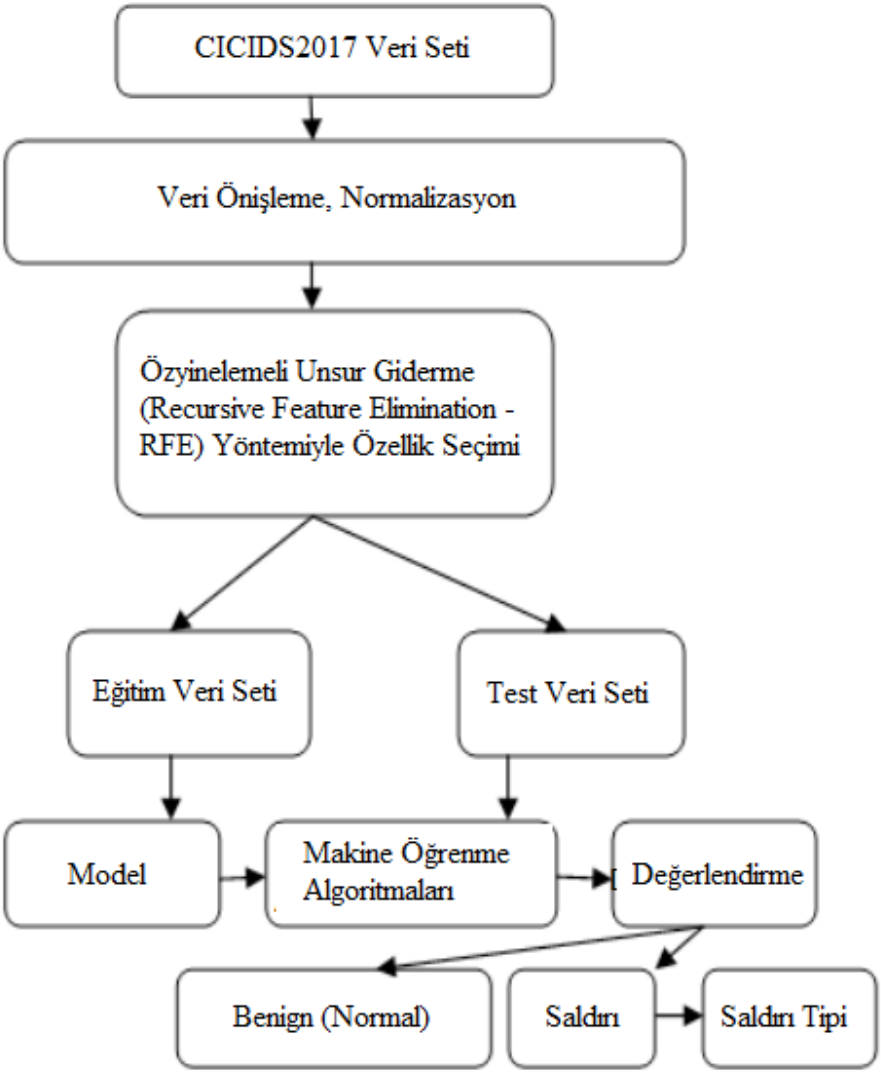
Şekil 4.9.'da gözüktüğü üzere eğitim veri kümesindeki etiket bilgisi makine öğrenme algoritmalarının eğitilmesinde önemli bir yer oluşturmaktadır. Diğer yandan test veri kümesindeki etiket bilgisi ise sınıflandırıcı sonucunda elde edilmiş, öngörülen etiket bilgisinin doğruluğunu kontrol etmek için kullanılmaktadır.

4.4. Modellerin Uygulanması

Bu bölümde, Bölüm 3'de yer alan makine öğrenmesi yöntemlerinin CICIDS2017 veri kümesi üzerine uygulanması ve elde edilen sonuçların analizi anlatılmaktadır. Modelleme aşamasında Karar Ağaçları, AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost kütüphaneleri kullanılmıştır.

Kullanılan karar ağaçları algoritması ile arttırma algoritmalarının karşılaştırılmasının objektif bir şekilde yapılabilmesi amacıyla algoritmalarda kullanılan temel parametreler her bir algoritma için aynı girilmiştir. Aynı değerlere sahip parametreler sırasıyla; 'n_estimators', 'learning_rate' ve 'max_depth' dir. Ayrıca, K-fold değeri 10 olarak belirlenerek Çapraz Doğrulama Değeri işlemi yapılmıştır. Çapraz doğrulama değeri, AUC değerinin hesaplanması aşamasında kullanılmaktadır. Çapraz doğrulama değeri işlemi bu tez çalışmasında yer alan bütün modellerde aynı parametre değerleriyle kullanılmıştır. Kullanılan parametreler sırasıyla; cv=10, n_jobs=-1 şeklindedir. Parametrelerde bulunan 'cv' parametresi K-fold değerinin sayısını belirtmektedir. Diğer yandan 'n_jobs' parametresi ise hesaplama işleminin paralel işlemciler kullanılarak yapılmasını bu sayede işlemlerin daha hızlı gerçekleşmesini sağlamaktadır. Bu parametrenin -1 değerini alması tüm CPU'ların kullanılmasını sağlamaktadır.

Şekil 4.10.'da CICIDS2017 veri kümesi üzerinde gerçekleştirilen makine öğrenme algoritmalarının uygulama sürecine ait işlemler gösterilmiştir.



Şekil 4.10. Anomali Tespit Süreci

4.4.1. Karar ağaçları

Karar ağaçları modeli için girilen parametreler ve tamamlanan eğitim süresi aşağıda belirtildiği gibidir.

- *Eğitim Süresi:* 1.5 dakika
- *Parametreler:* max_depth=50, random_state=0

Bu model için, çapraz doğrulama değerine ait hesaplama sonucu ve hesaplama süresi aşağıda belirtilmiştir.

- *Çapraz Doğrulama Değeri Hesaplama Süresi:* yaklaşık 1 dakika
- *Çapraz Doğrulama Değeri Hesaplama Sonucu:* [0.99851127 0.99843557 0.99810754 0.99818324 0.99830936 0.99846076 0.9985617 0.99795609 0.99830936 0.99803179]
- *Çapraz Doğrulama Değeri Ortalaması:* 0.9983

Veri kümesindeki başarımı göstermek amacıyla, elde edilen sonuçlar Şekil 4.11.'de belirtilmiştir. 'Label' özniteliğindeki değerler sayısala dönüştürüldüğü için etiket bilgisi 0-14 aralığında gösterilmiştir.

Karşılaştırma matrisi üzerinden elde edilen DP, YP ve YN değerleri sırasıyla; 847900, 1323, 1323 şeklindedir. Test veri kümesindeki toplam veri sayısı 849223'dür.

	0	1	2	3	4	5	...	9	10	11	12	13	14
0	681813	101	4	3	86	19	...	3	277	2	4	2	8
1	103	485	0	0	0	0	...	0	0	0	0	0	0
2	17	0	38221	0	0	0	...	0	0	0	0	0	0
3	8	0	0	3078	10	1	...	0	0	0	0	0	0
4	51	0	0	7	69141	0	...	0	12	0	1	0	0
5	8	0	0	0	2	1606	...	0	0	0	0	0	0
6	11	0	0	2	0	3	...	0	0	0	1	0	1
7	0	0	0	0	0	0	...	0	1	0	0	0	0
8	0	0	0	0	0	0	...	0	0	0	0	0	0
9	2	0	0	0	0	0	...	8	0	0	0	0	0
10	324	0	0	0	6	0	...	0	47258	0	0	0	2
11	1	0	0	0	0	0	...	0	0	1756	0	0	0
12	1	0	0	0	0	0	...	0	3	0	366	1	99
13	2	0	0	0	0	2	...	0	0	0	0	3	0
14	4	0	0	0	0	1	...	0	1	0	113	0	103

[15 rows x 15 columns]

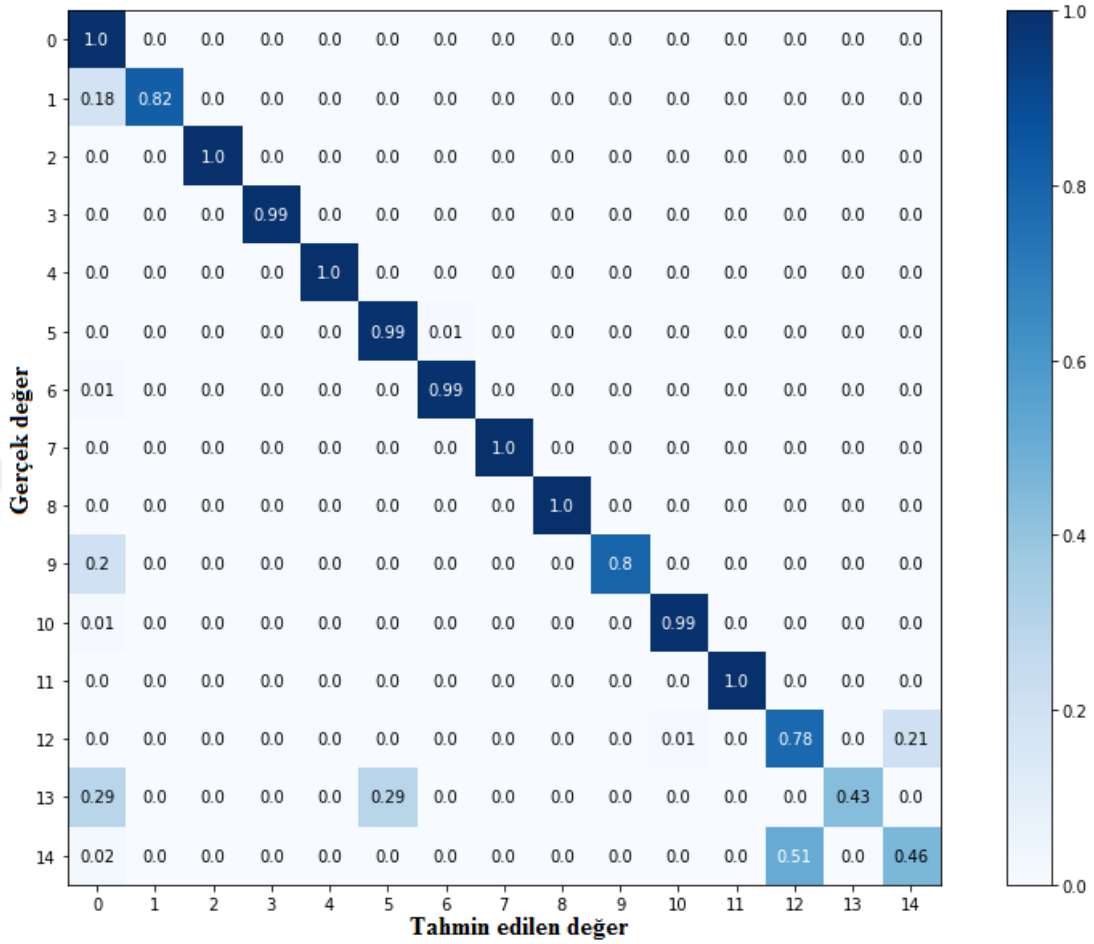
Şekil 4.11. Karar Ağaçları Algoritması için Karşılaştırma Matrisi

	pre	rec	spe	f1	geo	iba	sup
BENIGN	0.9992	0.9993	0.9968	0.9992	0.9980	0.9963	682324
Bot	0.8276	0.8248	0.9999	0.8262	0.9081	0.8103	588
DDoS	0.9999	0.9996	1.0000	0.9997	0.9998	0.9995	38238
DoS GoldenEye	0.9961	0.9939	1.0000	0.9950	0.9969	0.9932	3097
DoS Hulk	0.9985	0.9990	0.9999	0.9987	0.9994	0.9988	69212
DoS Slowhttptest	0.9841	0.9883	1.0000	0.9862	0.9941	0.9871	1625
DoS slowloris	0.9932	0.9898	1.0000	0.9915	0.9949	0.9888	1762
FTP-Patator	0.9996	0.9996	1.0000	0.9996	0.9998	0.9995	2317
Heartbleed	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	2
Infiltration	0.7273	0.8000	1.0000	0.7619	0.8944	0.7840	10
PortScan	0.9938	0.9930	0.9996	0.9934	0.9963	0.9920	47590
SSH-Patator	0.9989	0.9994	1.0000	0.9991	0.9997	0.9994	1757
Web Attack Brute Force	0.7546	0.7754	0.9999	0.7649	0.8805	0.7579	472
Web Attack Sql Injection	0.5000	0.4286	1.0000	0.4615	0.6547	0.4041	7
Web Attack XSS	0.4836	0.4640	0.9999	0.4736	0.6811	0.4390	222
avg / total	0.9984	0.9984	0.9974	0.9984	0.9979	0.9960	849223

Şekil 4.12. Karar Ağaçları Algoritması için Sınıflandırma Raporu

CICIDS2017 veri kümesi üzerinde gerçekleştirilen sınıflandırma işlemi sonucunda elde edilen sınıflandırma raporu Şekil 4.12.'de gösterilmiştir. Karar ağaçları algoritmasıyla gerçekleştirilen bu modelde etiket bilgisi olan 'Label' parametresi üzerinden yapılan karşılaştırma sonucunda elde edilen başarım oranları Şekil 4.12.'de bulunmaktadır. Sınıflandırma işleminde sadece saldırı veya normal paket ayrımı yapılmayıp ayrıca saldırı çeşitleri de tespit edilmiştir. Şekil 4.12. ve Şekil 4.13.'de saldırı çeşitleri için de başarım oranları gösterilmiştir. Şekil 4.13.'de, test veri kümesinde yer alan saldırı çeşitlerine ait değerler ile modelin tahmin ettiği saldırı çeşitlerine ait değerler karşılaştırılmıştır. Çizelge 4.6.'da 'Label' özniteliğinde bulunan sayısal değerlerin kategorik değer karşılıkları listelenmiştir. Şekil 4.13. incelendiğinde karar ağaçları algoritmasının 'Bot', 'Infiltration' ve 'Web Attack' başlıklı saldırılarda başarım oranlarının düşük olduğu görülmektedir.

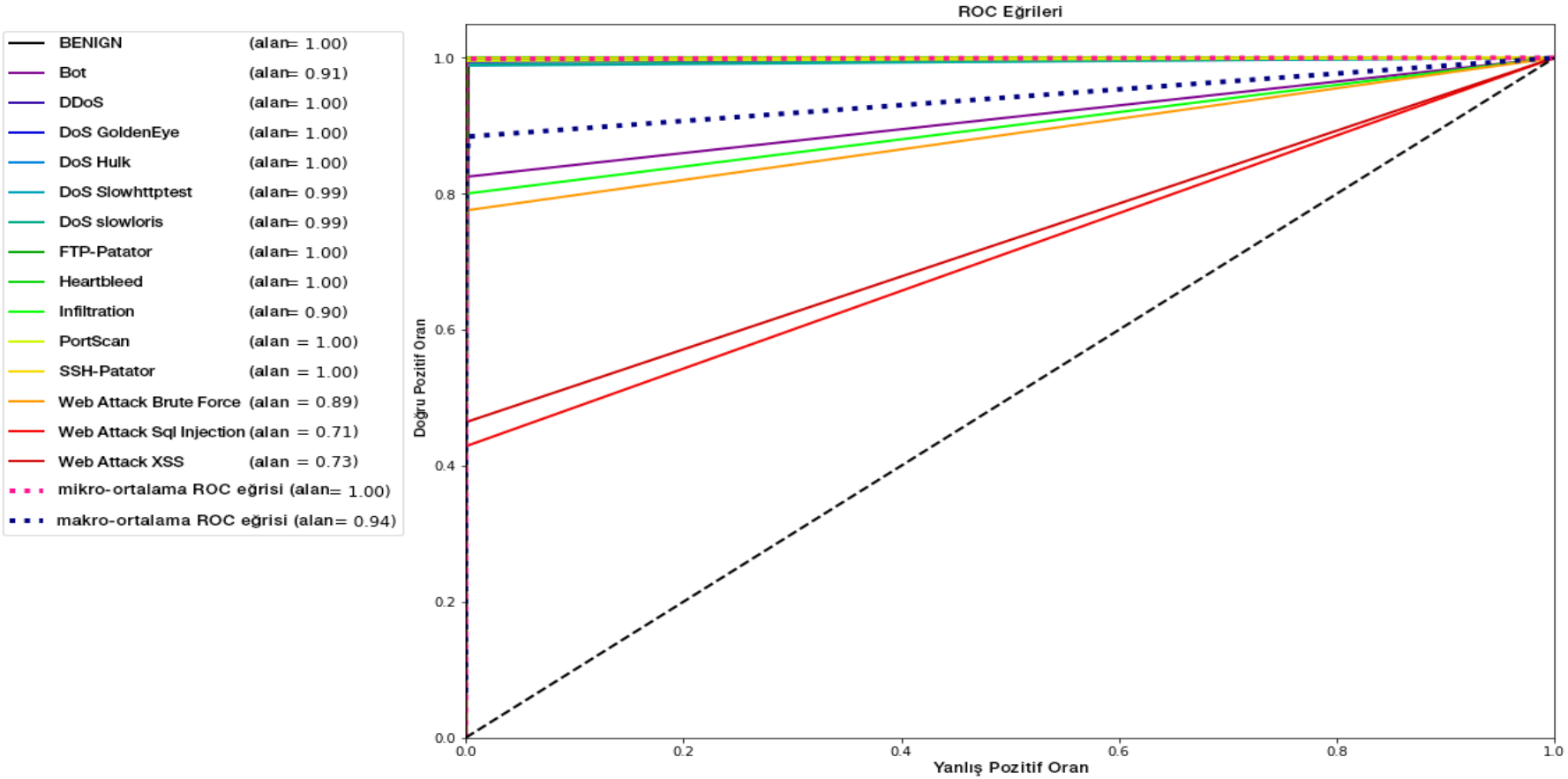
Şekil 4.14.'de ROC eğrisine ait grafik detaylı incelendiğinde ise etiketlenmiş saldırılar içerisinde 'Bot', 'Infiltration' ve 'Web Attack' başlıklı saldırıların AUC değerinin altında kaldığı gözlenmiştir. AUC değerinin altında kalan etiketlenmiş veriler için, karar ağaçları algoritması modelinin başarımını olumsuz etkilediği ve sınıflandırmada bir miktar tutarsızlık olduğu belirtilebilir. Ancak diğer saldırıların sınıflandırılmasında tutarlılık gözlemlenmiştir. AUC değerinin altında kalan saldırı çeşitlerinin sayısal yoğunluğu ile veri kümesinin tamamı karşılaştırılacak olursa, 827/849.223 işlemi sonucu %0.01 gibi çok düşük bir değer elde edilir. Bu değerin bu kadar düşük olmasının sebebi, bu saldırı çeşitlerini belirten verilerin sayısının veri kümesinde az olmasıdır.



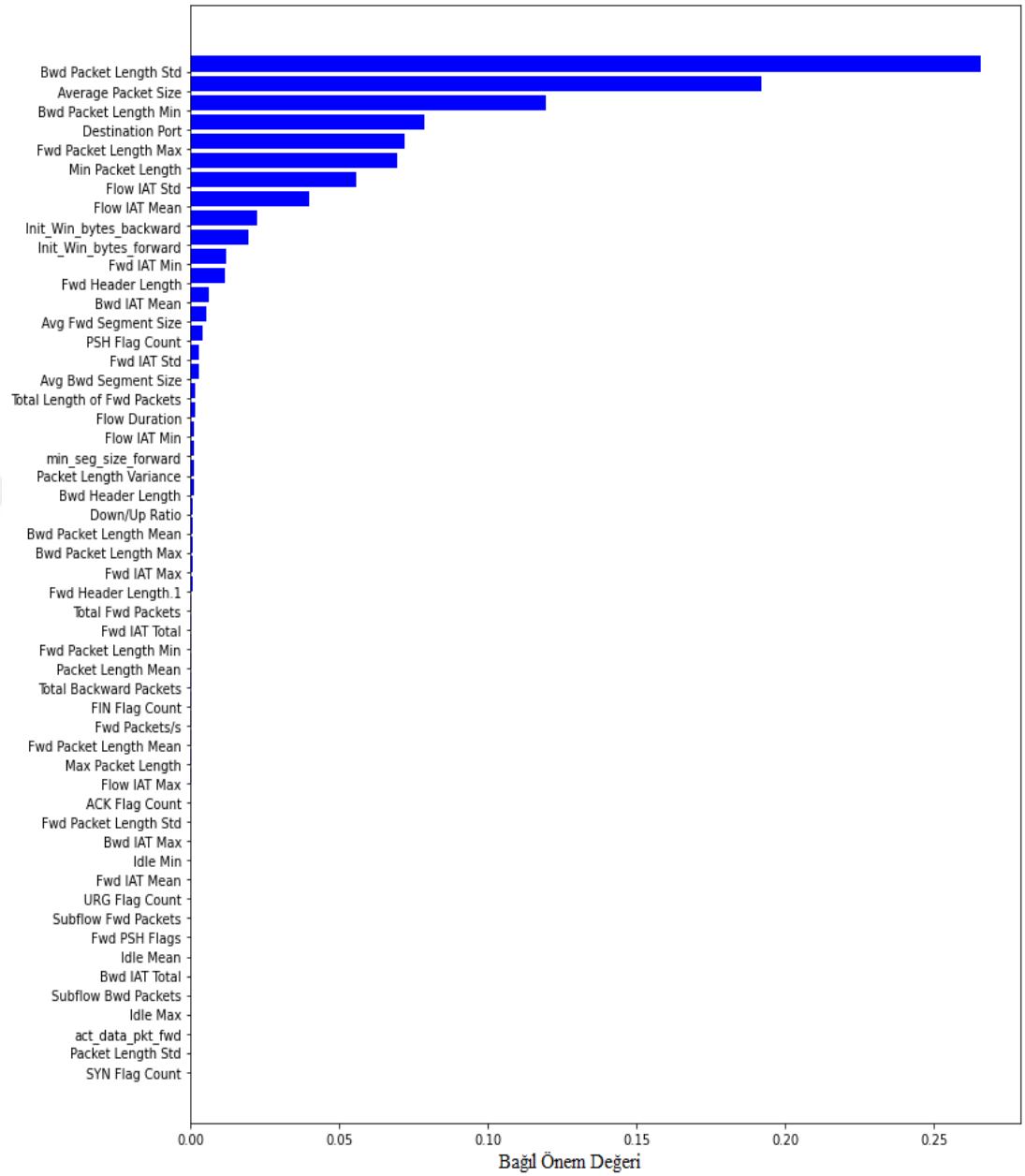
Şekil 4.13. Karar Ağaçları Algoritması için Karşılaştırma Matrisi

Bu modelde her bir etiket bilgisi için elde edilen ROC eğrisi grafiği Şekil 4.14.'de gösterilmiştir. AUC değeri ise testler sonucunda 0.9983 olarak bulunmuştur. AUC değeri 1.00'a ne kadar yaklaşırsa modelin ezberleme oranı daha düşük olmakta ve algoritmanın başarımı yükselmektedir.

Toplam 53 adet öznitelik içerisinde karar ağaçları algoritmasıyla oluşturulan bu modelde 'Bwd Packet Length Std' isimli öznitelik 0.2656 ile en yüksek değere sahip olmuştur. CICIDS2017 veri kümesinde yapılan sınıflandırma işleminde hesaplamalara etki eden özniteliklerin aldıkları önem değerleri Şekil 4.15.'de gösterilmiştir.



Şekil 4.14. Karar Ağaçları Algoritması için ROC Eğrisi



Şekil 4.15. Karar Ağaçları Algoritması için Özniteliklerin Önem Değeri

CICIDS2017 veri kümesi için sonuçlar incelendiğinde test veri kümesinin doğruluk değeri 0.9984 olarak tespit edilmiştir. Eğitim süresinin de diğer yöntemlerle kıyaslandığında 1.5 dakika ile en hızlı algoritma olduğu belirlenmiştir.

4.4.2. AdaBoost

AdaBoost modeli için girilen parametreler ve gerçekleşen eğitim süresi aşağıda belirtildiği gibidir.

- *Eğitim Süresi:* 249 dakika
- *Parametreler:* max_depth=50, learning_rate=0.1, n_estimators=100, random_state=0

Bu model için, çapraz doğrulama değerine ait hesaplama sonucu ve hesaplama süresi aşağıda belirtilmiştir.

- *Çapraz Doğrulama Değeri Hesaplama Süresi:* 76 dakika
- *Çapraz Doğrulama Değeri Hesaplama Sonucu:* [0.99871313 0.9985365 0.99813277 0.9983851 0.9987131 0.99851123 0.99876356 0.998486 0.99843553 0.99846076]
- *Çapraz Doğrulama Değeri Ortalaması:* 0.9985

Veri kümesindeki başarımı göstermek amacıyla, elde edilen sonuçlar Şekil 4.16.'da belirtilmiştir. 'Label' özniteliğindeki değerler sayısala dönüştürüldüğü için etiket bilgisi 0-14 aralığında gösterilmiştir.

Karşılaştırma matrisi üzerinden elde edilen DP, YP ve YN değerleri sırasıyla; 847968, 1255, 1255 şeklindedir. Test veri kümesindeki toplam veri sayısı 849223'dür.

	0	1	2	3	4	5	...	9	10	11	12	13	14
0	681827	107	2	2	90	10	...	0	277	0	3	0	2
1	101	487	0	0	0	0	...	0	0	0	0	0	0
2	13	0	38225	0	0	0	...	0	0	0	0	0	0
3	6	0	0	3081	10	0	...	0	0	0	0	0	0
4	44	0	0	7	69152	0	...	0	8	0	1	0	0
5	10	0	0	1	0	1604	...	0	0	0	0	0	1
6	5	0	0	2	0	5	...	0	1	0	1	1	1
7	0	0	0	0	0	0	...	0	0	0	0	0	0
8	0	0	0	0	0	0	...	0	0	0	0	0	0
9	4	0	0	0	0	0	...	6	0	0	0	0	0
10	270	0	0	0	6	0	...	0	47312	0	1	0	1
11	1	0	0	0	0	0	...	0	0	1756	0	0	0
12	4	0	0	0	0	0	...	0	3	0	355	2	107
13	1	0	0	0	0	1	...	0	1	0	0	4	0
14	4	0	0	0	0	2	...	0	4	0	118	0	94

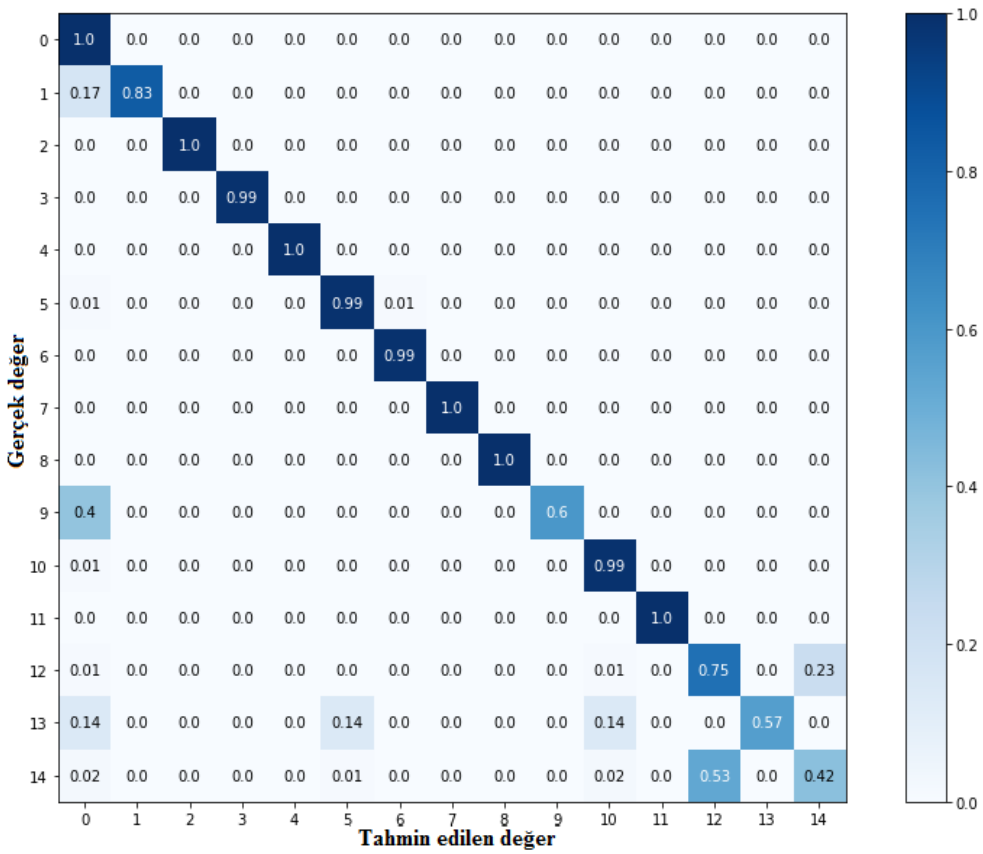
[15 rows x 15 columns]

Şekil 4.16. AdaBoost Algoritması için Karşılaştırma Matrisi

CICIDS2017 veri kümesi üzerinde gerçekleştirilen sınıflandırma işlemi sonucunda elde edilen sınıflandırma raporu Şekil 4.17.'de gösterilmiştir. AdaBoost algoritmasıyla gerçekleştirilen bu modelde etiket bilgisi 'Label' parametresi üzerinden yapılan karşılaştırma sonucunda elde edilen başarım oranları Şekil 4.17.'de bulunmaktadır. Sınıflandırma işleminde sadece saldırı veya normal paket ayrımı yapılmayıp ayrıca saldırı çeşitleri de tespit edilmiştir. Şekil 4.17. ve Şekil 4.18.'de saldırı tipleri için de başarım oranları gösterilmiştir. Şekil 4.18.'de, test veri kümesindeki gerçek değerler ile modelin tahmin ettiği değerler karşılaştırılmıştır. Çizelge 4.6.'da 'Label' özniteliğinde bulunan sayısal değerlerin kategorik değer karşılıkları listelenmiştir. Şekil 4.18. incelendiğinde Adaboost algoritmasının 'Bot', 'Infiltration' ve 'Web Attack' başlıklı saldırılarda başarım oranlarının daha düşük olduğu görülmektedir.

	precision	recall	f1-score	support
BENIGN	0.9993	0.9993	0.9993	682324
Bot	0.8199	0.8282	0.8240	588
DDoS	0.9999	0.9997	0.9998	38238
DoS GoldenEye	0.9961	0.9948	0.9955	3097
DoS Hulk	0.9985	0.9991	0.9988	69212
DoS Slowhttptest	0.9889	0.9871	0.9880	1625
DoS slowloris	0.9937	0.9909	0.9923	1762
FTP-Patator	0.9987	1.0000	0.9994	2317
Heartbleed	1.0000	1.0000	1.0000	2
Infiltration	1.0000	0.6000	0.7500	10
PortScan	0.9938	0.9942	0.9940	47590
SSH-Patator	1.0000	0.9994	0.9997	1757
Web Attack Brute Force	0.7411	0.7521	0.7466	472
Web Attack Sql Injection	0.5714	0.5714	0.5714	7
Web Attack XSS	0.4563	0.4234	0.4393	222
accuracy			0.9985	849223
macro avg	0.9039	0.8760	0.8865	849223
weighted avg	0.9985	0.9985	0.9985	849223

Şekil 4.17. AdaBoost Algoritması için Sınıflandırma Raporu

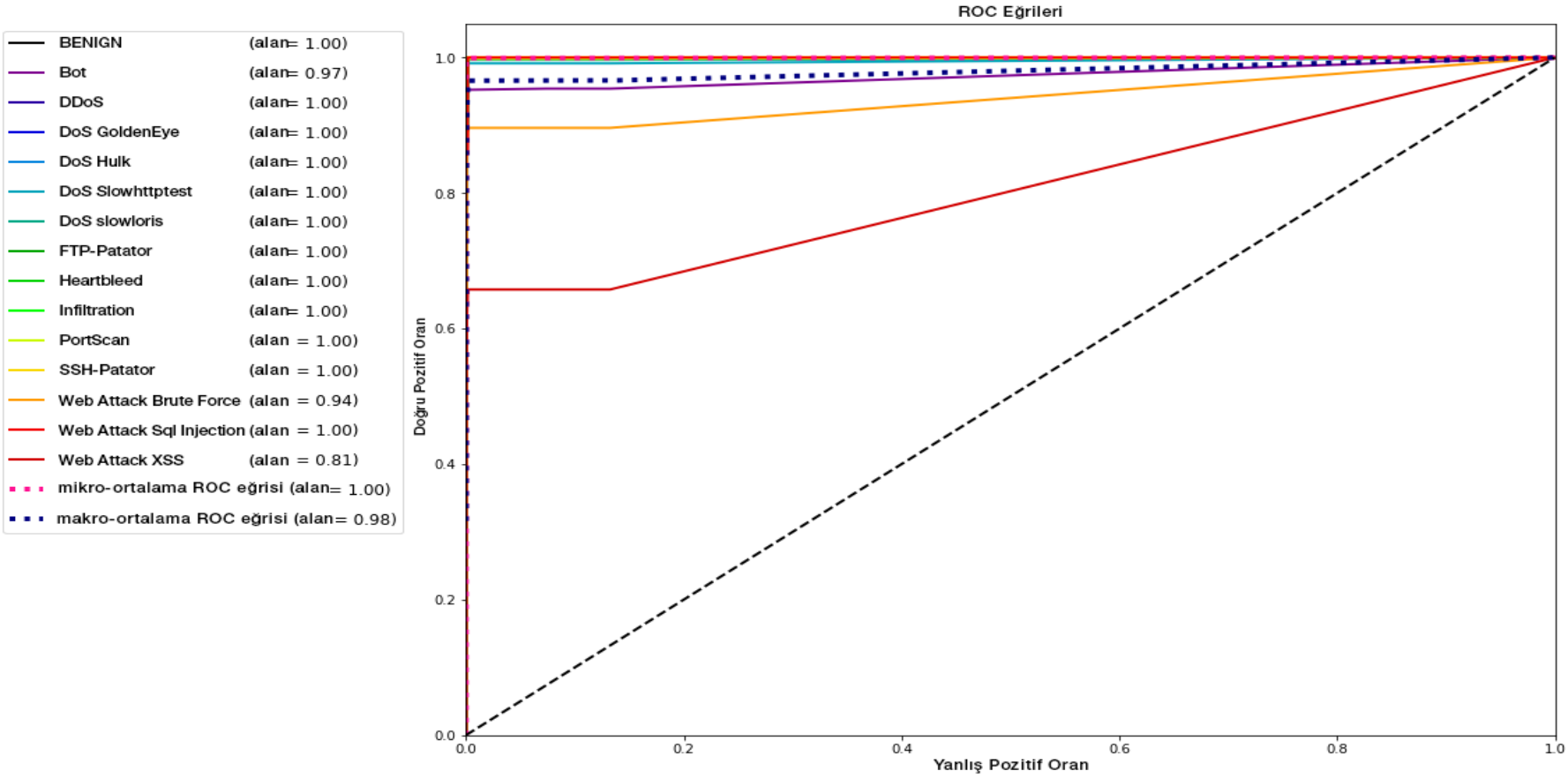


Şekil 4.18. AdaBoost Algoritması için Karşılaştırma Matrisi

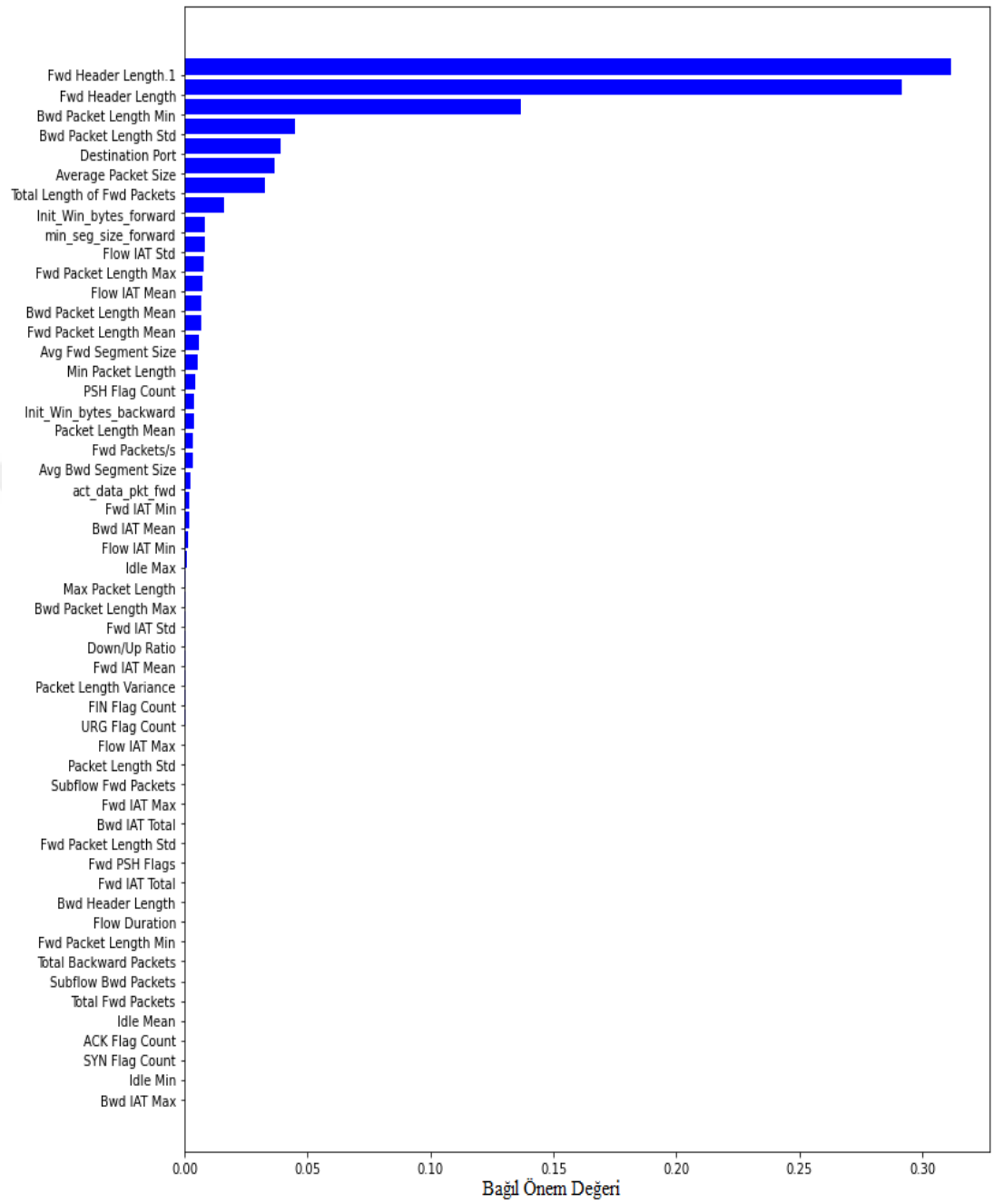
Bu modelde, her bir etiket bilgisi için elde edilen ROC Eğrisi grafiği Şekil 4.19.'da gösterilmiştir. AUC değeri ise testler sonucunda 0.9985 olarak bulunmuştur. AUC değeri 1.00'a ne kadar yaklaşırsa modelin ezberleme oranı daha düşük olmakta ve algoritmanın başarımı yükselmektedir.

Şekil 4.19.'da ROC eğrisine ait grafik detaylı incelendiğinde ise etiketlenmiş saldırılar içerisinde 'Bot' ve 'Web Attack' başlıklı saldırıların AUC değerinin altında kaldığı gözlenmiştir. AUC değerinin altında kalan etiketlenmiş veriler için, Adaboost algoritması modelinin başarımını olumsuz etkilediği ve sınıflandırmada bir miktar tutarsızlık olduğu belirtilebilir. Ancak diğer saldırıların sınıflandırılmasında tutarlılık gözlemlenmiştir. AUC değerinin altında kalan saldırı çeşitlerinin sayısal yoğunluğu ile veri kümesinin tamamı orantılanacak olursa, 1.282/849.223 işlemi sonucu %0.15 gibi düşük bir değer elde edilir. Bu değerin bu kadar düşük olmasının sebebi, bu saldırı çeşitlerini belirten verilerin sayısının veri kümesinde az olmasıdır.

Toplam 53 adet öznitelik içerisinde adaboost algoritmasıyla oluşturulan bu modelde 'Fwd Header Length.1' isimli öznitelik 0.3114 ile en yüksek değere sahip olmuştur. CICIDS2017 veri kümesinde yapılan sınıflandırma işleminde hesaplamalara etki eden özelliklerin aldıkları önem değerleri Şekil 4.20.'de gösterilmiştir.



Şekil 4.19. AdaBoost Algoritması için ROC Eğrisi



Şekil 4.20. AdaBoost Algoritması için Özelliklerin Önem Değeri

CICIDS2017 veri kümesi için sonuçlar incelendiğinde test veri kümesinin doğruluk değeri 0.9985 olarak tespit edilmiştir.

4.4.3. CatBoost

CatBoost modeli için girilen parametreler ve gerçekleşen eğitim süresi aşağıda belirtildiği gibidir.

- *Eğitim Süresi:* 19 dakika
- *Parametreler:* learning_rate=0.1, depth=8, eval_metric='Accuracy', iterations=100, random_state=0

Bu model için, çapraz doğrulama değerine ait hesaplama sonucu ve hesaplama süresi aşağıda belirtilmiştir.

- *Çapraz Doğrulama Değeri Hesaplama Süresi:* 29 dakika
- *Çapraz Doğrulama Değeri Hesaplama Sonucu:* [0.99795615 0.99790568 0.99742626 0.99800661 0.99777946 0.99793086 0.99808226 0.99813273 0.99790563 0.99795609]
- *Çapraz Doğrulama Değeri Ortalaması:* 0.9979

Veri kümesindeki başarımı göstermek amacıyla, elde edilen sonuçlar Şekil 4.21.'de belirtilmiştir. 'Label' özniteliğindeki değerler sayısala dönüştürüldüğü için etiket bilgisi 0-14 aralığında gösterilmiştir.

Karşılaştırma matrisi üzerinden elde edilen DP, YP ve YN değerleri sırasıyla; 847672, 1551, 1551 şeklindedir. Test veri kümesindeki toplam veri sayısı 849223'dür.

	0	1	2	3	4	5	6	...	8	9	10	11	12	13	14
0	681642	1	33	18	338	18	2	...	0	0	270	0	2	0	0
1	346	242	0	0	0	0	0	...	0	0	0	0	0	0	0
2	43	0	38194	0	1	0	0	...	0	0	0	0	0	0	0
3	54	0	0	3018	24	1	0	...	0	0	0	0	0	0	0
4	46	0	0	1	69164	0	0	...	0	0	0	0	1	0	0
5	12	0	0	1	2	1603	6	...	0	0	0	0	1	0	0
6	13	0	0	1	0	8	1737	...	0	0	0	0	3	0	0
7	4	0	0	0	0	0	0	...	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	...	1	0	0	0	0	0	0
9	10	0	0	0	0	0	0	...	0	0	0	0	0	0	0
10	17	0	0	0	12	0	0	...	0	0	47559	0	2	0	0
11	5	0	0	0	0	0	0	...	0	0	2	1748	0	0	0
12	23	0	0	0	0	0	0	...	0	0	0	0	449	0	0
13	7	0	0	0	0	0	0	...	0	0	0	0	0	0	0
14	27	0	0	1	0	0	0	...	0	0	0	0	192	0	2

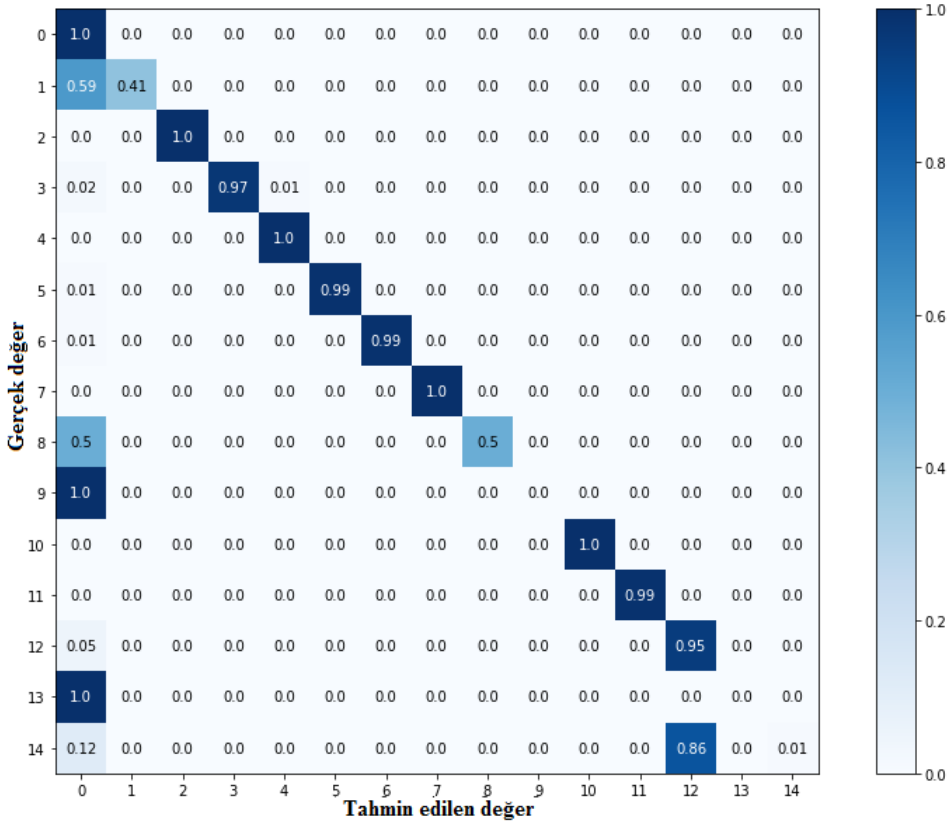
[15 rows x 15 columns]

Şekil 4.21. CatBoost Algoritması için Karşılaştırma Matrisi

CICIDS2017 veri kümesi üzerinde gerçekleştirilen sınıflandırma işlemi sonucunda elde edilen sınıflandırma raporu Şekil 4.22.'de gösterilmiştir. CatBoost algoritmasıyla gerçekleştirilen bu modelde etiket bilgisi 'Label' parametresi üzerinden yapılan karşılaştırma sonucunda elde edilen başarım oranları Şekil 4.22.'de bulunmaktadır. Sınıflandırma işleminde sadece saldırı veya normal paket ayrımı yapılmayıp ayrıca saldırı çeşitleri de tespit edilmiştir. Şekil 4.22. ve Şekil 4.23.'de saldırı tipleri için de başarım oranları gösterilmiştir. Şekil 4.23.'de, test veri kümesindeki gerçek değerler ile modelin tahmin ettiği değerler karşılaştırılmıştır. Çizelge 4.6.'da 'Label' özniteliğinde bulunan sayısal değerlerin kategorik değer karşılıkları listelenmiştir. Şekil 4.23. incelendiğinde Catboost algoritmasının 'Bot', 'Heartbleed', 'Infiltration' ve 'Web Attack' başlıklı saldırılarda yanılma oranlarının olduğu görülmektedir. Bu kadar fazla yanılma oranının olması YP ve YN değerleriyle doğrudan ilişkilidir. Başarım oranı değerlendirilirken bu değerler önem arz etmektedir.

	precision	recall	f1-score	support
BENIGN	0.9993	0.9993	0.9993	682324
Bot	0.8199	0.8282	0.8240	588
DDoS	0.9999	0.9997	0.9998	38238
DoS GoldenEye	0.9961	0.9948	0.9955	3097
DoS Hulk	0.9985	0.9991	0.9988	69212
DoS Slowhttptest	0.9889	0.9871	0.9880	1625
DoS slowloris	0.9937	0.9909	0.9923	1762
FTP-Patator	0.9987	1.0000	0.9994	2317
Heartbleed	1.0000	1.0000	1.0000	2
Infiltration	1.0000	0.6000	0.7500	10
PortScan	0.9938	0.9942	0.9940	47590
SSH-Patator	1.0000	0.9994	0.9997	1757
Web Attack Brute Force	0.7411	0.7521	0.7466	472
Web Attack Sql Injection	0.5714	0.5714	0.5714	7
Web Attack XSS	0.4563	0.4234	0.4393	222
accuracy			0.9985	849223
macro avg	0.9039	0.8760	0.8865	849223
weighted avg	0.9985	0.9985	0.9985	849223

Şekil 4.22. CatBoost Algoritması için Sınıflandırma Raporu

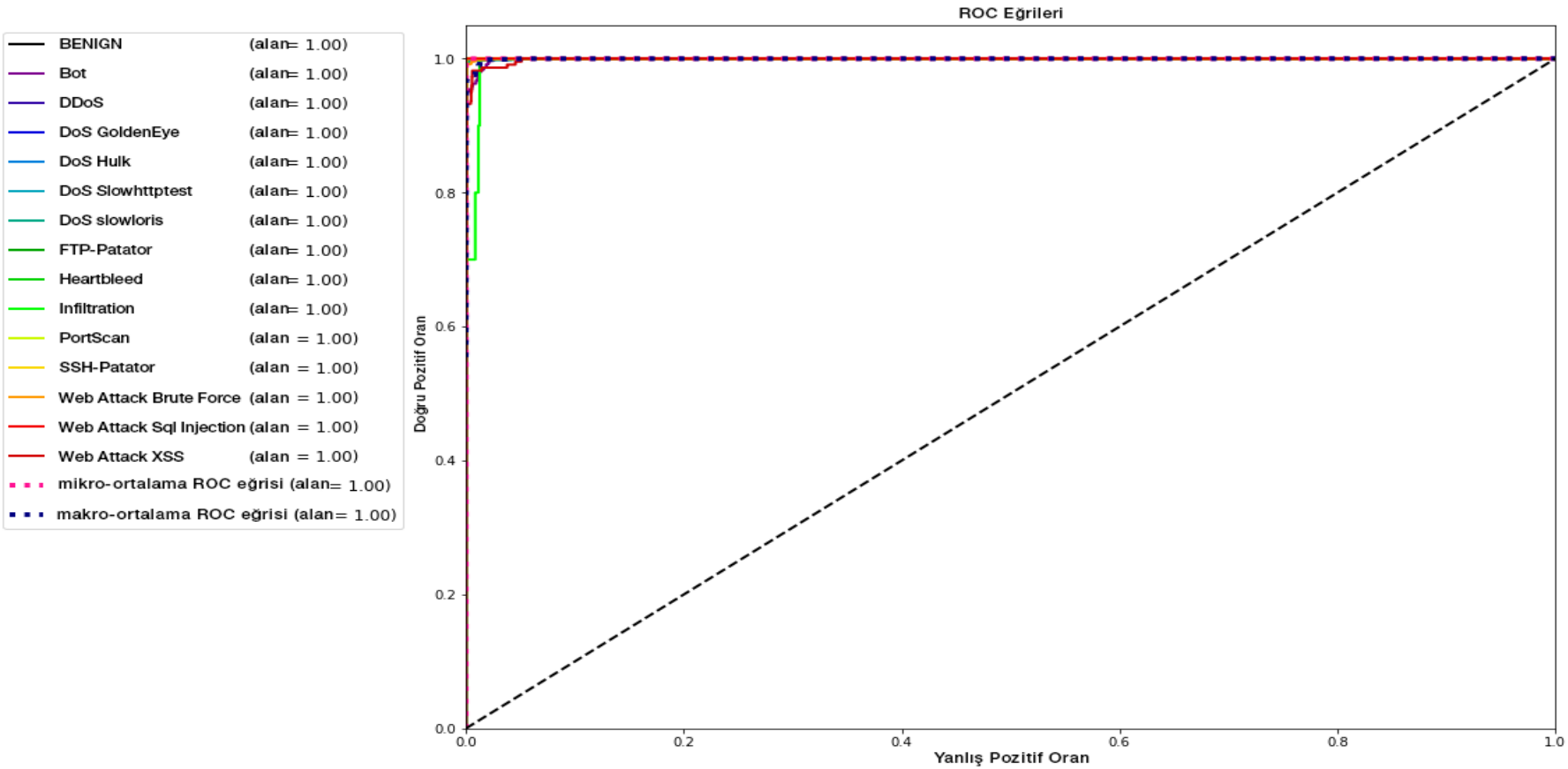


Şekil 4.23. CatBoost Algoritması için Karşılaştırma Matrisi

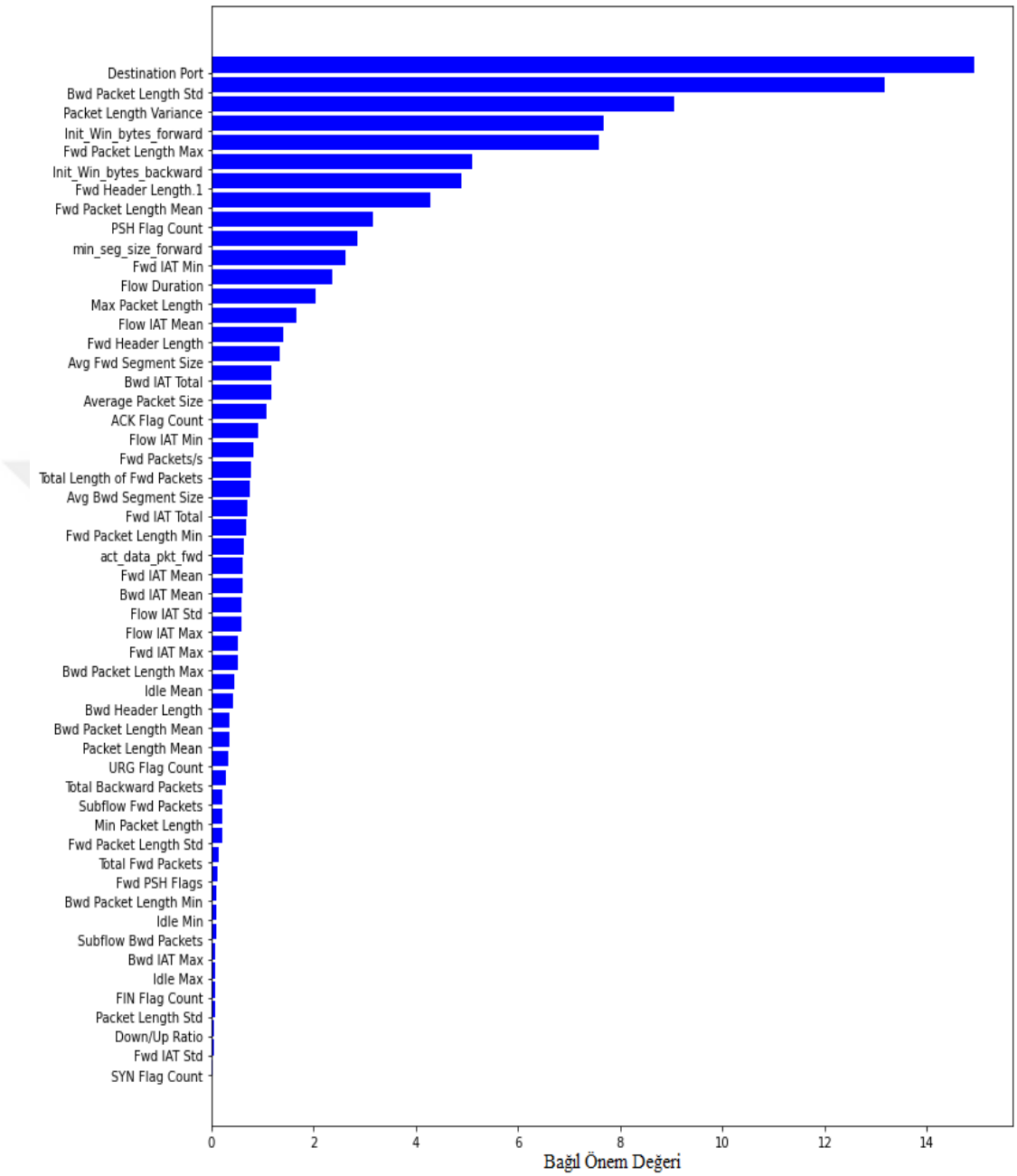
Bu modelde her bir etiket bilgisi için elde edilen ROC Eğrisi grafiği Şekil 4.24.'de gösterilmiştir. AUC değeri ise testler sonucunda 0.9979 olarak bulunmuştur. AUC değeri 1.00'a ne kadar yaklaşırsa modelin ezberleme oranı daha düşük olmakta ve algoritmanın başarımı yükselmektedir.

Şekil 4.24.'de ROC eğrisine ait grafik detaylı incelendiğinde ise etiketlenmiş saldırı çeşitlerine ait ROC eğrilerinin ideale yakın olduğu ve AUC değerlerinin de 1'e çok yakın olduğu görülmektedir. Grafik incelendiğinde CatBoost algoritmasına ait bu modelin sonuçlarında tutarlılık gösterdiği ve değerlendirmelerinde başarılı olduğu belirtilebilir. Ancak Şekil 4.23.'de göründüğü üzere bazı saldırı çeşitlerinde yanlış değerlendirme tespit edilmiştir. Bu durum CatBoost algoritmasının başarımını olumsuz etkilemektedir.

Toplam 53 adet öznitelik içerisinde CatBoost algoritmasıyla oluşturulan bu modelde 'Destination Port' isimli öznitelik 14.9181 ile en yüksek değere sahip olmuştur. CICIDS2017 veri kümesinde yapılan sınıflandırma işleminde hesaplamalara etki eden özniteliklerin aldıkları önem değerleri Şekil 4.25.'de gösterilmiştir.



Şekil 4.24. CatBoost Algoritması için ROC Eğrisi



Şekil 4.25. CatBoost Algoritması için Özelliklerin Önem Değeri

CICIDS2017 veri kümesi için sonuçlar incelendiğinde test veri kümesinin doğruluk değeri 0.9982 olarak tespit edilmiştir.

4.4.4. Gradyan arttırma

Gradyan Arttırma modeli için girilen parametreler ve gerçekleşen eğitim süresi aşağıda belirtildiği gibidir.

- *Eğitim Süresi:* 378 dakika
- *Parametreler:* learning_rate=0.1, max_depth=50, n_estimators=100, random_state=0,

Bu model için, çapraz doğrulama değerine ait hesaplama sonucu ve hesaplama süresi aşağıda belirtilmiştir.

- *Çapraz Doğrulama Değeri Hesaplama Süresi:* 172 dakika
- *Çapraz Doğrulama Değeri Hesaplama Sonucu:* [0.99823371 0.9983851 0.99815801 0.99813277 0.99820843 0.99798133 0.9984103 0.99835983 0.99823366 0.99846076]
- *Çapraz Doğrulama Değeri Ortalaması:* 0.9983

Veri kümesindeki başarımı göstermek amacıyla, elde edilen sonuçlar Şekil 4.26.'da belirtilmiştir. 'Label' özniteliğindeki değerler sayısala dönüştürüldüğü için etiket bilgisi 0-14 aralığında gösterilmiştir.

Karşılaştırma matrisi üzerinden elde edilen DP, YP ve YN değerleri sırasıyla; 847857, 1366, 1366 şeklindedir. Test veri kümesindeki toplam veri sayısı 849223'dür.

	0	1	2	3	4	5	...	9	10	11	12	13	14
0	681790	109	1	19	84	17	...	8	273	1	5	3	6
1	112	476	0	0	0	0	...	0	0	0	0	0	0
2	15	0	38223	0	0	0	...	0	0	0	0	0	0
3	16	0	0	3071	10	0	...	0	0	0	0	0	0
4	43	0	0	8	69152	2	...	0	6	0	0	0	1
5	19	0	0	1	0	1597	...	0	0	0	0	0	1
6	12	0	0	1	0	9	...	0	0	0	1	1	0
7	0	0	0	0	0	0	...	0	0	0	0	0	0
8	0	0	0	0	0	0	...	0	0	0	0	0	0
9	6	0	0	0	0	0	...	4	0	0	0	0	0
10	237	0	0	0	6	0	...	0	47346	0	0	0	1
11	1	0	0	0	0	0	...	0	0	1756	0	0	0
12	59	0	0	0	0	0	...	0	2	0	301	1	109
13	5	0	0	0	0	0	...	0	0	0	0	2	0
14	31	0	0	1	0	0	...	0	1	0	107	0	82

[15 rows x 15 columns]

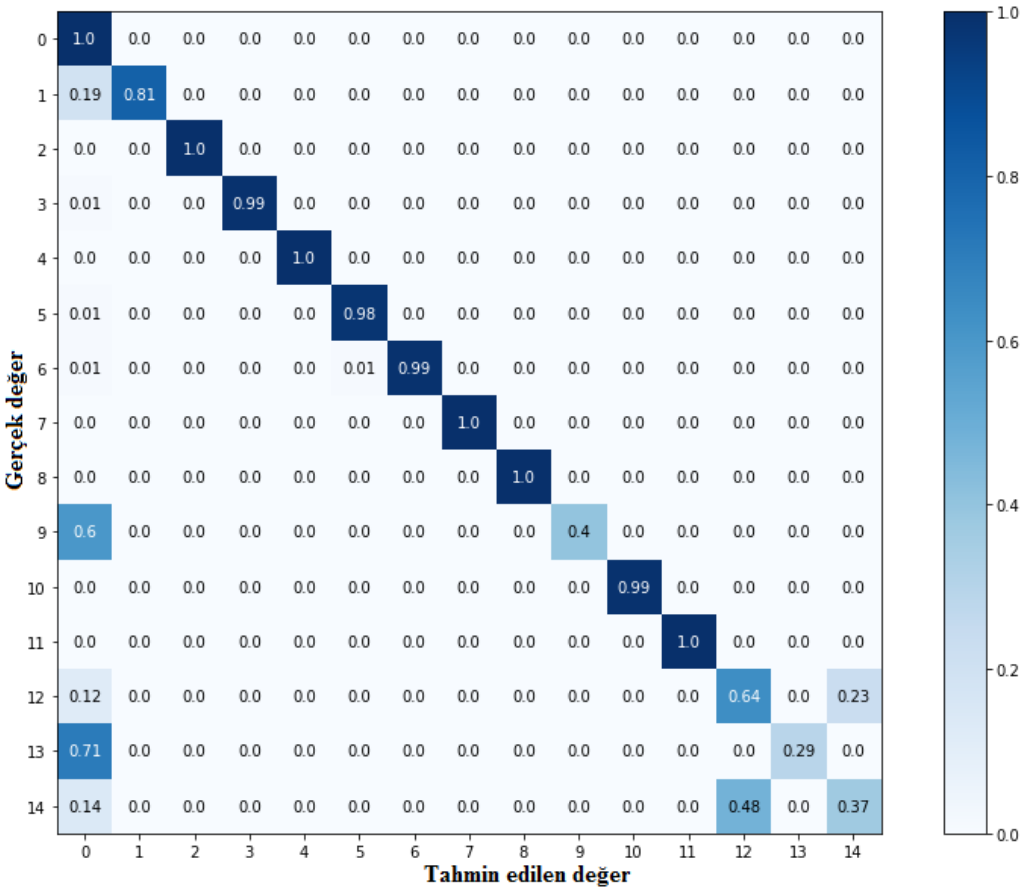
Şekil 4.26. Gradyan Arttırma Algoritması için Karşılaştırma Matrisi

CICIDS2017 veri kümesi üzerinde gerçekleştirilen sınıflandırma işlemi sonucunda elde edilen sınıflandırma raporu Şekil 4.27.'de gösterilmiştir.

Gradyan arttırma algoritmasıyla gerçekleştirilen bu modelde etiket bilgisi 'Label' parametresi üzerinden yapılan karşılaştırma sonucunda elde edilen başarım oranları Şekil 4.27.'de bulunmaktadır. Sınıflandırma işleminde sadece saldırı veya normal paket ayrımı yapılmayıp ayrıca saldırı çeşitleri de tespit edilmiştir. Şekil 4.27. ve Şekil 4.28.'de saldırı tipleri için de başarım oranları gösterilmiştir. Şekil 4.28.'de, test veri kümesindeki gerçek değerler ile modelin tahmin ettiği değerler karşılaştırılmıştır. Çizelge 4.6.'da 'Label' özniteliğinde bulunan sayısal değerlerin kategorik değer karşılıkları listelenmiştir. Şekil 4.28. incelendiğinde gradyan arttırma algoritmasının 'Infiltration' ve 'Web Attack' başlıklı saldırılarda başarım oranlarının daha düşük olduğu görülmektedir.

	precision	recall	f1-score	support
BENIGN	0.9992	0.9992	0.9992	682324
Bot	0.8137	0.8095	0.8116	588
DDoS	1.0000	0.9996	0.9998	38238
DoS GoldenEye	0.9903	0.9916	0.9910	3097
DoS Hulk	0.9986	0.9991	0.9988	69212
DoS Slowhttptest	0.9828	0.9828	0.9828	1625
DoS slowloris	0.9926	0.9864	0.9895	1762
FTP-Patator	0.9991	1.0000	0.9996	2317
Heartbleed	1.0000	1.0000	1.0000	2
Infiltration	0.3333	0.4000	0.3636	10
PortScan	0.9941	0.9949	0.9945	47590
SSH-Patator	0.9994	0.9994	0.9994	1757
Web Attack Brute Force	0.7271	0.6377	0.6795	472
Web Attack Sql Injection	0.2857	0.2857	0.2857	7
Web Attack XSS	0.4100	0.3694	0.3886	222
accuracy			0.9984	849223
macro avg	0.8351	0.8304	0.8322	849223
weighted avg	0.9984	0.9984	0.9984	849223

Şekil 4.27. Gradyan Arttırma Algoritması için Sınıflandırma Raporu

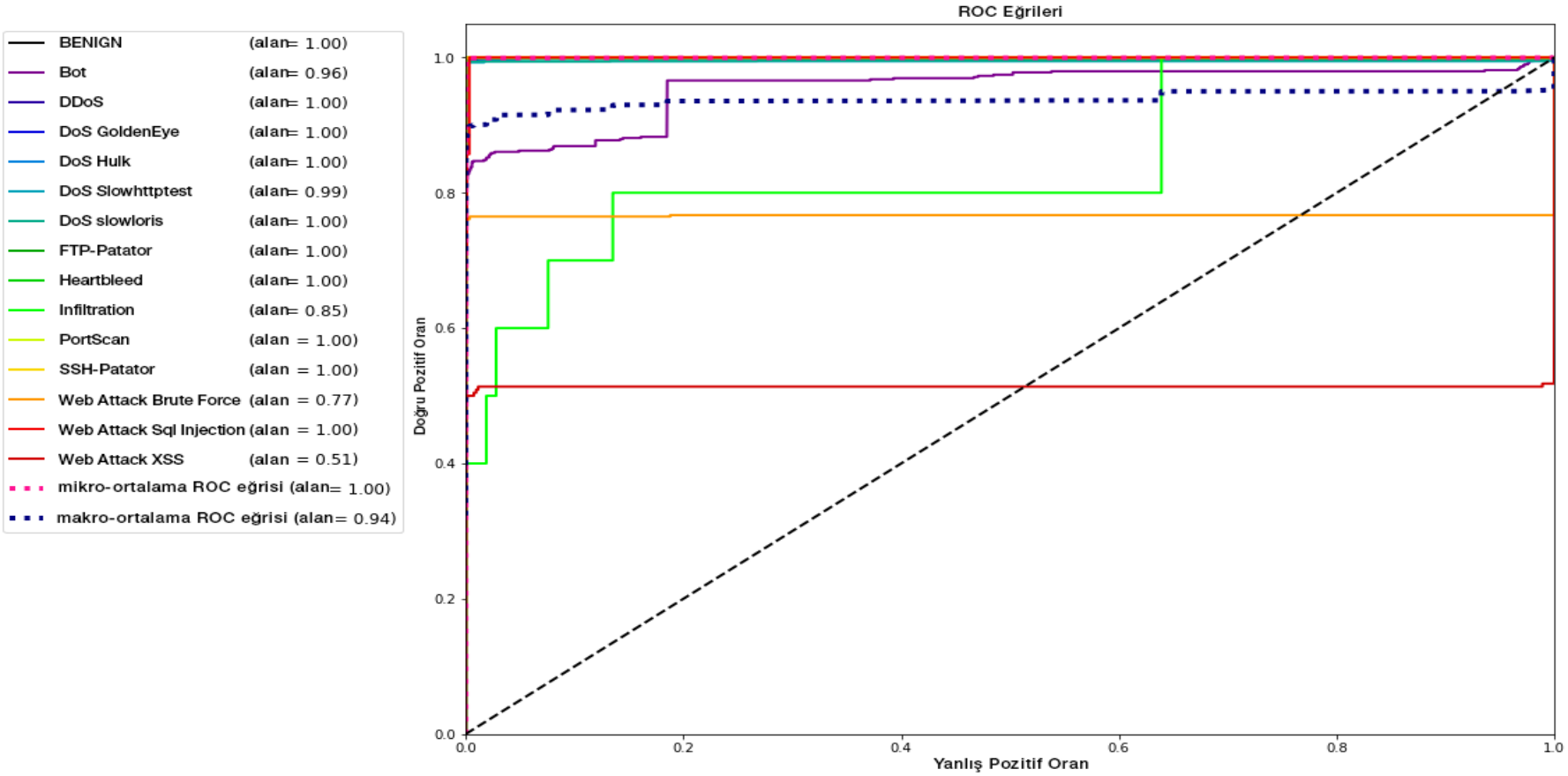


Şekil 4.28. Gradyan Arttırma Algoritması için Karşılaştırma Matrisi

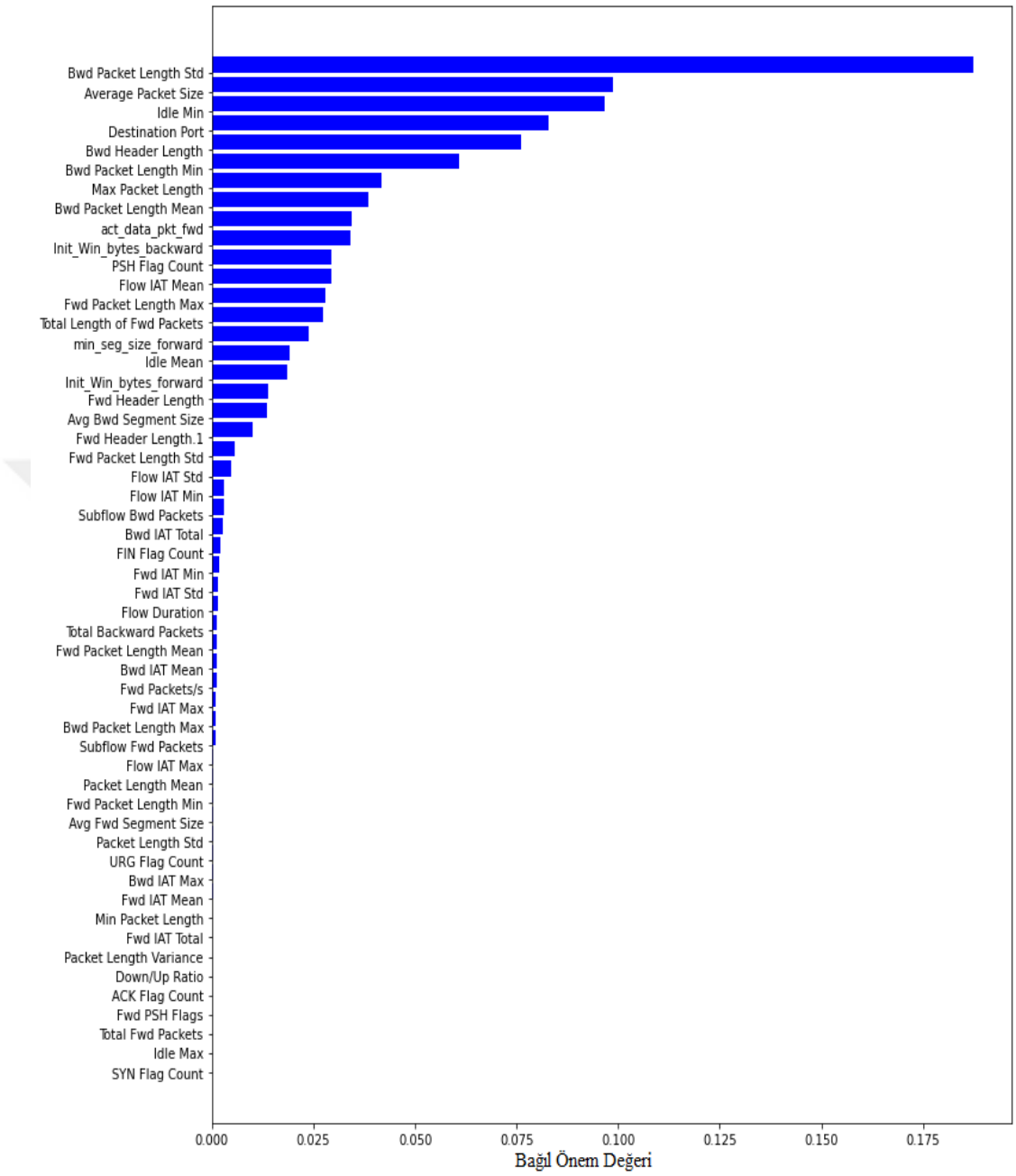
Bu modelde her bir etiket bilgisi için elde edilen ROC Eğrisi grafiği Şekil 4.29.'da gösterilmiştir. AUC değeri ise testler sonucunda 0.9983 olarak bulunmuştur. AUC değeri 1.00'a ne kadar yaklaşırsa modelin ezberleme oranı daha düşük olmakta ve algoritmanın başarımı yükselmektedir.

Şekil 4.29.'da ROC eğrisine ait grafik detaylı incelendiğinde ise etiketlenmiş saldırılar içerisinde 'Bot', 'Infiltration' ve 'Web Attack' başlıklı saldırıların AUC değerinin altında kaldığı gözlenmiştir. AUC değerinin altında kalan etiketlenmiş veriler için, gradyan arttırma algoritması modelinin başarımını olumsuz etkilediği ve sınıflandırmada ciddi ölçüde tutarsızlıkların olduğu belirtilebilir. Ancak diğer saldırıların sınıflandırılmasında tutarlılık gözlemlenmiştir. AUC değerinin altında kalan saldırı çeşitlerinin sayısal yoğunluğu ile veri kümesinin tamamı orantılanacak olursa, 1.292/849.223 işlemi sonucu %0.15 gibi düşük bir değer elde edilir. Bu değerin bu kadar düşük olmasının sebebi, bu saldırı çeşitlerini belirten verilerin sayısının veri kümesinde az olmasıdır.

Toplam 53 adet özellik içerisinde Gradyan Arttırma algoritmasıyla oluşturulan bu modelde 'Bwd Packet Length Std' isimli özellik 0.1873 ile en yüksek değere sahip olmuştur. CICIDS2017 veri kümesinde yapılan sınıflandırma işleminde hesaplamalara etki eden özniteliklerin aldıkları önem değerleri Şekil 4.30.'da gösterilmiştir.



Şekil 4.29. Gradyan Arttırma Algoritması için ROC Eğrisi



Şekil 4.30. Gradyan Arttırma Algoritması için Özelliklerin Önem Değeri

CICIDS2017 veri kümesi için sonuçlar incelendiğinde test veri kümesinin doğruluk değeri 0.9984 olarak tespit edilmiştir.

4.4.5. LightGBM

LightGBM modeli için girilen parametreler ve gerçekleşen eğitim süresi aşağıda belirtildiği gibidir.

- *Eğitim Süresi:* 4.5 dakika
- *Parametreler:* learning_rate=0.1, max_depth=50, n_estimators=100, n_jobs=-1, random_state=None,

Bu model için, çapraz doğrulama değerine ait hesaplama sonucu ve hesaplama süresi aşağıda belirtilmiştir.

- *Çapraz Doğrulama Değeri Hesaplama Süresi:* 9.5 dakika
- *Çapraz Doğrulama Değeri Hesaplama Sonucu:* [0.96755521 0.93821914 0.96494106 0.97137046 0.82312568 0.9487666 0.95541302 0.94567302 0.97493338 0.9549891]
- *Çapraz Doğrulama Değeri Ortalaması:* 0.9445

Veri kümesindeki başarımı göstermek amacıyla, elde edilen sonuçlar Şekil 4.31.'de belirtilmiştir. 'Label' özniteliğindeki değerler sayısala dönüştürüldüğü için etiket bilgisi 0-14 aralığında gösterilmiştir.

Karşılaştırma matrisi üzerinden elde edilen DP, YP ve YN değerleri sırasıyla; 829985, 19238, 19238 şeklindedir. Test veri kümesindeki toplam veri sayısı 849223'dür. YP ve YN değerleri bakımından diğer algoritmalarla karşılaştırıldığında en fazla değere sahip olan algoritma olarak başarımı en düşük olan algoritma olduğu tespit edilmiştir.

	0	1	2	3	4	5	...	9	10	11	12	13	14
0	674724	502	1322	613	1324	883	...	9	881	78	638	23	128
1	380	0	1	3	202	1	...	0	0	0	0	0	0
2	77	0	38010	23	31	58	...	0	2	0	0	0	7
3	186	384	0	2218	159	86	...	0	3	0	0	0	0
4	4879	2	62	78	63888	124	...	0	92	27	3	0	33
5	730	0	37	0	91	609	...	0	22	30	0	0	0
6	301	18	531	12	100	29	...	0	313	0	2	0	0
7	163	0	30	2	94	0	...	0	6	0	0	0	0
8	2	0	0	0	0	0	...	0	0	0	0	0	0
9	10	0	0	0	0	0	...	0	0	0	0	0	0
10	818	0	28	1	6	0	...	0	46542	101	2	0	68
11	93	0	0	0	0	0	...	0	0	1611	0	0	0
12	102	0	11	8	2	0	...	0	0	0	263	0	86
13	4	0	2	0	1	0	...	0	0	0	0	0	0
14	43	0	1	0	0	0	...	0	0	0	99	0	79

[15 rows x 15 columns]

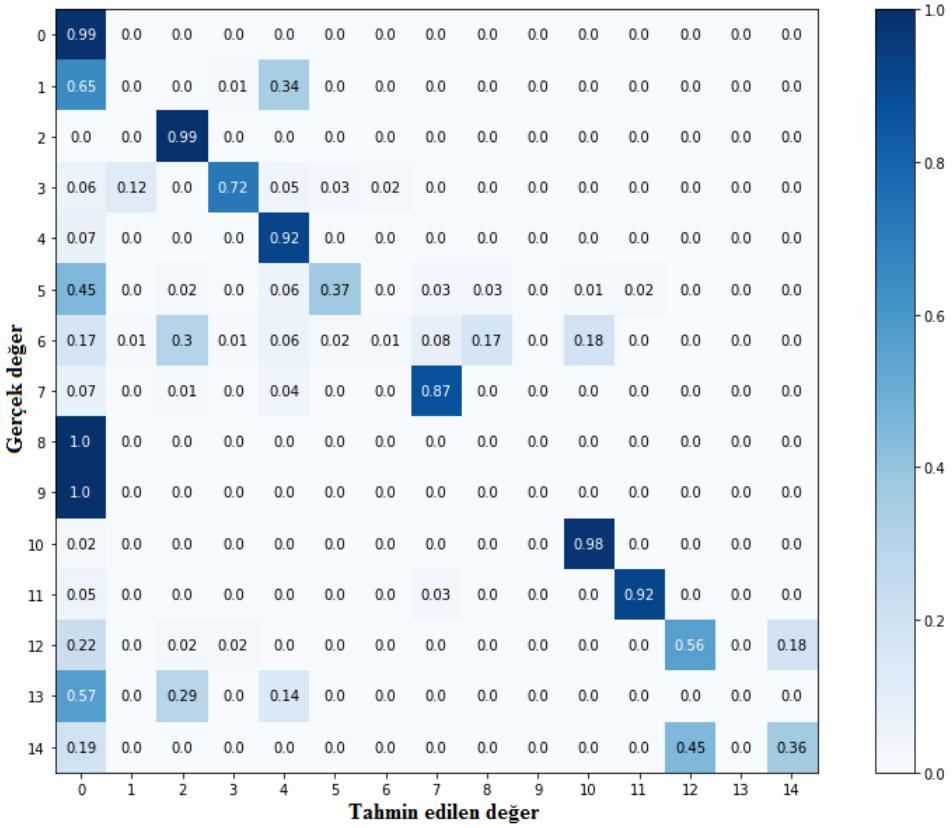
Şekil 4.31. LightGBM Algoritması için Karşılaştırma Matrisi

CICIDS2017 veri kümesi üzerinde gerçekleştirilen sınıflandırma işlemi sonucunda elde edilen sınıflandırma raporu Şekil 4.32.'de gösterilmiştir.

LightGBM algoritmasıyla gerçekleştirilen bu modelde etiket bilgisi 'Label' parametresi üzerinden yapılan karşılaştırma sonucunda elde edilen başarım oranları Şekil 4.32.'de bulunmaktadır. Sınıflandırma işleminde sadece saldırı veya normal paket ayrımı yapılmayıp ayrıca saldırı çeşitleri de tespit edilmiştir. Şekil 4.32. ve Şekil 4.33.'de saldırı çeşitleri için de başarım oranları gösterilmiştir. Şekil 4.33.'de, test veri kümesindeki gerçek değerler ile modelin tahmin ettiği değerler karşılaştırılmıştır. Çizelge 4.6.'da 'Label' özniteliğinde bulunan sayısal değerlerin kategorik değer karşılıkları listelenmiştir. Şekil 4.33. incelendiğinde LightGBM algoritmasının neredeyse bütün saldırılarda yanılma oranlarının olduğu görülmektedir. Bu kadar fazla yanılma oranının olması YP ve YN değerleriyle doğrudan ilişkilidir. Başarım oranı değerlendirilirken bu değerler önem arz etmektedir.

	precision	recall	f1-score	support
BENIGN	0.9886	0.9889	0.9887	682324
Bot	0.0000	0.0000	0.0000	588
DDoS	0.9494	0.9940	0.9712	38238
DoS GoldenEye	0.7498	0.7162	0.7326	3097
DoS Hulk	0.9695	0.9231	0.9457	69212
DoS Slowhttptest	0.3402	0.3748	0.3567	1625
DoS slowloris	0.0436	0.0108	0.0173	1762
FTP-Patator	0.6352	0.8727	0.7353	2317
Heartbleed	0.0000	0.0000	0.0000	2
Infiltration	0.0000	0.0000	0.0000	10
PortScan	0.9724	0.9780	0.9752	47590
SSH-Patator	0.8722	0.9169	0.8940	1757
Web Attack Brute Force	0.2612	0.5572	0.3556	472
Web Attack Sql Injection	0.0000	0.0000	0.0000	7
Web Attack XSS	0.1970	0.3559	0.2536	222
accuracy			0.9773	849223
macro avg	0.4653	0.5126	0.4817	849223
weighted avg	0.9778	0.9773	0.9774	849223

Şekil 4.32. LightGBM Algoritması için Sınıflandırma Raporu

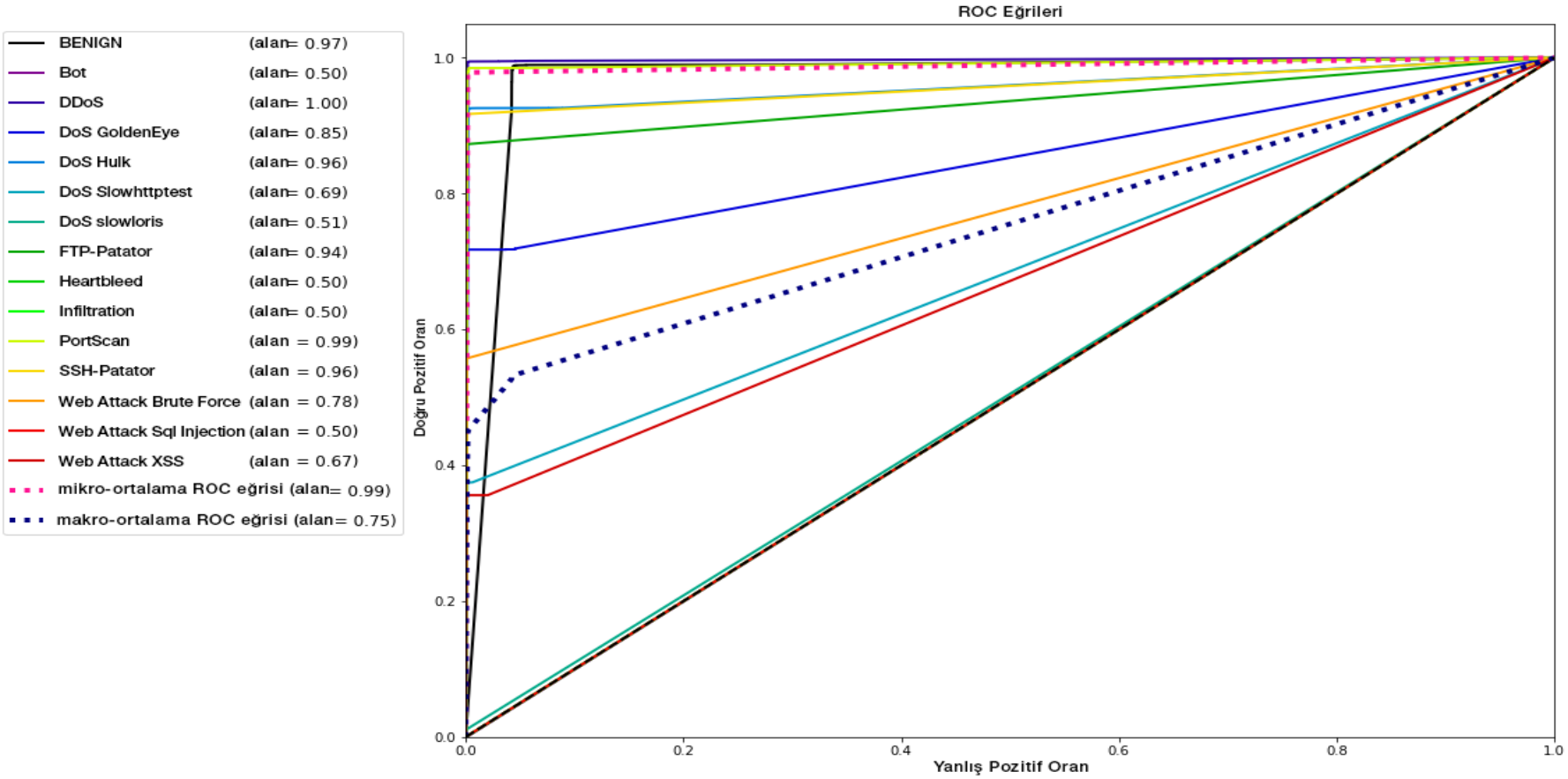


Şekil 4.33. LightGBM Algoritması için Karşılaştırma Matrisi

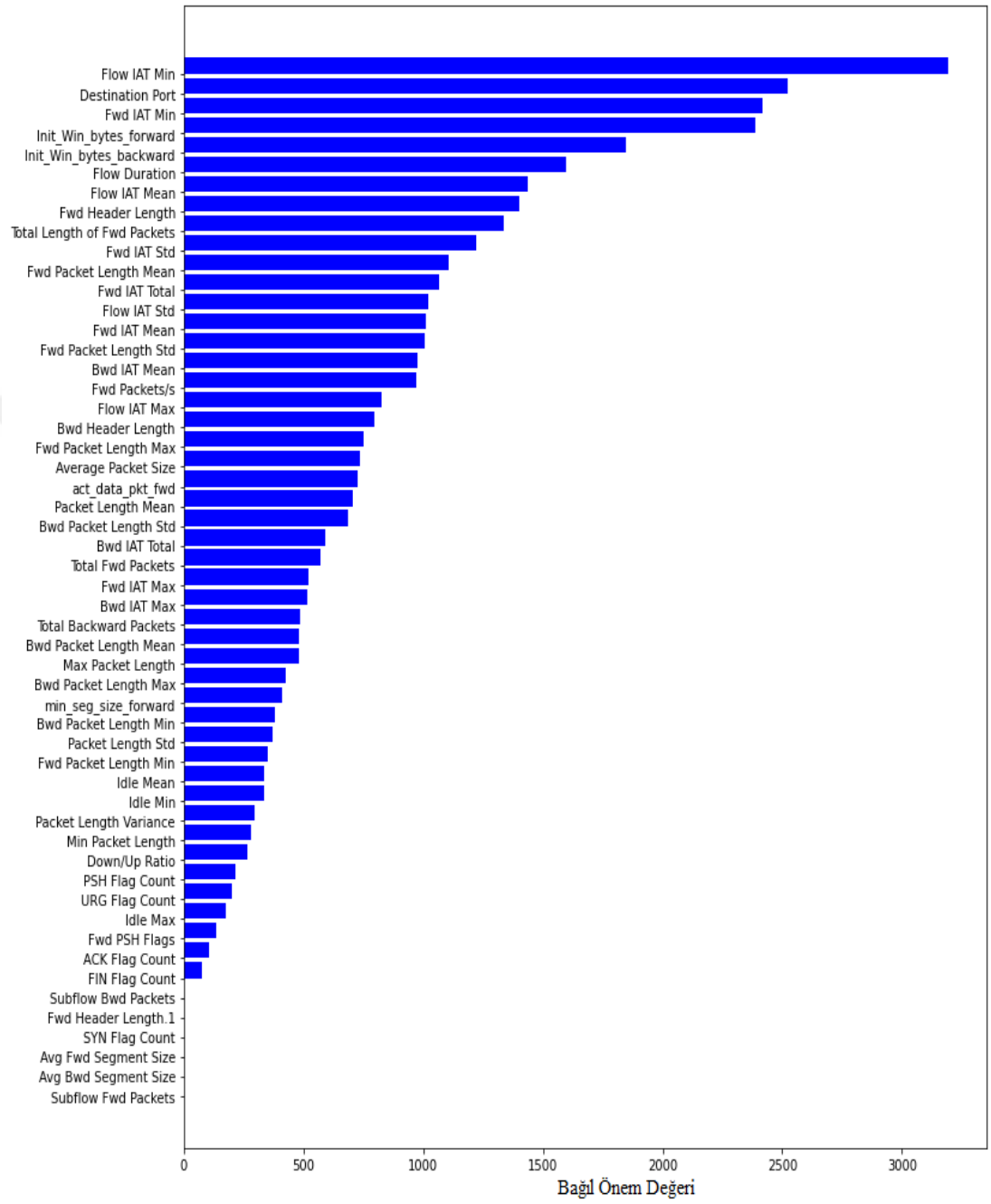
Bu modelde her bir etiket bilgisi için elde edilen ROC Eğrisi grafiği Şekil 4.34.'de gösterilmiştir. AUC değeri ise testler sonucunda 0.9445 olarak bulunmuştur. AUC değeri 1.00'a ne kadar yaklaşırsa modelin ezberleme oranı da daha düşük olmakta ve algoritmanın başarımı yükselmektedir.

Şekil 4.34.'de ROC eğrisine ait grafik detaylı incelendiğinde ise sadece 2 adet saldırı çeşidine ait ROC eğrisinin AUC değerinin üstünde olduğu gözlenmiştir. Bu durum sınıflandırma algoritmalarında istenmeyen bir durumdur. AUC değerinin altında bu kadar fazla sayıda değişkenin ROC eğrisi bulunması sebebiyle LightGBM algoritmasının tutarsızlık gösterdiği ve başarısının düşük olduğu belirtilebilir.

Toplam 53 adet öznitelik içerisinde LightGBM algoritmasıyla oluşturulan bu modelde 'Flow IAT Min' isimli öznitelik 3048 ile en yüksek değere sahip olmuştur. CICIDS2017 veri kümesinde yapılan sınıflandırma işleminde hesaplamalara etki eden özniteliklerin aldıkları önem değerleri Şekil 4.35.'de gösterilmiştir.



Şekil 4.34. LightGBM Algoritması için ROC Eğrisi



Şekil 4.35. LightGBM Algoritması için Özelliklerin Önem Değeri

CICIDS2017 veri kümesi için sonuçlar incelendiğinde test veri kümesinin doğruluk değeri 0.9773 olarak tespit edilmiştir.

4.4.6. XGBoost

XGBoost modeli için girilen parametreler ve gerçekleşen eğitim süresi aşağıda belirtildiği gibidir.

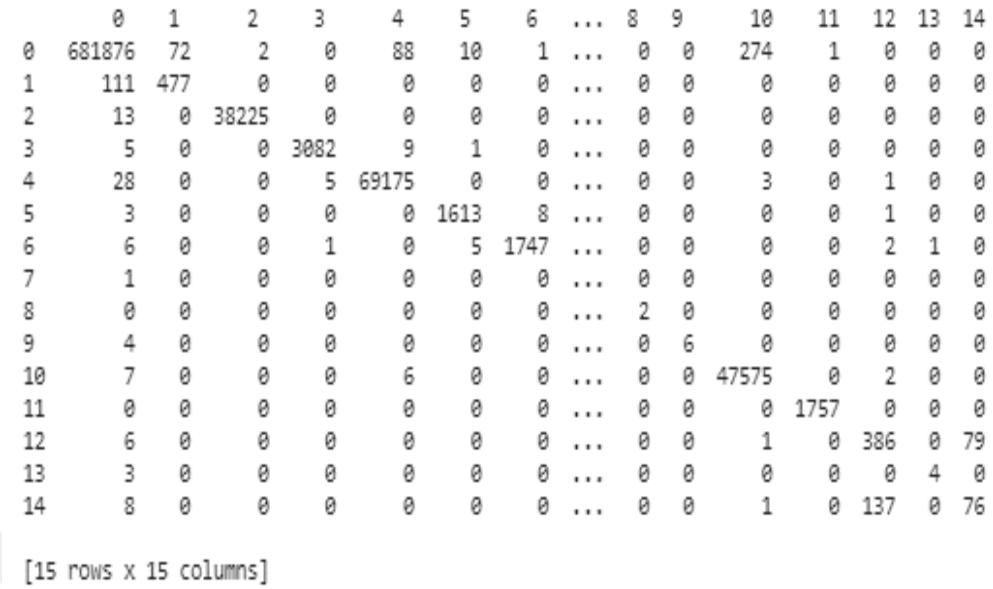
- *Eğitim Süresi:* 210 dakika
- *Parametreler:* learning_rate=0.1, max_depth=50, n_estimators=100, n_jobs=5, random_state=0

Bu model için, çapraz doğrulama değerine ait hesaplama sonucu ve hesaplama süresi aşağıda belirtilmiştir.

- *Çapraz Doğrulama Değeri Hesaplama Süresi:* 206 dakika
- *Çapraz Doğrulama Değeri Hesaplama Sonucu:* [0.99888976 0.99896546 0.9986122 0.99886453 0.99891496 0.99868786 0.99911683 0.99876356 0.99883926 0.99899066]
- *Çapraz Doğrulama Değeri Ortalaması:* 0.9989

Veri kümesindeki başarımı göstermek amacıyla, elde edilen sonuçlar Şekil 4.36.'da belirtilmiştir. 'Label' özniteliğindeki değerler sayısala dönüştürüldüğü için etiket bilgisi 0-14 aralığında gösterilmiştir.

Karşılaştırma matrisi üzerinden elde edilen DP, YP ve YN değerleri sırasıyla; 848317, 906, 906 şeklindedir. Test veri kümesindeki toplam veri sayısı 849223'dür.



	0	1	2	3	4	5	6	...	8	9	10	11	12	13	14
0	681876	72	2	0	88	10	1	...	0	0	274	1	0	0	0
1	111	477	0	0	0	0	0	...	0	0	0	0	0	0	0
2	13	0	38225	0	0	0	0	...	0	0	0	0	0	0	0
3	5	0	0	3082	9	1	0	...	0	0	0	0	0	0	0
4	28	0	0	5	69175	0	0	...	0	0	3	0	1	0	0
5	3	0	0	0	0	1613	8	...	0	0	0	0	1	0	0
6	6	0	0	1	0	5	1747	...	0	0	0	0	2	1	0
7	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	...	2	0	0	0	0	0	0
9	4	0	0	0	0	0	0	...	0	6	0	0	0	0	0
10	7	0	0	0	6	0	0	...	0	0	47575	0	2	0	0
11	0	0	0	0	0	0	0	...	0	0	0	1757	0	0	0
12	6	0	0	0	0	0	0	...	0	0	1	0	386	0	79
13	3	0	0	0	0	0	0	...	0	0	0	0	0	4	0
14	8	0	0	0	0	0	0	...	0	0	1	0	137	0	76

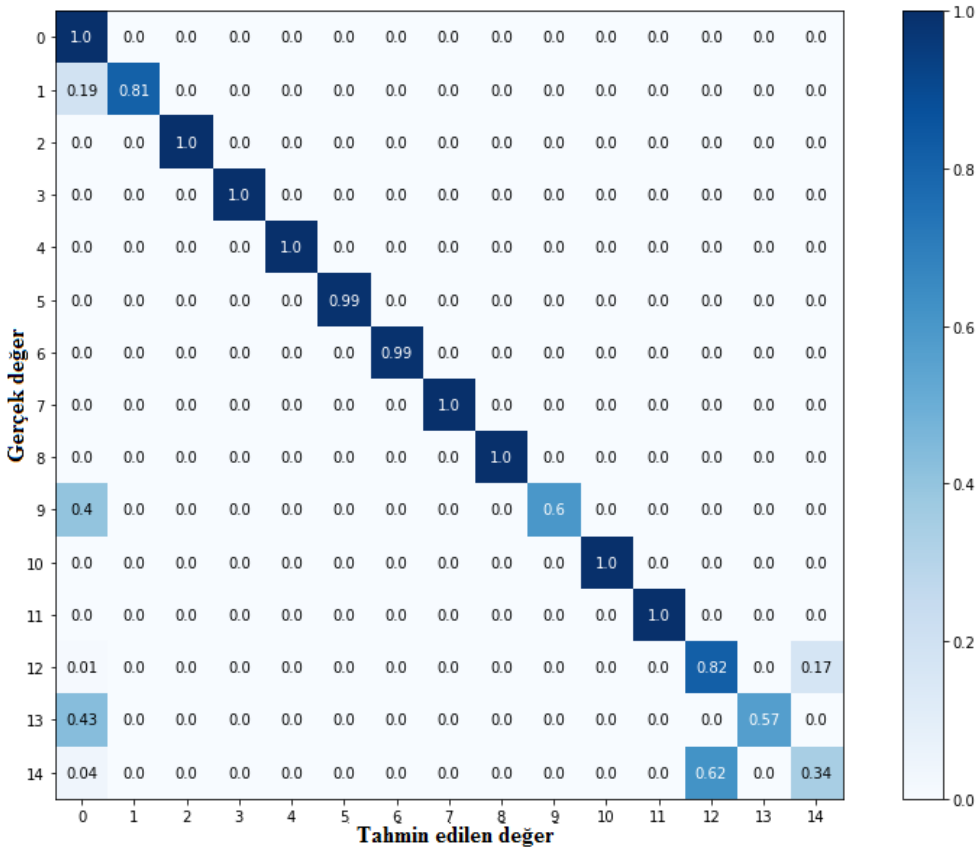
[15 rows x 15 columns]

Şekil 4.36. XGBoost Algoritması için Karşılaştırma Matrisi

CICIDS2017 veri kümesi üzerinde gerçekleştirilen sınıflandırma işlemi sonucunda elde edilen sınıflandırma raporu Şekil 4.37.'de gösterilmiştir. XGBoost algoritmasıyla gerçekleştirilen bu modelde etiket bilgisi 'Label' parametresi üzerinden yapılan karşılaştırma sonucunda elde edilen başarım oranları Şekil 4.37.'de bulunmaktadır. Sınıflandırma işleminde sadece saldırı veya normal paket ayrımı yapılmayıp ayrıca saldırı çeşitleri de tespit edilmiştir. Şekil 4.37. ve Şekil 4.38.'de saldırı çeşitleri için de başarım oranları gösterilmiştir. Şekil 4.38.'de, test veri kümesindeki gerçek değerler ile modelin tahmin ettiği değerler karşılaştırılmıştır. Çizelge 4.6.'da 'Label' özniteliğinde bulunan sayısal değerlerin kategorik değer karşılıkları listelenmiştir. Şekil 4.38. incelendiğinde XGBoost algoritmasının 'Infiltration' ve 'Web Attack' başlıklı saldırılarda başarım oranlarının daha düşük olduğu görülmektedir.

	precision	recall	f1-score	support
BENIGN	0.9997	0.9993	0.9995	682324
Bot	0.8689	0.8112	0.8391	588
DDoS	0.9999	0.9997	0.9998	38238
DoS GoldenEye	0.9981	0.9952	0.9966	3097
DoS Hulk	0.9985	0.9995	0.9990	69212
DoS Slowhttptest	0.9902	0.9926	0.9914	1625
DoS slowloris	0.9949	0.9915	0.9932	1762
FTP-Patator	1.0000	0.9996	0.9998	2317
Heartbleed	1.0000	1.0000	1.0000	2
Infiltration	1.0000	0.6000	0.7500	10
PortScan	0.9942	0.9997	0.9969	47590
SSH-Patator	0.9994	1.0000	0.9997	1757
Web Attack Brute Force	0.7297	0.8178	0.7712	472
Web Attack Sql Injection	0.8000	0.5714	0.6667	7
Web Attack XSS	0.4903	0.3423	0.4032	222
accuracy			0.9989	849223
macro avg	0.9242	0.8747	0.8937	849223
weighted avg	0.9989	0.9989	0.9989	849223

Şekil 4.37. XGBoost Algoritması için Sınıflandırma Raporu



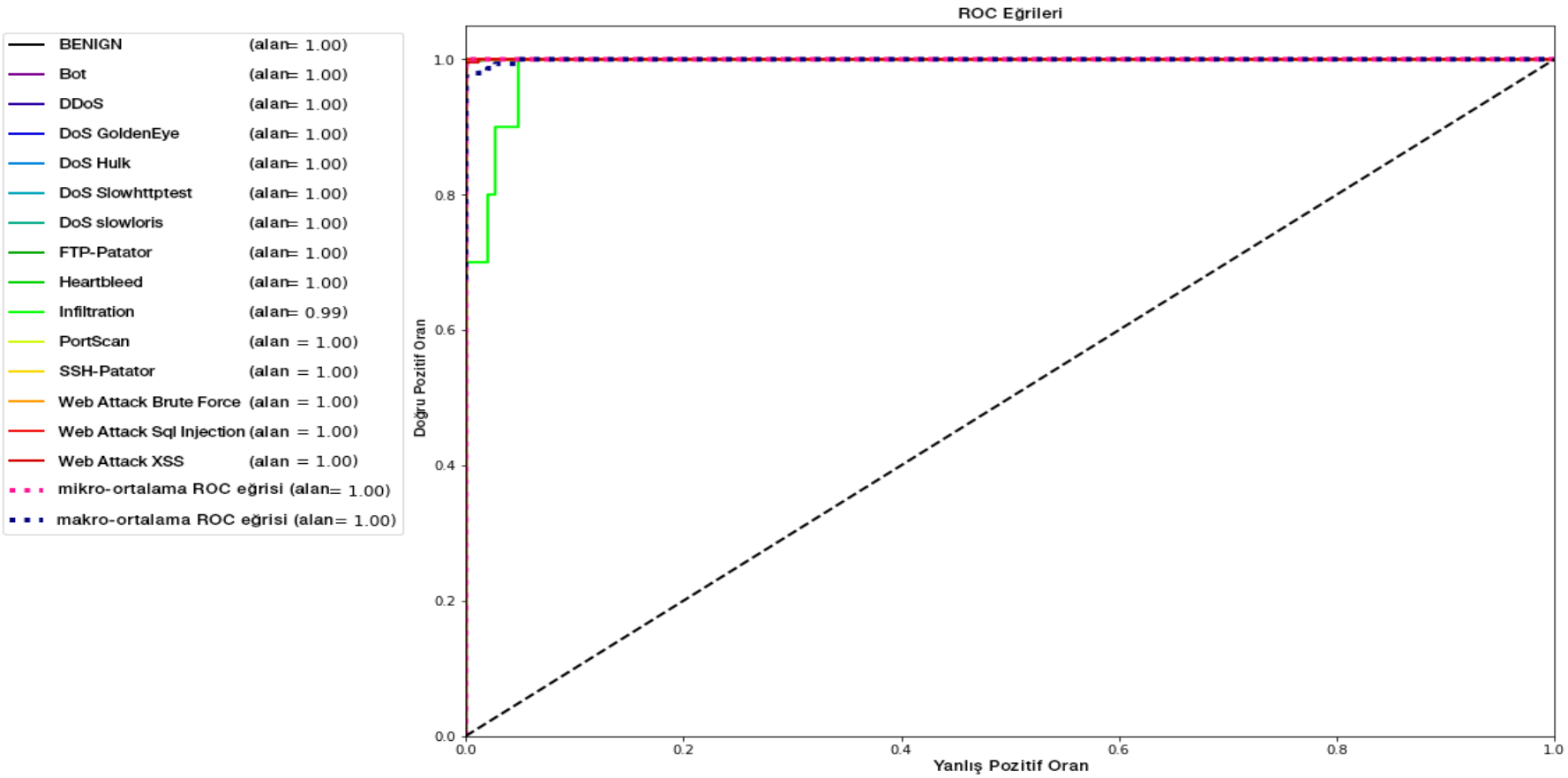
Şekil 4.38. XGBoost Algoritması için Karşılaştırma Matrisi

Bu modelde her bir etiket bilgisi için elde edilen ROC Eğrisi grafiği Şekil 4.39.'da gösterilmiştir. AUC değeri ise testler sonucunda 0.9989 olarak bulunmuştur. AUC değeri 1.00'a ne kadar yaklaşırsa modelin ezberleme oranı da daha düşük olmakta ve algoritmanın başarımı yükselmektedir.

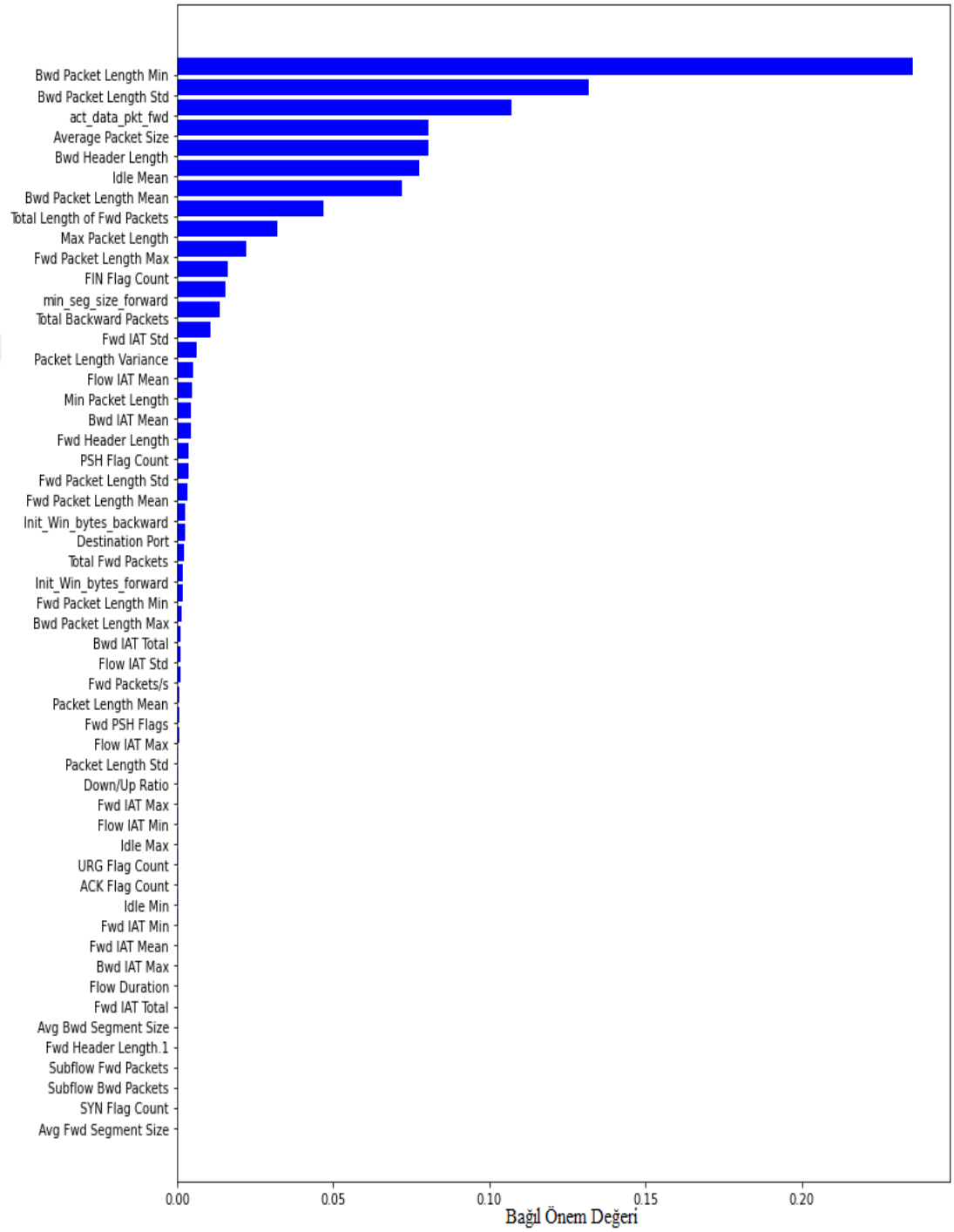
Şekil 4.39.'da ROC eğrisine ait grafik detaylı incelendiğinde ise etiketlenmiş saldırılar içerisinde yalnızca 'Infiltration' saldırısına ait ROC eğrisinin (0.99 değeri ile) AUC değeri altında kaldığı gözlenmiştir. Diğer bütün saldırı çeşitlerine ait ROC eğrileri AUC değerinin üzerinde bulunmaktadır. AUC değerinin altında kalan 'Infiltration' saldırısının sayısal yoğunluğu ile veri kümesinin tamamı orantılanacak olursa, 10/849.223 işlemi sonucu %0.001 ile çok düşük bir değer elde edilir. Bu değerin bu kadar düşük olmasının sebebi, bu saldırı çeşidine ait verilerin sayısının veri kümesinde çok az olmasıdır. Bu açıdan bakıldığında XGBoost algoritmasının değerlendirme ölçütleri bakımından tutarlılığı ve başarımının oldukça yüksek olduğu gözlenmiştir.

Toplam 53 adet özellik içerisinde XGBoost algoritmasıyla oluşturulan bu modelde 'Bwd Packet Length Min' isimli özellik 0.2354 ile en yüksek değere sahip olmuştur. CICIDS2017 veri kümesinde yapılan sınıflandırma işleminde hesaplamalara etki eden özniteliklerin aldıkları önem değerleri Şekil 4.40.'da gösterilmiştir.

CICIDS2017 veri kümesi için sonuçlar incelendiğinde test veri kümesinin doğruluk değeri 0.9989 olarak tespit edilmiştir. Doğruluk değeri açısından diğer tüm yöntemlerle kıyaslandığında en iyi sonucun elde edildiği algoritma olduğu belirlenmiştir.



Şekil 4.39. XGBoost Algoritması için ROC Eğrisi



Şekil 4.40. XGBoost Algoritması için Özniteliklerin Önem Değeri

5. BULGULAR

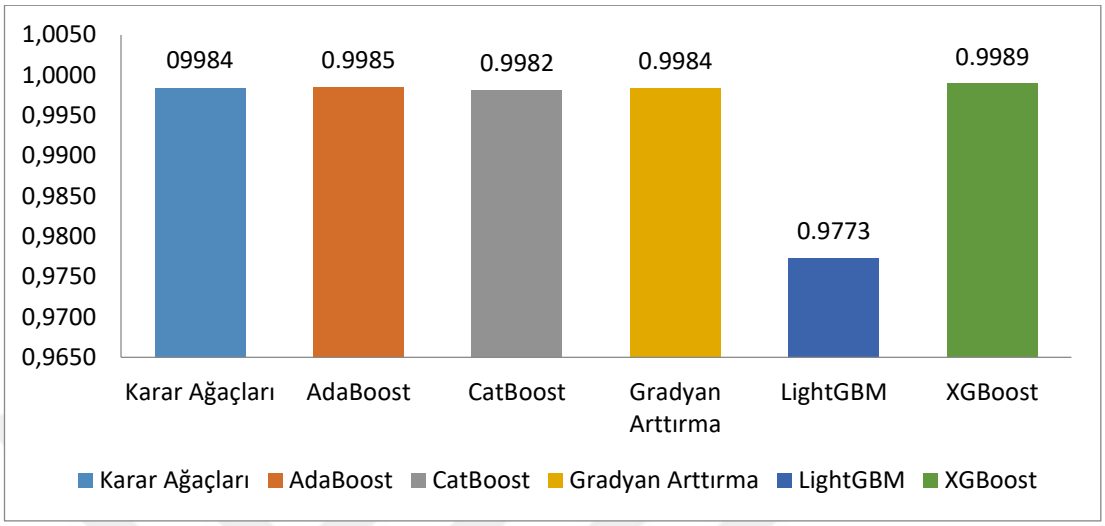
Bu bölümde, eğitim sonrası sınıflandırma yöntemlerinin performansları, test veri kümesi üzerinde doğruluk, kesinlik, duyarlılık, F1 değeri, Kappa değeri ve ROC eğrisi gibi performans değerlendirme ölçütleri kullanılarak değerlendirilmiştir. Algoritmalar, F1 değeri, Kappa ve AUC değerleri üzerinden karşılaştırılırken ayrıca algoritmaların öğrenme süreleri de değerlendirme ölçütü olarak kullanılmıştır. Performans değerlendirme ölçütlerine ait değerler Çizelge 5.1.'de gösterilmiştir.

Çizelge 5.1. Performans Değerlendirme Ölçütleri Karşılaştırması

	Performans Değerlendirme Ölçütleri						
Sınıflandırıcı	**Doğruluk**	**Kesinlik**	**Duyarlılık**	**F1 Değeri**	**Kappa Değeri**	**ROC AUC**	**Eğitim süresi (Dakika)**
Karar Ağaçları	0.9984	0.9984	0.9984	0.9984	0.9955	0.9983	1.5
AdaBoost	0.9985	0.9985	0.9985	0.9985	0.9957	0.9985	249
CatBooost	0.9982	0.9982	0.9982	0.9982	0.9947	0.9979	19
Gradyan Arttırma	0.9984	0.9984	0.9984	0.9984	0.9953	0.9983	378
LightGBM	0.9773	0.9773	0.9773	0.9773	0.9339	0.9445	4.5
XGBoost	**0.9989**	**0.9989**	**0.9989**	**0.9989**	**0.9969**	**0.9989**	210

Çizelge 5.1.'de görüldüğü üzere; CICIDS2017 veri kümesinin Karar Ağaçları, AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost algoritmalarının F1değeri oranları sırasıyla %99.84, %99.85, %99.82, %99.84, %97.73 ve %99.89'dur. Bu sonuçlara göre, en iyi değere %99.89 ile XGBoost algoritmasının sahip olduğu görülmektedir.

F1 değerine göre algoritmaların elde ettikleri sonuçlar grafiksel olarak Şekil 5.1.'de gösterilmiştir.

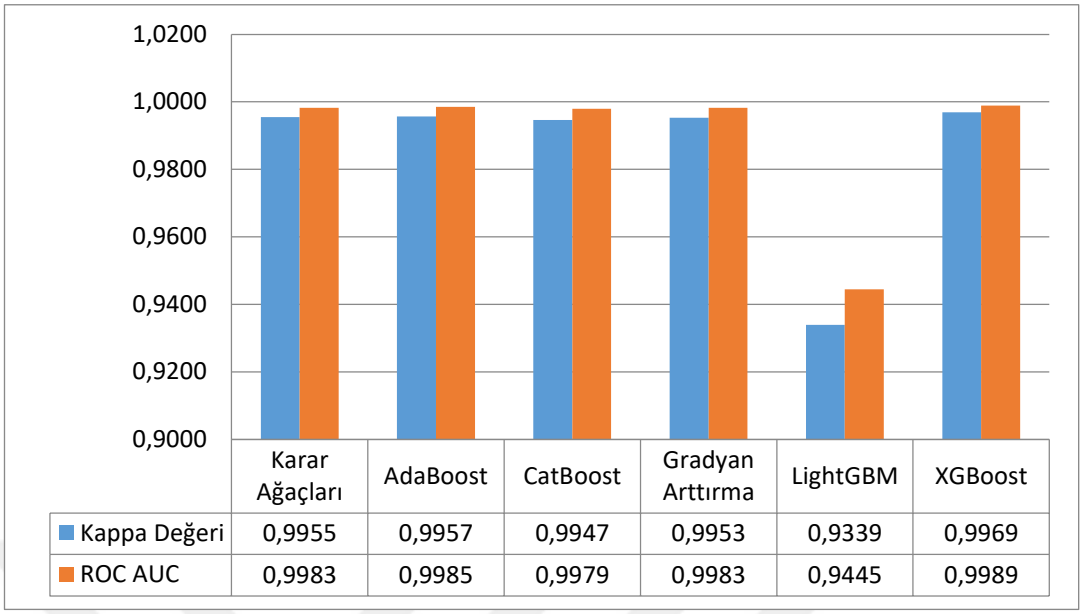


Şekil 5.1. Algoritmaların F1 Değeri Karşılaştırılması

Karar Ağaçları, AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost algoritmalarının performansları, bir diğer sınıflandırma performans ölçümü olan ROC eğrisi ve AUC değerleri ile de değerlendirilmiştir. Bölüm 4'de yer alan, her algoritmanın ROC eğrisi sonucunu gösteren grafikler incelendiğinde, algoritmaların ROC eğrilerinin ideale yakın olduğu ve AUC değerlerinin de 1'e çok yakın olduğu görülmektedir. AUC değerleri kapsamında karşılaştırıldıklarında en iyi sonucu 0.9989 ile XGBoost algoritmasının verdiği tespit edilmiştir.

Çizelge 5.1.'de belirtilen Kappa değerleri de incelendiğinde yine en yüksek değeri 0.9969 ile XGBoost algoritması almıştır.

Şekil 5.2.'de algoritmaların Kappa değerleri ile AUC değerleri birlikte gösterilmiştir. Şekil 5.2. incelendiğinde hem Kappa değeri hem de AUC değeri bakımından en yüksek değerlere XGBoost algoritmasının ulaştığı görülmektedir. Özellikle AUC değerinin bu denli yüksek olması ve F1 değeriyle yakın olması bu algoritmanın verdiği sonuçların yüksek başarıma sahip olduğunu göstermektedir.

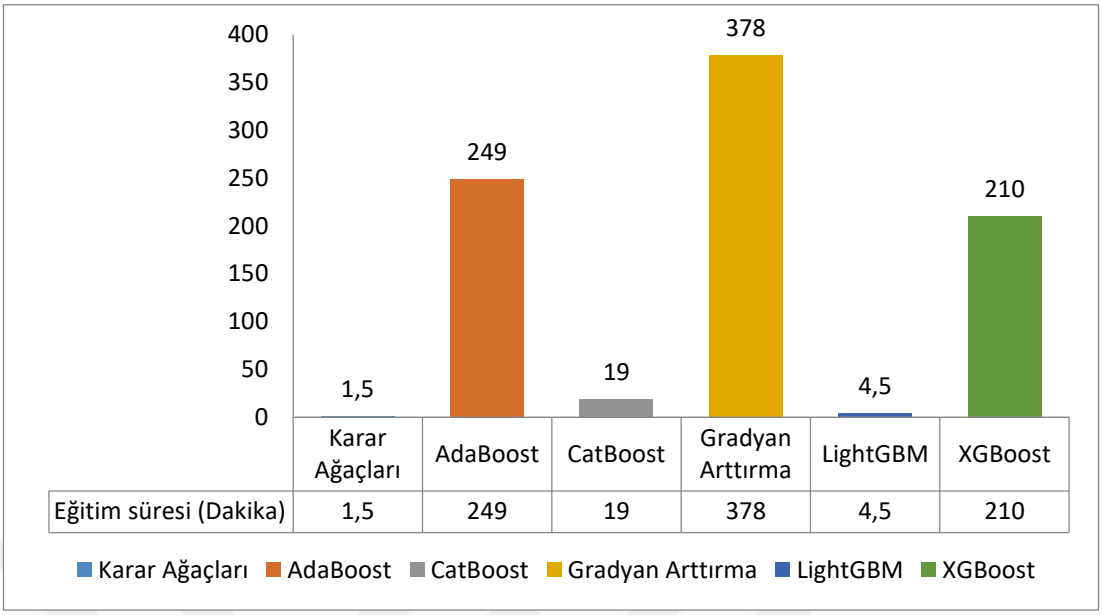


	Karar Ağaçları	AdaBoost	CatBoost	Gradyan Arttırma	LightGBM	XGBoost
Kappa Değeri	0,9955	0,9957	0,9947	0,9953	0,9339	0,9969
ROC AUC	0,9983	0,9985	0,9979	0,9983	0,9445	0,9989

Şekil 5.2. Algoritmaların Kappa ve AUC Değerlerinin Karşılaştırılması

Yapılan sınıflandırma işlemleri sonucunda modellerden LightGBM haricinde diğerlerinin kesinlik oranlarının 0.99'un üzerinde olması, doğru tahmin edilen pozitif gözlem sayısının toplam tahmin edilen pozitif gözlem sayısına oranının 1'e çok yakın olduğunu göstermektedir. Diğer adı duyarlılık olan ve doğru tahmin edilen pozitif gözlem sayısının gerçek sınıftaki tüm gözlem sayısına oranını veren duyarlılık değerlerinin de bu 5 sınıflandırıcı için 1'e çok yakın olduğu tespit edilmiştir. Hem YP hem de YN'leri dikkate alan ve kesinlik ile duyarlılık değerlerinin ağırlıklı ortalaması alınarak hesaplanan F1 değeri de yine aynı 5 sınıflandırıcı için 1'e oldukça yakındır. Yalnızca bir sınıflandırıcı 0.98'in de altında sonuç vermiştir. Hesaplamalarda 0.9773 değerini veren LightGBM algoritması eğitim süresi bakımından incelendiğinde 4.5 dakikalık bir süreyle karar ağaçları algoritmasının ardından en hızlı ikinci algoritma olarak gözlenmiştir.

Algoritmaların eğitim veri kümesi üzerinde öğrenme işlemleri süreleri bakımından karşılaştırıldıkları grafik Şekil 5.3.'de verilmiştir.



Şekil 5.3. Algoritmaların Öğrenme Sürelerinin Karşılaştırılması

Tez çalışmasında kullanılan sınıflandırmaların DP, DN, YN sayısal değerlerinin toplam test veri kümesi satır sayısı ile karşılaştırıldıkları matris Çizelge 5.2.'de gösterilmektedir. Çizelge 5.2.'de gözüktüğü üzere 906 satır sayısı ile en az YP ve YN değerlerine XGBoost algoritmasının sahip olduğu görülmektedir. Buna ek olarak LightGBM algoritması ise diğer algoritmalardan çok daha fazla sayıda YP ve YN değerlerine sahiptir. Bu bilgiler doğrultusunda XGBoost algoritmasının diğer algoritmalara göre daha iyi başarım sağladığı görülmüştür.

Çizelge 5.2. Algoritmaların DP, DN, YN Değerlerinin Karşılaştırılması

Algoritmaların DP, DN, YN Değerleri				
Sınıflandırıcı	**DP**	**YP**	**YN**	**Test Veri Kümesi**
Karar Ağaçları	847900	1323	1323	849223
AdaBoost	847968	1255	1255	849223
CatBoost	847672	1551	1551	849223
Gradyan Arttırma	847857	1366	1366	849223
LightGBM	829985	19238	19238	849223
XGBoost	**848317**	**906**	**906**	849223

6. TARTIŞMA VE SONUÇ

Bu tez çalışması kapsamında, günümüzde en çok karşılaşılan siber saldırı yöntemlerinden biri olan zararlı ağ trafiğinin tespit edilmesi için makine öğrenmesi algoritmalarından Karar Ağaçları, AdaBoost, CatBoost, Gradyan Arttırma, LightGBM ve XGBoost algoritmaları kullanılmıştır. Veri kümesi olarak da bu alandaki en güncel ve en kapsamlı veri kümelerinden biri olan CICIDS2017 kullanılmıştır. Eğitilen ve test edilen sınıflandırıcıların performansları doğruluk, kesinlik, duyarlılık, F1 değeri, Kappa değeri ve ROC eğrisi gibi performans değerlendirme ölçütleri ile incelenmiştir. Bunlar haricinde başka bir değerlendirme ölçütü olarak da algoritmaların öğrenme sürelerine de bakılmıştır.

Bu ölçümler neticesinde, karar ağacı ve diğer topluluk öğrenme yöntemlerine oranla XGBoost algoritmasının sınıflandırma performansının çok az bir farkla da olsa daha iyi olduğu gözlemlenmiştir.

Çalışma sonucunda elde edilen bir başka sonuç da, tespit edilen saldırı çeşitlerinin tespit edilme oranları karşılaştırma matrisi ile karşılaştırıldığında birbirleri arasında farklılıklar olduğu gözükmektedir. Ancak, bütün algoritmalarda ‘Infiltration’ ve ‘Web Attack SQL Injection’ isimli etiketlenmiş 2 saldırı çeşidi tespit edilirken algoritmalar tarafından zorlandığı görülmektedir. ‘Infiltration’ saldırı çeşidi kötü amaçlı dosya gönderimiyle gerçekleştirildiği için ağ trafiğinde gerçekleşen normal dosya gönderimi gibi algılanabilmekte ve bu yüzden algoritmalar tarafından ‘BENIGN’ olarak tanımlanabilmektedir. Bir diğer saldırı çeşidi olan ‘Web Attack SQL Injection’ saldırısında ise algoritmalar sınıflandırma yaparken diğer Web Attack ismini taşıyan saldırılarla karıştırmışlardır. ‘Infiltration’ ve ‘Web Attack SQL Injection’ saldırılarının sınıflandırılmasında yapılan hataların başlıca sebebi, sayıca az oldukları için tanımlanabilir. Tüm etiketlenmiş saldırıların sayıları ve açıklamaları

bulunan Çizelge 4.4. detaylı incelendiğinde bu 2 saldırı çeşidinin tüm veri kümesinde kapladığı alanın %0.002 olduğu görülmektedir. Algoritmaların başarım performansları karşılaştırıldığında XGBoost algoritması 0.9989 oranı ile en yüksek kesinlik değerine sahiptir.

Önerilen bu çalışmayla, gerçek dünyadaki çalışan sistemlerde ağ saldırılarının tespit edilmesi ve gerekli tedbirlerin alınması mümkündür. Önemli olan bir diğer konu da saldırı çeşitlerinin belirlenmesidir. Günümüzde teknolojinin hızla gelişmekte olduğu düşünülerek saldırı çeşitlerinin gittikçe artacağını ve tespit edilmelerinin zorlaşacağı düşünülmektedir.

Sunulan çalışmanın ilerleyen aşamalarında, CPU ve GPU kapasiteleri gün geçtikçe gelişen yüksek işlem gücüne sahip bilgisayarlar kullanılarak, makine öğrenmesi algoritmalarının yanı sıra derin öğrenme algoritmaları da kullanarak zararlı ağ trafiğinin incelenmesi ve saldırı çeşitlerine göre detaylı olarak analiz edilmesi amaçlanmaktadır.

KAYNAKLAR

Abdulkareem M.I. Comparative Study of Intrusion Detection System Using Machine Learning. - İstanbul : Altınbaş University, 2019. - Cilt Master of Science.

Abdulrahman A.A. ve Ibrahem M.K. Evaluation of DDoS Attacks Detection in a CICIDS2017 Dataset Based on Classification Algorithms [Dergi]. - Iraq : Iraqi Journal of Information and Communications Technology(IJICT), 2018. - 49-55 : Cilt Vol. 1, Issue 3.

Akçay A. Bilgi ve Belge Yönetiminde Veri Madenciliği. - İstanbul : İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, 2014. - Cilt Yüksek Lisans Tezi.

Aksu D. ve Aydin M.A. Detecting Port Scan Attempts with Comparative Analysis of Deep Learning and Support Vector Machine Algorithms [Konferans] // International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT). - Ankara : DOI: 10.1109/IBIGDELFT.2018.8625370, 2018. - Cilt 77-80. - s. 211-223.

Atalay N.S. [ve diğerleri] Adli Bilişim Alanında Ağ Analizi [Dergi]. - Elazığ : BEÜ Fen Bilimleri Dergisi, 2019. - 582-594 : Cilt 8 (2).

Atay R., Odabaş D.E. ve Pehlivanoğlu M.K. İki Seviyeli Hibrit Makine Öğrenmesi Yöntemi ile Saldırı Tespiti [Dergi] // Journal of Ambient Intelligence and Humanized Computing. - [basım yeri bilinmiyor] : Gazi Mühendislik Bilimleri Dergisi, 2019. - 258-272 : Cilt 5(3). - s. 1-8.

Aydın C. Makine Öğrenmesi Algoritmaları Kullanılarak İtfaiye İstasyonu İhtiyacının Sınıflandırılması [Dergi] // IEEE Wireless Communications . - İzmir : Avrupa Bilim ve Teknoloji Dergisi, 2018. - 14 : Cilt 169-175. - s. 6-28.

Baykara M. Bilişim Sistemleri için Saldırı Tespit ve Engelleme Yaklaşımlarının Tasarımı ve Gerçekleştirilmesi. - Elazığ : Fırat Üniversitesi, Fen Bilimleri Enstitüsü, 2016. - Cilt Doktora Tezi.

Budak İ., Şen B. ve Yıldırım M.Z. Lojistik Regresyon ile Bilgisayar Ağlarında Anomali Tespiti [Konferans] // Akademik Bilişim 2013 – XV. Akademik Bilişim Konferansı Bildirileri. - Antalya : Akdeniz Üniversitesi, 2013. - Cilt 813-818. - s. 1-28.

Canbek G. ve Sağıroğlu Ş. Bilgi, Bilgi Güvenliği ve Süreçleri Üzerine Bir İnceleme [Dergi]. - Ankara : Politeknik Dergisi, 2006. - 3, s.165-174 : Cilt 9.

Çalışır S. [ve diğerleri] Makine Öğrenmesi ve Derin Öğrenme Teknikleri ile Saldırı Tespiti [Dergi] // Applied Soft Computing Journal. - [basım yeri bilinmiyor] : (UBMK'19) 4rd International Conference on Computer Science and Engineering, 2019. - 4 : Cilt 978-1-7281-3964-7/19/$31.00 2019 IEEE. - s. 656-660.

Ekiz E. Makine Öğrenmesi Teknikleri ile Tahsilat Davranışı Tahmini: Telekomünikasyon Sektörü Örneği. - İstanbul : İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2019. - Cilt Yüksek Lisans Tezi.

Faker O. Intrusion Detection Using Big Data and Deep Learning Techniques. - Ankara : Çankaya University, 2019. - Cilt Master of Science.

Göker H. Makine Öğrenmesi Teknikleri Kullanılarak Çocukluk Çağı Dikkat Eksikliği ve Hiperaktivite Bozukluğunun Tahmin Edilmesine Yönelik Uzman Sistem Tasarımı. - Ankara : Gazi Üniversitesi, Bilişim Enstitüsü, 2019. - Cilt Doktora Tezi.

Gülaçar H. Nesnelerin İnterneti Platformları İçin Makine Öğrenmesi Tabanlı Bir Tahmin Modülü / dü. The 5th International Symposium on Emerging Inter-

networks Communication and Mobility. - İstanbul : İstanbul Tekni ; İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2018. - Cilt Yüksek Lisans Tezi.

Gülmez H.G. A Deep Reinforcement Learning Approach To Network Intrusion Detection. - Ankara : Computer Engineering Department, Middle East Technical University, 2019. - Cilt Master of Science.

Gümüştaş E. Kayıp Gözlem İçeren Dengesiz Veri Setlerinin Topluluk Öğrenme Algoritmaları İle Sınıflandırılması. - İstanbul : Mimar Sinan Güzel Sanatlar Üniversitesi, 2019. - Cilt Yüksek Lisans Tezi.

Güven E.Y. Kenar Bilişim İçin Siber Saldırıları Tespit Ve Önleme Yöntemleri. - İstanbul : Springer ; Fatih Sultan Mehmet Vakıf Üniversitesi, Mühendislik ve Fen Bilimleri Enstitüsü, 2018. - Cilt Yüksek Lisans Tezi.

Hosseini S. ve Seilani H. Anomaly Process Detection Using Negative Selection Algorithm and Classification Techniques [Dergi]. - Kerman, Iran : Evolving Systems, 2019. - Cilt https://doi.org/10.1007/s12530-019-09317-1.

Hossen S. ve Janagam A. Analysis of Network Intrusion Detection System with Machine Learning Algorithms (Deep Reinforcement Learning Algorithm). - Karlskrona, Sweden : Faculty of Computing, Blekinge Institute of Technology, 2018. - Cilt Master of Science.

Kaya K. Veri Madenciliği Yöntemleri Kullanarak Hava Kirliliği Tahmini. - İstanbul : IEEE ; İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2019. - Cilt Yüksek Lisans Tezi.

McKay R. [ve diğerleri] Machine Learning Algorithms on Botnet Traffic: Ensemble and Simple Algorithms [Konferans] // The International Conference on Compute and Data Analysis 2019 (ICCDA). - Kahului, HI, USA : https://doi.org/10.1145/3314545.3314569, 2019. - Cilt 31-35. - s. 90-98.

Özekes S. ve Karakoç E.N. Makine Öğrenmesi Yöntemleriyle Anormal Ağ Trafiğinin Tespit Edilmesi [Dergi] // International Journal of Distributed. - Düzce : Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 2019. - 7 : Cilt 566-576.

Sharafaldin I., Lashkari A.H. ve Ghorbani A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization [Konferans] // 4th International Conference on Information Systems Security and Privacy. - [basım yeri bilinmiyor] : DOI: 10.5220/0006639801080116, 2018. - Cilt 108-116.

Tama B.A. [ve diğerleri] An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble [Dergi]. - South Korea : IEEE Access Special Section on Machine Learning Designs, Implementations and Techniques, 2020. - 24120-24134 : Cilt Volume 8.

Tomak L. ve Bek Y. İşlem karakteristik eğrisi analizi ve eğri altında kalan alanların karşılaştırılması [Dergi]. - Samsun : Journal of Experimental and Clinical Medicine - Deneysel ve Klinik Tıp Dergisi, 2009. - 58-65 : Cilt 27.

Tozlu İ. Simplifying Balance Sheet Adjustment Process in Commercial Loan Applications Using Machine Learning Methods. - İstanbul : İstanbul Teknik Üniversitesi, 2019. - Cilt Yüksek Lisans Tezi.

Turgut S. Makine Öğrenmesi Yöntemleri Kullanarak Kanser Teşhisi. - İstanbul : İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, 2017. - Cilt Yüksek Lisans Tezi.

Türkmenoğlu C. Türkçe Metinlerde Duygu Analizi. - İstanbul : İstanbul Teknik Üniversitesi, 2015. - Cilt Yüksek Lisans Tezi.

Wankhede S. ve Kshirsagar D. DoS Attack Detection using Machine Learning and Neural Network [Konferans] // Fourth International Conference on Computing

Communication Control and Automation (ICCUBEA). - Pune, India : DOI: 10.1109/ICCUBEA.2018.8697702, 2018. - Cilt 1-5. - s. 926 – 931 .

Yangın G. Xgboost ve Karar Ağacı Tabanlı Algoritmaların Diyabet Veri Setleri Üzerine Uygulaması. - İstanbul : Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Bilimleri Enstitüsü, 2019. - Cilt Yüksek Lisans Tezi.

Yiğidim H.A. Makine Öğrenme Algoritmalarını Kullanarak Ağ Trafiğinin Sınıflandırılması / dü. Sciences Proceedings of the 33rd Hawaii International Conference on System. - Ankara : TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, 2012. - Cilt Yüksek Lisans Tezi.

Yıldırım M.Z. Makine Öğrenmesi Yöntemleri ile Network Üzerinde Saldırı Tespiti. - Karabük : Karabük Üniversitesi, Fen Bilimleri Enstitüsü, 2017. - Cilt Yüksek Lisans Tezi.

Yılmaz S. Topluluk Öğrenme Yöntemini Kullanarak Twitter Verisi Üzerinde Duygu Algılama ve Tanıma. - İzmir : Computers and Electrical Engineering ; Ege Üniversitesi, Fen Bilimleri Enstitüsü, 2019. - Cilt Yüksek Lisans Tezi.

Yulianto A., Sukarno P. ve Suwastika N.A. Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset [Konferans] // Journal of Physics: Conference Series, 1192. - [basım yeri bilinmiyor] : DOI:10.1088/1742-6596/1192/1/012018, 2018. - s. 206-213.

EKLER

Ek A. Veri Önişleme

```
df1=pd.read_csv("/content/drive/My Drive/koray_tez/Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv", low_memory=False,  sep=',',
error_bad_lines=False,  index_col=False, dtype='unicode')
.
.
.
df8=pd.read_csv("/content/drive/My Drive/koray_tez/Wednesday-workingHours.pcap_ISCX.csv", low_memory=False, sep=',', error_bad_lines=False, index_col=False, dtype='unicode')

df=pd.concat([df1,df2,df3,df4,df5,df6,df7,df8])
df.replace(' ',0.0, inplace=True)
df.replace('inf', 0.0, inplace=True)
df.replace('NaN',0.0, inplace=True)
labels = df[' Label'].astype('category').cat.categories.tolist()
replace_labels = {' Label' : {k: v for k,v in zip(labels,list(range(0,len(labels)+1)))}}
filename = '/content/drive/My Drive/koray_tez/df_zip.bz2'
sfile = bz2.BZ2File(filename, 'w')
pickle.dump(norm_X_train, sfile)
sfile.close()
```

Ek B. Veri Normalizasyon

```
cols = list(df_pkl.columns)
for col in cols:
    norm_df_pkl[col] =stats.zscore(df_pkl[col])
norm_df_pkl[' Label'].value_counts().plot(kind= 'pie', startangle=30, fontsize = 18,
rotatelabels=90, labeldistance=1.05,figsize=(20,20), autopct='%.1f%%', labels =
['BENIGN', 'Bot', 'DDoS', 'DoS GoldenEye', 'DoS Hulk', 'DoS Slowhttptest',
'DoS slowloris', 'FTP-Patator', 'Heartbleed', 'Infiltration', 'PortScan', 'SSH-Patator',
'Web Attack Brute Force', 'Web Attack Sql Injection', 'Web Attack XSS'])
plt.tight_layout()
plt.legend(['BENIGN','Bot','DDoS','DoS GoldenEye','DoS Hulk','DoS Slowhttptest',
'DoS slowloris', 'FTP-Patator', 'Heartbleed', 'Infiltration', 'PortScan', 'SSH-Patator',
'Web Attack Brute Force', 'Web Attack Sql Injection', 'Web Attack XSS'],
          fontsize = 12, labelspacing=2, loc = 'upper left');
filename = '/content/drive/My Drive/koray_tez/norm_df-zip.bz2'
sfile = bz2.BZ2File(filename, 'w')
pickle.dump(norm_df_pkl, sfile)
sfile.close()
```

SECTION 5: Calculated ENTROPY, INFORMATION GAIN

```
ent=pd.DataFrame(drv.entropy(X_train))
print(" time for " ," entropy:: ", (t2-t1)/60 , " minutes")
cols = list(X_train.columns)
for col in cols:
    print(col,'=', drv.entropy(X_train[col],fill_value=np.nan))
    print(col,'(Information Gain)','=',ent[0] -
 (drv.entropy(X_train[col],fill_value=np.nan)))
print(" time for " ," entropy and InformationGain:: ", (t2-t1)/60 , " minutes")
pca = PCA(n_components=1)
fit = pca.fit(X_train)
print("Explained Variance: ", fit.explained_variance_ratio_)
print(fit.components_)
print(); print(" time for " ," entropy and InformationGain:: ", (t2-t1)/60 , " minutes")
```

SECTION 6: FEATURE SELECTION

```
model = ExtraTreesClassifier(criterion='entropy', bootstrap=True, n_jobs=-1)
model.fit(X_train, y_train)
print(); print(" time for " ," entropy and InformationGain:: ", (t2-t1)/60 , " minutes")
model.score(X_train, y_train)
feature_selection= pd.DataFrame(model.feature_importances_)
feature_selection=feature_selection.T
X = X_train.drop(columns=['Bwd PSH Flags','Bwd URG Flags',
'Fwd Avg Bytes/Bulk','Fwd Avg Packets/Bulk','Fwd Avg Bulk Rate',
'Bwd Avg Bytes/Bulk','Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate',
'RST Flag Count','ECE Flag Count','CWE Flag Count',
'Fwd URG Flags','Flow Bytes/s','Flow Packets/s','Active Std',
'Active Mean','Active Min','Bwd IAT Min','Idle Std','Active Max',
'Total Length of Bwd Packets','Subflow Bwd Bytes',
'Subflow Fwd Bytes','Bwd IAT Std','Bwd Packets/s'], axis=1, inplace=True)
sns.heatmap(X_train.corr(), vmin=0, vmax=1, fmt='.1g',annot=False, square=True,
robust=True)
```

Ek Ç. Veri Kümesi Ayrıştırma

SECTION 4: Splitted Data as X_train, y_train, X_test, y_test

```
X_train,X_test=train_test_split(norm_df_pkl,test_size=0.3,random_state=0)
y_train=X_train[X_train.columns[-1]]
X_train=X_train[X_train.columns[0:-1]]
y_test=X_test[X_test.columns[-1]]
X_test=X_test[X_test.columns[0:-1]]
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Ad Soyad	:Koray ÇOŞKUN
Uyruk	: T.C.
Doğum Yeri ve Tarihi	: İzmir - 1984
Medeni Hali	: Evli
Telefon	:+90 555 835 02 22
E-posta	: koraycoskun@msn.com

Eğitim

Alınan Derece	Aldığı Kurum/Üniversite	Mezuniyet Yılı
Lise	Mithatpaşa Endüstri Meslek Lisesi	2001
Önlisans	Dokuz Eylül Üniversitesi – Bilgisayar Programlama	2004
Lisans	Muğla Sıtkı Koçman Üniversitesi – Elektronik ve Bilgisayar Eğitimi	2011

Yabancı Dil

Dil (İngilizce)	Başlangıç	Orta	İleri
Okuma/Yazma		X	
Konuşma/Anlama		X	

İş Tecrübesi

Yıl	Yer	Görev
2019	HAVATEK A.Ş. (İzmir)	Doküman Kontrol Sorumlusu
2015-2019	STAR Rafineri A.Ş. (İzmir)	Doküman Kontrol Yazılım Uzmanı
2012-2016	Yaşar Üniversitesi (İzmir)	Öğretim Görevlisi
2014-2015	Dokuz Eylül Üniversitesi (İzmir)	Öğretim Görevlisi
2012-2013	Bilgitürk Bilgisayar Teknolojileri A.Ş. (İzmir)	Sistem Destek Uzmanı
2006-2007	Altek Bilgi Teknolojileri Ltd.Şti. (İzmir)	Bilgi Teknolojileri Personeli
2001-2006	Datateknik Bilgisayar A.Ş. (İzmir)	Teknik Servis Personeli