
GAZİ ÜNİVERSİTESİ
Gazi
1926

OY BİRLİĞİ VE ÖZELLEŞMİŞ SINIFLANDIRICILAR İLE ZARARLI YAZILIM TESPİTİ

Sercan GÜLBURUN

YÜKSEK LİSANS TEZİ
BİLGİ GÜVENLİĞİ MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KASIM 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza

Sercan GÜLBURUN

31/10/2022

OY BİRLİĞİ VE ÖZELLEŞMİŞ SINIFLANDIRICILAR İLE ZARARLI YAZILIM TESPİTİ

(Yüksek Lisans Tezi)

Sercan GÜLBURUN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Kasım 2022

ÖZET

Dijital dünyaya yönelik tehditlerin en yaygın olarak karşılaşılan çeşitlerinden bir tanesi zararlı yazılımlardır. Bu tür yazılımlar, saldırganların kötücül amaçlarını gerçekleştirmek için kullandığı kodlardır. Mevcut ve yeni zararlı yazılımların bilgi varlıklarına zarar vermeden tespit edilmesi ve engellenmesi büyük önem arz etmektedir. Zararlı yazılımların tespit edilmesi için makine öğrenmesi yaklaşımları etkin bir şekilde kullanılmaktadır. Bu tez çalışmasında, denetimli ve denetimsiz öğrenme algoritmalarının birlikte kullanıldığı bir model sunulmaktadır. Sunulan model yüksek doğruluk ve f1 skoruyla mümkün olan en kısa sürede tahmin gerçekleştirmektedir. Modelin ilk aşamasında veriler K-ortalamalar algoritmasıyla kümelenmektedir. İkinci aşamasında ise ilgili küme için en iyi tahmin performansına sahip sınıflandırıcı kombinasyonu ile tahmin gerçekleştirilmektedir. İkinci aşamada kümelere göre en iyi sınıflandırıcılar seçilirken on makine öğrenme algoritmasının (Kernel Destek Vektör Makinesi, K-En Yakın Komşu, Naïve Bayes, Karar Ağacı, Rastgele Orman, Ekstra Gradian Yükseltme, Kategorik Yükseltme, Adaptif Yükseltme, Ekstra Ağaçlar ve Gradyan Yükseltme) üçlü kombinasyonu alınmaktadır. Seçilen üçlü sınıflandırıcı kombinasyonu iki kademede konumlandırılmaktadır. Tahmin süresi en yüksek olan sınıflandırıcının ikinci kademede olacak şekilde konumlandırılması sayesinde modelin tahmin süresi iyileştirilmektedir. Modelin tahmin performansı, BODMAS veri seti, Kaggle Zararlı Yazılım Tespiti veri seti, EMBER 2018 veri seti ve özgün veri setiyle, doğruluk ve f1 skoru değerleri kullanılarak sunulmaktadır. Model BODMAS veri setinde %99,74 doğruluk ve %99,77 f1 skoru, EMBER veri setinde %96,77 doğruluk ve %96,77 f1 skoru sağlamakta olup, aynı veri setleri kullanılarak gerçekleştirilen çalışmalardan daha iyi performans göstermektedir.

Bilim Kodu	:	92403
Anahtar Kelimeler	:	Zararlı yazılım tespiti, toplu öğrenme, sınıflandırma, kümeleme, özel sınıflandırıcı
Sayfa Adedi	:	127
Danışman	:	Doç. Dr. Murat DENER

MALWARE DETECTION USING CONSENSUS LEARNING AND SPECIALIZED CLASSIFIERS

(M. Sc. Thesis)

Sercan GÜLBURUN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

November 2022

ABSTRACT

One of the most common types of threats to the digital world is malicious software. This type of software is the code that attackers use to accomplish their malicious purposes. It is of great importance to detect and prevent existing and new malware without damaging information assets. Machine learning approaches are used effectively to detect malicious software. In the study, a model in which supervised and unsupervised learning algorithms used together is presented. The presented model makes predictions in the shortest possible time with high accuracy and f1 score. In the first stage of the model, the data are clustered with the K-means algorithm. In the second stage, the prediction is made with the combination of the classifier with the best prediction performance for the related cluster. In the second step, while choosing the best classifiers for the given clusters, a triple combination of ten machine learning algorithms (Kernel SVM, K-Nearest Neighbor, Naïve Bayes, Decision Tree, Random Forest, Extra Gradient Boosting, Categorical Boosting, Adaptive Boosting, Extra Trees, and Gradient Boosting) is used. The selected triple classifier combination is positioned in two stages. The prediction time of the model is improved by positioning the classifier with the highest prediction time to be in the second stage. The prediction performance of the model is presented using the BODMAS dataset, the Kaggle Malware Detection Dataset, EMBER 2018 dataset and the original dataset, and the accuracy and f1-score values are presented. The proposed methods has 99,74% accuracy and 99,77% f1 score for BODMAS dataset, and 96,77% accuracy and 96,77% f1 score for EMBER dataset. The proposed method's prediction performance is better than the rest of the studies in the literature in which BODMAS and EMBER datasets are used.

Science Code	:	92403
Key Words	:	Malware detection, ensemble learning, classification, clustering, specialized classifier
Page Number	:	127
Supervisor	:	Assoc. Prof. Dr. Murat DENER

TEŞEKKÜR

Çalışmalarım süresince tez danışmanlığımı üstlenerek bana yol gösteren, tecrübeleri ile beni yönlendiren, çalışmanın yürütülmesinde katkı ve destek sunan danışmanım Doç.Dr. Murat DENER'e, destekleriyle her zaman yanımda olan başta eşim Merve ALTIN GÜLBURUN olmak üzere değerli aileme ve arkadaşlarıma teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmaktadır.

Kısaltmalar	Açıklamalar
ADAB	Adaptive Boosting
API	Application Programming Interface
CATB	Categorical Boosting
CLS	Cluster (Küme)
CNN	Convolutional Neural Networks
DN	Doğru Negatif
DP	Doğru Pozitif
DT	Decision Tree
EXT	Extra Trees
GB	Gradient Boosting
KNN	K-Nearest Neighbour
KSVM	Kernel Support Vector Machines
LR	Logistic Regression
NB	Naive Bayes
P2P	Peer-to-Peer
PE	Portable Executable
RF	Random Forest
RFC	Request for Comments
RPC	Remote Procedure Call
TTL	Time-to-Live
XGB	Extreme Gradient Boosting
YN	Yanlış Negatif
YP	Yanlış Pozitif

1. GİRİŞ

Bilgi teknolojilerinin amacı insan hayatını kolaylaştırmaktır. Eğitimden sağlığa, güvenlikten tarıma, vatandaşlık hizmetlerinden iletişim ve haberleşmeye, ulusal altyapılardan kişisel kullanıma kadar, hemen hemen insan hayatını ilgilendiren her noktada bilgi teknolojilerinden faydalanılmaktadır. Bilgi teknolojileri ilk ortaya çıktığı dönemde az sayıda bulunan, zor erişilebilen ve ulusal nitelikteki kurum ve kuruluşların birtakım işlerinin daha etkin bir şekilde gerçekleştirilmesi için kullanılmaktaydı. Günümüzde ise bilgi teknoloji varlıklarının kullanımı her bir birey için kaçınılmaz ve zorunludur. Bilgi toplumunun her üyesi birer e-birey olup, e-vatandaş olarak e-devlet hizmetlerinden faydalanmakta ve e-toplumun bir parçasını teşkil etmektedir. Bireylerin yanı sıra kamu kurum ve kuruluşları ile birçok özel şirket etkin olarak dijital dünyada yer almaktadır. TUİK tarafından gerçekleştirilen Hanehalkı Bilişim Teknolojileri Kullanım Araştırması sonuçlarına göre, 2011 yılında %42 olan hane internet erişim oranının 2021'de %92'ye, aynı dönemde bireysel internet kullanım oranının ise %45'ten %82'ye çıktığı görülmektedir. Özel amaçla kamu kurum ya da kuruluşları ile iletişime geçmek veya kamu hizmetlerinden yararlanmak için 2020 yılı Nisan ayı ile 2021 yılı Mart ayını kapsayan on iki aylık dönemde İnterneti kullanan bireylerin oranı ise %58,9 olup ve bir yıl öncesine göre gerçekleşen artış oranı %7,4'tür [1].

Birey, kurum ve kuruluşların dijital ortamda var olmasına paralel olarak; dijital dünyada hem birey, kurum ve kuruluşlara hem de bunların dijital ortamda var olmasını sağlayan bilgi varlıklarının gizlilik, bütünlük ve erişilebilirliğine yönelik tehditler ortaya çıkmaktadır. İlk kez 1974 yılında, Rabbit virüsü ile ortaya çıkan zararlı yazılımlar, bugün hem ülkelerin ulusal güvenliğini hem de bireyleri ciddi tehditlere maruz bırakan bir noktadadır. Bu tehditler sadece dijital ortamları etkilememekte, siber fiziksel sistemleri etkileyen zararlılar nedeniyle gerçek dünyayı da ciddi bir şekilde etkilemektedir. Tespit edilen zararlı yazılımların yıllara sari istatistiği incelendiğinde, 2013 yılında tespit edilen zararlı sayısının 182 milyon olduğu, bu sayının 2021'e gelindiğinde 1 milyar 312 milyona çıktığı görülmektedir [2].

Bilgi varlıklarının güvenliğinin temellerini gizlilik, bütünlük ve erişilebilirlik oluşturmaktadır. Saldırganlar bilgi varlıklarının gizliliğini ortadan kaldırmaya, bütünlüğünü bozmaya ve erişilebilirliğini engellemeye yönelik saldırılarda bulunurlar. Zararlı yazılımlar,

saldırganların amaçlarını gerçekleştirmesi için kullandıkları kodlardır. Virüs, solucan, truva atı, fidye yazılım ve casus yazılım gibi çeşitli zararlı yazılım türleri bulunmaktadır. Zararlı yazılımların tespiti için imza tabanlı ve sezgisel tespit yaklaşımları kullanılmaktadır. İmza tabanlı yaklaşımlar, bilinen zararlı yazılımların kendilerine özgü davranışlarına dayanarak tespit yaparlar. Anomali tabanlı yaklaşımlar temel aldıkları normalin dışındaki davranışlardan faydalanarak tespit gerçekleştirirler. Bu yaklaşımlar ayrı ayrı kullanılabileceği gibi birlikte de kullanılabilmektedir. Zararlının incelenmesi statik ve dinamik olarak yapılmaktadır. Statik analizde, kod otomatik ya da manuel olarak kodun çalıştırılmadığı durumda incelenmektedir. Dinamik analizde ise kodun çalışma ortamındaki davranış ve özellikleri incelenmektedir.

Saldırganlar tarafından üretilen zararlılar sürekli olarak değişmekte ve gelişmektedir. Bu değişim ve gelişim sebebiyle imza tabanlı zararlı yazılım tespiti yaklaşımları özellikle yeni zararlıların tespit edilmesi kapsamında yetersiz kalmaktadır. İmza tabanlı zararlı yazılım tespiti yaklaşımının yetersiz kaldığı durumlarda zararlı yazılım tespitinin gerçekleştirilebilmesi için sezgisel yaklaşım kullanılır.

Zararlı yazılımların yaygınlaşması, siber ya da fiziksel dünya ayrımı olmadan günümüz insanlarını siber saldırganların bilinçli ya da rastgele hedefi haline getirmektedir. Bu durum, zararlı yazılımların kötücül eylemlerini gerçekleştirmeden tespit edilmesini ve durdurulmasını zorunlu kılmaktadır. Geçmişte kullanılan ve tespit edilen zararlı yazılımların tespiti, imza tabanlı tespit gerçekleştiren güncel bir anti virüs programı ile kolaylıkla sağlanmaktadır. Ancak ilk kez kullanılan ve tespit yaklaşımlarına karşı kaçınma teknikleri uygulayan zararlılara karşı imza tabanlı tespit yaklaşımları yetersiz kalmaktadır.

Her yıl milyonlarca yeni zararlı yazılımın ortaya çıkması, geleneksel zararlı yazılım tespit yaklaşımlarını yetersiz kılmaktadır. Makine öğrenmesi yöntemlerinin uygulanabilirliğinin artması ise zararlı yazılım tespiti kapsamında yeni yaklaşımların ortaya konulmasına imkân vermektedir. Çalışmanın amacı, geleneksel tespit yaklaşımlarıyla tespit edilemeyen zararlıların, kümeleme ve toplu öğrenme metotları kullanılarak, yüksek doğruluk ve hızla tespit edilmesini sağlamaktır.

Bu kapsamda, birincisi kümeleme ikincisi tahmin aşaması olan iki ana aşamalı bir yöntem sunulmaktadır. Tahmin aşaması da iki alt kademeye ayrılmıştır. Bu aşamaların ilkinde, bu

kademeye konumlandırılan iki sınıflandırıcının aynı tahmini yapması durumunda sınıflandırma gerçekleştirilir. İki sınıflandırıcının farklı tahmin yaptığı durumda ise ikinci kademedeki sınıflandırıcı sınıflandırma işlemini gerçekleştirmektedir.

Çalışmada kullanılan "özelleşmiş sınıflandırıcı" terimi, bir örneklemin, o örneklemin sınıflandırılmasına yönelik özel olarak eğitilmiş sınıflandırıcılar tarafından sınıflandırılması yaklaşımı ifade etmek için kullanılan bir terimdir. Çalışmada kullanılan "oy birliği" terimi, iki kademeli tahmin safhasının ilk kademesindeki sınıflandırıcıların aynı sınıflandırmayı yapmasını ifade etmektedir.

Çalışmanın özgünlüğü ve literatüre katkıları;

- Zararlı yazılım tespiti kapsamında denetimli ve denetimsiz öğrenme algoritmalarının birlikte kullanılması,
- Oluşturulan farklı alt kümelerde farklı sınıflandırıcı kombinasyonlarının kullanılması,
- Özelleşmiş sınıflandırıcı yaklaşımı ile veri setinin alt kümelerinde daha iyi tahmin performansı ortaya koyabilen alt sınıflandırıcıların eğitilmesi,
- Erken oybirliği yaklaşımı ile tahmin süresinin düşürülmesi,
- Sunulan tespit yaklaşımının geçerliliğinin farklı veri setleri üzerinde test edilerek ortaya konulmasıdır.

Çalışmanın ikinci bölümünde denetimli ve denetimsiz öğrenme algoritmalarının birlikte kullanılması, toplu öğrenmeyle zararlı yazılım tespiti ve çalışmada kullanılan veri setleri kapsamında literatürde yer alan çalışmalara, üçüncü bölümünde zararlı yazılımın tanımı, türleri, örnekleri tespit ve analiz yöntemleri ile zararlı yazılım tespitinden kaçınma yöntemlerine değinilmiştir. Dördüncü bölümde, çalışmada kullanılan makine öğrenmesi algoritmaları ve metrikler, beşinci bölümde kullanılan veri setleri, altıncı bölümde ise kullanılan araçlar anlatılmıştır. Yedinci bölümde önerilen zararlı yazılım tespit yaklaşımı, sekizinci bölümde ise bu yaklaşım kullanılarak elde edilen sonuçlar sunulmaktadır. Son bölümde, çalışmanın genel değerlendirmesi yapılmıştır.

2. İLGİLİ ÇALIŞMALAR

İlgili çalışmalar bölümü üç alt başlıkta incelenmiştir. Sırasıyla, toplu öğrenme ile zararlı yazılım tespiti kapsamında yapılan çalışmalar, kümeleme ve sınıflandırma algoritmalarının birlikte kullanıldığı çalışmalar, BODMAS ve EMBER 2018 veri setlerinin kullanıldığı çalışmalar sunulmaktadır.

2.1. Toplu Öğrenme ile Zararlı Yazılım Tespiti Kapsamında Yapılan Çalışmalar

Zararlı yazılımların hem çeşitliliğinin artması hem de karmaşıklaşması geleneksel tespit yaklaşımlarını yetersiz kılmaktadır. Bu sebeple zararlı yazılım tespiti için makine öğrenmesi yöntemleri sıklıkla kullanılmaktadır [3-5]. Teorik olarak uzun yıllar önce ortaya konan birçok makine öğrenme algoritması, son on yılda, hesaplama gücünün artmasına paralel olarak, birçok alanda uygulanmaktadır. Son dönemde makine öğrenmesi algoritmalarından faydalanılarak gerçekleştirilen zararlı yazılım tespit çalışmaların incelendiğinde, toplu öğrenme yönteminin yaygın olarak kullanıldığı görülmektedir.

Çalışmada [6], kötücül yazılımların görüntü tabanlı analizi ve tespitine yönelik ölçeklenebilir hibrit bir toplu öğrenme yöntemi sunulmuştur. Sınıflandırıcı olarak Random Forest ve Extra-Trees algoritmaları kullanılmıştır. Maligm ve Kaggle's Big 2015 Malware veri setleri kullanılarak eğitilen model ile %98,91 doğrulukla tahmin gerçekleştirilmiştir.

Çalışmada [7], Geliştirilmiş İki-gizli-katmanlı aşırı öğrenme makinesi (improved Two-hidden-layer Extreme Learning Machine - Improved TELM) yöntemi sunulmuştur. TELM metoduna zararlı yazılımların kod dizilişinden yararlanma ve öğrenme sürecinde geri yayılımdan kaçınma özelliği kazandırılmıştır. Sunulan metot ile herhangi bir veri ön işlemeye gerek kalmaksızın model eğitilebilmektedir. Farklı veri setleri ile yapılan testlerde, sunulan modelin fidye yazılımı veri setinde %98,96, Kaggle Big 2015 Malware veri setinde ise %97,29 doğrulukla tahmin gerçekleştirdiği görülmüştür.

Çalışmada [8], zararlı yazılım tespiti kapsamında, düşük boyutlu özelliklerin ve ağaç tabanlı çoklu öğrenme metotlarının karşılaştırmalı analizi yapılmıştır. Düşük boyutlu özellikler kapsamında 2-gram, 2-gramM, API-DLL API ve WEM kullanılmıştır.

Yapılan karşılaştırmalar neticesinde, düşük boyutlu özelliklerle zararlı yazılım tespiti kapsamında en iyi performansı WEM'in, sınıflandırma algoritmalarında ise XGB algoritmasının gösterdiği görülmüştür.

Zararlı JPEG dosyalarının tespitine özel olarak geliştirilen zararlı yazılım tespit modeli [9] iki aşamadan oluşmaktadır. İlk aşamada, MalJPEG özellik çıkarıcıyla özellikler çıkarılmış, ikinci aşamada ise LightGBM algoritması veri seti kullanılarak eğitilmiştir. Sunulan metot 0,951 hassassiyet ile tahmin gerçekleştirmiştir.

Çalışmada [10] davranış tabanlı bir zararlı yazılım tespit modeli sunulmuştur. Model; çalışma zamanı özelliklerinin yakalanması, özellik çıkarımı, özellik işleme ve sınıflandırıcı eğitimi aşamalarından oluşmaktadır. PSI, API çağrıları, dosya operasyonları, kayıt defteri değişiklikleri gibi hususlar özellik işleme aşamasında kullanılmıştır. Çıkarılan özellikler kullanılarak model eğitilmiştir. Değişen sayılarda örneklemle gerçekleştirilen testlerde en iyi sonucu Adaboost (%99,54 doğruluk, %99,82 duyarlılık) algoritması vermiştir.

Mevcut dinamik analiz yöntemlerinin kaçınma saldırılarına zafiyeti bulunması hususunu temel alarak gerçekleştirilen çalışmada [11], üç aşamalı, BiGRU ve VGG 19 algoritmalarını kullanan SMASH modeli geliştirilmiştir. Alt seviye donanım özellikleri de kullanılarak geliştirilen gelişmiş sınıflandırma modeliyle gerçekleştirilen tahminlerde, %94,9 doğruluk ve %94,1 f1 skoru elde edilmiştir.

Toplu öğrenme tabanlı hibrit bir sınıflandırma modeli sunulan çalışmada [12], ClaMP (Classification of Malware with PE Headers) veri seti kullanılmıştır. İki aşamadan oluşan modelin ilk aşamasında CNN ve istifli toplu öğrenme kullanılarak sınıflandırma yapılmıştır. İkinci aşamada 13 makine öğrenmesini içeren meta sınıflandırıcı kullanılmıştır. Sunulan metodun ilk aşamasında toplu öğrenme ve CNN, meta öğrenme aşamasında ekstra ağaçlar kullanıldığında %99,99 doğruluk, kesinlik ve f1 skoru, %99,98 duyarlılıkla tespit gerçekleştirilmiştir.

Yazılımların statik ve dinamik özelliklerinin birlikte kullanılmasıyla tespit gerçekleştirilmesi amaçlanan çalışmada [13], tahmin için biri ağırlıklı oylamaya diğeri istiflemeye dayalı iki metot sunulmuştur. Sınıf doğruluklarına göre sıralı birleştirme ağırlıklı (WeightedVoting_RACA) oylama metodu, %99,5 doğruluk ve %99,6 TPR ile tahmin

gerçekleştirilmiştir.

Çalışmada [14], görselleştirme tabanlı iki aşamalı bir zararlı yazılım sınıflandırma metodu sunulmuştur. İlk aşamada, doku özellikleri çıkarma ve bir matris üzerine gri ölçeğin mekânsal konumlandırmasının tanımlanması ile gri-seviyeli mekânsal bağımlılık matrisini hesaplanması işlemleri gerçekleştirilmektedir. İkinci aşamada ise toplu öğrenme metoduyla sınıflandırma gerçekleştirilmektedir. Malimg veri seti üzerinde %95,58 doğrulukla tahmin gerçekleştirilmiştir.

Çalışmada [15], özel kaçınma tekniklerine karşı dayanıklılık sağlamak için genel ve özel tespit ediciler kullanılan bir toplu öğrenme modeli sunulmuştur. Bilinmeyen zararlı ailelerine ait zararlıların ve değişim gösteren zararlıların etkin bir şekilde tespit edilmesi amaçlanmıştır. Eğitilen özelleşmiş tespit edicilerin, özellikle evrilen zararlıların tespitinde her zaman genel tespit ediciler kadar iyi performans sergilememesinden ötürü, genel ve özelleşmiş tespit edicilerin birlikte kullanıldığı istifleme tabanlı bir model sunulmuştur. Modelde kullanılan genel tespit ediciler tüm veri setiyle eğitilirken, diğer yandan her bir zararlı ailesi için ayrı bir özelleşmiş sınıflandırıcı eğitilmiştir. Genel ve özelleşmiş tespit edicilerin sonuçlarını girdi alarak tahmin yapan bir meta sınıflandırıcı ile örneklemlerin nihai sınıflandırması yapılmıştır. Analiz süreci sırasında gözlem penceresinin uzunluğunun değiştirilebilmesi ve farklı dedektör kombinasyonlarının kullanımına imkân sağlaması sayesinde, sunulan tespit yönteminin öngörülebilirliğinin azaltılması mümkün olmaktadır.

Çalışmada [16], özel bir kayıt kaybı fonksiyonlu LightGBM ile zararlı yazılım tespiti modeli sunulmuştur. Zararlı yazılım tespiti kapsamında statik yöntemlerin doğruluk performansları incelenmiştir. EMBER veri seti üzerinde yapılan çalışmada, önerilen maliyet duyarlı kayıp fonksiyonuyla çalıştırılan normal kayıp fonksiyonuyla çalıştırılan modele göre daha iyi tahmin değerlerine (AUC=0,979) sahip olduğu görülmüştür. Özel kayıp fonksiyonunda yanlış pozitif ve yanlış negatif değerlerinden faydalanılmıştır.

Makine öğrenmesi temelli tespit metotlarının başarısının eğitim verilerine dayanmakta ve bu durum eğitim setinde bulunmayan yeni zararlı ailelerine ait zararlı yazılımların tespitini zorlaştırmaktadır. Bu probleminin çözümü için, sınıflandırılmış sonuçların değerlendirilmesi maksadıyla karışık uzaklık kriteri kullanan yeni bir değerlendirme yaklaşımı olan yumuşak ilişki yaklaşımı sunulmuştur [17]. Sunulan model, Big 2015 veri

seti kullanılarak eğitilmiş ve %99,8 doğrulukla tahmin gerçekleştirilmiştir.

Çalışmada [18], sıfırıncı gün açıklıklarını istismar eden zararlıların etkin bir şekilde tespit edilebilmesi amaçlanmıştır. Bu maksatla, torbalama ve yükseltme yöntemlerinin birlikte kullanıldığı bir toplu öğrenme modeli sunulmuştur. Makine öğrenmesi modelinin tahminleriyle korele edilebilmesi ve makine öğrenmesi modelinin tahmininde en etkili özelliklerin tespit edilebilmesi için Shapley değerleri olasılık ölçeğine dönüştürülmüştür. Sınıflandırıcı olarak XGB, LightGBM, RF ve EXT algoritmalarının kullanıldığı çalışmada, XGB algoritmasının farklı veri setleri için %97,87, %97,5 ve %98,49 doğrulukla diğer sınıflandırma algoritmalarından daha iyi tahmin performansı sergilediği görülmüştür.

Toplu öğrenme ile zararlı yazılım tespiti kapsamında incelenen çalışmalar çizelge 2.1'de sunulmaktadır.

Çizelge 2.1. Toplu öğrenme ile zararlı yazılım tespiti kapsamında incelenen çalışmalar

ÇALIŞMA	VERİ SETİ	KULLANILAN ALGORİTMALAR	KULLANILAN METRİKLER
Roseline, S.A. [6]	Kaggle's Big 2015	RF EXT	Doğruluk - %98,91
Dai, Y. ve diğerleri [11]	Özgün Veri Seti	SMASH (BiGRU, VGG 19)	Doğruluk - %94,9, F1 Skoru – 94,1
Jahromi, N.A ve diğerleri [7]	Kaggle's Big 2015, Fidye Yazılım Veri Seti, Özgün Veri Seti	Geliştirilmiş TELM	Doğruluk - %97,29, Doğruluk - %98,96
Euh, S. ve diğerleri [8]	Özgün Veri Seti	XGB	Doğruluk(WEM) – %100 Doğruluk(API) - %94,7
Damaševičius, R. ve diğerleri [12]	ClaMP	CNN İstifleme	Doğruluk - %99,99, Kesinlik - %99,99, F1 Skoru - %99,99, Duyarlılık - %99,98

Çizelge 2.1. (devam) Toplu öğrenme ile zararlı yazılım tespiti kapsamında incelenen çalışmalar

ÇALIŞMA	VERİ SETİ	KULLANILAN ALGORİTMALAR	KULLANILAN METRİKLER
Cohen, A. ve diğerleri [9]	Özgün Veri Seti	LightGBM	Doğruluk - %95,1
Gupta ve Rani [13]	Özgün Veri Seti	Torbalama (Weighted Voting_RACA)	Doğruluk - %99,5, TPR - %99,6, MCC - 0,991
Verma, V. ve diğerleri [14]	Malimg	Kernel-Based Extreme Machine Learning	Doğruluk - %98,58 Kesinlik - %98,06 F1 Skoru - %98,05
Singh, J. ve Singh, J. [10]	Özgün Veri Seti	AdaBoost	Doğruluk - %99,54
Ficco, M. [15]	Drebrin, Malgenome, CICAnd-Mal2017 Özgün Veri Seti	DNN İstifleme	Doğruluk - %87,0 Kesinlik - %92,7 Hassasiyet - %82,6
Gao, Y. ve diğerleri [16]	EMBER 2017 FFRI	LightGBM	Doğruluk - %99,81 Doğruluk - %99,84
Zhang, Y. ve diğerleri [17]	Kaggle’s Big 2015	S-value	Doğruluk - %99,8
Kumar, R. ve Subbiah, G. [18]	Özgün Veri Seti (D1, D2, D3)	RF LightGBM XGB* EXT	Doğruluk - %97,87, Doğruluk - %97,5, Doğruluk - %98,49

İncelenen çalışmalarda, farklı veri setlerinin ve zararlı yazılımların farklı özellikleri kullanıldığı görülmektedir. Çalışmalarda, ikili sınıflandırma, zararlı yazılım ailesinin belirlenmesi ve sadece bir türe ait zararlı yazılımın belirlenmesine yönelik metotlarla karşılaşılmaktadır. Çalışmaların, önemli bir bölümünde benzer çalışmalarla değil aynı çalışma içerisinde sonuçları elde edilen mevcut sınıflandırıcılarla karşılaştırmalar yapılmaktadır.

2.2. Kümeleme ve Sınıflandırma Algoritmalarının Birlikte Kullanımı

Kümeleme, verilerin benzerliklerine göre küme ya da sınıflara bölündüğü, yüksek benzerlik gösteren verilerin aynı küme ya da sınıfa atandığı bir denetimsiz öğrenme metodudur. Zararlı yazılım analizi kapsamında genel olarak kullanılan kümeleme algoritmaları, K Means, DBScan ve hiyerarşik kümeleme algoritmalarıdır. Sınıflandırma, veri setinin karakteristiğine bağlı olarak, bir örneklemin ön tanımlı sınıf ya da gruplarla eşleştiren denetimli öğrenme yöntemidir [19].

Makine öğrenmesi kullanılarak zararlı yazılım tespiti gerçekleştiren metotlar incelendiğinde çalışmaların çok büyük bir kısmında sınıflandırma algoritmalarının, küçük bir kısmında ise kümeleme algoritmalarının kullanıldığı gözlemlenmiştir. Kümeleme ve sınıflandırma algoritmalarının birlikte kullanımının ise çok nadir olduğu görülmüştür.

Çalışmada [20], Android işletim sistemine yönelik zararlı yazılımların sınıflandırma etkinliğinin geliştirilmesi amaçlanmıştır. Sunulan modelde, bulanık C-Means kümeleme algoritması ve LightGBM sınıflandırma algoritması birlikte kullanılmıştır. Bulanık kümeleme algoritması kullanılarak Android uygulama izinlerinden yeni özellikler üretilmiştir. LightGBM sınıflandırma algoritmasıyla, üretilen bu özellikler kullanılarak sınıflandırma gerçekleştirilmiştir. Önerilen metot karar ağacı, KNN, SVM sınıflandırıcılarından daha iyi performansla tahmin gerçekleştirmiştir.

Çalışmada [21], yazılımların statik özellikleri ile Android zararlı yazılımlarının tespiti için geliştirilen CENDroid modeli sunulmuştur. Model, kümeleme ve toplu öğrenmenin birlikte kullanılması suretiyle Android uygulamalarının zararlı ya da faydalı olduğunu ortaya koymaya çalışmaktadır. Kümeleme algoritması olarak k-mode algoritması, sınıflandırıcı olarak karar ağacı, ELM, lojistik regresyon, Ripper ve lineer destek vektör makineleri kullanılmıştır. Eğitim setinde gerçekleştirilen kümelemede, tek tip sınıfa sahip olan kümelerdeki üyelerinin sınıflandırılması ve farklı tip sınıflara sahip olan kümelerdeki üyelerin sınıflandırılması için farklı sınıflandırma aşamaları kullanılmıştır. Uygulama izinleri, uygulama çağrısı etiketleri ve her ikisinin birlikte kullanımı ile sınıflandırma gerçekleştirildiği durumlar incelenmiştir. Her iki özellik grubunun birlikte kullanıldığı durumlarda temel sınıflandırıcıların daha yüksek doğruluk, kesinlik ve duyarlılıkla tahmin gerçekleştirdiği ortaya konmuştur. Ortalama, ağırlıklı ortalama ve istifleme metotları

kullanılarak CENDroid yöntemiyle gerçekleştirilen tahminler ise temel sınıflandırıcılar ile gerçekleştirilen tahminlerden daha yüksek doğruluk, kesinlik ve duyarlılıkla gerçekleşmiştir.

Çalışmada [22], zararlı bir yazılımın tespit edilmesi ve tespit edilen bu zararlı yazılımın hangi aileye ait olduğunu ortaya konulması amaçlanmıştır. Bu amaca ulaşmak için iki ayrı model önerilmiştir. İlk modelde, iki safhalı istifleme yöntemi kullanılmış ve ağırlıklı ortalama ile bir örneğin zararlı olup olmadığı tahmin edilmiştir. İkinci modelde ise, t-SNE algoritmasıyla boyutu azaltılan örneklemlerin ailesinin belirlenmesi ve K-means algoritmasının küme sayısının ideale ulaşması için sürekli besleme yapılan bir model oluşturulmuştur. Doğruluk, duyarlılık, kesinlik ve F1 skoru metrikleri kapsamında, rastgele ağaç, destek vektör makineleri, XGBoosting gibi sınıflandırıcılarla alınan sonuçlarla yapılan karşılaştırmalarda önerilen her iki modelin de diğer temel sınıflandırıcılardan daha iyi sonuç verdiği görülmüştür.

Çalışmada [23], Android zararlılarının tespiti için üç aşamalı bir model sunulmuştur. Modelin ilk aşamasında, görüntü bazlı bir kümeleme deseni oluşturmak için eğitim setinin her görünümünde tekrar eden bir kümeleme safhası yer almaktadır. Bu desen sayesinde, ilgili görünümün özellik vektörü temel alınarak eğitim örnekleri ayrık kümelere bölünür. Modelin ikinci aşaması, bir görünümde öğrenilen her kümeleme desenini soldaki görünümlere aktarmak için gerçekleştirilen bir bilgi değişim aşamasıdır. Modelin son aşaması, zararlı yazılım tespit deseninin öğrenilmesi için önceki iki safhada ortaya konulan kümeleme desenlerinden faydalanan istifleme tabanlı bir sınıflandırma yaklaşımı kullanılmıştır. Çalışmada kümeleme algoritması olarak K-means algoritması, sınıflandırıcı olarak rastgele orman algoritması kullanılmıştır. Literatürdeki diğer çalışmalarla yapılan karşılaştırmada, ortalama AUC değeri açısından tüm çalışmalardan daha iyi sonuç verirken, hatalı pozitif oranında ise birçok çalışmadan daha kötü performans ortaya koyduğu görülmüştür.

Kümeleme ve sınıflandırma algoritmalarının birlikte kullanımı kapsamında incelenen çalışmalar çizelge 2.2'de sunulmaktadır.

Çizelge 2.2. Kümeleme ve sınıflandırma algoritmalarının birlikte kullanımı kapsamında incelenen çalışmalar

ÇALIŞMA	VERİ SETİ	KULLANILAN ALGORİTMALAR	KULLANILAN METRİKLER
Taha, A.A. ve Malebary, S.J. [20]	Özgün Veri Seti	Kümeleme (C-Means) Yükseltme (LigthGBM)	Doğruluk - %94,63 Kesinlik - %97,70 AUC - %98,74
Badhani, S. ve Mutto, S.K. [21]	Özgün Veri Seti	Kümeleme (k-mode) Torbalama (Oy Çokluğu, Ağırlıklı Oylama)	Doğruluk - %99,41 Kesinlik - %99,3 Hassasiyet - %98,97 F-1 Skoru - %99,13
Yang, H. ve diğerleri [22]	Datacon 2019	t-SNE, İstifleme	Doğruluk - %99,67, Duyarlılık - %99,67, F1 Skoru - %99,67
Appice, A. ve diğerleri [23]	Özgün Veri Seti	Kümeleme, İstifleme	Hassasiyet - %96 AUC - %96,6

Çalışmalar incelendiğinde, denetimli ve denetimsiz öğrenme algoritmalarının birlikte kullanımının standart bir yaklaşımı olmadığı görülmektedir. Bu tez çalışmasında sunulan model, oluşturulan alt kümelere göre sınıflandırıcı belirlenmesi ile denetimli ve denetimsiz öğrenme algoritmalarının kademeli kullanımı açısından çizelge 2.1 ve çizelge 2.2'de sunulan çalışmalardan farklılık göstermektedir.

2.3. BODMAS ve EMBER 2018 Veri Setleri Kullanılarak Yapılan Çalışmalar

Bu bölümde, BODMAS ve EMBER 2018 veri setleri kullanılarak gerçekleştirilen çalışmalar incelenmektedir. BODMAS veri seti kullanılarak gerçekleştirilen çalışmalar çizelge 2.3'te, EMBER 2018 veri seti kullanılarak gerçekleştirilen çalışmalar 2.4'te sunulmaktadır. Çalışmada kullanılan Kaggle Zararlı Yazılım Tespiti veri setiyle ilgili akademik çalışma bulunmadığı için bu bölümde ele alınmamaktadır.

Fauzan ve diğerleri tarafından yapılan çalışmada [24], oy çokluğunun kullanıldığı hibrit bir sınıflandırma modeli sunulmaktadır. Çalışmada LightGBM, XGBM ve LR algoritmaları kullanılmaktadır. Zararlı yazılımların statik analizine yönelik gerçekleştirilen çalışmada

BODMAS veri setinden faydalanılmaktadır. Sınıflandırıcıların tekil ve hibrit performansının karşılaştırılması neticesinde doğruluk, f1 skoru, kesinlik ve hassasiyet metrikleri açısından daha iyi performansa sahip olan sınıflandırıcı başarılı kabul edilmektedir. BODMAS veri seti üzerinde, rastgele orman algoritması kullanılarak 2381 özelliğin 375 tanesi seçilmektedir. Özellik seçimini müteakip sınıflandırıcılar tarafından elde edilen sonuçlar incelendiğinde, XGB ve LGBM+XGB+LR'nın %99,63 doğruluk ve %99,56 f1 skoru ile aynı performansı gösterdiği görülmektedir.

Zararlı yazılımların ikili sınıflandırılması kapsamında yapılan çalışmada [25], dönüştürücü tabanlı iki farklı model sunulmaktadır. Sunulan modellerin ilki, ikili dosyalardan ayıklanan bayt dizilerini kullanarak sıra tabanlı bir sınıflandırma yapmaktadır. İkinci model olarak ise, ikili dosyaları görüntülere dönüştüren ve imaj tabanlı sınıflandırma yapan bir model sunulmaktadır. BODMAS veri setinde bulunan zararlılar, zararlı ailelerine göre gruplandırılarak 1000'den fazla örnekleme sahip zararlı ailelerine üye örneklemler ile BODMAS'ın bir alt kümesi oluşturulmaktadır. Oluşturulan alt veri seti üzerinde yapılan sınıflandırmada, %97,00 doğruluk ve %96,99 ağırlıklandırılmış f-1 skoru değeri elde edildiği görülmektedir.

Çizelge 2.3. BODMAS veri seti kullanılarak gerçekleştirilen çalışmalar

ÇALIŞMA	VERİ SETİ	ÖZELLİK SAYISI	KULLANILAN ALGORİTMALAR	KULLANILAN METRİKLER
Fauzan ve diğerleri [24]	BODMAS	375	LightGBM + XGB + LR	Doğruluk - %99,63 Kesinlik - %97,29 Hassasiyet – %99,50 F-1 Skoru – %99,56
Qikai, L. [25]	BODMAS	2381	ImgConvAttn-Frequency + SeqConvAttn	Doğruluk - %97,00 F-1 Skoru - %96,99

Artan veri boyutları, zararlı yazılım tespiti kapsamında kullanılan makina öğrenmesi algoritmalarının etkin bir şekilde kullanılmasını kısıtlamaktadır. Çalışmada [26], makina öğrenmesi modellerinin daha etkin ve hızlı bir şekilde eğitilmesi maksadıyla, yeni bir model

sunulmuştur. MalConv yönteminin Küresel Kanal Geçişi (Global Channel Gating) ile birlikte kullanılması sayesinde, hem tahmin süresi makul bir seviyede tutulurken tahmin performansı iyileştirilmektedir. Sunulan model, EMBER 2018 veri seti üzerinde %93,29 doğruluk ve %98,04 AUC ile tahmin gerçekleştirmektedir.

Zararlı yazılımların kullandıkları aldatma teknikleri, bu tür varyantların tespit edilmesinde özel dikkat gerektirmektedir. Geleneksel yöntemler, bu tür zararlıların tespiti için ihtiyaç duyulan araçlara sahip değildir. Çalışmada [27], ikili dosyaların gri tonlamalı görüntüye dönüştürülmesine dayanan bir model sunulmaktadır. EMBER veri seti üzerinde yapılan çalışmada, gri tonlamalı görüntüyle CNN ve RF sınıflandırıcılarının kullanıldığı model, %88,04 doğruluk ve %89,14 f1 skoru sağlamaktadır. Aynı veri seti için kırmızı-yeşil-mavi tonlamalı görüntüye dönüştürülen veri setinde %94 doğruluk ve %93,4 f1 skoru ile tahmin gerçekleştirilmektedir.

Çalışmada [28], zararlı yazılımların statik olarak analiz ve tespitinin etkin bir şekilde gerçekleştirilmesi amaçlanmaktadır. Mevcut problemlemlerin ortadan kaldırılması maksadıyla, bir veri setinden özellik seti çıkaran ve statik PE dosyalarını etkin şekilde sınıflandıran bir model sunulmaktadır. Çalışmada, model oluşturmaktan çok özellik çıkarımının önemi vurgulanmakta, iyi çıkarılmış özelliklerle beslenen yapay sinir ağlarının daha iyi performans verdiği belirtilmektedir. Ember 2018 veri seti üzerinde gerçekleştirilen tahminde, %94,09 doğruluk ve %88,66 f1 skoru performansı sağlanmaktadır.

Bilgi sistem varlıklarının korunması için zararlı yazılımların çok düşük yanlış pozitif oranlarıyla tespiti büyük önem arz etmektedir. Çalışmada [29], çeşitli veri setleri, modeller ve özellik tipleri için zararlı yazılım tespiti kapsamında belirsizliğin kullanımına yönelik araştırma sunulmaktadır. Elde edilen sonuçlar incelendiğinde, EMBER 2018 veri seti için en iyi doğruluğun %94,72 ile Bayesyen LR tarafından, en iyi AUC performansının %98.62 LightGBM (toplu öğrenme) tarafından sağlandığı görülmektedir.

Akıllı polimorfik zararlı yazılımların yaygınlaşmasıyla, zararlıların etkin bir şekilde tespit edilmesi ve karantinaya alınmasına yönelik sistemler geliştirmek bir zorunluluk haline gelmiştir. Çalışmada [30], statik analiz ve makina öğrenmesi kullanılarak zararlı yazılım tespiti probleminin çözülmesi amaçlanmaktadır. Bu kapsamda iki adımlı bir model sunulmaktadır. Birinci adımda, GB sınıflandırıcısı ve EMBER 2018 veri seti kullanılarak

oluşturulan ikili sınıflandırma modeli, ikinci adımda CNN ve Virüs MNIST veri seti kullanılarak oluşturulan aile sınıflandırma modeli yer almaktadır. Elde edilen sonuçlar incelendiğinde, EMBER 2018 veri seti için en iyi tahmin performansının %96 doğruluk ve %96 f1 skoruyla GB sınıflandırıcısı tarafından gerçekleştirildiği görülmektedir.

Çizelge 2.4. EMBER 2018 veri seti kullanılarak gerçekleştirilen çalışmalar

ÇALIŞMA	VERİ SETİ	SEÇİLEN ÖZELLİK SAYISI	KULLANILAN ALGORİTMALAR	KULLANILAN METRİKLER
Raff, E. ve diğerleri [26]	EMBER 2018	Veri Yok	MalConv w/ GCG	Doğruluk - %93,29 AUC - %98,04
Marais, B. ve diğerleri [27]	EMBER 2018	2381	CNN	Doğruluk - %94 F1 Skoru - %93,4
Sumit, S.L. ve Adamuthe, A.C. [28]	EMBER 2018	2381	Yoğun Sinir Ağı	Doğruluk - %94,09 Kesinlik - %90,14 Hassasiyet - %88,85 F1 Skoru - %88,66
Nguyen, A.T. ve diğerleri [29]	EMBER 2018	2381	Bayesci LR	Doğruluk - %94,72 AUC - %98,15
Thosar, K. ve diğerleri [30]	EMBER 2018	2381	GB	Doğruluk - %96 Kesinlik - %96 Hassasiyet - %96 F1 Skoru - %96

3. ZARARLI YAZILIMLAR

Bu bölümde, zararlı yazılımların türleri, analiz yöntemleri, geleneksel tespit yaklaşımları ve zararlı yazılım tespitinden kaçınma yöntemleri ele alınmaktadır.

3.1. Zararlı Yazılım Türleri

Siber saldırganlar tarafından yaygın olarak kullanılan kötücül yazılımlar, kullanım amacı, hedef sistemdeki faaliyetleri, yayılma biçimleri gibi çeşitli konularda farklılıklar göstermektedirler. Virüs, solucan, truva atı, casus yazılım, fidye yazılım, botnet, rootkit, dosyasız yazılım, arka kapı, zaman/mantık bombası gibi çeşitli zararlı yazılım türleri bulunmaktadır.

Virus

Virüsler, sistem veya kullanıcı tarafından tetiklenerek bir programdan diğerine ya da bir bilgisayardan başka bir bilgisayara yayılabilen, kodunu diğer programlara enjekte ederek kendini çoğaltan kötü amaçlı yazılımlardır [31].

Solucan

Solucanlar, kendilerini kopyalayarak bilgisayarlar arasında yayılabilen ve kurban bilgisayardaki dosya ve verilere zarar veren kötü amaçlı programlardır. Solucanlar dosya şifreleme ya da gereksiz eposta gönderme gibi eylemler gerçekleştirmektedir. Bir programa enjekte edilmiş şekilde yayılan virüslerin aksine, solucanlar kendi konteynırlarında yayılmaktadır [32].

Truva atı

Truva atları, meşru bir program olarak çalışıyor görünürken gizli ve istenmeyen faaliyetler gerçekleştiren programlardır. Truva atları, saldırganların kurban bilgisayarlara erişmesine, parola ve bankacılık bilgileri gibi kullanıcıların gizli bilgilerinin saldırganlar tarafından ele geçirilmesine imkân tanımaktadır [31].

Casus yazılım

Casus yazılımlar, kullanıcıların faaliyetleri gizli bir şekilde sürekli olarak gözlemleyen kötücül yazılımlardır. Kullanıcılar için gizlilik ifade eden kullanıcı adı, parola, bankacılık bilgileri, ziyaret edilen web siteleri gibi çeşitli bilgileri toplayan casus yazılımlar, bu bilgileri saldırgana iletmektedirler [33].

Rootkit

Rootkit, bir bilgisayar sistemine erişmek ve diğer kötü amaçlı yazılımların sisteme girmesini sağlamak için programlanmış, normal kullanıcıların sahip olduğu yetkilerden daha yüksek yetkilerle işlem gerçekleştiren bir kötücül yazılım türüdür [34].

Fidye yazılım

Fidye yazılımlar, bilgisayar korsanlarının bilgisayarları kilitlemesini, kurbanın kendisi için önemli sayılan verilere erişimini kısıtlamasını sağlayan bir kötücül yazılım türüdür. Fidye yazılımlar, erişilebilirliğini engellediği verilerin yeniden erişilebilir olması için kurbanlardan ödeme talep etmektedirler [35].

Reklam yazılımları

Reklam yazılımı, kullanıcıların önüne istekleri dışında reklam getiren bir tür kötücül yazılımdır. Bu tür yazılımlar genellikle ücretsiz olarak indirilen yazılımlar ve oyunlar ile gelmektedir [36].

Botnet

Bilgisayarları, akıllı telefonları ve nesnelerin interneti cihazlarını uzaktan kontrol ederek bu cihazların kötücül amaçlar doğrultusunda kullanılmasına imkân tanıyan zararlı yazılımlardır. Hedef cihazlara bulaşan virüs ile saldırgan cihazın kontrolünü sağlar. Botnetler genellikle hizmet dışı bırakma saldırıları ve spam eposta saldırıları için kullanılmaktadır. Bu tür saldırıların kurbanlarının, cihazlarının enfekte olduğunun farkında olması çok zordur [37].

Mantık bombaları

Bilgisayar dünyasındaki saldırıların önemli bir bölümü dış dünyadan gelmektedir. Fakat organizasyon içi kaynaklı saldırıların da olabilmesi mümkündür. Bu tür saldırılar, dış kaynaklı saldırılara göre daha ciddi sonuçlara sebep olmaktadır. Mantık bombaları, iç tehditler tarafından oluşturulan, gizlenen, genellikle başka bir kodun içine gömülen, belirtilen mantıksal durum ortaya çıkıncaya kadar uykuda kalan ve tespit edilmesi oldukça zor olan kötücül kodlardır [38].

Arka kapılar

Kimlik doğrulama mekanizmalarını işlevsiz kılarak, saldırganlara hedef sisteme erişme imkânı tanıyan zararlı kodlardır. Bu tür kodların bir kısmının bizzat geliştirici tarafından kasten bırakılmış olmasından ötürü tespiti oldukça zordur [39].

Dosyasız zararlı yazılımlar

Dosyasız zararlı yazılımların, klasik zararlı yazılımlardan farklı olarak, dosya sistemi üzerinde herhangi bir izdüşümü yoktur. Bu tür yazılımlar faaliyetlerini bellek üzerinde gerçekleştirirler. Dosya sisteminde herhangi bir işlem gerçekleştirilmediği için imza tabanlı tespit yaklaşımları bu tür zararlıların tespitinde etkisiz kalmaktadır. Ayrıca, sağladıkları süreklilik sayesinde kurbanlar için ciddi seviyede tehdit oluşturmaktadırlar [40].

3.2. Önemli Zararlı Yazılım Örnekleri

Zararlı yazılımlar, bilgisayar sistemini kesintiye uğratmak, veri çalmak, hassas verileri silmek veya şifrelemek, bilgisayar korsanlığı yapmak, temel bilgi işlem işlevlerini değiştirmek veya izlemek gibi amaçlarla tasarlanmaktadır. Bu yazılımlar, hedef bilgisayar üzerinde kullanıcıların rızası olmadan çalıştırılmaktadır [41].

Bilinen ilk zararlı yazılım, deneysel amaçlarla geliştirilmiş olan, kendisini uzak sistemlere kopyalayabilen bir yazılım olan Creeper'dır [42]. Virüs, solucan, fidye yazılımı, truva atı gibi birçok çeşit zararlı yazılım bulunmaktadır ve bu tür zararlıların kullanıldığı saldırılar son dönemde büyük artış göstermektedir [41]. Kullanım şekilleri, teknikleri ve etkileri

nedeniyle en çok bilinen zararlıların bir kısmı aşağıda sunulmuştur.

StuxNet

Stuxnet, Siemens SCADA sistemlerini hedef almak için özel olarak geliştirilmiş olan, yüksek karmaşıklığa sahip bir zararlı yazılımdır. İran'ın uranyum zenginleştirme faaliyetlerini sekteye uğratmak için özel olarak geliştirilen bu zararlı yazılımda, daha önce kullanılmamış 4 adet sıfırıncı gün açıklığı istismar edilmiştir. Natanz zenginleştirme tesislerindeki santrifüjleri hedef alan Stuxnet'in hedef ağa geçişi USB bellekler ile gerçekleşmiştir. Zararlı, MS08-067 SMB zafiyeti, MS10-061 print spooler servisi zafiyeti ve ağ paylaşımı ile ağ içerisinde yayılmıştır. Siemens SIMATIC Step7 sistemini, sistemin DLL'lerini ve çalıştırılabilir kodlarını enfekte eden zararlı, kullanıcı ve kernel seviyesinde yetkilerle hedef sistemler üzerinde işlemler gerçekleştirmiştir. Hedef sistemlere kontrollü bir şekilde zarar veren Stuxnet, izleme sistemlerine gerçek durumla örtüşmeyen verilen göndererek eylemlerini başarılı bir şekilde gizlemiştir. Stuxnet'in yaklaşık bin kadar santrifüjün bozulmasına sebep olduğu ve İran'ın nükleer programını ciddi bir şekilde sekteye uğrattığı değerlendirilmektedir [43].

WannaCry

WannaCry, kullanıcıların dosyalara veya sistemlere erişimini engelleyen, dosyaları veya tüm cihazı şifreleme kullanarak rehin tutan, kullanıcılar tarafından şifre çözme anahtarı için bir fidye ödemesi karşılığında şifrelenmiş dosyalara erişimine izin veren bir zararlı yazılımdır. 2017 yılında ortaya çıkan WannaCry fidye yazılımı, kendisinden önce gelen benzer zararlılardan çok daha yıkıcı sonuçlara sebebiyet vermiştir. Birçok hastane, üniversite, şirket ve kamu kurumunu etkileyen zararlının iki milyondan fazla kurbanı bulunmaktadır. Dosyaların yedeklerinin bulunmadığı senaryolarda, kurbanların verilerini elde etmesinin tek yolu, istenen fidyeyi ödemektir [44].

Zeus

Zbot, WSNPOEM, NTOS, PRG gibi isimlerle de anılan Zeus zararlısı, genellikle finans sektörünü hedef alan bir truva atıdır. İlk olarak 2007 yılında görülmüş, 2011 yılında ise zararlının kaynak kodu internete sızmıştır. Zararlının kaynak kodunun erişilebilir hale

gelmesinden sonra ICE IX, KINS, Citadel gibi çok sayıda Zeus varyantı ortaya çıkmıştır. Klavye vuruşlarını kaydederek veri çalan, kendisini mesajlaşmayla ve anlık olarak diğer sistemlere kopyalayabilen Zeus, bilgisayar korsanlarının enfekte olan bilgisayarları kontrol etmesine ve izlemesin imkân tanımaktadır [45]. Finansal işlemler sırasında çevrimiçi kötü amaçlı yazılım bulaşmalarının %44'ünden ve banka dolandırıcılıklarının yaklaşık %90'ından Zeus botnetleri sorumlu bulunmuştur [46].

ILOVEYOU

2000 yılının mayıs ayında ortaya çıkan virüs, içinde Dünya Sağlık Örgütü ve Pentagon gibi kuruluşlarında bulunduğu çok sayıda organizasyonu kişisel bilgisayarları enfekte etmiştir. Virüsün sadece İngiltere'de on milyarlarca poundluk zarara sebebiyet verdiği değerlendirilmektedir. Eposta ile yayılan "the Love Bug" virüsü, sadece Microsoft yazılımları kullanan bilgisayarları etkilemiştir. Çok hızlı bir şekilde yayılan ve kırkbeş milyondan fazla bilgisayarı etkileyen bu virüs, bilgi teknolojileri dünyasında tek üreticiye bağlılığın yeniden değerlendirilmesine neden olmuştur [47].

Melissa

1999 yılının mart ayında çok hızlı bir şekilde yayılmaya başlayan Melissa virüsü, çok hızlı bir şekilde yayılabilen zararlı yazılımların öncülerinden biridir. Kolay bir şekilde yaratılan, Word ve VBA gibi popüler teknolojilerin zayıflıklarını istismar eden virüs, mail yoluyla hızlı bir şekilde yayılmıştır. Kullanıcılar tarafından Microsoft Outlook uygulamasında dokümanlar açıldığı esnada, virüsün aktifleşmesi suretiyle yayılım gerçekleştirmiştir. Bir milyondan fazla bilgisayarı, 50 binden fazla sunucuyu ve binlerce şirketi etkileyen virüs, yüz milyonlarca dolarlık hasara neden olmuştur [48].

CovidLock

CovidLock, Covid-19'un getirdiği panik ortamından yararlanan bir fidye yazılımdır. Kullanıcıları, pandemiyi takip edebilecekleri bir uygulama indirdiklerine inandırarak kullanıcı cihazlarına yerleşen bu zararlı, indirilmeyi müteakip kullanıcı dosyalarını şifreler ve cihazın şifresini değiştirir. Cihaza ve dosyalara erişim için fidye talep eden virüs, fidye zararlılarının en güncel örneklerindendir [49].

Emotet

Çok gelişmiş modüler bir truva atı olan Emotet zararlısı, öncelikli olarak kullanıcıların banka hesap bilgilerini oltalama mailleri göndererek ele geçirmeyi amaçlamaktadır. Bunun yanında Trickbot, Ursnif ve IceDiD gibi diğer bankacılık truva atlarını da indirerek hedef sistemleri istismar eder. Ayrıca, Ryuk fidye yazılımı ile kurbanların hassas dosyalarını şifreleyerek siber saldırganlara fayda sağlar. CrowdStrike tarafından yapılan araştırmaya göre, Emotet'tin neden olduğu bir siber olayın etkilerinin giderilmesi için yaklaşık 1 milyon dolara ihtiyaç duyulmaktadır. Yayılımının çok büyük bir kısmını e-posta yoluyla gerçekleştiren Emotet, zararlı URL linklerinden, sahte PDF dosyalarından ve makro ayarları aktif edilmiş Microsoft Word dokümanlarından faydalanmıştır [50].

CryptoLocker

CryptoLocker, Eylül 2013'te P2P Zeus zararlısını kullanarak hızlı bir şekilde yayılan yayılmıştır. Zararlının temel amacı, kurbanların dosyalarını şifreleyip bu şifrelerin çözülmesi karşılığında para kazanılmasını sağlamaktır. Mayıs 2014'e kadar devam eden saldırılar Windows işletim sistemini hedef almıştır. Dosyaların çözülmesi için gerekli anahtarlar sadece CryptoLocker sunucularında tutulmakta ve talep edilen fidyenin ödenmesi dosyaların sağlıklı bir şekilde elde edileceğini garanti etmemekteydi [51]. Eposta yoluyla yayılan zararlıyı farklı programlar vasıtasıyla sistemden kaldırmak mümkün olsa da kullanılan şifreleme algoritmasının gücü sebebiyle dosyaların anahtar olmadan kurtulabilmek mümkün değildi [52]. Virüsün Nisan 2014'e kadar 234000'den fazla bilgisayarı etkilediği değerlendirilmektedir [45].

MyDoom

2004 yılının ocak ayında ilk kez karşılaşılan ve çok hızlı bir şekilde yayılan MyDoom solucanı yayılma vektörü olarak epostayı kullanmıştır. Zararlı, birkaç saat içerisinde 142 ülkeye ve milyonlarca cihaza yayılan MyDoom, sıkıştırılmış bir eposta eklentisi içerisinde farklı dosya uzantılarına sahip bir şekilde hedef sistemi enfekte etmektedir. Hedef sisteme ulaşmasını müteakip kendini kopyalayan bu zararlı, sistem dosyalarında ve kayıt defterinde değişiklik yapmaktadır [53]. Saatler içerisinde çok büyük bir etki seviyesine ulaşan MyDoom'un 38 milyar doların üstünde bir zarara sebep olduğu değerlendirilmektedir [54].

Code red

Temmuz 2001 yılında karşılaşılan Code Red solucanı, 14 saatten kısa bir süre içerisinde 359 binden fazla bilgisayarı enfekte etmiştir. Bu virüsün etkilerinin 2,6 milyar doların üzerinde olduğu değerlendirilmektedir. Code Red solucanı, Haziran 2021'de Microsoft IIS sunucularına yönelik yayımlanan bir zafiyeti istismar etmiştir. Yayılımını rastgele belirlediği IP adresleri üzerinden gerçekleştiren Code Red solucanı, ayın ilk 19 günü yayılma eylemi yaparken, yirminci günden itibaren saldırının ikinci aşaması olan hizmet dışı bırakma saldırısını uygulamıştır [55].

3.3. Zararlı Yazılım Analiz Yaklaşımları

Bilgi teknoloji varlıklarının korunması ve zararlı yazılımların engellenmesi için yalnızca güvenlik duvarı filtrelemesi veya imza tabanlı IPS uygulanması yeterli değildir. Zararlı yazılım analizi alanında eğitim gerektiren ciddi bir tersine mühendislik yapılması gerekmektedir. Zararlı yazılım analizi, bir zararlı yazılımın kaynağını, foksiyonalitesini, olası etkilerini belirlemek için yapılan çalışma ya da süreçlerdir. Bu nedenle, kötü amaçlı yazılım analizi ve sınıflandırması, en popüler araştırma alanlarından biridir [41].

3.3.1. Statik analiz

Bir yazılımı çalıştırmadan analiz etme işlemine statik analiz denir. Statik analiz teknikleri, yazılımların farklı şekillerine uygulanabilmektedir. Eğer yazılımın kaynak kodu mevcutsa, statik analiz araçları bellek bozulmasına sebep olacak kusurlar bulunabilmekte ve yazılımların sistemler için doğruluğunu kanıtlanabilmektedir. Statik analiz, bir yazılımın derlenmiş gösterimi için de uygulanabilmektedir. Yazılımların kaynak kodlarının erişilebilir olmaması durumunda, statik analiz sadece derlenmiş gösterime uygulanabilmektedir. Statik analizin derlenmiş gösterim üzerinde gerçekleştireceği analizden elde edebileceği veri, kaynak kod analizinden elde edilebilecek veriye göre kısıtlıdır. Bilgisayar korsanları, bu kısıtları göz önünde bulundurarak zararlı yazılımlar geliştirmektedirler [56].

3.3.2. Dinamik analiz

Bir programın çalışma zamanında gösterdiği özellikleri analiz etme yöntemi dinamik analiz olarak adlandırılmaktadır. Dinamik analizde, uygulama çağrılarının izlenmesi, fonksiyon parametrelerinin analizi, veri akışının takibi, komut takibi, otomatik başlatma gibi eylemler gerçekleştirilmektedir. Zararlı yazılımların dinamik olarak analiz edilmesi, canlı ortamlardan izole bir ortamda genellikle kum havuzu olarak adlandırılan sistemlerde gerçekleştirilmektedir. Etkili bir dinamik analiz için, bilgisayar korsanları tarafından kullanılan kaçınma tekniklerini işlevsiz hale getirecek analiz ortamlarına ihtiyaç duyulmaktadır [56].

3.4. Geleneksel zararlı yazılım tespit yaklaşımları

Zararlı yazılım tarihi, Creeper isimli, deneysel olarak geliştirilen ve kötü bir amaç için kullanılmayan bir kodla başlamaktadır. Zararlı yazılım zaman içerisinde bilgi teknoloji varlıklarına, bilgi teknolojisi kullanıcılarına, siber fiziksel sistemlere ve dolayısıyla fiziki dünyaya yıkıcı zararlar verebilen bir noktaya gelmiştir [57]. Zararlı yazılımların yarattığı problemlerin önüne geçmek için anti virüs teknolojisi geliştirilmiştir. Anti virüsler, genel olarak, imza tabanlı ve sezgisel zararlı yazılım tespiti tekniklerini kullanmaktadırlar. İmza, bir uzman tarafından zararlı yazılım analizi sonucu oluşturulur. Sezgisel yaklaşım, zararlı yazılımlarla normal yazılımları ayırmak için zararlı yazılım analistleri tarafından geliştirilen çeşitli kurallardan meydana gelir.

3.4.1. İmza tabanlı tespit

İmza tabanlı tespit, kötü amaçlı yazılımın imza biçimindeki benzersiz özelliklerinden faydalanılarak tespitini gerçekleştiren yaklaşımdır. İmza tabanlı tespit yaklaşımında kullanılan özellikler zararlı yazılım analizi süreciyle elde edilmektedir. Zararlı yazılım imzasına en basit örneklerden bir tanesi mesaj özeti (hash) değeridir. Bu metot, analiz edilen yazılım özelliklerinin bir zararlı yazılım veri tabanındaki girdiyle eşleşmesi durumunu kullanmaktadır [57]. Bu yöntem yeni ve bilinmeyen türdeki zararlı yazılımları tespit edememektedir [58].

3.4.2. Sezgisel tespit

Sezgisel tespit yaklaşımıyla, bilinmeyen bir örneğin zararlı olup olmadığının tespiti yapılmaktadır. Bu yaklaşım, ilgili örneğin şüpheli karakteristikleri olup olmadığını ortaya koyarak tespit yapabilecek, uzman bir zararlı yazılım analisti tarafından tanımlanan bir kurallar kümesini kullanmaktadır. Sezgisel tespit yaklaşımı, imza tabanlı tespit yaklaşımından farklı olarak yeni zararlıları ve zararlı varyantlarını tespit edebilmektedir. Sezgisel tespit yaklaşımında statik ve dinamik sezgisel teknikler kullanılabilmektedir. Sezgisel tespit yaklaşımları yeni zararlıları ve zararlı varyantlarını tespit etme yeteneğine sahiptir. Diğer taraftan, hatalı pozitif tespitler yaparak faydalı uygulamaları zararlı olarak görebilmekte, ayrıca yeni kaçınma teknikleri tarafından aldatılabilmektedir [59].

3.5. Zararlı Yazılım Tespitinden Kaçınma Yöntemleri

Zararlı yazılımların tespiti için, faaliyetlerinin ve özelliklerinin incelenmesi suretiyle yapılan taramalar, günümüzün güvenlik mekanizmasının temelidir. Bu tarama yöntemleri dinamik analiz, statik analiz, istatistiksel ve içerik analizi gibi farklı tekniklere dayanmaktadır. Ancak, bu güvenlik tekniklerinin farkında ve tedbirlere aşina olan bilgisayar korsanları, bu tespit yöntemlerinden kolay kaçabilen yeni zararlı yazılım varyantları yaratabilmektedir [41]. Günümüzde, belirli bir organizasyona yönelik olarak geliştirilen zararlı yazılımların sayısı artmakta ve zararlılar da giderek karmaşık olmaktadır. Saldırganlar tarafından, geliştirilen tespit sistemlerini atlatmak için gizleme, paket parçalama ve oturum bölme, uygulamaya özel istismarlar, protokol istismarları, sızma tespit sistemlerine trafik enjeksiyonu, hizmet dışı bırakma ve kod tekrar kullanımı gibi teknikler kullanılmaktadır [60].

Gizleme (Obfuscation)

Gizleme yöntemi, zararlıların tespitten kaçınması kapsamında en yaygın kullanılan yöntemlerden biridir. Tür dönüştürme, karakter manipülasyonu, şifreleme ve polimorfik kodlama ile gerçekleştirilebilmektedir. [60]

Paket parçalama ve oturum bölme

Paket parçalama ve oturum bölme tekniği, ağ seviyesinde tespitten kaçınma tekniğidir. Bu teknik, sistemlerin MTU değerlerini manipule etme yeteneğinden faydalanarak, TCP/IP paketlerinin standart dışı boyutlarda ve zararlı kodun aynı oturum içerisinde farklı paketlerle hedefe gönderilmesi suretiyle uygulanır.

Uygulamaya özel istismarlar

Uygulamalara özel istismarlar, uygulamaların RFC'lerin olması gerektiği şekilde zorunlu kılınmamasından kaynaklanan zayıflıkların istismarına dayanmaktadır. HTTP formatının standart olarak uygulanmaması ya da GET/POST metotlarının birbirleri yerine kullanılması bu tür istismarlara bir örnektir [60].

Protokol istismarları

Protokol istismarları, gönderilen bir paketin sızma tespit sisteminde ve hedef cihazda farklı şekillerde ele alınması zayıflığını istismar etmektedir. Özellikle RPC protokolünün istismarına yönelik çok sayıda örnek mevcuttur [60].

Sızma tespit sistemlerine trafik enjeksiyonu

Bu yöntemde, hedef sisteme gönderilen paketin sızma tespit sistemine ulaşması ancak hedefe cihaza ulaşmaması amaçlanmaktadır. Bunun gerçekleştirilebilmesi için, paketlerin TTL değerleri manipüle edilmekte ve nihai olarak sızma tespit sisteminin paketi uygun şekilde işleyememesi sağlanmaktadır [60].

Hizmet dışı bırakma

Sızma tespit sistemleri, bir ağda en çok kaynağa ihtiyaç duyan sistemlerin başında gelmektedir. Tüm ağ trafiğini analiz eden bu sistemler, kaynak kullanımı istismarına açık sistemlerdir. Bu tür sistemler genel olarak, sistemin durması durumunda ağ kullanımının devam edebileceği şekilde yapılandırılmaktadırlar. Bu cihazların işlemci, bellek, ağ kapasitesi gibi kaynaklarının tüketilmesine yönelik yapılan saldırılar sistemleri işlevsiz

kılmaktadır [60].

Kod tekrar kullanımı

Kod tekrar kullanımı tekniği gücünü, eylemleri gerçekleştirirken hedef sisteme bir zararlı yükleme ihtiyacı duymamasından almaktadır. Bu tekniğin kullanıldığı durumlarda, zararlı eylemler bir yerel işlemin yaptığı geçerli sistem çağrıları olarak görülmektedir. Bu sebeple, zararlı tespiti yapmaya çalışan geleneksel yaklaşımlar işlevsiz kalmaktadır [60].

4. KULLANILAN MAKİNE ÖĞRENMESİ ALGORİTMALARI VE METRİKLER

Bu bölümün ilk kısmında çalışmada kullanılan makine öğrenmesi algoritmaları, ikinci kısmında ise sınıflandırıcıların performanslarını değerlendirmek maksadıyla kullanılan metrikler açıklanmıştır.

4.1. Makine Öğrenmesi Algoritmaları

Çalışmada kullanılan makine öğrenmesi algoritmaları kümeleme, sınıflandırma ve toplu öğrenme algoritmaları başlıkları altında ele alınmıştır.

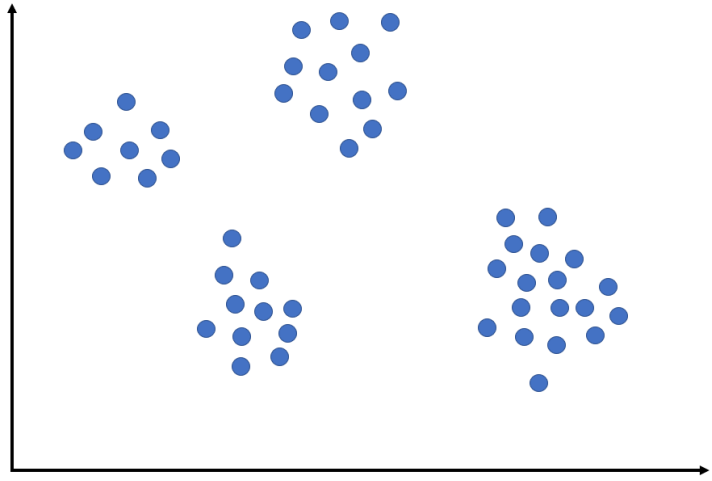
4.1.1. Kümeleme algoritması

Kümeleme, ön bilginin hiç olmadığı ya da yetersiz olduğu durumlarda verileri anlamlı bir şekilde gruplandırmak için uygulanan bir yöntemdir. Kümeleme algoritmaları benzerlik/uzaklık tabanlı, hiyerarşik tabanlı, hataların karesi tabanlı, kernel tabanlı gibi çeşitli alt gruplara ayrılmaktadır [61]. Çalışmada, benzerlik/uzaklık tabanlı kümeleme algoritmalarından K-means algoritması kullanılmıştır.

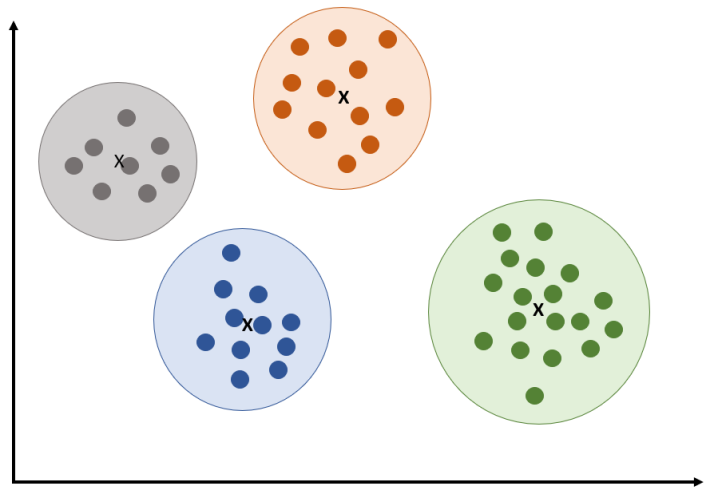
K-means (K-ortalamalar)

En basit denetimsiz öğrenme algoritmalarından biri olan K-means algoritması, kümeleme problemlerinin çözümü için yaygın olarak kullanılmaktadır. Genel olarak veri setinin etiketlenmediği durumlarda kullanılan K-means algoritması, uygun bir küme sayısı belirlenmesi ve devinimli olarak hesaplanan küme merkezlerinin konumuna göre her örneğin, en yakın olduğu küme merkezine göre gruplandığı bir algoritmadır [62].

İki boyutlu bir düzlemde, örneklemlerin etiketlenmemiş hali Şekil 4.1'de, K-means algoritmasıyla kümelenmiş hali Şekil 4.2.'de sunulmuştur.



Şekil 4.1. Örneklemlerin K-means algoritmasıyla kümelenmeden önceki gösterimi



Şekil 4.2. Örneklemlerin K-means algoritmasıyla kümelenmiş gösterimi

Şekil 4.1'de sınıflandırılmamış şekilde bulunan örneklem grubunun iki boyutlu düzlemdeki konumları yer almaktadır. Dört küme merkezine sahip K-means algoritmasıyla kümeleme gerçekleştirildiği durumda ortaya çıkan kümeler şekil 4.2'de sunulmaktadır. K-means algoritması, her bir örneklemi, örneklemin en yakın olduğu küme merkezine göre kümelemektedir.

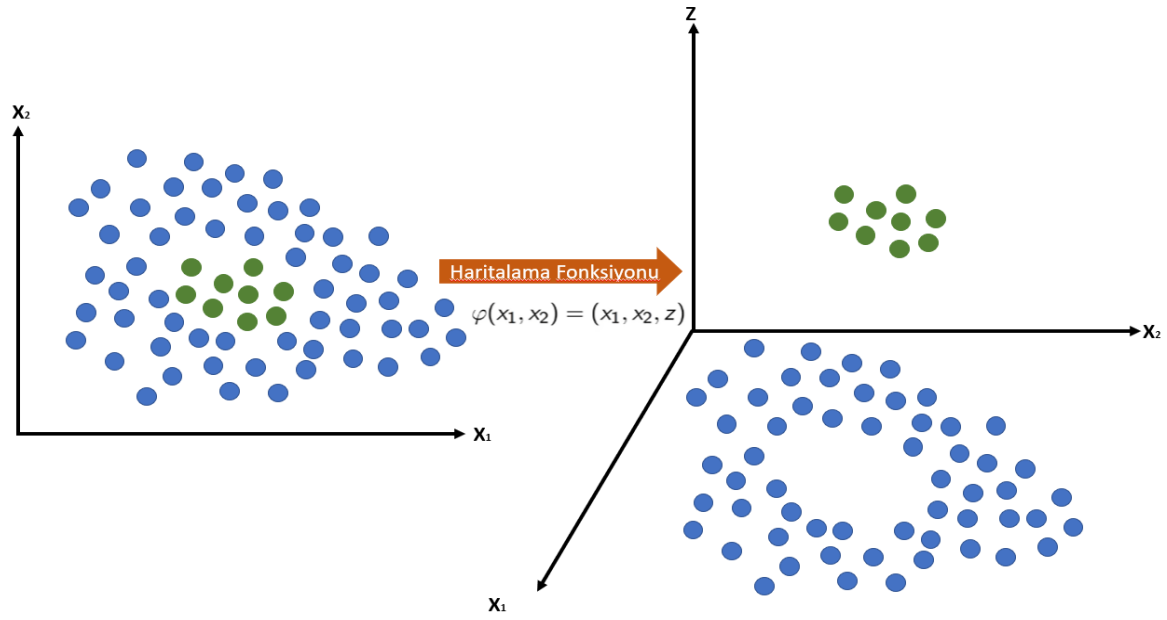
4.1.2. Sınıflandırma algoritmaları

Sınıflandırma algoritmaları, denetimli öğrenme algoritmaları olup, ikili ya da çoklu sınıflandırma gerçekleştirebilmektedir. Önerilen tespit yaklaşımında ikili sınıflandırma yaklaşımı benimsenmiş olup, modelde çekirdek destek vektör makinesi, k-en yakın komşu,

naive-bayes ve karar ağacı sınıflandırma algoritmaları kullanılmaktadır.

Çekirdek destek vektör makinesi (KSVM)

Bir çekirdek fonksiyonu modifiye edilerek destek vektör makinesi sınıflandırıcısının performansının artırılması sonucu ortaya çıkan çekirdek destek vektör makinesi sınıflandırıcısı, çekirdek fonksiyonu tarafından indüklenen Riemann geometrisinin yapısına dayanmaktadır [63]. Kernel fonksiyonu, tek boyutlu bir verinin iki boyutlu gibi sınıflandırılmasına imkân veren bir fonksiyon olup, Çekirdek destek vektör makinelerinin gerçek dünya uygulamalarında çok iyi sonuçlar elde edilmiştir [64]. KSVM algoritmasının, haritalama fonksiyonuyla örneklemleri ayırması durumu Şekil 4.3'te gösterilmektedir.



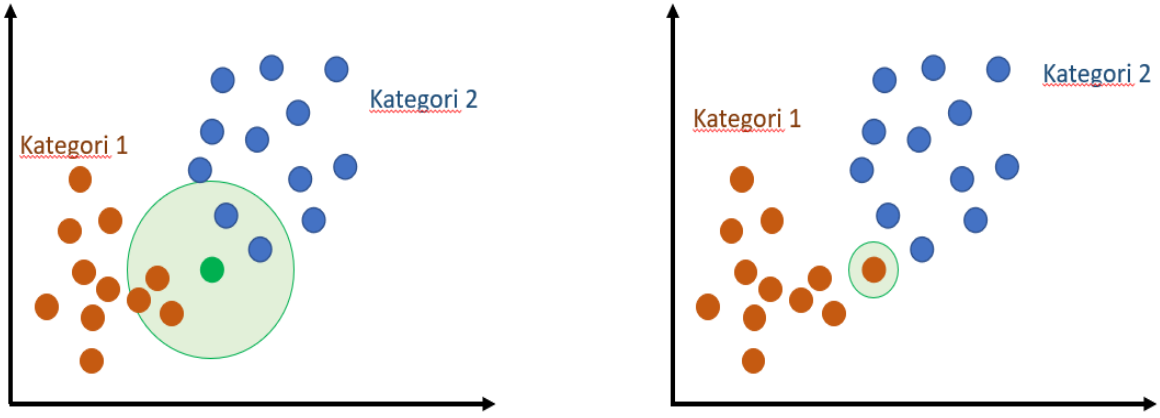
Şekil 4.3. Kernel SVM fonksiyonu ile lineer ayrımın gösterimi

Şekil 4.3'te görüldüğü üzere, KSVM ile normal koşullarda lineer olarak ayrılamayan örneklemlerin, bir haritalama fonksiyonu yardımıyla boyutunun artırılarak lineer olarak ayrılabilmesini sağlanmaktadır.

K-en yakın komşu (KNN)

Herhangi bir eğitim adımına ihtiyaç duyulmayan algoritmada, örneklemler en yakınlarındaki "k" adet komşu sınıf etiketlerine bakılarak sınıflandırılmaktadır. Sınıflandırma, eldeki

verilerin birbirlerine benzerliği veya uzaklığı kullanılarak gerçekleştirilmektedir. Örneklemlerin benzerlikleri değerlendirilirken genellikle Euclidean, Manhattan ve Minkowski yöntemlerinden biri kullanılmaktadır [65]. KNN ile örnek sınıflandırma şekil 4.4'te sunulmaktadır.

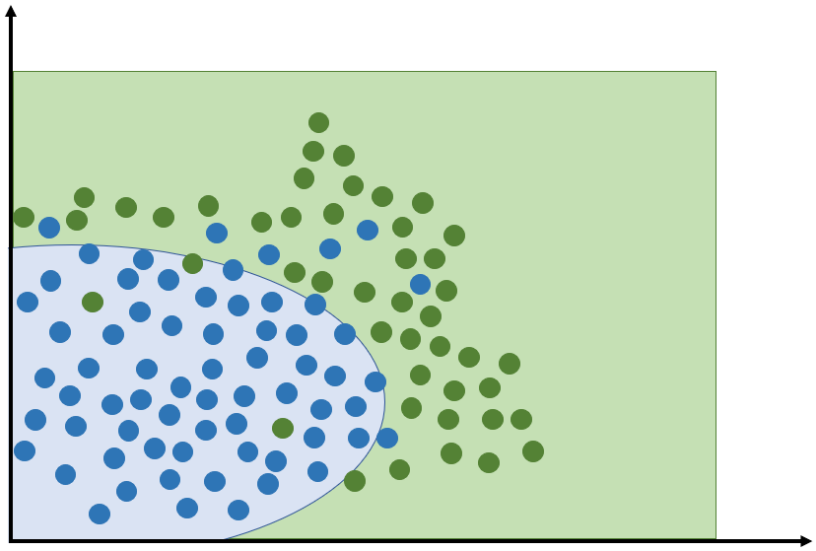


Şekil 4.4. KNN ile sınıflandırma gösterimi

Şekil 4.4'te yer alan ilk düzlemde kategori 1 (kırmızı) ve kategori 2 (mavi) olmak üzere 2 farklı kategoriye sahip bir örneklem grubu ve henüz sınıflandırması yapılmamış bir örneklem (yeşil) bulunmaktadır. KNN ile sınıflandırmada, yeni gelen örneklemin, tek sayı olmak koşuluyla, en yakınındaki belirli sayıdaki örnekleme göre sınıflandırma yapılmaktadır. Yeni dahil olan örneklemin en yakın 5 komşuya göre sınıflandırılması gerektiği durumda, komşulardan 3 tanesi kategori 1'e 2 tanesi ise kategori 2'ye ait olduğu için, yeni dahil olan örneklem kategori 1'e sınıflandırılmıştır.

Naive bayes (NB)

Bir denetimli öğrenme algoritması olan Naive Bayes algoritması, Bayes teoremi temel alınarak geliştirilmiştir. Süreklilik arz eden değerler üretebileceği gibi çoklu ve ikili sınıflandırma da gerçekleştirebilmektedir. Değişkenlerin bağımlı gerçekleşme olasılıklarından faydalanarak sınıflandırma gerçekleştirilmektedir. Çok boyutlu verilere, dengesiz veri setlerine uygunluğu, kolay uygulanabilirliği ve hızı sebebiyle gerçek zamanlı sistemler dahil olmak üzere birçok durumda yaygın olarak kullanılmaktadır [65]. Örnek Naive Bayes sınıflandırmasının iki boyutlu düzlemde gösterimi Şekil 4.5'te sunulmaktadır.



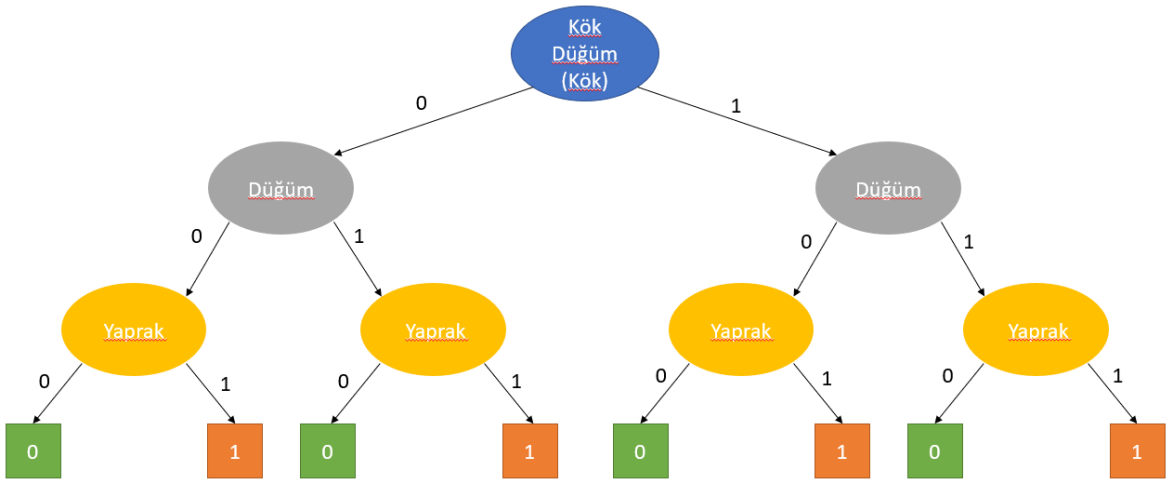
Şekil 4.5. Naive Bayes sınıflandırıcı ile sınıflandırma gösterimi

NB algoritmasının iki boyutlu düzlemdeki sınıflandırma alanlarının ve iki kategorideki örneklerin gösterildiği Şekil 4.5'te, sınıflandırılmak üzere yeni gelen bir örneklem iki boyutlu düzlemde konumlandığı noktayı kapsayan renk kategorisinde sınıflandırılmaktadır.

Karar ağacı (DT)

Karar ağaçları, örneklemleri özellik değerlerine göre sınıflandıran ağaçlardır. Karar ağaçlarındaki her bir düğüm bir özelliğe, her dal bir değere, son safhadaki değerler ise yaprağa denk gelmektedir. Örneklem, düğümün mantıksal işlemine göre kendi özelliğine uygun dal tarafında sınıflandırılmaktadır. Her örneklem, en kuvvetli bağımsız değişkenin hesaplanması ile ortaya çıkan kök düğümden itibaren bu mantıksal sınıflandırmaya tabi olmaktadır. Örneklem, en uçtaki yaprağa geldiğinde nihai sınıflandırması tamamlanmaktadır [65].

Karar ağaçları farklı derinliklerde sınıflandırma gerçekleştirebilmektedir. Derinliği 3 olan örnek bir karar ağacı sınıflandırmasının gösterimi şekil 4.6'da sunulmaktadır.



Şekil 4.6. Karar ağacı sınıflandırması gösterimi

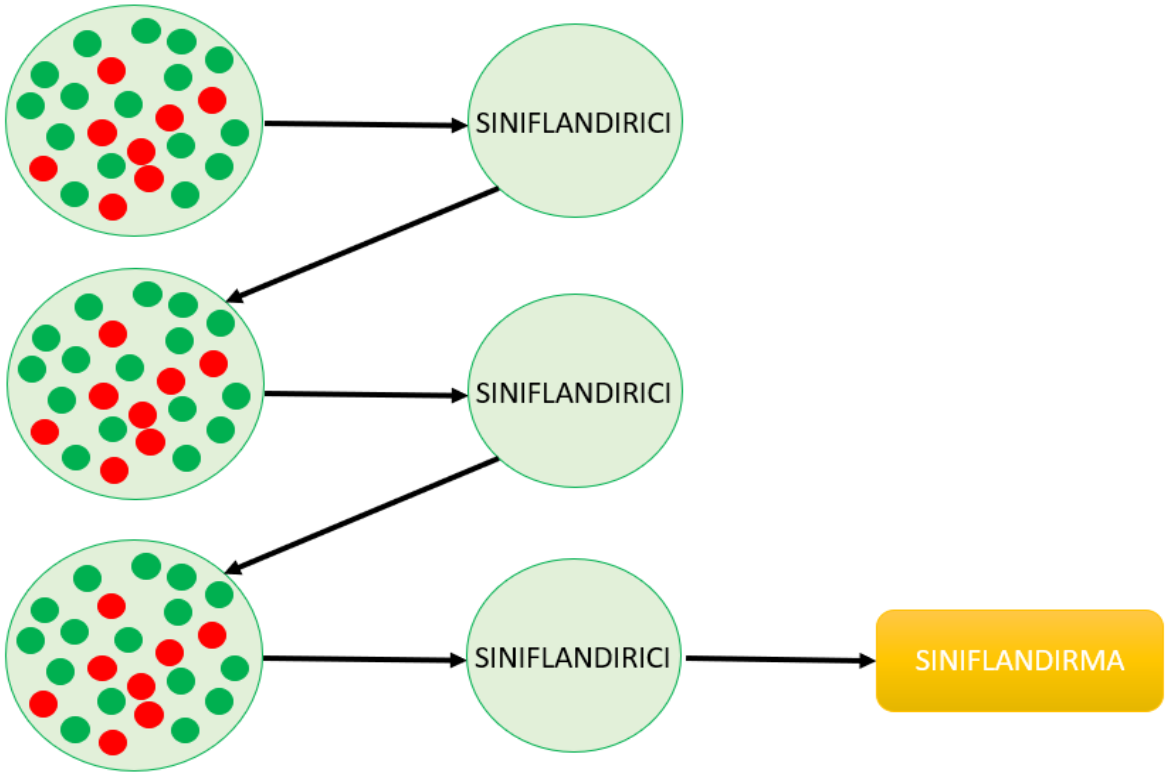
Şekil 4.6'da derinliği üç olan bir karar ağacı görülmektedir. Yeni bir örneklemin sınıflandırılması gerektiği durumda, sırasıyla önce kök düğümdeki koşula, daha sonra düğümdeki koşula ve nihai olarak yapraktaki koşula göre sınıflandırma gerçekleştirilmektedir.

4.1.3. Toplu Öğrenme

Toplu öğrenme, temel makine öğrenme algoritmalarının farklı yöntemler altında birlikte kullanılması suretiyle, daha etkin sonuçlar almayı amaçlayan yöntemlerdir. Toplu öğrenme yöntemleri, boosting (yükseltme), bagging (torbalama) ve stacking (istifleme) olmak üzere üç ana başlıkta incelenmiştir.

Yükseltme

Yükseltme yöntemi, temel sınıflandırıcılar kullanılarak daha güçlü bir sınıflandırıcı ortaya konması yaklaşımını ifade etmektedir. Yükseltme algoritması, farklı şekilde kümelenmiş ya da ağırlıklandırılmış eğitim setleri kullanılarak eğitilen temel sınıflandırıcıların tahmin yapması suretiyle uygulanmaktadır [66]. Çalışmada, Adaptif Yükseltme (AdaB), Gradyan Yükseltme (GB), Kategorik Yükseltme (CATB) ve Ekstra Gradyan Yükseltme (XGB) algoritmaları kullanılmıştır. Yükseltme yaklaşımının genel modeli şekil 4.7'de sunulmaktadır.



Şekil 4.7. Yükseltme yöntemi ile sınıflandırma gösterimi

Şekil 4.7'de örnek modeli sunulan yükseltme tabanlı modellerde, birden fazla zayıf öğrenici (sınıflandırıcı) bulunmaktadır. Her zayıf öğrenici aynı veriyle ve buna ek olarak kendisinden önce gelen zayıf öğrenicilerin hatalarıyla eğitilmektedir. Nihai olarak, zayıf öğrenicilerin hepsinden daha güçlü sınıflandırma yapan bir güçlü sınıflandırıcı elde edilmektedir.

Adaptif yükseltme (ADAB)

En iyi performansa sahip yükseltme algoritmalarından biri olan AdaBoost algoritması, sağlam teorik temelleri olan ve pratik uygulamalarda çok iyi sonuçlar veren bir algoritmadır. AdaBoost, zayıf bir öğrenme algoritmasını, bu algoritmanın tasarımına yeni bir yöntem getirerek, rastgele tahminden daha iyi bir doğrulukla, güçlü bir öğrenme algoritmasına dönüştürebilmektedir [67].

Gradyan yükseltme (GB)

Gradyan yükseltme, birden fazla sıralı zayıf öğrenicinin, bir öğrenici kendinden önceki öğrenicinin hatalarından öğrenerek performansını iyileştirmesi suretiyle tahmin

gerçekleştiren bir algoritmadır. Zayıf öğreniciler arasındaki ilişki, kullanılan bir kayıp algoritmasıyla sağlanmaktadır [68].

Ekstra gradyan yükseltme (XGBoost)

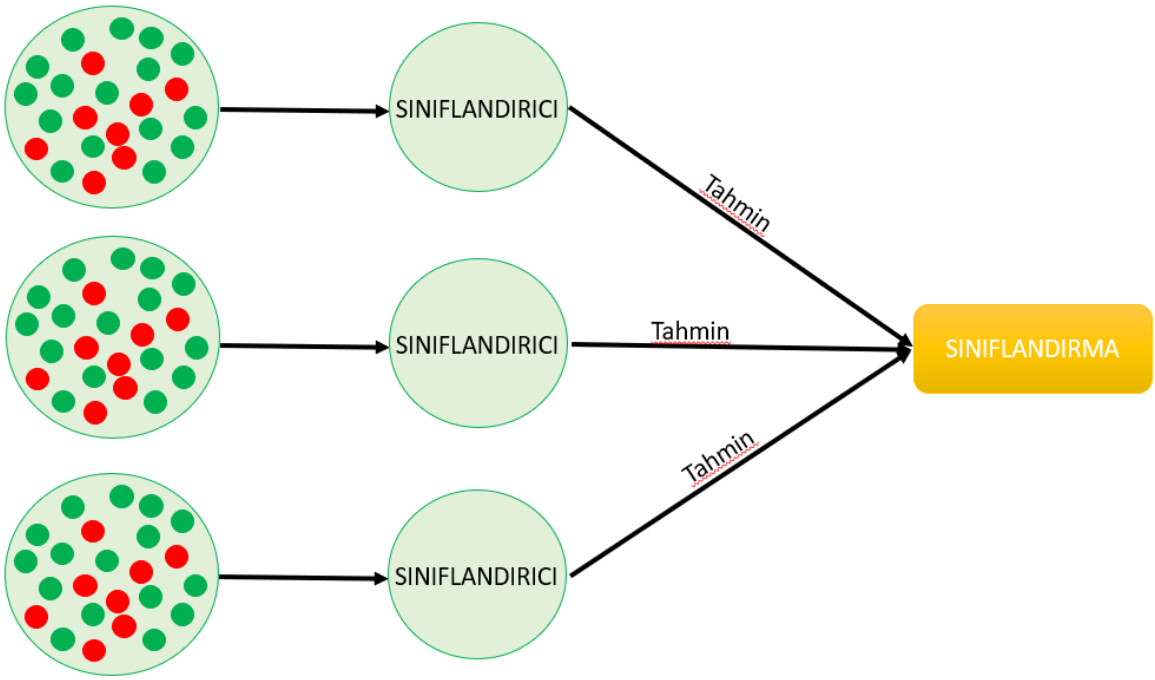
Ağaç temelli ölçeklenebilir bir yükseltme algoritması olan XGB, farklı işletim sistemlerinde paralel işlem gerçekleştirebilme özelliğiyle çok hızlı tahmin gerçekleştirebilen bir toplu öğrenme algoritmasıdır [69].

Kategorik yükseltme (CatBoost)

Klasik yükseltme algoritmalarından farklı olarak sıralı yükseltme ve permütasyon tabanlı yükseltme kullanan CatBoost, kategorik özelliklerin işlenmesi açısından yenilikçi bir algoritmadır [70].

Torbalama

Önyükleme ve bütünleme işlemlerini birleştiren torbalama, bir çeşit paralel toplu öğrenme yaklaşımı sunmaktadır. İkili ve çoklu sınıflandırmalar için kullanılabilen torbalama yaklaşımı, sınıflandırmalar için oylama, tahmin için ortalama yöntemleriyle sonuç elde etmektedir. Çalışmada, bu gruptaki algoritmalar arasında rastgele orman algoritması kullanılmıştır [66]. Torbalama yaklaşımının örnek modeli şekil 4.8'de sunulmaktadır.



Şekil 4.8. Torbalama yöntemi ile sınıflandırma

Şekil 4.8'de bir örneği sunulan torbalama yönteminde, farklı sınıflandırıcılar aynı veri seti kullanılarak eğitilmektedirler. Yeni bir örneklemin sınıflandırılması gerektiğinde, sınıflandırıcılar ayrı ayrı tahmin yapmaktadırlar. Sınıflandırıcılar tarafından ayrı ayrı yapılan tahminler, oy çokluğu, ağırlıklı oylama gibi yöntemlerle birleştirilerek nihai sınıflandırma gerçekleştirilmektedir.

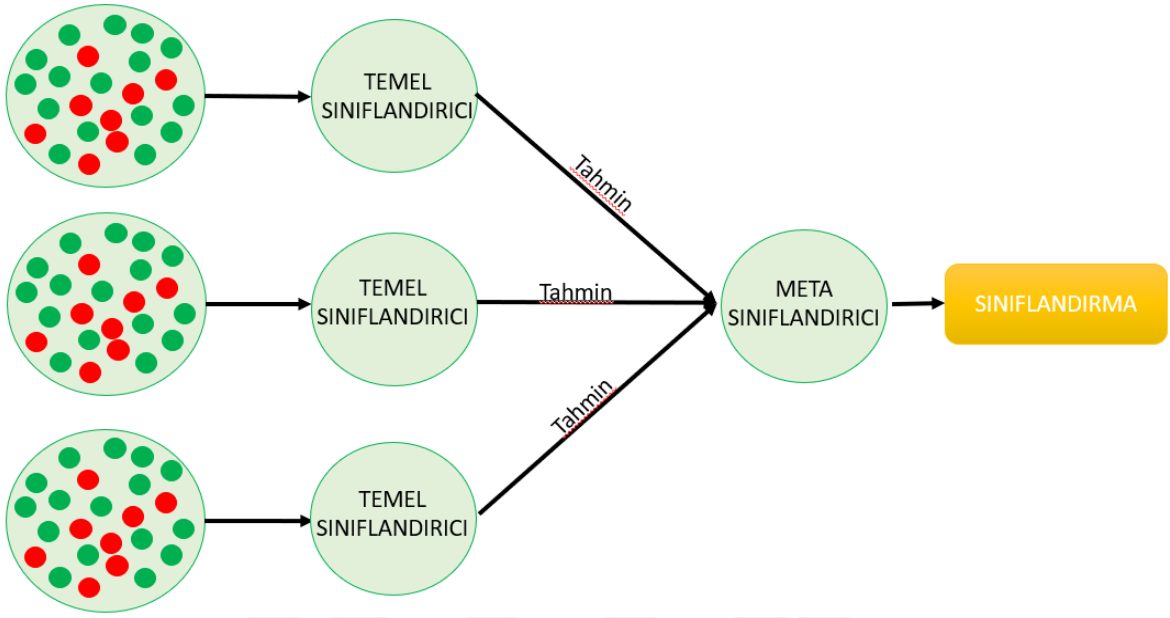
Rastgele orman (RF)

Basit ve etkili bir makine öğrenmesi algoritması olan rastgele orman algoritması, zararlı yazılım tespiti ve sınıflandırılmasında yaygın olarak kullanılmaktadır. Rastgele orman algoritması, çok sayıda alt sınıflandırıcıdan oluşan bir meta sınıflandırıcıdır. Bir girdinin sınıflandırılması, alt sınıflandırıcılar (rastgele ağaç) tarafından yapılan tahminden faydalanılarak meta sınıflandırıcı tarafından gerçekleştirilmektedir [71].

<u>İstifleme</u>

İstifleme yönteminde, bağımsız öğreniciler bir meta öğrenici tarafından birleştirilmektedir. İstiflemedeki temel yaklaşım, bağımsız öğrenicilerin eğitilmesi suretiyle meta öğrenici tarafından işlenecek yeni bir veri oluşturmaktır. Meta öğreniciler, bağımsız öğreniciler

tarafından oluşturulan bu yeni veriyi kullanarak sınıflandırma gerçekleştirmektedirler [66]. İstifleme yaklaşımının örnek modeli şekil 4.9'da sunulmaktadır.



Şekil 4.9. İstifleme yöntemi ile sınıflandırma

Şekil 4.9'da örnek modeli sunulan istifleme yönteminde iki aşama sınıflandırıcı kullanılmaktadır. Birinci aşama sınıflandırıcılar temel sınıflandırıcılar olarak adlandırılmakta ve bu aşamadaki tüm sınıflandırıcılar aynı veri ile tahmin gerçekleştirmektedir. İkinci aşamada yer alan sınıflandırıcı, meta sınıflandırıcı olarak tanımlanmakta ve ilk aşamada eğitilen sınıflandırıcıların çıktıları ikinci aşama sınıflandırıcıya girdi olarak verilmektedir. Meta sınıflandırıcı tarafından yapılan sınıflandırma nihai sınıflandırma olarak kabul edilmektedir.

4.2. Metrikler

Makine öğrenme modellerinin etkinliklerinin değerlendirilebilmesi için çeşitli metrikler kullanılmaktadır. Önerilen tespit yaklaşımında, ikili sınıflandırma yapan algoritmaların etkinliğinin değerlendirilmesi için, doğruluk ve f1 skoru metrikleri kullanılmıştır.

Karmaşıklık Matrisi

Karmaşıklık matrisi bir metrik olmamakla birlikte, metriklerin daha iyi anlaşılması maksadıyla çalışmada kullanılmıştır. Karmaşıklık matrisi şekil 4.10'da sunulmaktadır.

Şekil 4.10. Karmaşıklık matrisi

İkili sınıflandırmalar için kullanılan karmaşıklık matrisi (Şekil 4.10), 2x2'lik bir matris olup, doğru-pozitif (DP), doğru-negatif (DN), yanlış-pozitif (YP) ve yanlış-negatif (YN) değerlerinden oluşmaktadır.

Doğruluk (Accuracy)

Doğruluk, yapılan doğru tahminlerin tüm tahminlere oranını ifade eden metriktir. Doğruluk hesaplama formülü eşitlik 4.1'de sunulmaktadır.

$$\text{Doğruluk} = (DP+DN) / (DP+DN+YP+YN) \quad (4.1)$$

F1 Skoru (F1 Score)

F1 skoru kesinlik ve duyarlılık metriklerinin harmonik ortalamasının hesaplanması suretiyle elde edilen metriktir. Duyarlılık hesaplama formülü eşitlik 4.2'de, kesinlik hesaplama formülü eşitlik 4.3'te, f1 skoru hesaplama formülü eşitlik 4.4'te sunulmaktadır.

$$\text{Duyarlılık} = DP/ (DP + YN) \quad (4.2)$$

$$\text{Kesinlik} = DP/ (DP + YP) \quad (4.3)$$

$$\text{F1 Skoru} = (2 * \text{Duyarlılık} * \text{Kesinlik}) / (\text{Duyarlılık} + \text{Kesinlik}) \quad (4.4)$$

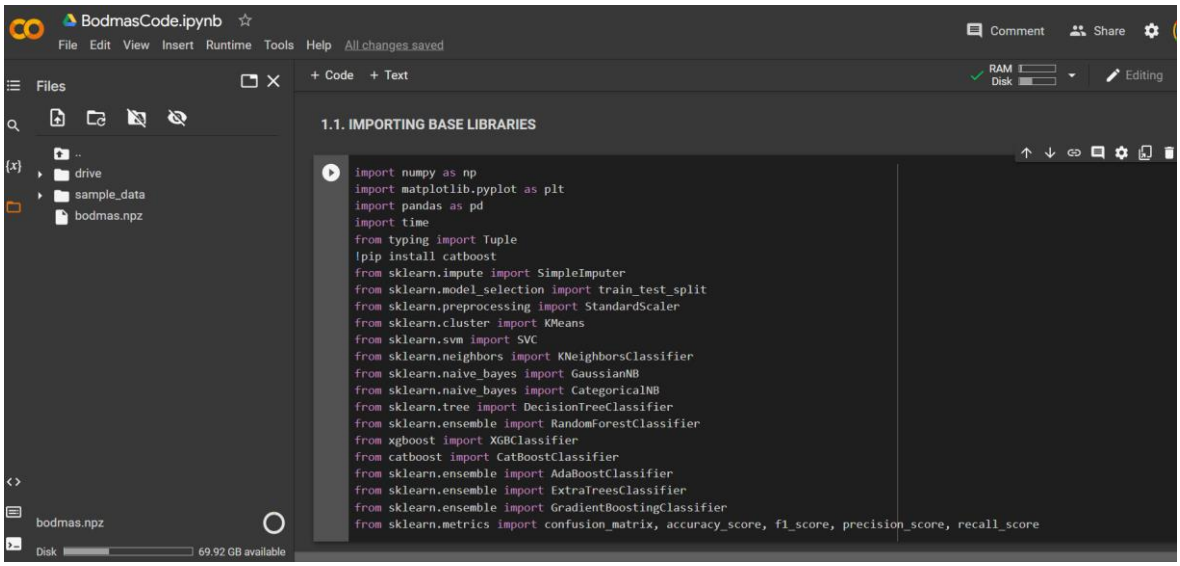
5. KULLANILAN ARAÇLAR

Modelin ortaya konması ve uygulanması kapsamında kullanılan kaynaklar müteakip alt başlıklar altında sunulmaktadır.

5.1. Google Colaboratory

Colab olarak da bilinen Google Colaboratory [72], Google firması tarafından sunulan bir web uygulamasıdır. Bir tarayıcı aracılığıyla, Python kodlarının çalıştırılmasına imkân sağlayan Colab, makine öğrenmesi, veri analizi ve eğitim amaçlarıyla yaygın olarak kullanılmaktadır.

Python'ın çeşitli versiyonlarını ve kütüphanelerinin büyük kısmını destekleyen uygulama, önerilen tespit yaklaşımının geliştirilmesinde kullanılmıştır. İçerdiği hazır kütüphaneler ve donanımsal kaynaklardan etkin bir şekilde istifade edilmiştir. Resim. 5.1'de BODMAS veri setinin eğitilmesi kapsamında ihtiyaç duyulan kütüphanelerin eklenmesi aşamasının görüntüsü yer almaktadır.



Resim 5.1. Colaboratory ortamında ihtiyaç duyulan kütüphanelerin yüklenmesi

5.2. Windows 8.1

Uygulama aşamasında ihtiyaç duyulan faydalı dosyaların bir kısmı sanal ortamda kurulan Windows 8.1 işletim sistemine sahip bir istemcide yer alan dosyalardan alınmıştır. Windows 8.1 işletim sisteminin ana akım desteği 2018 yılı itibariyle tamamlanmıştır. Diğer taraftan, uygulamanın geliştirilmesi kapsamında kullanılan zararlı yazılımlar, Windows 8.1 işletim sisteminin aktif olarak kullanıldığı bir dönemde toplanmıştır. Bu sebeple, çalışmada faydalı yazılım olarak, Windows 8.1 işletim sisteminin sistem dosyaları kullanılmıştır.

5.3. Windows 10

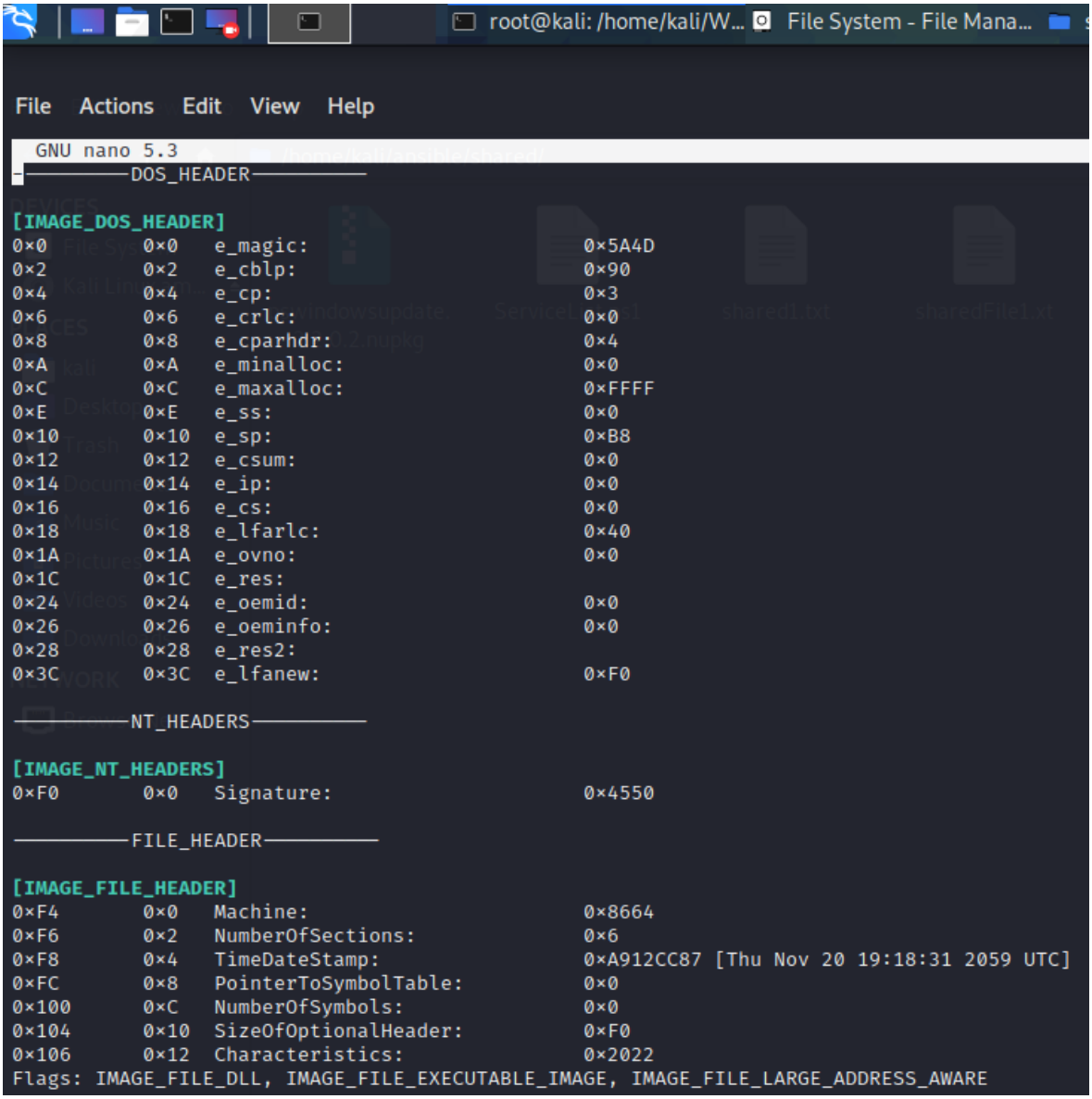
Modelin geliştirilmesi ve uygulanmasına yönelik çalışmalar, HP Pavilion (Intel® Core™ i7-8750H 2,20 GHZ CPU, 16 GB RAM, 64-bit İşletim Sistemi) üzerinde çalışan Windows 10 işletim sistemi ve bu işletim sisteminde çalışan VMWare Workstation sanallaştırma uygulaması üzerine kurulmuş sanal makineler kullanılarak gerçekleştirilmektedir.

5.4. Kali Linux

Kali Linux, temel olarak sızma testi ve güvenlik denetlemelerinde kullanılmak üzere geliştirilmiş, Debian tabanlı, açık kaynak bir Linux dağıtımıdır. İçinde bulundurduğu hazır araçlar sayesinde sızma testi, güvenlik araştırmaları, adli bilişim ve tersine mühendisliğe imkân tanımaktadır.

Tespit modelinin oluşturulması ve etkinliğinin ortaya konması için Windows PE dosyaları kullanılmıştır. Kullanılan zararlıların işletim sistemine zarar vermemesi maksadıyla, modelin gerçek ortamda uygulanmasına yönelik çalışmalar VM Workstation içerisinde oluşturulan sanal Kali Linux 2021.1 istemcisi üzerinde gerçekleştirilmektedir.

Kali Linux ortamında incelenen bir dosyanın PE başlık bilgileri Resim 5.2’de sunulmaktadır.



Resim 5.2. Kali Linux ortamında dosya PE başlık bilgilerinin incelenmesi

Resim 5.2 incelendiğinde, DOS_HEADER, NT_HEADERS ve FILE_HEADER bölümleri ile bu bölümlerin içinde yer alan ve ilgili yazılımın özelliklerini gösteren çeşitli değerler görülmektedir. FILE_HEADER bölümü altında yer alan ve ekranda görünen “Machine”, “NumberOfSections”, “NumberOfSymbols”, “SizeOfOptionalHeader” ve “Characteristics” değerleri özgün veri seti ile gerçekleştirilen çalışmalarda kullanılmıştır.

5.5. Python

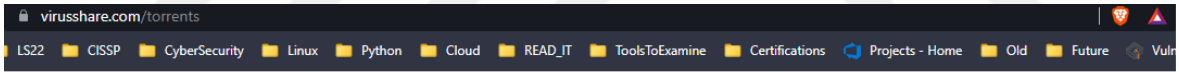
Python programlama dili veri bilimi alanında en yaygın olarak kullanılan, yorumlayıcı kullanan, nesne tabanlı, dinamik semantiğe sahip üst düzey bir programlama dilidir.

Dosyalardan veri çıkarımı, veri ön işlemesi, sınıflandırma algoritmalarının eğitilmesi, modelin dışa aktarılarak başka ortamlarda kullanılmasına imkân vermesi gibi konularda birçok hazır kütüphanesi bulunmaktadır.

Çalışmanın hem model geliştirilmesi aşamasında hem de uygulama aşamasında etkin bir şekilde kullanılmıştır. Modelin geliştirilmesi ve gerçek dünya uygulamasının gerçekleştirilmesi sürecinde Sci-kit Learn [73], PeFile [74], Numpy, Pandas, Time, Typing, Catboost, XGBoost, Drive, Files, CSV, Math, Sys ve Pickle kütüphaneleri aktif olarak kullanılmıştır.

5.6. Virus Share

Virus Share (Resim 5.3), zararlı yazılım üzerine çalışma gerçekleştiren araştırmacılara, olay müdahalecilere ve adli bilişim analistlerine zararlı yazılım örnekleri sağlayan bir depodur. Platform her ne kadar ücretsiz olsa da üyelik süreci, yapılan bir talep ve bu talebin incelenmesini müteakip olumlu olarak neticelenmesine bağlıdır.



Special Requests		
File	**Size**	**Added**
apilog_2019-07-14.zip	11.16 GB	2019-07-16 09:49:47
Hexacorn sandbox reports.		
VirusShare_Mediyes_000.zip	431.66 MB	2012-07-08 19:26:49
Request for all 'Mediyes' matches. 1875 samples.		
VirusShare_InstallCore_000.zip	6.03 GB	2012-07-14 12:49:23
Request for all 'InstallCore' matches. 11444 samples.		
VirusShare_APT1_293.zip	16.66 MB	2013-03-04 17:27:09
293 of the 1007 samples listed in the Mandiant APT1 report.		
VirusShare_x86-64_WinEXE_20130711.zip	505.78 MB	2013-07-11 22:12:04
Request for all 64-Bit Windows PE files. 997 samples.		
VirusShare_Citadel-Zeus_PE-Arc_20130113-20130712.zip	940.75 MB	2013-07-13 00:47:35
Request for newer Citadel / Zeus binaries and archives discovered during the 6-month period 2013-01-13 to 2013-07-12. 2031 samples.		
VirusShare_ELF_20140617.zip	129.41 MB	2014-06-17 17:25:45
Request for newer ELF binaries. 2778 samples.		
VirusShare_dotNET_20140618	93.32 GB	2014-06-22 14:57:40
Request for all .NET binaries. 220,997 samples.		

Resim 5.3. Virusshare.com web sayfası [75]

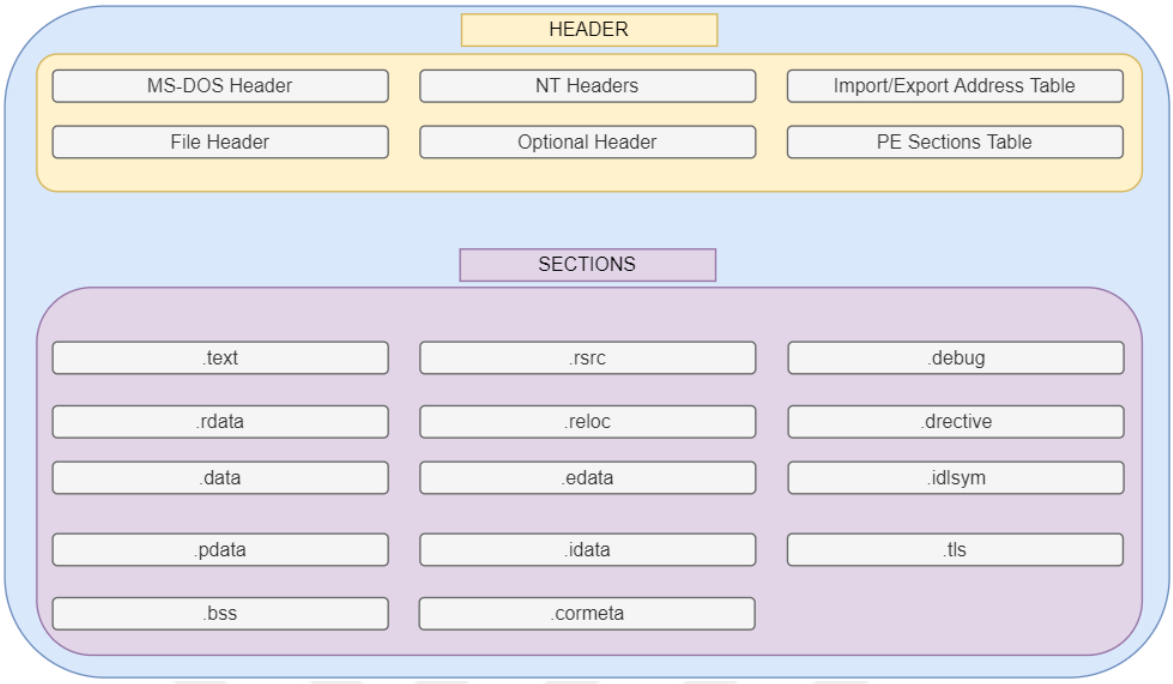
Çalışma kapsamında Virus Share platformuna üye olunmuş ve bu platform üzerinden Windows PE formatındaki 997 zararlı yazılım örneğini içeren VirusShare_x86-64_WinEXE_20130711.zip (Resim 5.3) klasörü edinilmiştir.

5.7. Windows PE Dosyaları

PE dosyaları, 32-bit ve 64-bit Windows işletim sistemlerinde çalıştırılabilir olan, ".acm, .ax, .dll, .efi, .exe, .ocx, .sys" gibi uzantılara sahip dosyalardır. PE dosyalarının veri yapısı, ilgili dosyanın Windows işletim sistemi tarafından çalıştırılabilmesini sağlamaktadır. PE dosyaları, değişen sayılarda başlık ve bölümden oluşmakta olup, bu bölümlerde yer alan değerlerin bir kısmı zararlı yazılım tespit modeline girdi teşkil edecek özellikler olacak şekilde kullanılmıştır.

PE başlıklarında, bir Windows işletim sisteminin, çalıştırılabilir bir kodu yönetmesini sağlayacak bilgiler yer almaktadır. Bu bilgiler, dll referansları, API çağırma tabloları, kaynak yönetim verisi, gibi hususları içerir ve dosyanın bellek üzerine nasıl yerleşeceği de dahil olmak üzere çalışma zamanında ihtiyaç duyulacak dosya özelliklerini içermektedir.

Windows PE dosyasının başlık ve bölümler olmak üzere iki ana kısmı bulunmaktadır. Başlığın altında "MS-DOS Header, NT Header, File Header, Optional Header, PE Sections Table, Import/Export Address Table" bölümleri, bölümler kısmının altında ".data, .rdata, .edata, idata, .pdata .text, .bss, .reloc, .rsrc" alt bölümleri bulunabilmektedir [76]. Windows PE başlık yapısı Şekil 5.1.'te sunulmaktadır.



Şekil 5.1. Windows PE başlık yapısı

Şekil 5.1'de genel yapısı sunulan PE dosyasının alt bölümleri aşağıdaki amaçlarla kullanılmaktadır.

- ".text" bölümüde programın çalıştırılabilir kodu,
- ".rdata" bölümünde hata giderme dizini bilgileri,
- ".data" bölümünde dosya tarafından kullanılacak sabitler,
- ".pdata" bölümünde istisna yönetimi için kullanılan fonksiyon tablo dizisi,
- ".bss" bölümünde tanımlanmamış veriler,
- ".rsrc" bölümünde tüm kaynak dizin tabloları,
- ".reloc" bölümünde imaj dosyaları için relokasyon bilgileri,
- ".edata" bölümünde dışar aktarım bilgileri,
- ".idata" bölümünde içer aktarım bilgileri,
- ".cormeta" bölümünde dosyanın yönetilen kod içeriğinin belirtilmesi için kullanılan metadatalar,
- ".debug" bölümünde derleyici tarafından üretilen hata bilgileri,
- ".drective" bölümünde bir imaj dosyası oluşturulması durumunda silinecek bölüm bilgisi,
- ".idlsym" bölümünde arayüz tanımlama dosyalarına yönelik yapısal istisna yönetim bilgileri,
- ".tls" bölümünde PE dosyaları için statik TLS (Thread local storage) desteklemesi ihtiyaç duyulan bilgiler tutulmaktadır.

6. VERİ SETLERİ

Çalışmada, Kaggle Zararlı Yazılım Tespiti veri seti, BODMAS veri seti ve EMBER 2018 veri seti kullanılmıştır. Ayrıca uygulama aşamasında VirusShare'dan alınan zararlılar ve Windows sistem dosyaları kullanılarak çalışmaya özgü yeni bir veri seti de oluşturulmuştur. Kullanılan veri setleri müteakip maddelerde açıklanmıştır.

6.1. Kaggle Zararlı Yazılım Tespiti Veri Seti

Çalışmada ilk olarak, Kaggle zararlı yazılım tespiti veri seti [77] kullanılmıştır. Kaggle zararlı yazılım veri setinin ilk olarak kullanılmasının amacı, az sayıda özelliğe sahip olmasıdır. Bu sayede modelin tasarlanması aşamasında ihtiyaç duyulan hız sağlanmıştır. Ek olarak, test veri seti üzerinde gerçekleştirilen tahminle, modelin performansı çevrimiçi olarak kontrol edilebilmektedir. Kaggle zararlı yazılım veri setinde hem faydalı hem de zararlı yazılımlardan çıkarılmış özellikler bulunmaktadır. Veri setinde bulunan özellikler sadece ikili sınıflandırmaya imkân vermekte, zararlı yazılım ailelerine ait özellik ya da bilgi bulunmamaktadır. Veri seti, Kaggle web sitesi tarafından düzenlenen bir zararlı yazılım tespit yarışmasında kullanılmak üzere 2018 yılında üretilmiştir. Veri seti hazırlanırken statik analiz yaklaşımı kullanılmıştır. PE özelliklerinin çıkarılması ve dosyaların farklı bölümlerindeki entropinin hesaplanması ile veri seti oluşturulmuştur. Kaggle zararlı yazılım tespiti veri setinde (eğitim seti) bulunan faydalı ve zararlı örneklem sayısı bilgisi Çizelge 6.1'de sunulmaktadır.

Çizelge 6.1. Kaggle veri seti için faydalı ve zararlı örneklem sayısı tablosu

	Faydalı	Zararlı	Toplam
Örneklem Sayısı	75 503	140 849	216 352

Kaggle Zararlı Yazılım Tespiti veri seti, çevrimiçi platformda, eğitim ve test veri setlerine ayrılmış şekilde bulunmaktadır. Çevrimiçi plaformda bulunan test veri setinde sınıf bilgisi yer almamaktadır. Çevrimiçi test veri setine yönelik tahminlerin kontrolü, bu tahminlerin çevrimiçi olarak yüklenmesiyle mümkündür. Bu sebeple, tespit yaklaşımının oluşturulması sürecinde eğitim veri seti ikiye bölünerek hem eğitim hem de test seti olarak kullanılmıştır. Tespit yaklaşımının oluşturulmasından sonra, etkinliğinin kontrolü için orijinal test seti ile ilgili sınıflandırma yapılarak doğruluk değeri çevrimiçi platformda hesaplanmıştır.

Kaggle zararlı yazılım tespiti veri setinde 56 özellik bulunmakta ve çalışmada tüm özellikler kullanılmaktadır. Veri setinin sahip olduğu özellikler Çizelge 6.2'de sunulmaktadır.

Çizelge 6.2. Kaggle zararlı yazılım tespiti veri seti özellikleri

ÖZELLİK	AÇIKLAMA
Md5	Dosyanın hash değeri
Machine	Dosyanın hangi işletim sistemine uygun olduğu bilgisi
Size of Optional Header	Opsiyonel başlık boyutu
Characteristics	Dosya özelliklerini içeren değer bitleri
MajorLinkerVersion	Bağlayıcı ana sürüm numarası
MinorLinkerVersion	Bağlayıcı alt sürüm numarası
SizeOfCode	Kod (Metin) bölümünün boyutu
SizeOfInıtializedData	Başlatılan veri bölümünün boyutu veya birden çok veri bölümü varsa bu tür bölümlerin toplam boyutu
SizeOfUninitializedData	Başlatılmamış veri bölümünün (BSS) boyutu veya birden fazla BSS bölümü varsa bu tür tüm bölümlerin toplamı
AddressOfEntryPoint	Yürütülebilir dosya belleğe yüklendiğinde, görüntü tabanına göre giriş noktasının adresi
BaseOfCode	Belleğe yüklendiğinde kodun başlangıcı bölümünün imaj tabanına göreli adresi
BaseOfData	Belleğe yüklendiğinde, veri başlangıcı bölümünün imaj tabanına göreli adresi
ImageBase	Belleğe yüklendiğinde görüntünün ilk baytının tercih edilen adresi
SectionAlignment	Bölümlerin belleğe yüklendiğinde hizalanması (bayt cinsinden)
FileAlignment	İmaj dosyasındaki bölümlerin ham verilerini hizalamak için kullanılan hizalama faktörü (bayt cinsinden)
MajorOperatingSystemVersion	Gerekli işletim sisteminin ana sürüm numarası
MinorOperatingSystemVersion	Gerekli işletim sisteminin alt sürüm numarası
MajorImageVersion	İmajın ana sürüm numarası
MinorImageVersion	İmajın alt sürüm numarası
MajorSubsystemVersion	Alt sistemin ana sürüm numarası
MinorSubsystemVersion	Alt sistemin alt sürüm numarası
SizeOfImage	İmaj belleğe yüklenirken tüm başlıklar dahil imajın boyutu (bayt cinsinden)
SizeOfHeaders	Bir MS-DOS saplamasının, PE başlığının ve bölüm başlıklarının birleştirilmiş boyutu
CheckSum	İmaj dosyası sağlama toplamı
Subsystem	Bu imajı çalıştırmak için gerekli olan alt sistem.
DllCharacteristics	Dll karakteristikleri
SizeOfStackReserve	Ayrılacak yığının boyutu
SizeOfStackCommit	İşlenecek yığının boyutu

Çizelge 6.2. (devam) Kaggle zararlı yazılım tespiti veri seti özellikleri

ÖZELLİK	AÇIKLAMA
SizeOfHeapCommit	İşlenecek yerel yığın alanının boyutu
LoaderFlags	Yükleme bilgileri için ayrılmış değerler
NumberOfRvaAndSizes	İsteğe bağlı başlığın geri kalanındaki veri dizini girişlerinin sayısı
SizeOfHeapReserve	Ayrılacak yerel yığın alanının boyutu
SectionsNb	Bölüm sayısı
SectionsMeanEntropy	Bölümlerin ortalama entropisi
SectionsMinEntropy	En düşük bölüm entropisi
SectionsMaxEntropy	En yüksek bölüm entropisi
SectionsMeanRawsize	Bölümlerin ortalama ham boyutu
SectionMinRawsize	En düşük bölüm ham boyutu
SectionMaxRawSize	En yüksek bölüm ham boyutu
SectionsMeanVirtualsize	Bölümlerin ortalama sanal boyutu
SectionsMinVirtualsize	En düşük bölüm ham sanal boyutu
SectionMaxVirtualsize	En yüksek bölüm ham sanal boyutu
ImportsNbDLL	İçe aktarılan DLL sayısı
ImportsNb	İçe aktarım sayısı
ImportsNbOrdinal	İçe aktarılan Ordinal sayısı
ExportNb	Dışa aktarım sayısı
ResourcesNb	Kaynak sayısı
ResourcesMeanEntropy	Kaynakların ortalama entropisi
ResourcesMinEntropy	En düşük kaynak entropisi
ResourcesMaxEntropy	En yüksek kaynak entropisi
ResourcesMeanSize	Oralama kaynak boyutu
ResourcesMinSize	En düşük kaynak boyutu
ResourcesMaxSize	En yüksek kaynak boyutu
LoadConfigurationSize	Yüklenen yapılandırma bilgisi boyutu
VersionInformationSize	Versiyon bilgisi boyutu
Legitimate	Sınıf

6.2. BODMAS Veri Seti

BODMAS zararlı yazılım veri seti [78], PE zararlı yazılım veri setlerindeki zaman damgalı zararlı örneklem eksikliği ve iyi düzenlenmiş bir zararlı aile bilgisinin bulunmaması eksikliklerinden hareketle oluşturmuştur. Bu veri seti, her bir örneklem için 2381 özelliğe sahip güncel bir veri setidir. Veri seti oluşturulurken kullanılan örneklemler Ağustos 2019 - Eylül 2020 tarihleri arasında toplanmıştır [79]. Veri seti herkes tarafından erişilebilir olarak bulunmaktadır. BODMAS veri setinde bulunan faydalı ve zararlı örneklem sayısı bilgisi çizelge 6.3'te sunulmaktadır.

Çizelge 6.3. BODMAS veri seti için faydalı ve zararlı örneklem sayısı tablosu

	Faydalı	Zararlı	Toplam
Örneklem Sayısı	77 142	57 293	134 435

LIEF projesi, ELF, PE ve MachO formatlarındaki çalıştırılabilir dosyaların farklı platformlarda ayrıştırılmasını amaçlayan bir proje ve bu projeye bağlı üretilen kodlar bütünüdür. BODMAS veri setini oluşturma sürecinde özellik çıkarımı yapılırken LIEF projesinden faydalanılmıştır [80].

BODMAS veri seti üzerinde yapılan çalışmada özellik seçmek maksadıyla Random Forest algoritması kullanılmaktadır. Random Forest algoritması kullanılarak özelliklerin önem derecesi elde edilmektedir. Veri setinde mevcut olan 2381 özellikten 448 adedi seçilerek kullanılmaktadır. Çalışmada kullanılan özellikler çizelge 6.4'te sunulmaktadır.

Çizelge 6.4. BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
ByteHistorgram-0	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-1	
ByteHistorgram-2	
ByteHistorgram-3	
ByteHistorgram-4	
ByteHistorgram-5	
ByteHistorgram-6	
ByteHistorgram-7	
ByteHistorgram-8	
ByteHistorgram-9	
ByteHistorgram-11	
ByteHistorgram-12	
ByteHistorgram-13	
ByteHistorgram-14	
ByteHistorgram-16	
ByteHistorgram-18	
ByteHistorgram-19	
ByteHistorgram-20	
ByteHistorgram-21	
ByteHistorgram-22	
ByteHistorgram-27	
ByteHistorgram-32	
ByteHistorgram-33	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
ByteHistorgram-35	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-36	
ByteHistorgram-38	
ByteHistorgram-40	
ByteHistorgram-41	
ByteHistorgram-43	
ByteHistorgram-44	
ByteHistorgram-46	
ByteHistorgram-47	
ByteHistorgram-48	
ByteHistorgram-49	
ByteHistorgram-50	
ByteHistorgram-52	
ByteHistorgram-53	
ByteHistorgram-54	
ByteHistorgram-55	
ByteHistorgram-56	
ByteHistorgram-57	
ByteHistorgram-60	
ByteHistorgram-61	
ByteHistorgram-62	
ByteHistorgram-64	
ByteHistorgram-65	
ByteHistorgram-67	
ByteHistorgram-68	
ByteHistorgram-70	
ByteHistorgram-71	
ByteHistorgram-72	
ByteHistorgram-73	
ByteHistorgram-74	
ByteHistorgram-76	
ByteHistorgram-78	
ByteHistorgram-83	
ByteHistorgram-87	
ByteHistorgram-90	
ByteHistorgram-92	
ByteHistorgram-94	
ByteHistorgram-95	
ByteHistorgram-98	
ByteHistorgram-101	
ByteHistorgram-103	
ByteHistorgram-105	
ByteHistorgram-106	
ByteHistorgram-107	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
ByteHistorgram-108	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-109	
ByteHistorgram-110	
ByteHistorgram-111	
ByteHistorgram-112	
ByteHistorgram-113	
ByteHistorgram-114	
ByteHistorgram-115	
ByteHistorgram-116	
ByteHistorgram-117	
ByteHistorgram-121	
ByteHistorgram-122	
ByteHistorgram-125	
ByteHistorgram-129	
ByteHistorgram-131	
ByteHistorgram-133	
ByteHistorgram-134	
ByteHistorgram-135	
ByteHistorgram-137	
ByteHistorgram-139	
ByteHistorgram-141	
ByteHistorgram-142	
ByteHistorgram-149	
ByteHistorgram-150	
ByteHistorgram-151	
ByteHistorgram-154	
ByteHistorgram-155	
ByteHistorgram-157	
ByteHistorgram-158	
ByteHistorgram-159	
ByteHistorgram-162	
ByteHistorgram-163	
ByteHistorgram-164	
ByteHistorgram-165	
ByteHistorgram-171	
ByteHistorgram-172	
ByteHistorgram-173	
ByteHistorgram-175	
ByteHistorgram-176	
ByteHistorgram-179	
ByteHistorgram-180	
ByteHistorgram-181	
ByteHistorgram-185	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
ByteHistorgram-186	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-188	
ByteHistorgram-189	
ByteHistorgram-190	
ByteHistorgram-191	
ByteHistorgram-192	
ByteHistorgram-193	
ByteHistorgram-194	
ByteHistorgram-195	
ByteHistorgram-196	
ByteHistorgram-199	
ByteHistorgram-200	
ByteHistorgram-201	
ByteHistorgram-202	
ByteHistorgram-203	
ByteHistorgram-204	
ByteHistorgram-207	
ByteHistorgram-209	
ByteHistorgram-211	
ByteHistorgram-212	
ByteHistorgram-214	
ByteHistorgram-215	
ByteHistorgram-216	
ByteHistorgram-222	
ByteHistorgram-223	
ByteHistorgram-224	
ByteHistorgram-225	
ByteHistorgram-227	
ByteHistorgram-229	
ByteHistorgram-231	
ByteHistorgram-232	
ByteHistorgram-235	
ByteHistorgram-236	
ByteHistorgram-238	
ByteHistorgram-239	
ByteHistorgram-240	
ByteHistorgram-242	
ByteHistorgram-245	
ByteHistorgram-247	
ByteHistorgram-248	
ByteHistorgram-249	
ByteHistorgram-251	
ByteHistorgram-252	
ByteHistorgram-253	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
ByteHistorgram-255	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteEntropyHistorgram-0	0-256 arası belirtilen numaralar için tekrar eden bayt sayısına göre entropi değeri
ByteEntropyHistorgram-50	
ByteEntropyHistorgram-66	
ByteEntropyHistorgram-72	
ByteEntropyHistorgram-78	
ByteEntropyHistorgram-96	
ByteEntropyHistorgram-112	
ByteEntropyHistorgram-114	
ByteEntropyHistorgram-115	
ByteEntropyHistorgram-116	
ByteEntropyHistorgram-119	
ByteEntropyHistorgram-130	
ByteEntropyHistorgram-132	
ByteEntropyHistorgram-134	
ByteEntropyHistorgram-135	
ByteEntropyHistorgram-136	
ByteEntropyHistorgram-137	
ByteEntropyHistorgram-138	
ByteEntropyHistorgram-144	
ByteEntropyHistorgram-145	
ByteEntropyHistorgram-146	
ByteEntropyHistorgram-147	
ByteEntropyHistorgram-148	
ByteEntropyHistorgram-149	
ByteEntropyHistorgram-150	
ByteEntropyHistorgram-151	
ByteEntropyHistorgram-155	
ByteEntropyHistorgram-159	
ByteEntropyHistorgram-160	
ByteEntropyHistorgram-166	
ByteEntropyHistorgram-168	
ByteEntropyHistorgram-179	
ByteEntropyHistorgram-192	
ByteEntropyHistorgram-196	
ByteEntropyHistorgram-197	
ByteEntropyHistorgram-199	
ByteEntropyHistorgram-200	
ByteEntropyHistorgram-202	
ByteEntropyHistorgram-203	
ByteEntropyHistorgram-204	
ByteEntropyHistorgram-208	
ByteEntropyHistorgram-209	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
ByteEntropyHistorgram-210	0-256 arası belirtilen numaralar için tekrar eden bayt sayısına göre entropi değeri
ByteEntropyHistorgram-211	
ByteEntropyHistorgram-212	
ByteEntropyHistorgram-213	
ByteEntropyHistorgram-216	
ByteEntropyHistorgram-217	
ByteEntropyHistorgram-218	
ByteEntropyHistorgram-220	
ByteEntropyHistorgram-221	
ByteEntropyHistorgram-222	
ByteEntropyHistorgram-223	
ByteEntropyHistorgram-224	
ByteEntropyHistorgram-225	
ByteEntropyHistorgram-227	
ByteEntropyHistorgram-228	
ByteEntropyHistorgram-230	
ByteEntropyHistorgram-232	
ByteEntropyHistorgram-233	
ByteEntropyHistorgram-234	
ByteEntropyHistorgram-235	
ByteEntropyHistorgram-237	
ByteEntropyHistorgram-238	
ByteEntropyHistorgram-240	
ByteEntropyHistorgram-241	
ByteEntropyHistorgram-242	
ByteEntropyHistorgram-243	
ByteEntropyHistorgram-244	
ByteEntropyHistorgram-245	
ByteEntropyHistorgram-246	
ByteEntropyHistorgram-247	
ByteEntropyHistorgram-248	
ByteEntropyHistorgram-249	
ByteEntropyHistorgram-250	
ByteEntropyHistorgram-251	
ByteEntropyHistorgram-252	
ByteEntropyHistorgram-253	
ByteEntropyHistorgram-254	
ByteEntropyHistorgram-255	
# of consecutive printable strings	Ardışık yazdırılabilir karakter sayısı
avg length of consecutive printable strings	Yazdırılabilir karakter sayılarının ortalama uzunluğu
sum of frequencies of all the printable characters	Tüm yazdırılabilir karakterlerin frekanslarının toplamı
freq of printable characters -	İlgili yazdırılabilir karakterin frekansı

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
freq of printable characters - !	İlgili yazdırılabilir karakterin frekansı
freq of printable characters - "	
freq of printable characters - $	
freq of printable characters - '	
freq of printable characters -)	
freq of printable characters - +	
freq of printable characters - -	
freq of printable characters - .	
freq of printable characters - /	
freq of printable characters - 0	
freq of printable characters - 1	
freq of printable characters - 2	
freq of printable characters - 3	
freq of printable characters - 4	
freq of printable characters - 5	
freq of printable characters - 6	
freq of printable characters - 7	
freq of printable characters - 8	
freq of printable characters - ;	
freq of printable characters - <	
freq of printable characters - =	
freq of printable characters - >	
freq of printable characters - B	
freq of printable characters - C	
freq of printable characters - D	
freq of printable characters - E	
freq of printable characters - F	
freq of printable characters - G	
freq of printable characters - H	
freq of printable characters - I	
freq of printable characters - J	
freq of printable characters - K	
freq of printable characters - L	
freq of printable characters - N	
freq of printable characters - O	
freq of printable characters - P	
freq of printable characters - Q	
freq of printable characters - R	
freq of printable characters - T	
freq of printable characters - U	
freq of printable characters - V	
freq of printable characters - W	
freq of printable characters - Y	
freq of printable characters - Z	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
freq of printable characters - [	İlgili yazdırılabilir karakterin frekansı
freq of printable characters - \	
freq of printable characters -]	
freq of printable characters - ^	
freq of printable characters - _	
freq of printable characters - `	
freq of printable characters - b	
freq of printable characters - c	
freq of printable characters - d	
freq of printable characters - e	
freq of printable characters - g	
freq of printable characters - j	
freq of printable characters - n	
freq of printable characters - o	
freq of printable characters - p	
freq of printable characters - q	
freq of printable characters - t	
freq of printable characters - u	
freq of printable characters - v	
freq of printable characters - w	
freq of printable characters - x	
freq of printable characters - z	
freq of printable characters - {	
freq of printable characters - •	
entropy of the byte histogram	Bayt histogramı entropisi
occurrences of "http://" or "https://". (ignore case)	"http://" ve "https://" görülme sayısı
occurrences of "MZ"	"MZ" görülme sayısı
length of the bytes of the file	Bayt cinsinden dosya boyutu
virtual size of the lief parsed binary	Ayrıştırılmış ikili dosyanın sanal boyutu
whether the binary has a debug section	İkili dosyanın bir hata ayıklama bölümü olup olmaması durumu
# of exported functions	Dışa aktarılan fonksiyon sayısı
# of imported functions	İçe aktarılan fonksiyon sayısı
whether the binary use "Relocation"	Relokasyon kullanılıp kullanılmaması durumu
whether the binary has a "Resources" object	İkili dosyanın "Kaynaklar" nesnesine sahip olup olmaması durumu
whether the binary is signed	İkili dosyanın imzalı olup olmaması durumu
when the file was created	Dosyanın oluşturulma tarihi
hash COFF machine type - 5	İlgili COFF makine tipi için özet değeri
hash COFF machine type - 6	
hash COFF characteristics - 0	İlgili COFF karakteristiği için özet değeri
hash COFF characteristics - 3	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
hash COFF characteristics - 4	İlgili COFF karakteristiği için özet değeri
hash COFF characteristics - 5	
hash COFF characteristics - 6	
hash optional subsystem - 7	İlgili numaraya sahip opsiyonel alt sistemin özet değeri
hash optional subsystem - 8	
hash optional dll_characteristics - 1	İlgili numaralı DLL karakteristik özelliğinin özet değeri
hash optional dll_characteristics - 3	
hash optional dll_characteristics - 8	
hash optional magic number - 1	Opsiyonel dosya tipi numarasının özet değeri
hash optional magic number - 4	
major_image_version	İmaj ana sürüm numarası
major_linker_version	Bağlayıcı ana sürüm numarası
minor_linker_version	Bağlayıcı alt sürüm numarası
major_operating_system_version	İşletim sistemin ama sürüm numarası
major_subsystem_version	Alt sistem sürüm numarası
sizeof_code	Kod boyutu
total # of sections	Toplam bölüm sayısı
# of sections with zero size	"0" boyutlu bölüm sayısı
# of sections with empty name	İsimsiz bölüm sayısı
# of sections with "props" including "MEM_READ" and "MEM_EXECUTE"	"MEM_READ" ve MEM_EXECUTE" proplarına sahip bölüm sayısı
# of sections with "props" including "MEM_WRITE"	"MEM_WRITE" propuna sahip bölüm sayısı
hash on pair of section name and size - 0	İlgili numaralı bölüm ismi ve boyutunun özet değeri
hash on pair of section name and size - 5	
hash on pair of section name and size - 14	
hash on pair of section name and size - 22	
hash on pair of section name and size - 32	
hash on pair of section name and size - 35	
hash on pair of section name and size - 41	
hash on pair of section name and size - 43	
hash on pair of section name and entropy - 5	İlgili numaralı bölüm ismi ve entropisinin özet değeri
hash on pair of section name and entropy - 14	
hash on pair of section name and entropy - 32	
hash on pair of section name and entropy - 35	
hash on pair of section name and entropy - 41	
hash on pair of section name and entropy - 42	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
hash on pair of section name and entropy - 43	İlgili numaralı bölüm isminin ve entropisinin özet değeri
hash on pair of section name and virtual size - 0	İlgili numaralı bölüm isminin ve sanal boyutunun özet değeri
hash on pair of section name and virtual size - 2	
hash on pair of section name and virtual size - 5	
hash on pair of section name and virtual size - 14	
hash on pair of section name and virtual size - 16	
hash on pair of section name and virtual size - 22	
hash on pair of section name and virtual size - 32	
hash on pair of section name and virtual size - 34	
hash on pair of section name and virtual size - 35	
hash on pair of section name and virtual size - 41	
hash on pair of section name and virtual size - 43	
hash on name of entry point - 9	İlgili numaralı giriş noktası adının özet değeri
hash on name of entry point - 31	
hash on name of entry point - 39	
hash on name of entry point - 43	
hash on list of properties of entry point - 13	İlgili numaralı giriş noktası özellik listesinin özet değeri
hash on list of properties of entry point - 23	
hash on list of properties of entry point - 37	
hash on list of properties of entry point - 47	
hash on list of unique imported libraries - 25	
hash on list of unique imported libraries - 30	
hash on list of unique imported libraries - 32	
hash on list of unique imported libraries - 43	
hash on list of unique imported libraries - 117	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
hash on list of library:function - 55	İlgili fonksiyon numarasına sahip kütüphane listesinin özet değeri
hash on list of library:function - 67	
hash on list of library:function - 68	
hash on list of library:function - 74	
hash on list of library:function - 138	
hash on list of library:function - 185	
hash on list of library:function - 212	
hash on list of library:function - 217	
hash on list of library:function - 228	
hash on list of library:function - 232	
hash on list of library:function - 254	
hash on list of library:function - 265	
hash on list of library:function - 317	
hash on list of library:function - 320	
hash on list of library:function - 347	
hash on list of library:function - 359	
hash on list of library:function - 360	
hash on list of library:function - 428	
hash on list of library:function - 446	
hash on list of library:function - 476	
hash on list of library:function - 496	
hash on list of library:function - 542	
hash on list of library:function - 558	
hash on list of library:function - 564	
hash on list of library:function - 619	
hash on list of library:function - 629	
hash on list of library:function - 636	
hash on list of library:function - 652	
hash on list of library:function - 658	
hash on list of library:function - 670	
hash on list of library:function - 678	
hash on list of library:function - 689	
hash on list of library:function - 768	
hash on list of library:function - 785	
hash on list of library:function - 794	
hash on list of library:function - 805	
hash on list of library:function - 821	
hash on list of library:function - 838	
hash on list of library:function - 859	
hash on list of library:function - 876	
hash on list of library:function - 902	
hash on list of library:function - 960	
hash on list of library:function - 1011	

Çizelge 6.4. (devam) BODMAS veri setinden seçilen özellikler

ÖZELLİK	AÇIKLAMA
hash on list of library:function - 1022	İlgili fonksiyon numarasına sahip kütüphane listesinin özet değeri
size of data directories - 0	İlgili numaralı veri dizinlerinin boyutu ve ilgili numaralı veri dizinlerinin sanal boyutu
virtual size of data directories - 0	
size of data directories - 1	
virtual size of data directories - 1	
size of data directories - 2	
virtual size of data directories - 2	
size of data directories - 3	
virtual size of data directories - 3	
size of data directories - 4	
virtual size of data directories - 4	
size of data directories - 5	
virtual size of data directories - 5	
size of data directories - 6	
virtual size of data directories - 6	
virtual size of data directories - 9	
size of data directories - 10	
virtual size of data directories - 10	
size of data directories - 12	
virtual size of data directories - 12	
size of data directories - 14	
virtual size of data directories - 14	

BODMAS veri setinde bulunan zararlılar 581 farklı zararlı ailesine gruplanmaktadır. Veri setinde en sık kullanılan zararlı aileleri, bu ailelerin türleri ve zararlı ailesi ile ilgili açıklamalar çizelge 6.5'te sunulmaktadır.

Çizelge 6.5. BODMAS veri setinde bulunan zararlı aileleri (en çok kullanılan 10 aile)

Aile	Adet	Tür	Açıklama
Sfone [81]	4 729	Solucan	Bu aileye ait zararlılar, bir sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Bulunduğu sistemde yetki kazanmak için kaba kuvvet saldırısı uygulamaktadır. Kendini bilinen dosya adlarıyla paylaşılan ağlarda yaymaya çalışmaktadır.

Çizelge 6.5. (devam) BODMAS veri setinde bulunan zararlı aileleri

Aile	Adet	Tür	Açıklama
Wacatac [82]	4 694	Truva atı	Bu aileye ait zararlılar, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Sisteme çeşitli çalıştırılabilir dosyalar eklemekte ve klasörler oluşturmaktadır. Windows User StartUp klasörüne eklediği bir dosya sayesinde sistem yeniden başladıktan sonra da aktif olarak kalmaktadır.
Upatre [83]	3 901	Truva atı	Bu aileye ait yazılımlar, sistemlere istenmeyen posta iletilerine eklenen kötü amaçlı dosyalar olarak veya kullanıcıları zararlının barındığı kötü amaçlı bir web sitesini ziyaret etmesi suretiyle ulaşmaktadır. Zararlılar, bir sistemi etkiledikten sonra farklı kötücül yazılımların ilgili sisteme indirilmesini sağlamaktadır. Zararlılar tarafından indirilen kötü amaçlı yazılımlardan bazıları ZEUS, CRILOCK, DYREZA ve ROVNIX türevleridir.UPATRE'nin güncel varyantları etkilenen sistemin bilgisayar adı ve işletim sistemi gibi sistem bilgilerini çalabilmektedir.
Wabot [84]	3 673	Truva atı	Bu aileye ait zararlılar, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Zararlı, kendini sistem klasörünün altına kopyalamakta ve çeşitli kötücül yazılımları sisteme indirmektedir. Kayıt defterinde Winlogon klasörü altına eklediği bir kayıt ile süreklilik sağlamaktadır.
Smal [85]	3 339	Truva atı	Bu aileye ait zararlılar, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Kendisini “%Windows%\Explores.exe” dosyası olarak sisteme eklemektedir. Kayıt defterinde “Run” klasörü altına eklediği bir kayıtla sistemdeki sürekliliğini sağlamaktadır.
Ganelp [86]	2 232	Solucan	Bu aileye ait zararlılar, çıkarılabilir sürücüler aracılığıyla yayılmaktadır. Ayrıca, diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak da sistemlere ulaşabilmektedir. Bıraktığı kopyaları yürüten zamanlanmış bir görev eklemek için Windows Görev Zamanlayıcı'yı kullanmaktadır. Kendi kopyalarını çıkarılabilir sürücülere bırakmaktadır. Bu bırakılan kopyalar, dosya adları için söz konusu sürücülerde bulunan klasörlerin adlarını kullanmaktadır.

Çizelge 6.5. (devam) BODMAS veri setinde bulunan zararlı aileleri

Aile	Adet	Tür	Açıklama
Dinwod [87]	2 057	Truva atı	Bu aileye ait zararlılar, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Sistemi enfekte etmeyi müteakip, sisteme çeşitli işlemler eklemektedir. Sistemin yeniden başlatılması durumunda sürekliliği sağlamak için kayıt defterine kendisini başlatacak kayıt eklemektedir. Müteakip zararlı faaliyetler için sisteme çeşitli dosyalar indirmektedir.
Mira [88]	1 960	Solucan	Bu zararlı ailesi, Windows sistemleri etkileyen bir fidye yazılım ailesidir. Sisteme sızmayı müteakip, dosyaları şifreler ve dosya uzantılarını ".mira" olarak değiştirmektedir. Enfekte ettiği her sistem için asimetrik şifreleme algoritmalarıyla bir anahtar oluşturulmaktadır. Dosyaların şifrelerinin çözülmesi için, komuta kontrol merkezinde tutulan şifre çözücü anahtar karşılığında, kurbanlardan fidye talep edilmektedir.
Berbew [89]	1 749	Arka kapı	Bu zararlı ailesi, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Zararlı, öncelikle kendisinin kopyalarını Windows sistem klasörü altına indirmektedir. Daha sonra kayıt defterine yeni kayıt ekleme ve mevcut bazı kayıtların güncellenmesi işlemlerini gerçekleştirmektedir.
Sillyp2p [90]	1 616	Solucan	Bu zararlı ailesi, eposta, çıkarılabilir sürücüler ve P2P uygulamaların paylaşılan klasörleri aracılığıyla yayılmaktadır. Bu aileye mensup zararlılar, etki ettiği sistemin güvenlik ayalarını düşürmekte ve belirli güvenlik yazılımlarını ve hizmetlerini devre dışı bırakmaktadır. Sistem klasörü altına eklediği "Googleuy.exe" dosyası ve kayıt defterinde yaptığı değişiklikle süreklilik sağlamaktadır.

6.3. EMBER 2018 Veri Seti

EMBER zararlı yazılım veri seti [91], Windows işletim sistemlerinde bulunan ya da bu sistemleri hedef alan çalıştırılabilir dosyalar kullanılarak elde edilmiş veri setidir. Özellik çıkarımı aşamasında, BODMAS veri setinde olduğu gibi, LIEF [80] projesinden faydalanılmıştır. EMBER veri setinin üç farklı versiyonu bulunmaktadır. İlk olarak, 2017 ve öncesindeki faydalı ve zararlı dosyalar kullanılarak elde edilen orjinal EMBER 2017 (v1) veri seti yayımlanmıştır. EMBER 2018 veri seti [92], EMBER 2017 veri setinden farklı

olarak 2018 yılında toplanan faydalı ve zararlı örneklemleri de içermektedir. EMBER 2018 veri setiyle, orijinal veri setine göre, makine öğrenmesi algoritmaları tarafından daha zor sınıflandırma yapılması amaçlanmaktadır [93]. Çalışmada, en güncel versiyon olan EMBER 2018 veri seti kullanılmaktadır. EMBER 2018 veri setinde bulunan faydalı, zararlı ve etiketlenmemiş örneklem sayısı bilgileri çizelge 6.6'da sunulmaktadır.

Çizelge 6.6. EMBER 2018 veri seti için örneklem bilgileri tablosu

	Eğitim Veri Seti			Test Veri Seti		Toplam
Etiket	Faydalı	Zararlı	Etiketsiz	Faydalı	Zararlı	
Örneklem Sayısı	300 000	300 000	200 000	100 000	100 000	1 000 000

EMBER 2018 veri seti üzerinde Random Forest algoritması kullanılarak özelliklerin önem derecesi elde edilmekte ve 2381 özellikten 448 adedi seçilerek kullanılmaktadır. Çalışmada kullanılan özellikler çizelge 6.7'de sunulmaktadır.

Çizelge 6.7. EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-0	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-1	
ByteHistorgram-2	
ByteHistorgram-3	
ByteHistorgram-4	
ByteHistorgram-5	
ByteHistorgram-6	
ByteHistorgram-7	
ByteHistorgram-8	
ByteHistorgram-9	
ByteHistorgram-10	
ByteHistorgram-11	
ByteHistorgram-12	
ByteHistorgram-13	
ByteHistorgram-14	
ByteHistorgram-15	
ByteHistorgram-16	
ByteHistorgram-17	
ByteHistorgram-18	
ByteHistorgram-19	
ByteHistorgram-20	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-21	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-22	
ByteHistorgram-23	
ByteHistorgram-24	
ByteHistorgram-25	
ByteHistorgram-26	
ByteHistorgram-27	
ByteHistorgram-28	
ByteHistorgram-29	
ByteHistorgram-30	
ByteHistorgram-31	
ByteHistorgram-32	
ByteHistorgram-33	
ByteHistorgram-34	
ByteHistorgram-35	
ByteHistorgram-36	
ByteHistorgram-37	
ByteHistorgram-38	
ByteHistorgram-39	
ByteHistorgram-40	
ByteHistorgram-41	
ByteHistorgram-42	
ByteHistorgram-43	
ByteHistorgram-44	
ByteHistorgram-45	
ByteHistorgram-46	
ByteHistorgram-47	
ByteHistorgram-48	
ByteHistorgram-49	
ByteHistorgram-50	
ByteHistorgram-51	
ByteHistorgram-52	
ByteHistorgram-53	
ByteHistorgram-54	
ByteHistorgram-55	
ByteHistorgram-56	
ByteHistorgram-57	
ByteHistorgram-58	
ByteHistorgram-59	
ByteHistorgram-60	
ByteHistorgram-61	
ByteHistorgram-62	
ByteHistorgram-63	
ByteHistorgram-64	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-65	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-66	
ByteHistorgram-67	
ByteHistorgram-68	
ByteHistorgram-69	
ByteHistorgram-70	
ByteHistorgram-71	
ByteHistorgram-72	
ByteHistorgram-73	
ByteHistorgram-74	
ByteHistorgram-75	
ByteHistorgram-76	
ByteHistorgram-77	
ByteHistorgram-78	
ByteHistorgram-79	
ByteHistorgram-80	
ByteHistorgram-81	
ByteHistorgram-82	
ByteHistorgram-83	
ByteHistorgram-84	
ByteHistorgram-85	
ByteHistorgram-86	
ByteHistorgram-87	
ByteHistorgram-88	
ByteHistorgram-89	
ByteHistorgram-90	
ByteHistorgram-91	
ByteHistorgram-92	
ByteHistorgram-93	
ByteHistorgram-94	
ByteHistorgram-95	
ByteHistorgram-96	
ByteHistorgram-97	
ByteHistorgram-98	
ByteHistorgram-99	
ByteHistorgram-100	
ByteHistorgram-101	
ByteHistorgram-102	
ByteHistorgram-103	
ByteHistorgram-104	
ByteHistorgram-105	
ByteHistorgram-106	
ByteHistorgram-107	
ByteHistorgram-108	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-109	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-110	
ByteHistorgram-111	
ByteHistorgram-112	
ByteHistorgram-113	
ByteHistorgram-114	
ByteHistorgram-115	
ByteHistorgram-116	
ByteHistorgram-117	
ByteHistorgram-118	
ByteHistorgram-119	
ByteHistorgram-120	
ByteHistorgram-121	
ByteHistorgram-122	
ByteHistorgram-123	
ByteHistorgram-124	
ByteHistorgram-125	
ByteHistorgram-126	
ByteHistorgram-127	
ByteHistorgram-128	
ByteHistorgram-129	
ByteHistorgram-130	
ByteHistorgram-131	
ByteHistorgram-132	
ByteHistorgram-133	
ByteHistorgram-134	
ByteHistorgram-135	
ByteHistorgram-136	
ByteHistorgram-137	
ByteHistorgram-138	
ByteHistorgram-139	
ByteHistorgram-140	
ByteHistorgram-141	
ByteHistorgram-142	
ByteHistorgram-143	
ByteHistorgram-144	
ByteHistorgram-145	
ByteHistorgram-146	
ByteHistorgram-147	
ByteHistorgram-148	
ByteHistorgram-149	
ByteHistorgram-150	
ByteHistorgram-151	
ByteHistorgram-152	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-153	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-154	
ByteHistorgram-155	
ByteHistorgram-156	
ByteHistorgram-157	
ByteHistorgram-158	
ByteHistorgram-159	
ByteHistorgram-160	
ByteHistorgram-161	
ByteHistorgram-162	
ByteHistorgram-163	
ByteHistorgram-164	
ByteHistorgram-165	
ByteHistorgram-166	
ByteHistorgram-167	
ByteHistorgram-168	
ByteHistorgram-169	
ByteHistorgram-170	
ByteHistorgram-171	
ByteHistorgram-172	
ByteHistorgram-173	
ByteHistorgram-174	
ByteHistorgram-175	
ByteHistorgram-176	
ByteHistorgram-177	
ByteHistorgram-178	
ByteHistorgram-179	
ByteHistorgram-180	
ByteHistorgram-181	
ByteHistorgram-182	
ByteHistorgram-183	
ByteHistorgram-184	
ByteHistorgram-185	
ByteHistorgram-186	
ByteHistorgram-187	
ByteHistorgram-188	
ByteHistorgram-189	
ByteHistorgram-190	
ByteHistorgram-191	
ByteHistorgram-192	
ByteHistorgram-193	
ByteHistorgram-194	
ByteHistorgram-195	
ByteHistorgram-196	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-197	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-198	
ByteHistorgram-199	
ByteHistorgram-200	
ByteHistorgram-201	
ByteHistorgram-202	
ByteHistorgram-203	
ByteHistorgram-204	
ByteHistorgram-205	
ByteHistorgram-206	
ByteHistorgram-207	
ByteHistorgram-208	
ByteHistorgram-209	
ByteHistorgram-210	
ByteHistorgram-211	
ByteHistorgram-212	
ByteHistorgram-213	
ByteHistorgram-214	
ByteHistorgram-215	
ByteHistorgram-216	
ByteHistorgram-217	
ByteHistorgram-218	
ByteHistorgram-219	
ByteHistorgram-220	
ByteHistorgram-221	
ByteHistorgram-222	
ByteHistorgram-223	
ByteHistorgram-224	
ByteHistorgram-225	
ByteHistorgram-226	
ByteHistorgram-227	
ByteHistorgram-228	
ByteHistorgram-229	
ByteHistorgram-230	
ByteHistorgram-231	
ByteHistorgram-232	
ByteHistorgram-233	
ByteHistorgram-234	
ByteHistorgram-235	
ByteHistorgram-236	
ByteHistorgram-237	
ByteHistorgram-238	
ByteHistorgram-239	
ByteHistorgram-240	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteHistorgram-241	0-256 arası belirtilen numaralar için tekrar eden ilgili bayt değeri sayısı
ByteHistorgram-242	
ByteHistorgram-243	
ByteHistorgram-244	
ByteHistorgram-245	
ByteHistorgram-246	
ByteHistorgram-247	
ByteHistorgram-248	
ByteHistorgram-249	
ByteHistorgram-250	
ByteHistorgram-251	
ByteHistorgram-252	
ByteHistorgram-253	
ByteHistorgram-254	
ByteHistorgram-255	
ByteEntropyHistorgram-56	0-256 arası belirtilen numaralar için tekrar eden bayt sayısına göre entropi değeri
ByteEntropyHistorgram-113	
ByteEntropyHistorgram-115	
ByteEntropyHistorgram-116	
ByteEntropyHistorgram-117	
ByteEntropyHistorgram-118	
ByteEntropyHistorgram-119	
ByteEntropyHistorgram-130	
ByteEntropyHistorgram-131	
ByteEntropyHistorgram-132	
ByteEntropyHistorgram-133	
ByteEntropyHistorgram-134	
ByteEntropyHistorgram-143	
ByteEntropyHistorgram-144	
ByteEntropyHistorgram-145	
ByteEntropyHistorgram-146	
ByteEntropyHistorgram-147	
ByteEntropyHistorgram-152	
ByteEntropyHistorgram-154	
ByteEntropyHistorgram-157	
ByteEntropyHistorgram-159	
ByteEntropyHistorgram-161	
ByteEntropyHistorgram-162	
ByteEntropyHistorgram-163	
ByteEntropyHistorgram-166	
ByteEntropyHistorgram-168	
ByteEntropyHistorgram-172	
ByteEntropyHistorgram-174	
ByteEntropyHistorgram-175	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
ByteEntropyHistorgram-178	0-256 arası belirtilen numaralar için tekrar eden bayt sayısına göre entropi değeri
ByteEntropyHistorgram-179	
ByteEntropyHistorgram-180	
ByteEntropyHistorgram-181	
ByteEntropyHistorgram-182	
ByteEntropyHistorgram-183	
ByteEntropyHistorgram-191	
ByteEntropyHistorgram-195	
ByteEntropyHistorgram-210	
ByteEntropyHistorgram-211	
ByteEntropyHistorgram-214	
ByteEntropyHistorgram-217	
ByteEntropyHistorgram-218	
ByteEntropyHistorgram-223	
ByteEntropyHistorgram-227	
ByteEntropyHistorgram-228	
ByteEntropyHistorgram-229	
ByteEntropyHistorgram-230	
ByteEntropyHistorgram-231	
ByteEntropyHistorgram-233	
ByteEntropyHistorgram-235	
ByteEntropyHistorgram-237	
ByteEntropyHistorgram-240	
ByteEntropyHistorgram-241	
ByteEntropyHistorgram-242	
ByteEntropyHistorgram-244	
ByteEntropyHistorgram-245	
ByteEntropyHistorgram-246	
ByteEntropyHistorgram-247	
ByteEntropyHistorgram-253	
ByteEntropyHistorgram-255	
# of consecutive printable strings	Ardışık yazdırılabilir karakter sayısı
avg length of consecutive printable strings	Yazdırılabilir karakter sayılarının ortalama uzunluğu
sum of frequencies of all the printable characters	Tüm yazdırılabilir karakterlerin frekanslarının toplamı
freq of printable characters -	İlgili yazdırılabilir karakterin frekansı
freq of printable characters - !	
freq of printable characters - "	
freq of printable characters - #	
freq of printable characters - $	
freq of printable characters - %	
freq of printable characters - &	
freq of printable characters - '	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
freq of printable characters - (	İlgili yazdırılabilir karakterin frekansı
freq of printable characters -)	
freq of printable characters - *	
freq of printable characters - +	
freq of printable characters - ,	
freq of printable characters - -	
freq of printable characters - .	
freq of printable characters - /	
freq of printable characters - 0	
freq of printable characters - 1	
freq of printable characters - 2	
freq of printable characters - 3	
freq of printable characters - 4	
freq of printable characters - 5	
freq of printable characters - 6	
freq of printable characters - 7	
freq of printable characters - 8	
freq of printable characters - 9	
freq of printable characters - :	
freq of printable characters - ;	
freq of printable characters - <	
freq of printable characters - =	
freq of printable characters - >	
freq of printable characters - ?	
freq of printable characters - @	
freq of printable characters - A	
freq of printable characters - B	
freq of printable characters - C	
freq of printable characters - D	
freq of printable characters - E	
freq of printable characters - F	
freq of printable characters - G	
freq of printable characters - H	
freq of printable characters - I	
freq of printable characters - J	
freq of printable characters - K	
freq of printable characters - L	
freq of printable characters - M	
freq of printable characters - N	
freq of printable characters - O	
freq of printable characters - P	
freq of printable characters - Q	
freq of printable characters - R	
freq of printable characters - S	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
freq of printable characters - T	İlgili yazdırılabilir karakterin frekansı
freq of printable characters - U	
freq of printable characters - V	
freq of printable characters - W	
freq of printable characters - X	
freq of printable characters - Y	
freq of printable characters - Z	
freq of printable characters - [	
freq of printable characters - \	
freq of printable characters -]	
freq of printable characters - ^	
freq of printable characters - _	
freq of printable characters - `	
freq of printable characters - a	
freq of printable characters - b	
freq of printable characters - c	
freq of printable characters - d	
freq of printable characters - e	
freq of printable characters - f	
freq of printable characters - g	
freq of printable characters - h	
freq of printable characters - i	
freq of printable characters - j	
freq of printable characters - k	
freq of printable characters - l	
freq of printable characters - m	
freq of printable characters - n	
freq of printable characters - o	
freq of printable characters - p	
freq of printable characters - q	
freq of printable characters - r	
freq of printable characters - s	
freq of printable characters - t	
freq of printable characters - u	
freq of printable characters - v	
freq of printable characters - w	
freq of printable characters - x	
freq of printable characters - y	
freq of printable characters - z	
freq of printable characters - {	
freq of printable characters - |	
freq of printable characters - }	
freq of printable characters - ~	
freq of printable characters - •	

Çizelge 6.7. (devam) EMBER 2018 veri setinden seçilen özellikler

Özellik	Açıklama
entropy of the byte histogram	Bayt histogramı entropisi
occurrences of "http://" or "https://". (ignore case)	"http://" ve "https://" görülme sayısı
occurrences of "MZ"	"MZ" görülme sayısı
length of the bytes of the file	Bayt cinsinden dosya boyutu
virtual size of the lief parsed binary	Ayrıştırılmış ikili dosyanın sanal boyutu
# of imported functions	İçe aktarılan fonksiyon sayısı
when the file was created	Dosyanın oluşturulma tarihi
hash COFF machine type - 5	İlgili COFF makine tipi için özet değeri
hash COFF characteristics - 0	İlgili COFF karakteristiği için özet değeri
hash optional subsystem - 8	8 numaralı opsiyonel alt sistemin özet değeri
hash optional dll_characteristics - 1	1 numaralı DLL karakteristik özelliğinin özet değeri
major_linker_version	Bağlayıcı ana sürüm numarası
sizeof_code	Kod boyutu
# of sections with "props" including "MEM_READ" and "MEM_EXECUTE"	"MEM_READ" ve MEM_EXECUTE" proplarına sahip bölüm sayısı
# of sections with "props" including "MEM_WRITE"	"MEM_WRITE" propuna sahip bölüm sayısı
hash on pair of section name and size - 43	43 numaralı bölüm ismi ve boyutunun özet değeri
hash on pair of section name and entropy - 14	İlgili numaralı bölüm ismi ve entropisinin özet değeri
hash on pair of section name and entropy - 32	
hash on pair of section name and entropy - 41	
hash on pair of section name and entropy - 43	
hash on pair of section name and virtual size - 43	43 numaralı bölüm isminin ve sanal boyutunun özet değeri
hash on list of properties of entry point - 37	37 numaralı giriş noktası özellik listesinin özet değeri
hash on list of library:function - 144	İlgili fonksiyon numarasına sahip kütüphane listesinin özet değeri
hash on list of library:function - 398	
hash on list of library:function - 942	
virtual size of data directories - 1	İlgili numaralı veri dizinlerinin boyutu ve ilgili numaralı veri dizinlerinin sanal boyutu
size of data directories - 2	
size of data directories - 4	
virtual size of data directories - 4	
virtual size of data directories - 6	
size of data directories - 12	
size of data directories - 13	
virtual size of data directories – 14	

EMBER 2018 veri setinde bulunan zararlılar 3227 farklı zararlı ailesine gruplanmaktadır. Veri setinde en sık kullanılan zararlı aileleri, bu ailelerin türleri ve zararlı ailesi ile ilgili açıklamalar çizelge 6.8'de sunulmaktadır.

Çizelge 6.8. EMBER 2018 veri setinde bulunan zararlı aileleri (en çok kullanılan 10 aile)

Aile	Adet	Tür	Açıklama
Xtrat [94]	41 564	Arka Kapı	Bu aileye mensup zararlılar, sistem güvenliğini bozmakta ve ilgili sistemden bilgi çalmaktadır. İlgili zararlılar, uygulama verileri, sistem dosyaları ve geçici kullanıcı dosyaları altına indirdikleri zararlı yüklerini alışıldık dosya isimlerine benzer dosya isimleriyle sistemde tutmaktadır. Genellikler, kendilerini "Iexplore.exe" ve "svchost.exe" gibi dosyalara enjekte ederek çalışmaktadır. Kayıt defterinde çeşitli değişiklikler yaparak sürekliliği sağlamaktadır.
Zbot [95]	27 656	Casus Yazılım	Bu aileye ait zararlılar çoğunlukla kurbanların tıklayacağı bağlantılarla e-posta yoluyla yayılmaktadır, ancak istismar kitleri de bu casus yazılımı yayabilmektedir. Zararlılar, Microsoft Windows sistemleri hedeflemekte ve genellikle finansal verileri çalmak için kullanılmaktadır. Bu zararlı ailesine ait ilk varyant 2007 yılında tespit edilmiştir. Bu zararlı yazılım ailesine mensup zararlıların bir kısmı, dünya çapındaki botnet yazılımlarının önemli parçalarından birini oluşturmaktadır.
Ramnit [96]	23 999	Virüs Solucan Truva atı	Ramnit, virüsleri, solucanları ve Truva atlarını barındıran çok yönlü bir ailedir. Etkilenen bir sistemdeki EXE, DLL ve HTML dosyalarına bulaşma yeteneğine sahiptirler. İlk Ramnit varyantı 2010 yılında görülmüştür. Varyantın özelliklerine göre, sistemden bilgi çalmaktan hedef makineyi botnet ağının bir parçası haline getirmeye kadar çeşitli etkileri bulunmaktadır.

Çizelge 6.8. (devam) EMBER 2018 veri setinde bulunan zararlı aileleri

Aile	Adet	Tür	Açıklama
Installmonster [97]	22 199	Reklam Yazılımı	InstallMonster, sistemlere reklam yazılımları ve potansiyel olarak istenmeyen programları (PUP'lar) yükleyen bir paketleyici ailesi için genel adlandırmadır. Kayıt defterinde değişiklikler yaparak sürekliliği sağlamaktadır. Komuta kontrol sunucusuna bağlantı yapmaya çalışarak internet bağlantısı kontrolü, dosya indirme ve yürütme, yapılandırma bilgileri edinme, saldırganlardan komut alma gibi işlemleri gerçekleştirmektedir.
Sality [98]	21 750	Virus	Bu aile, polimorfik bulaşıcı virüsleri içermektedir. Sality ailesinin ilk varyantı Temmuz 2003'te tespit edilmiştir. Zararlılar zaman içerisinde değişikliklere uğramış olup, şifre çözme algoritmaları ve programlara bulaşma yöntemleri önemli şekilde gelişmiştir. Virüsün gövdesi, virüslü programın son bölümünün sonunda bulunmaktadır. Virüsün ilk kısmı büyük ölçüde karartılmaltadır ve diğer kodun şifresini çözmektedir. Bu aileye ait zararlıların kötü amaçlı işlevleri, URL'lerden veya P2P ağ aracılığıyla indirilebilen ayrı modüller tarafından uygulanmaktadır.
Zusy [99]	18 766	Casus Yazılım	Bu aileye mensup zararlılar, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Zararlılar, yüklerini, uygulama verileri klasörü altına yerleştirmektedir. Sürekliliğin sağlanması maksadıyla, sistem başlatıldığında otomatik olarak çalışması için kayıt defterine yeni kayıt eklemektedir.
Vtflooder [100]	16 164	Truva atı	Bu aileye mensup zararlılar, sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Zararlı, yüklerini geçici kullanıcı dosyaları altına “.exe” ve “.reg” uzantılı dosyalar olarak indirmektedir. Sistem başlatıldığında otomatik olarak çalışması için kayıt defterine yeni kayıt eklemektedir.

Çizelge 6.8. (devam) EMBER 2018 veri setinde bulunan zararlı aileleri

Aile	Adet	Tür	Açıklama
Emotet [101]	15 624	Truva atı	Emotet, öncelikle kimlik avı e-posta ekleri ve tıklandığında yükü başlatan bağlantılar yoluyla yayılan gelişmiş bir truva atı ailesidir. Bu aileye mensup zararlılar sistemlere girdikten sonra, kullanıcı kimlik bilgilerini kaba kuvvet yöntemiyle ele geçirmeye ve paylaşılan sürücülere yazarak ağ içinde çoğalmaya çalışmaktadır. Emotet ailesine mensup zararlılar, yeteneklerini sürekli olarak geliştirmek ve güncellemek için modüler DLL'leri kullanmaktadır.
Fareit [102]	14 382	Truva atı	Bu aileye ait zararlılar, spam eposta yoluyla yayılmakta ve epostada gönderilen bağlantıya tıklayan kullanıcıları zararlının bulunduğu internet sayfasına yönlendirmek suretiyle zararlının sisteme indirilmesi sağlanmaktadır. Bu aileye ait zararlıların temel amacı, enfekte olan bilgisayarlardaki FTP istemcilerinin, bulut depolama hizmetlerinin ve eposta hizmetlerinin hesap bilgileri ile tarayıcı verilerini toplamaktır. Bu ailedeki bazı zararlılar, başka zararlıları indirip çalıştırabilmektedirler.
Adposhel [103]	11 756	Reklam Yazılımı	Bu aileye ait zararlılar, bir sisteme diğer kötü amaçlı yazılımlar tarafından bırakılan bir dosya olarak veya kötü amaçlı siteleri ziyaret ederken kullanıcılar tarafından bilmeden indirilen bir dosya olarak gelmektedir. Ayrıca, bir kullanıcı tarafından elle de yüklenebilmektedir. Zararlılar, program klasörleri, uygulama verisi klasörleri ve geçici kullanıcı klasörleri altına yerleşmektedir. Etkiledikleri sistemlerde sürekliliği sağlamak için kayıt defterine eklemelerde bulunmaktadır.

6.4. Özgün Veri Seti

Çalışmanın uygulama aşamasında kullanılmak üzere, VirusShare [77] üzerinden indirilen 997 zararlının 977'si ile Windows 8 işletim sistemine sahip bir istemciden elde edilen faydalı dosyalardan 5323 tanesi kullanılarak, geliştirilen Python kodunun yardımıyla, çalışmaya özgü bir veri seti oluşturulmuştur. Bu veri setinin oluşturulmasının amacı, uygulama aşamasında kullanılacak modeldeki sınıflandırıcıların eğitilmesi ve canlı ortamda zararlı

yazılım tespiti yapabilecek bir model geliştirilmesidir. Özgün veri setinde bulunan faydalı ve zararlı örneklem sayısı bilgisi Çizelge 6.9'da sunulmaktadır.

Çizelge 6.9. Özgün veri seti için faydalı ve zararlı örneklem sayısı tablosu

	Faydalı	Zararlı	Toplam
Örneklem Sayısı	5 323	977	6 300

Özgün veri seti oluşturulurken dosyaların PE başlık bilgileri ile kaynak ve bölüm entropilerinden faydalanılmaktadır. Özgün veri seti kapsamında çıkarılan özellikler Çizelge 6.10'da sunulmaktadır.

Çizelge 6.10. Özgün veri setinde kullanılan özellikler

ÖZELLİK	AÇIKLAMA
Characteristics	Dosya özelliklerini içeren değer bitleri
MajorLinkerVersion	Bağlayıcı ana sürüm numarası
MinorLinkerVersion	Bağlayıcı alt sürüm numarası
SizeOfCode	Kod (Metin) bölümünün boyutu
SizeOfInitializedData	Başlatılan veri bölümünün boyutu veya birden çok veri bölümü varsa bu tür bölümlerin toplam boyutu
SizeOfUninitializedData	Başlatılmamış veri bölümünün (BSS) boyutu veya birden fazla BSS bölümü varsa bu tür tüm bölümlerin toplamı
AddressOfEntryPoint	Yürütülebilir dosya belleğe yüklendiğinde, görüntü tabanına göre giriş noktasının adresi
BaseOfCode	Belleğe yüklendiğinde kodun başlangıcı bölümünün imaj tabanına göreli adresi
BaseOfData	Belleğe yüklendiğinde, veri başlangıcı bölümünün imaj tabanına göreli adresi
ImageBase	Belleğe yüklendiğinde görüntünün ilk baytının tercih edilen adresi
SectionAlignment	Bölümlerin belleğe yüklendiğinde hizalanması (bayt cinsinden)
FileAlignment	İmaj dosyasındaki bölümlerin ham verilerini hizalamak için kullanılan hizalama faktörü (bayt cinsinden)
MajorOperationSystemVersion	Gerekli işletim sisteminin ana sürüm numarası
MinorOperatingSystemVersion	Gerekli işletim sisteminin alt sürüm numarası
MajorImageVersion	İmajın ana sürüm numarası
MinorImageVersion	İmajın alt sürüm numarası
MajorSubsystemVersion	Alt sistemin ana sürüm numarası
MinorSubsystemVersion	Alt sistemin alt sürüm numarası
SizeOfImage	İmaj belleğe yüklenirken tüm başlıklar dahil imajın boyutu (bayt cinsinden). SectionAlignment'ın katı olmalıdır

Çizelge 6.10. (devam) Özgün veri setinde kullanılan özellikler

ÖZELLİK	AÇIKLAMA
SizeOfHeaders	Bir MS-DOS saplamasının, PE başlığının ve bölüm başlıklarının birleştirilmiş boyutu
Checksum	İmaj dosyası sağlama toplamı
Subsystem	Bu imajı çalıştırmak için gerekli olan alt sistem
DllCharacteristics	DLL Karakteristikleri
SizeOfStackReserve	Ayrılacak yığının boyutu
SizeOfStackCommit	İşlenecek yığının boyutu
SizeOfHeapReserve	Ayrılacak yerel yığın alanının boyutu
SizeOfHeapCommit	İşlenecek yerel yığın alanının boyutu
LoaderFlags	Ayrılmış, sıfır olmalıdır
NumberOfRVAAndSizes	İsteğe bağlı başlığın geri kalanındaki veri dizini girişlerinin sayısı
NumberOfSections	Bölüm sayısı
SectionMeanEntropy	Bölümlerin ortalama entropisi
SectionMinEntropy	En düşük bölüm entropisi
SectionMaxEntropy	En yüksek bölüm entropisi
SectionMeanRawSize	Bölümlerin ortalama ham boyutu
SectionMinRawSize	En düşük bölüm ham boyutu
SectionMaxRawSize	En yüksek bölüm ham boyutu
SectionMeanVirtualSize	Bölümlerin ortalama sanal boyutu
SectionMinVirtualSize	En düşük bölüm ham sanal boyutu
SectionMaxVirtualSize	En yüksek bölüm ham sanal boyutu
ImportsDllCount	İçe aktarılan DLL sayısı
ImportsCount	İçe aktarım sayısı
ExportCount	Dışa aktarım sayısı
ResourceNb	Kaynak sayısı
ResourceMeanEntropy	Kaynakların ortalama entropisi
ResourceMinEntropy	En düşük kaynak entropisi
ResourceMaxEntropy	En yüksek kaynak entropisi
ResourceMeanSize	Oralama kaynak boyutu
ResourceMinSize	En düşük kaynak boyutu
ResourceMaxSize	En yüksek kaynak boyutu
LoadConfigurationSize	Yüklenen yapılandırma bilgisi boyutu
Legitimate	Sınıf

Özgün veri setinde bulunan zararlılar 112 farklı zararlı ailesine gruplanmaktadır. Veri setinde en sık kullanılan zararlı aileleri, bu ailelerin türleri ve zararlı ailesi ile ilgili açıklamalar çizelge 6.11'de sunulmaktadır.

Çizelge 6.11. Özgün veri setinde bulunan zararlı aileleri

Aile	Adet	Tür	Açıklama
Runouce [104]	84	Solucan	Bu aileye ait zararlılar, çalıştırıldıklarına kendilerini sistem dizinine kopyalamakta ve Windows kayıt defterini değiştirmektedirler. Bu sayede etkiledikleri sistemlerde sürekliliği sağlamaktadırlar. Hedef olarak kullanıcıların sabit disklerindeki HTML dosyalarını, zararlı yüklerden birini tetikleyecek şekilde değiştirmektedirler. Eposta üzerinden bir ses dosyası olarak yayılmaktadırlar.
Chir [105]	55	Solucan, Virus	Bu zararlı yazılım ailesi hem solucan hem de virüs bileşenlerine sahiptir. Windows sistemlerde bulunan ve güvenlik yamaları geçilmemiş olan bir açıklığı (MS01-020) istismar etmektedir. Virüs bileşeni, yerel ve uzak sürücülerdeki .EXE ve .SCR dosyalarına bulaşmaktadır. Ayrıca hedef bilgisayarlarda depolanan .HTM ve .HTML dosyaları zararlılar tarafından düzenlenmekte ve bu dosyalar açıldığında zararlının da dosyayla birlikte çalıştırıldığı bilinmektedir.
Virut [106]	54	Virüs	Bu zararlı yazılım ailesi, ilk kez 2006 yılında görülen bir botnet zararlı yazılım ailesidir. Bu aileye ait zararlılar, erişilen herhangi bir yürütülebilir dosyaya veya ekran koruyucu dosyaya kod enjekte ederek yayılmaktadırlar. Ayrıca HTML, PHP ve ASP dosyalarına kötü amaçlı iframe'ler enjekte etmektedirler. Bu sayede, zararlılar ağ paylaşımları, çıkarılabilir sürücüler ve kötü amaçlı reklam yoluyla ek cihazlara dağıtılabilmektedir. Ailenin en güncel varyantlar, wiınlogon.exe işlemine kod enjekte edebilmektedir.
Madangel [107]	47	Truva Atı, Virüs	Bu aileye ait zararlılar, sistemde bulunan çalıştırılabilir dosyaları enfekte etmektedir. Enfekte olan çalıştırılabilir dosyalar, zararlıyı daha da yaymak için kötü amaçlı kodu çalıştırmaktadırlar. Enjekte edilen kod, hedef sistemin enfekte edildiğini bildirmek ve/veya kendini güncellemek için kullanabileceği sabit kodlanmış URL'ler içermektedir.

Çizelge 6.11. (devam) Özgün veri setinde bulunan zararlı aileleri

Aile	Adet	Tür	Açıklama
Autoit [108]	37	Solucan	Bu aileye mensup zararlılar, çeşitli yıkıcı görevleri çalıştıran bir AutoIt betiğinden oluşmaktadır. Kötü amaçlı yazılım, kendisini okumaya/yazmaya açık klasörlere (varsa) kopyalayarak ağ kaynakları veya çıkarılabilir medya aracılığıyla yayılmaktadır. Autoit ailesindeki varyantlar, sisteme diğer zararlı programları indirmeye ve yüklemeye çalışmaktadır. Çoğu türev ayrıca web tarayıcısı (genellikle Internet Explorer) için başlangıç ve varsayılan arama sayfalarını değiştirmeye çalışmaktadır.
Msil [109]	36	Truva Atı	Bu aileye ait zararlılar, kullanıcıların izni ve rızası olmadan kullanıcıya ait bilgilerin imha edilmesi, kullanıcı bilgilerine erişimin engellenmesi, değiştirilmesi veya kopyalanması, bilgisayarların ve bilgisayar ağlarının işleyişinin bozulması gibi eylemleri gerçekleştirmektedir. Hedef kullanıcılardan maddi kazanç sağlamak için bir fidye yazılım olarak kullanılabilmektedir.
Sirefef [110]	26	Truva Atı	Bu aileye mensup zararlılar, genellikle yazılım lisanslarını etkisiz kılmak için kullanılan korsan uygulamalar tarafından ya da diğer zararlılar tarafından hedef sisteme indirilmektedirler. Yeni varyantların bir kısmı sisteme “GoogleUpdate.exe” olarak ya da bir sürücü dosyasının yerine yerleşmekte, kayıt defterinde yapılan değişiklik ile sistemde süreklilik sağlamaktadırlar. Hedef sistemde, yapılan arama sonuçlarını değiştirme, kripto madenciliği ve reklam tıklama gibi işlemler gerçekleştirmektedir.
Nimda [111]	26	Solucan	Bu aileye mensup zararlılar, Microsoft Windows işletim sistemlerinin belirli sürümlerine sahip bilgisayarları hedeflemektedirler. Zararlılar, Web içeriği belgelerine bulaşarak ve kendisini e-postalara ekleyerek yayılmak için Microsoft Güvenlik Bülteni MS01-020'de açıklanan Windows güvenlik açığından yararlanmaktadırlar. Zararlılar, ayrıca yürütülebilir dosyalara bulaşarak ve kendisini arka kapılar aracılığıyla yerel klasörlere, ağ paylaşımlarına ve uzak bilgisayarlara kopyalayarak yayılmaktadırlar. Sistemin enfekte edilmesini müteakip, C sürücüsünü paylaşıma açmak ve yönetici izinleriyle bir Konuk hesabı oluşturmak suretiyle sistem güvenliğini güvenliği tehlikeye atmaktadırlar.

Çizelge 6.11. (devam) Özgün veri setinde bulunan zararlı aileleri

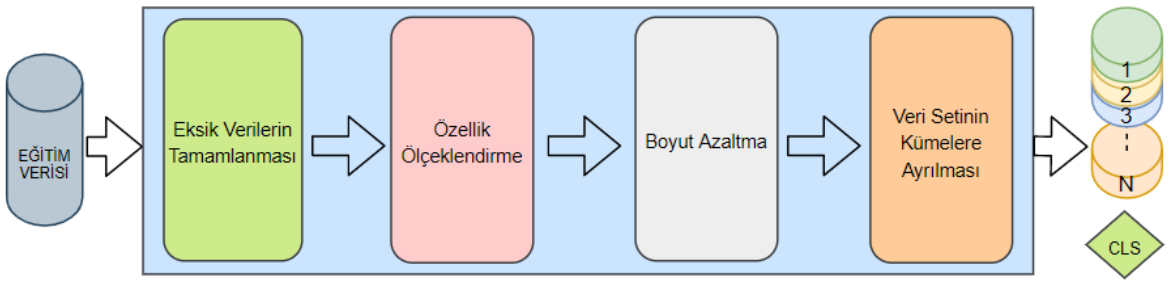
Aile	Adet	Tür	Açıklama
Slugin [112]	17	Virüs	Bu aileye mensup zararlılar, kendilerini enfekte edilmiş bir bilgisayardaki çeşitli dosyalara ekleyebilmektedir. Zararlılar, sistem ayarlarını değiştirmek ve tarayıcıyı kötü amaçlı sitelere yönlendirmek gibi kötü niyetli eylemler gerçekleştirebilmektedir. Zararlılar, sistem başlangıcında yüklenebilmektedir. Ayrıca, epostalar ve virüslü dosyalar aracılığıyla yayılabilmektedirler.
Zbot [95]	12	Casus Yazılım	Bu aileye ait zararlılar çoğunlukla kurbanların tıklayacağı bağlantılarla e-posta yoluyla yayılmaktadır, ancak istismar kitleri de bu casus yazılımı yayabilmektedir. Zararlılar, Microsoft Windows sistemleri hedeflemekte ve genellikle finansal verileri çalmak için kullanılmaktadır. Bu zararlı ailesine ait ilk varyant 2007 yılında tespit edilmiştir. Bu zararlı yazılım ailesine mensup zararlıların bir kısmı, dünya çapındaki botnet yazılımlarının önemli parçalarından birini oluşturmaktadır.

7. GELİŞTİRİLEN MODEL

Çalışmanın bu bölümünde önerilen tespit yaklaşımı ve oluşturulma aşamaları anlatılmaktadır. Sunulan tespit yaklaşımında, farklı makine öğrenmesi algoritmalarından faydalanılarak yüksek doğruluk ve f1 skoru ile zararlı yazılım tespiti yapılması amaçlanmaktadır. Sunulan metot, veri önişleme ve kümeleme, sınıflandırıcıların eğitimi ve seçimi, modelin oluşturulması olmak üzere üç safhadan oluşmaktadır.

7.1. Veri Önişleme ve Kümeleme

Veri önişleme aşamasında, veri setinin makine öğrenmesi algoritmaları tarafından etkin bir şekilde işlenebilir hale getirilmesi amaçlanmaktadır. Bu aşamada, eksik ve/veya hatalı verilerin düzeltilmesini içeren "eksik verilerin tamamlanması", algoritmaların performanslarının artırılması için "özellik ölçeklendirme" ve "Boyut Azaltma" adımları bulunmaktadır. Veri önişleme adımlarının tamamlanmasından sonra kümeleme adımı gelmektedir. Bu adımda veri seti bir kümeleme algoritması kullanılarak alt veri setlerine bölünmektedir. Veri önişleme ve kümeleme adımları Şekil 7.1'de sunulmaktadır.



Şekil 7.1. Veri önişleme ve kümeleme

Şekil 7.1'de gösterilen veri önişleme bölümünün alt adımlarının açıklamaları müteakip başlıklar altında sunulmaktadır.

Eksik Verilerin Tamamlanması

Bir zararlı yazılım dosyası kullanılarak özellik çıkarma işlemi gerçekleştirilirken, her bir zararlı için tüm özellik değerlerinin elde edilmesi mümkün olamamaktadır. Bu durumda, veri setindeki örneklemlerin bir kısmı elde edilmek istenen özellik değerlerinin hepsine sahipken, bir kısmı ise bazı özellik değerlerinden yoksun olmaktadır. Bir veri setinde eksik bir veri bulunması durumunda, iki farklı yöntem izlenmektedir. Bunlardan birincisi, ilgili

örneklemin veri setinden çıkarılması, ikincisi ise eksik veri yerine yeni bir veri atanmasıdır.

Çalışmada kullanılan Bodmas ve EMBER 2018 veri setlerinde eksik veri bulunmamaktadır. Kaggle Zararlı Yazılım Tespiti veri setindeki eksik verilerin tamalanması için, "SimpleImputer" kütüphanesi kullanılarak, ilgili sütunun ortalamasının alınması ve eksik veri yerine sütun ortalamasının koyulması yaklaşımı izlenmektedir.

Özellik Ölçeklendirme

Normalizasyon olarak da adlandırılan özellik ölçeklendirme aşamasının maksadı, ham veride bulunan değerlerin, çeşitli makine öğrenmesi algoritmaları tarafından etkin bir şekilde işlenebilmesini sağlamaktır. Minimum-maksimum normalizasyonu, ortalama normalizasyonu ve z-skoru normalizasyonu gibi ölçeklendirme yaklaşımları bulunmaktadır. Çalışmada ortalama normalizasyonu kullanılmaktadır.

Bu aşamada, veri seti, eğitim veri seti ve test veri seti olmak üzere iki veri setine ayrılmaktadır. Eğitim ve test verilerinin ayrılmasından sonra, ölçekleme modeli eğitim veri setinin ölçeklendirilmesi suretiyle eğitilmektedir. Müteakiben, eğitilmiş ölçekleme modeli ile test veri seti ölçeklendirilmektedir. Özellik ölçeklendirme aşamasında "StandardScaler" kütüphanesinden faydalanılmaktadır.

Boyut Azaltma

Boyut azaltma işlemiyle, makine öğrenmesi algoritmalarının tahmin sürecinde anlam ifade etmeyen ya da çok düşük anlam ifade eden özelliklerin ortaya çıkarabileceği ek iş yükünün azaltılması amaçlanmaktadır. Bu sayede hem tahmin performansının artması hem de tahmin süresinin kısalması hedeflenmektedir.

Çalışmada, Bodmas ve EMBER 2018 veri setleri için, rastgele orman yöntemi kullanılarak boyut azalma işlemi gerçekleştirilmektedir. İlk adım olarak, her iki veri setinde, tüm özellikler için önem derecesi hesaplanmaktadır. İkinci adımda, 2381 özellik arasından, en yüksek önem derecesine sahip 448 özellik tespit edilerek, veri setinden bu özelliklerin olduğu kolonlar çekilmekte ve makine öğrenmesi algoritmaları tarafından işlenecek veri setleri oluşturulmaktadır.

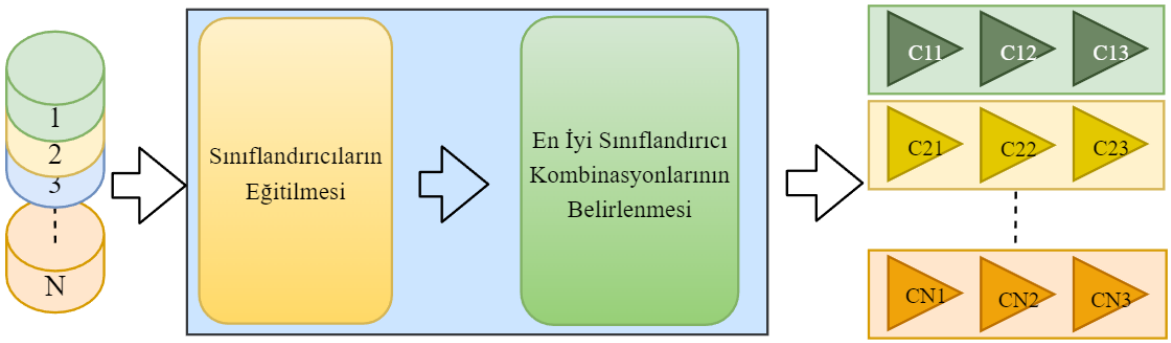
56 özelliğe sahip Kaggle Zararlı Yazılım Tespiti veri seti ve 51 özelliğe sahip özgün veri seti için boyut azaltma işlemi uygulanmamaktadır.

Kümeleme

Kümeleme, bir denetimsiz öğrenme yaklaşımı olup, genellikle sınıflandırılmamış veriler üzerinde işlem yapmak için kullanılmaktadır. Bu tez çalışmasında, sınıflandırma işleminden önce bir kümeleme işlemi gerçekleştirmek suretiyle tespit yaklaşımının tahmin performansının artırılması hedeflenmektedir. İdeal küme sayısının tespit edilebilmesi için "Dirsek Grafiği" yöntemi kullanılmaktadır. Dirsek grafiğinin bir eksenini küme sayısı diğer eksenini örneklemlerin küme merkezlerine uzaklığının karelerinin toplamı oluşturmaktadır. Veri setlerinin hiçbirisinde, uzaklık karelerinin toplamında belirgin bir kırılma ortaya çıkmamaktadır. Bu sebeple dirsek noktası tespit edilememiştir. Önerilen tespit yaklaşımının etkin bir şekilde sunulabilmesi maksadıyla, çalışmada kullanılan veri setleri üç alt kümeye ayrılmaktadır. Bu aşamada, veri seti K-means algoritması kullanılarak kümelere ayrılmaktadır.

7.2. Sınıflandırıcıların Eğitilmesi ve Seçimi

Metodun bu aşamasında hem tüm eğitim verisi hem de veri setinin alt kümeleri kullanılarak farklı makine öğrenme algoritmaları eğitilmektedir. Daha sonra her bir alt küme için en iyi tahmin yapan üçlü sınıflandırıcı kombinasyonu belirlenmektedir. Sınıflandırıcıların eğitilmesi ve en iyi sınıflandırıcı kombinasyonunun belirlenmesi adımları Şekil 7.2'de sunulmaktadır.



Şekil 7.2. Sınıflandırıcıların eğitilmesi ve en iyi sınıflandırıcı seçimi

Şekil 7.2'de gösterilen sınıflandırıcıların eğitilmesi ve seçimi bölümünün alt adımlarının açıklamaları müteakip başlıklar altında sunulmaktadır.

Sınıflandırıcıların Eğitilmesi

Bu aşamada, eğitim veri seti tümü ve bir önceki adımda elde edilen alt kümeleriyle sınıflandırıcılar eğitilmektedir. Eğitilen sınıflandırma algoritmaları Kernel SVM, K-nearest-neighbor, Naive Bayes, Decision Tree, Random Forest, XGBoost, CatBoost, AdaBoost, Extreme Trees ve Gradient Boosting algoritmalarıdır. Bu aşama neticesinde, her bir algoritma için, bir tanesi bütün eğitim setiyle, üç tanesi eğitim setinin farklı alt kümeleriyle eğitilmiş olan dört sınıflandırıcı eğitilmektedir. Tüm eğitim setiyle eğitilmiş sınıflandırıcılar genel sınıflandırıcı, eğitim setinin alt kümeleriyle eğitilen sınıflandırıcılar ise özelleşmiş sınıflandırıcı olarak adlandırılmaktadır.

En İyi Sınıflandırıcı Kombinasyonlarının Belirlenmesi

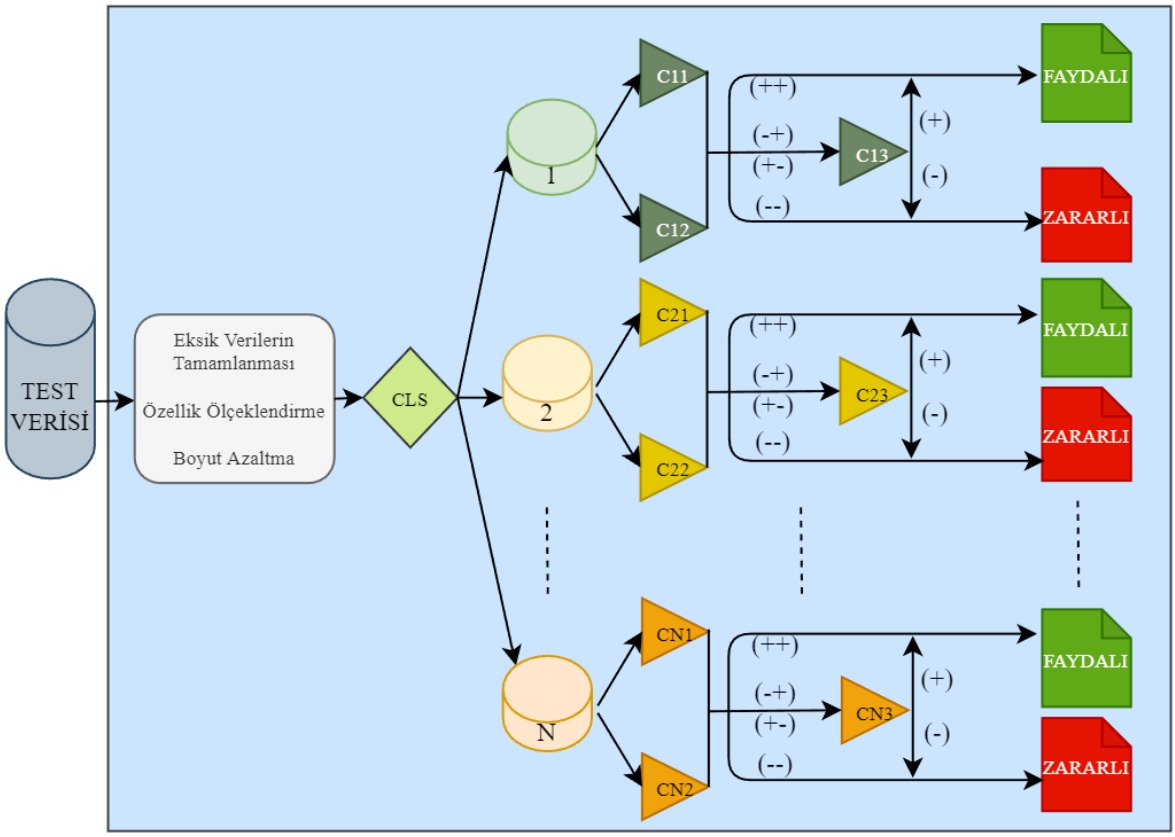
Bu aşamada, genel ve özelleşmiş sınıflandırıcıların ayrı ayrı üçlü kombinasyonu alınarak her bir kombinasyonun tahmin performansları değerlendirilmektedir. Her sınıflandırıcı kombinasyonu için doğruluk ve f1 skorunun ortalaması alınarak en yüksek oranda tespit gerçekleştiren sınıflandırıcı kombinasyonu belirlenmektedir. İlgili kombinasyon, ilgili küme için en iyi sınıflandırıcı kombinasyonu olarak kabul edilmektedir. Bu belirleme işlemi hem tüm veri seti için hem de veri setinin alt kümeleri için ayrı ayrı yapılmaktadır.

Sınıflandırıcıların eğitilmesi ve seçimi adımının çıktısı, hem tüm veri seti hem de veri setinin her bir alt kümesi için, en iyi sınıflandırmayı gerçekleştiren sınıflandırıcı kombinasyonlarıdır. Şekil 7.2'de yer alan "C11 C12 C13" gösterimi, bir numaralı küme için en iyi sınıflandırmayı gerçekleştiren üçlü sınıflandırıcı kombinasyonunu ifade etmektedir. Aynı şekilde, "CN1 CN2 CN3" ile "N" numaralı küme için en iyi sınıflandırmayı gerçekleştiren üçlü sınıflandırıcı kombinasyonu gösterilmektedir.

Bu aşamada seçilen üçlü sınıflandırıcı kombinasyonları için, her bir üçlünün en yavaş tahmin süresine sahip sınıflandırıcısı belirlenmektedir. Belirlenen bu en yavaş sınıflandırıcının, sınıflandırma esnasında ikinci kademede kullanılması suretiyle tahmin süresi performansı iyileştirilmektedir.

7.3. Modelin oluşturulması

Veri önişleme ve sınıflandırıcı seçimleri tamamlandıktan sonra; veri önişleme, kümeleme ve sınıflandırma modelleri birleştirilerek zararlı yazılım tespit modeli oluşturulmaktadır. Oluşturulan model şekil 7.3'te sunulmaktadır.



Şekil 7.3. Oy birliği ve toplu öğrenme ile zararlı yazılım tespit modeli

Şekil 7.3'te sunulan modelin bir örneklemi sınıflandırması aşağıdaki sırayla gerçekleşmektedir:

- Modele gelen bir örneklem öncelikle veri önişleme adımlarına tabi tutulmaktadır. Bu aşamada, örneklemdeki eksik veriler tamamlanmakta, örneklem özellikleri ölçeklendirilmekte ve boyut azaltma işlemi gerçekleştirilmektedir. Bu adımı tamamlayan bir örneklem, modeldeki kümeleme ve sınıflandırma algoritmaları tarafından işlenebilir hale gelmektedir.
- Kümeleme (CLS) adımına gelen örneklem, eğitim verisi kullanılarak eğitilen K-Means algoritması tarafından, mevcut kümelerden birine (1,2, ... N) kümelenmektedir. Bu

adımda, örneklemin, kendisini en doğru şekilde sınıflandıracak sınıflandırıcılara yönlendirilmesi amaçlanmaktadır.

- Sınıflandırma aşamasında, her alt küme için 3 sınıflandırıcı (CN1, CN2, CN3) bulunmaktadır. Bu aşamada hangi sınıflandırıcıların kullanılacağına, ilgili küme için en iyi tahmin performansını sergileyen üçlü sınıflandırıcı kombinasyonu bulunarak karar verilmektedir. Eğer bir örneklem "1" numaralı kümeye kümelendirilirse, ilgili örneklem C11-C12-C13 sınıflandırıcıları tarafından sınıflandırılmaktadır. Sınıflandırma aşamasında kullanılan sınıflandırıcılar, tahmin süresinin kısaltılması maksadıyla, iki kademeli olarak konumlandırılmaktadır.
- Sınıflandırma aşamasının birinci kademesinde CN1 ve CN2 sınıflandırıcıları bulunmaktadır. CN1 ve CN2 sınıflandırıcıları belirlenirken, ilgili kümede en iyi tahmin performansını sağlayan sınıflandırıcı kombinasyonu içerisindeki sınıflandırıcılardan en hızlı tahmin gerçekleştiren iki sınıflandırıcı CN1 ve CN2 olarak, en yavaş tahmin gerçekleştiren sınıflandırıcı CN3 olarak belirlenmektedir. Birinci aşamada bulunan CN1 ve CN2 sınıflandırıcıları bir örneklem üzerinde aynı tahmini (faydalı/zararlı) yaptıkları durumda, örneklem yapılan bu tahminle etiketlenmektedir. Bu aşamada, erken oy birliği ile yapılan sınıflandırma sayesinde, modelin tahmin süresi performansında iyileşme sağlanmaktadır.
- Sınıflandırma aşamasının birinci kademesinde, CN1 ve CN2 sınıflandırıcıları tarafından, ortak bir tahmin yapılamadığı durumda, nihai sınıflandırma (faydalı/zararlı) CN3 sınıflandırıcısı tarafından yapılmaktadır.

8. DENEYSEL SONUÇLAR

Bu bölümde, önerilen tespit yaklaşımının Google Colaboratory ortamında Kaggle Zararlı Yazılım Tespiti veri seti, BODMAS veri seti, EMBER 2018 veri seti ve özgün veri seti üzerinde sağladığı tahmin performansları sunulmaktadır.

Müteakip çizelgelerde yer alan sütunların ve kısaltmaların açıklamaları:

- Sınıflandırıcı : Eğitilmiş makine öğrenmesi algoritmaları
- Küme-n : İlgili veri setinin "n" numaralı alt kümesi
- Tüm Veri Seti : İlgili veri setinin tamamı
- Birleştirilmiş Tahmin: Bir algoritma için, veri setinin alt kümelerine göre eğitilmiş sınıflandırıcıların ayrı ayrı tahminlerinin birleştirilmesi sonucu elde edilen tahmin performansı
- Tahmin süresi : Tek bir örneklem için ilgili sınıflandırıcının tahmin gerçekleştirme süresi (Toplam tahmin süresi sınıflandırılan örneklem sayısına bölünerek saniye cinsinde elde edilmektedir)
- Küme : Veri setinin alt kümesi
- Sıra : İlgili kümede gerçekleştirilen tahmin performansı sırası
- D : Doğruluk
- F1 : F1 Skoru
- Kombinasyon : Üçlü sınıflandırıcı kombinasyonu

Sonuçların sunumunda, BODMAS, EMBER 2018 ve Kaggle Zararlı Yazılım Tespiti veri setlerinin her biri için 5 ayrı çizelge kullanılmaktadır. Bu çizelgelerin açıklaması aşağıda sunulmaktadır.

Genel Sınıflandırıcılar için Tahmin Performansları Çizelgesi

Bu tez kapsamında kullanılan genel sınıflandırıcı tabiri, tüm veri seti kullanılarak eğitilen sınıflandırıcıları ifade etmektedir. "Genel Sınıflandırıcılar için Tahmin Performansları" çizelgesinde genel sınıflandırıcıların tahmin performansları sunulmaktadır. "Küme-N" olarak ifade edilen sütunlarda, bir genel sınıflandırıcının ilgili kümedeki tüm örneklemler için gerçekleştirdiği tahmin performası sunulmaktadır. "Tüm veri seti" sütununda test veri setinin tamamı kullanılarak gerçekleştirilen tahmin performası sunulmaktadır. "Tahmin Süresi" sütununda, ilgili sınıflandırıcının bir örneklem üzerinde tahmin gerçekleştirmek için harcadığı ortalama süre verilmektedir. Tahmin süresi, bir sınıflandırıcının test veri seti

üzerinde tahmin gerçekleştirirken harcadığı toplam sürenin test verisi örneklem sayısına bölünmesi suretiyle hesaplanmaktadır.

Özelleşmiş Sınıflandırıcılar İçin Tahmin Performansları Çizelgesi

Bu tez kapsamında kullanılan özelleşmiş sınıflandırıcı tabiri, veri setinin alt kümelerinden biri kullanılarak eğitilmiş ve o alt kümeye kümelenen örneklemler üzerinde tahmin gerçekleştiren sınıflandırıcıları ifade etmek için kullanılmaktadır. "Özelleşmiş Sınıflandırıcılar İçin Tahmin Performansları" çizelgesinde özelleşmiş sınıflandırıcıların tehmin performansları sunulmaktadır. "Küme-N" olarak ifade edilen sütunlarda, "N" numaralı kümeye ait eğitim seti tarafından eğitilmiş ve kümeleme algoritması tarafından "N" numaralı kümeye kümelenen örneklemler üzerinde sınıflandırma gerçekleştiren sınıflandırıcının tahmin performası sunulmaktadır. "Birleştirilmiş Tahmin" sütununda, ilgili sınıflandırma algoritmasından türetilen özelleşmiş sınıflandırıcıların tahminlerinin, karmaşıklık matrisi kullanılmak suretiyle birleştirilmesi suretiyle ortaya çıkan performans değerleri sunulmaktadır.

Genel Sınıflandırıcılar ile En İyi 5 Kombinasyon Çizelgesi (Tüm Veri Seti İçin)

Bu tez çalışmasında 10 adet sınıflandırma algoritması kullanılmaktadır. Örneklemlerin en yüksek doğruluk ve f1 skoru ile sınıflandırılabilmesi maksadıyla, kullanılan 10 algoritmanın 3'lü kombinasyonları alınarak, her 3'lü kombinasyon için tahmin performansı elde edilmektedir. Elde edilen 3'lü sınıflandırıcı kombinasyonlarından, tüm veri seti için en iyi tahmin performansını gösteren ilk 5 tanesi "Genel Sınıflandırıcılar ile En İyi 5 Kombinasyon" çizelgesinde sunulmaktadır. "Tüm Veri Seti" sütununda ilgili sınıflandırı kombinasyonunun test veri seti üzerinde gerçekleştirdiği tahmin performası, "Küme-N" sütunları ise sınıflandırıcı kombinasyonunun "N" numaralı kümeye kümelenen örneklemler üzerinde gerçekleştirdiği tahmin performansı sunulmaktadır.

Genel Sınıflandırıcılar ile En İyi 5 Kombinasyon Çizelgesi (Her Küme İçin)

"Genel Sınıflandırıcılar ile En İyi 5 Kombinasyon (Her Küme İçin)" çizelgesinde, her küme için en iyi tahmin performansını sağlayan 5 genel sınıflandırıcı kombinasyonu sunulmaktadır. Sıralama gerçekleştirilirken doğruluk ve f-1 skoru değerlerinin aritmetik

ortalaması alınmaktadır. Eğer sınıflandırıcı kombinasyonlarının tahmin performansları eşitse, tahmin süresi performansı dikkate alınarak sıralama gerçekleştirilmektedir.

Özelleşmiş Sınıflandırıcılar ile En İyi 5 Kombinasyon Çizelgesi (Her Küme İçin)

"Özelleşmiş Sınıflandırıcılar ile En İyi 5 Kombinasyon (Her Küme İçin)", her küme için en iyi tahmin performansını sağlayan 5 özelleşmiş sınıflandırıcı kombinasyonu sunulmaktadır. Sıralama gerçekleştirilirken doğruluk ve f-1 skoru değerlerinin aritmetik ortalaması alınmaktadır. Eğer sınıflandırıcı kombinasyonlarının tahmin performansları eşitse, tahmin süresi performansı dikkate alınarak sıralama gerçekleştirilmektedir.

Bu tez çalışmasında kullanılan veri setlerine yönelik tespit modeli oluşturulurken, "Genel Sınıflandırıcılar ile En İyi 5 Kombinasyon (Her Küme İçin)" ve "Özelleşmiş Sınıflandırıcılar ile En İyi 5 Kombinasyon (Her Küme İçin)" tablolarından faydalanılmaktadır. Bu iki çizelge incelenerek, ilgili alt kümede en iyi sınıflandırmayı gerçekleştiren sınıflandırıcı kombinasyonu seçilmekte ve bu sınıflandırıcı kombinasyonu, modelde, ilgili kümeye kümelenen örneklemleri sınıflandırmak için kullanılmaktadır.

8.1. Kaggle Zararlı Yazılım Tespiti Veri Seti ile Elde Edilen Sonuçlar

Önerilen tespit yaklaşımı kapsamında, Kaggle Zararlı Yazılım Tespiti veri seti üzerinde elde edilen sonuçlar Çizelge 8.1-5'te, en iyi tahmin modeli Şekil 8.1'de sunulmaktadır.

Çizelge 8.1. Genel sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Tüm Veri Seti	Tahmin Süresi (s)
KSVM	D : 0,9420 F1: 0,9605	D : 0,9859 F1: 0,8	D : 0,9688 F1: 0,8755	D : 0,9601 F1: 0,9428	0,00188
KNN	D : 0,9679 F1: 0,9780	D : 0,9874 F1: 0,8391	D : 0,9800 F1: 0,9254	D : 0,9761 F1: 0,9660	0,00428
NB	D : 0,7327 F1: 0,8443	D : 0,2071 F1: 0,0881	D : 0,2237 F1: 0,2526	D : 0,4155 F1: 0,5437	1,94e-06
DT	D : 0,9788 F1: 0,9854	D : 0,09910 F1: 0,8888	D : 0,9866 F1: 0,9490	D : 0,9840 F1: 0,9771	6,39e-07
RF	D : 0,9861 F1: 0,9904	D : 0,9931 F1: 0,9090	D : 0,9914 F1: 0,9672	D : 0,9895 F1: 0,9850	1,7e-05

Çizelge 8.1. (devam) Genel sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Tüm Veri Seti	Tahmin Süresi (s)
XGB	D : 0,9682 F1: 0,9781	D : 0,9894 F1: 0,8529	D : 0,9667 F1: 0,8591	D : 0,9692 F1: 0,9550	6,63e-05
CatB	D : 0,9860 F1: 0,9903	D : 0,9915 F1: 0,8865	D : 0,9914 F1: 0,9673	D : 0,9893 F1: 0,9847	1,41e-06
AdaB	D : 0,9557 F1: 0,9696	D : 0,9862 F1: 0,8088	D : 0,9807 F1: 0,9248	D : 0,9717 F1: 0,9593	4,64e-05
EXT	D : 0,9845 F1: 0,9893	D : 0,9928 F1: 0,9052	D : 0,9913 F1: 0,9670	D : 0,9889 F1: 0,9840	3,83e-05
GB	D : 0,9836 F1: 0,9887	D : 0,9907 F1: 0,8780	D : 0,9895 F1: 0,9599	D : 0,9873 F1: 0,9818	3,49e-06

Çizelge 8.1'de sunulduğu üzere, genel sınıflandırıcılar için tahmin performansları incelendiğinde, en iyi tahmin performansı %98,95 doğruluk ve %98,5 f1 skoru ile RF sınıflandırıcısı tarafından gerçekleştirilmektedir. En iyi tahmin süresi performansı ise örneklem başına 6,39e-07 saniye ile DT sınıflandırıcısı tarafından sağlanmaktadır. En kötü tahmin performansı %41,55 doğruluk ve %54,37 f1 skoru ile NB sınıflandırıcısı tarafından, en kötü tahmin süresi performansı ise örneklem başına 0,0042 saniye ile KNN sınıflandırıcısı tarafından gerçekleştirilmektedir.

Çizelge 8.2. Özelleşmiş sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş Tahmin
KSVM	D : 0,9438 F1: 0,9617	D : 0,9770 F1: 0,5837	D : 0,9727 F1: 0,8939	D : 0,9621 F1: 0,9458
KNN	D : 0,9682 F1: 0,9782	D : 0,9875 F1: 0,8339	D : 0,9797 F1: 0,9240	D : 0,9761 F1: 0,9659
NB	D : 0,7427 F1: 0,8488	D : 0,0684 F1: 0,0774	D : 0,2380 F1: 0,2563	D : 0,4148 F1: 0,5429
DT	D : 0,9792 F1: 0,9856	D : 0,9915 F1: 0,8888	D : 0,9858 F1: 0,9458	D : 0,9838 F1: 0,9767
RF	D : 0,9858 F1: 0,9902	D : 0,9939 F1: 0,9187	D : 0,9911 F1: 0,9661	D : 0,9893 F1: 0,9847
XGB	D : 0,9682 F1: 0,9781	D : 0,9894 F1: 0,8529	D : 0,9667 F1: 0,8591	D : 0,9692 F1: 0,9550
CatB	D : 0,9862 F1: 0,9905	D : 0,9939 F1: 0,9198	D : 0,9911 F1: 0,9664	D : 0,9895 F1: 0,9850
AdaB	D : 0,9618 F1: 0,9737	D : 0,9881 F1: 0,8339	D : 0,9821 F1: 0,9311	D : 0,9749 F1: 0,9640
EXT	D : 0,9841 F1: 0,9890	D : 0,9926 F1: 0,9020	D : 0,9910 F1: 0,9657	D : 0,9885 F1: 0,9835

Çizelge 8.2. (devam) Özelleşmiş sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş Tahmin
GB	D : 0,9845 F1: 0,9893	D : 0, 9912 F1: 0, 8858	D : 0,9903 F1: 0,9633	D : 0,9882 F1: 0,9831

Çizelge 8.2'de görüldüğü gibi, özelleşmiş sınıflandırıcılar incelendiğinde, her 3 küme için de en iyi tahmin performansı CATB sınıflandırıcıları tarafından gerçekleştirilmektedir. CATB algoritmasıyla, 0 numaralı küme için %98,62 doğruluk ve %99,05 f1 skoru, 1 numaralı küme için %99,39 doğruluk ve %91,98 f1 skoru, 2 numaralı küme için %99,11 doğruluk ve %96,64 f1 skoru sağlanmaktadır. Her bir küme için ayrı ayrı eğitilmiş sınıflandırıcıların, birleştirilmiş tahmin performansları değerlendirildiğinde, CATB sınıflandırıcılarının %98,95 doğruluk ve %98,5 f1 skoru ile en iyi tahmin performansını gerçekleştirdiği görülmektedir. Özelleşmiş CATB sınıflandırıcılarının birleştirilmiş tahmin performansı, genel CATB sınıflandırıcısının tahmin performansından daha yüksektir. Özelleşmiş sınıflandırıcıların birleştirilmiş performansları genel sınıflandırıcı performanslarıyla karşılaştırılmıştır. En belirgin iyileşmenin ADAB sınıflandırıcısında gerçekleştiği görülmektedir. İlgili sınıflandırıcının doğruluk performansının %97,11'den %97,49'a, f1 skoru %95,93'ten %96,4'e yükseldiği görülmektedir. Diğer sınıflandırıcıların performansları incelendiğinde, KSVM, KNN ve GB sınıflandırıcılarında kısmi performans artışı, DT, RF ve EXT sınıflandırıcılarında ise performans azalışı görülmektedir.

Çizelge 8.3. Genel sınıflandırıcılar ile en iyi 5 kombinasyon (tüm veri seti için)

Kombinasyon	Küme-0	Küme-1	Küme-2	Tüm Veri Seti
DT+CATB+EXT	D : 0,9866 F1: 0,9908	D : 0,9944 F1: 0,9268	D : 0,9921 F1: 0,9700	D : 0,9902 F1: 0,9860
RF+ CATB+EXT	D : 0,9869 F1: 0,9909	D : 0,9939 F1: 0,9192	D : 0,9918 F1: 0,9690	D : 0,9901 F1: 0,9859
NB+ CATB+EXT	D : 0,9866 F1: 0,9907	D : 0,9928 F1: 0,9072	D : 0,9921 F1: 0,9705	D : 0,9901 F1: 0,9859
DT+RF+CATB	D : 0,9866 F1: 0,9907	D : 0,9936 F1: 0,9166	D : 0,9919 F1: 0,9692	D : 0,9900 F1: 0,9857
RF+EXT+GB	D : 0,9864 F1: 0,9906	D : 0,9936 F1: 0,9154	D : 0,9917 F1: 0,9683	D : 0,9898 F1: 0,9854

Çizelge 8.3'te görüldüğü gibi, genel sınıflandırıcılar kullanılarak en iyi tahmin performansı sağlayan üçlü sınıflandırıcı kombinasyonu, %99,02 doğruluk ve %98,60 f1 skoruyla "DT-

CATB-EXT" kombinasyonudur. Bu kombinasyonla, tüm genel ve özelleşmiş sınıflandırıcıların performanslarından daha iyi tahmin performansı elde edilmektedir.

Çizelge 8.4. Genel sınıflandırıcılar ile en iyi 5 kombinasyon (her küme için)

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	RF+ CATB+EXT	0,9869	0,9909
0	2	DT+CATB+EXT	0,9866	0,9908
0	3	NB+ CATB+EXT	0,9866	0,9907
0	4	DT+RF+CATB	0,9866	0,9907
0	5	RF+EXT+GB	0,9864	0,9906
1	1	DT+CATB+EXT	0,9944	0,9268
1	2	DT+EXT+GB	0,9941	0,9236
1	3	RF+ CATB+EXT	0,9939	0,9192
1	4	DT+RF+GB	0,9936	0,9172
1	5	DT+RF+CATB	0,9936	0,9166
2	1	DT+CATB+EXT	0,9921	0,9705
2	2	NB+ RF+CATB	0,9921	0,9701
2	3	NB+CATB+EXT	0,9920	0,9701
2	4	KNN+RF+CATB	0,9920	0,9695
2	5	DT+RF+CATB	0,9919	0,9692

Çizelge 8.4'te sunulan, genel sınıflandırıcı kombinasyonlarının her bir küme için performansları incelendiğinde, 0 numaralı küme için "RF-CATB-EXT" sınıflandırıcı kombinasyonu, 1 ve 2 numaralı kümeler için "DT-CATB-EXT" sınıflandırıcı kombinasyonu en iyi tahmin performansını sağlamaktadır. Her bir küme için, en iyi sınıflandırıcı kombinasyonlarının tahmin performansının, ilgili kümede en iyi sınıflandırma performansına sahip genel ve özelleşmiş sınıflandırıcılardan daha iyi olduğu görülmektedir.

Çizelge 8.5. Özelleşmiş sınıflandırıcılar ile en iyi 5 kombinasyon

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	CATB0+EXT0+GB0	0,9867	0,9908
0	2	RF0+CATB0+GB0	0,9866	0,9908
0	3	DT0+RF0+CATB0	0,9866	0,9907
0	4	DT0+CATB0+EXT0	0,9864	0,9906
0	5	RF0+ CATB0+EXT0	0,9864	0,9906
1	1	KNN1+DT1+CATB1	0,9950	0,9526
1	2	DT1+EXT1+GB1	0,9945	0,9481
1	3	KNN1+DT1+EXT1	0,9945	0,9478
1	4	DT1+RF1+CATB1	0,9945	0,9473
1	5	KSVM1+DT1+CATB1	0,9945	0,9473
2	1	NB2+RF2+CATB2	0,9917	0,9688
2	2	NB2+CATB2+EXT2	0,9916	0,9684

Çizelge 8.5. (devam) Özelleşmiş sınıflandırıcılar ile en iyi 5 kombinasyon

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
2	3	KNN2+RF2+CATB2	0,9916	0,9682
2	4	KSVM2+CATB2+EXT2	0,9916	0,9681
2	5	RF2+ CATB2+EXT2	0,9916	0,9680

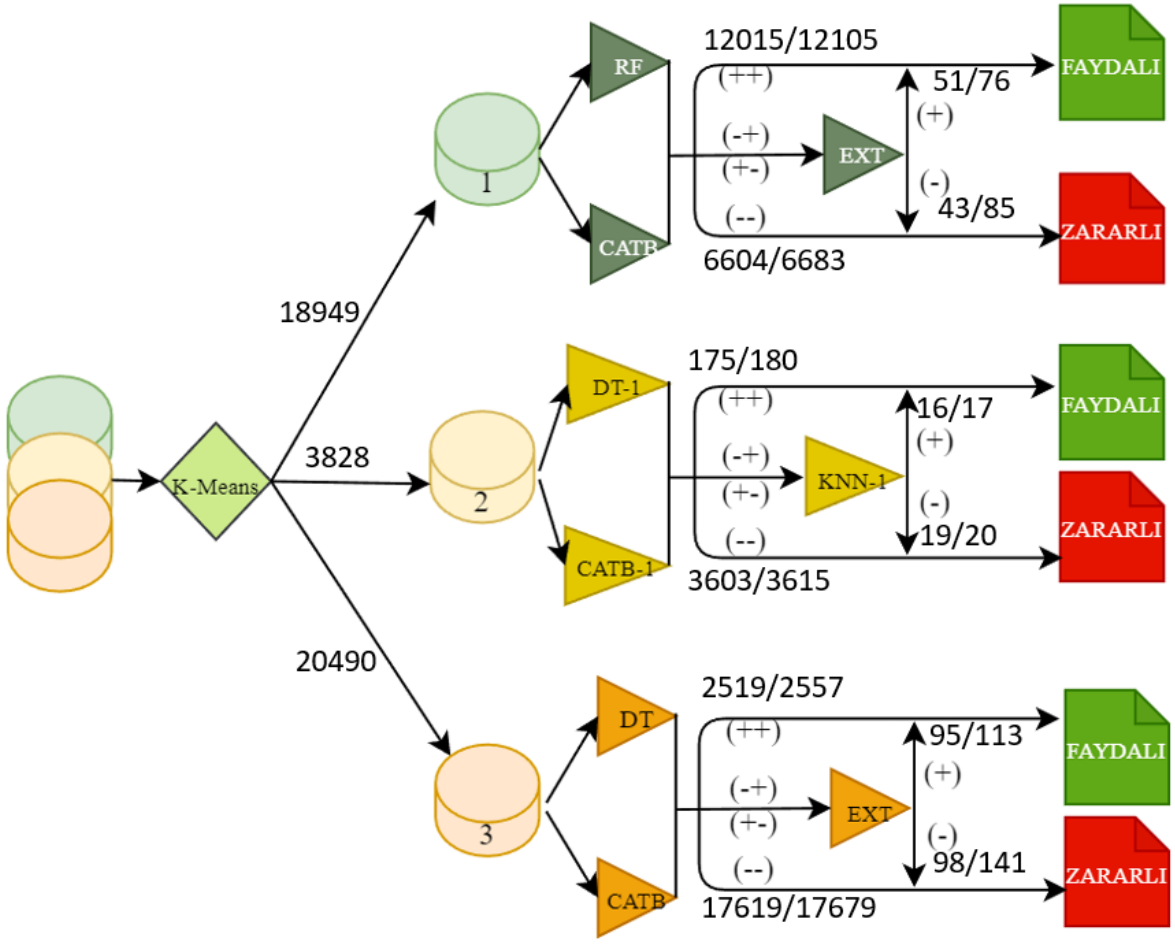
Çizelge 8.5'te sunulan, özelleştirilmiş sınıflandırıcı kombinasyonlarının tahmin performansı incelendiğinde, en iyi sonuçların;

- 0 numaralı küme için, "CATB0-EXT0-GB0" kombinasyonu tarafından %98,67 doğruluk ve %99,08 f1 skoruyla,
- 1 numaralı küme için, "KNN1-DT1-CATB1" kombinasyonu tarafından %99,50 doğruluk ve %95,25 f1 skoruyla,
- 2 numaralı küme için, "NB2-RF2-CATB2" kombinasyonu tarafından %99,17 doğruluk ve %96,88 f1 skoruyla sağlandığı görülmektedir.

Elde edilen tüm veriler göz önünde bulundurulduğunda, en yüksek tahmin performanslarının;

- 0 numaralı küme için "RF-CATB-EXT" (genel sınıflandırıcı kombinasyonu),
- 1 numaralı küme için "KNN1-DT1-CATB1" (özelleşmiş sınıflandırıcı kombinasyonu),
- 2 numaralı küme için "DT-CATB-EXT" (genel sınıflandırıcı kombinasyonu) tarafından sağlandığı görülmektedir.

Yukarıda belirtilen sınıflandırıcı kombinasyonlarının kullanıldığı model, %99,04 doğruluk ve %98,63 f1 skoru ile hem genel ve özelleşmiş sınıflandırıcılardan hem de üçlü sınıflandırıcı kombinasyonlarından daha iyi tahmin performansı sağlamaktadır. Kaggle zararlı yazılım tespit veri seti için oluşturulan bu model Şekil 8.1'de sunulmaktadır.



Şekil 8.1. Kaggle zararlı yazılım tespit veri seti için tahmin modeli

Şekil 8.1'de yer alan model incelendiğinde; 43267 (15108 faydalı, 28163 zararlı) örneklem, K-means kümeleme algoritması kullanılarak 3 ayrı kümeye ayrılmaktadır. İlk kümede 18949 (12187 faydalı, 6762 zararlı), ikinci kümede 3828 (204 faydalı, 3624 zararlı) ve üçüncü kümede 20490 (2717 faydalı, 17773 zararlı) örneklem kümelenmektedir.

Birinci kümede yapılan sınıflandırma işleminde; ilk aşamada %99,1 doğrulukla 18788 sınıflandırma, ikinci aşamada %58,4 doğrulukla 161 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %99,1'u birinci aşamada %0,9'u ikinci aşamada gerçekleştirilmektedir. Erken oybirliği ile kademeli sınıflandırma yaklaşımı sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %66,96 azalma sağlanmaktadır.

İkinci kümede yapılan sınıflandırma işleminde; ilk aşamada %99,5 doğrulukla 6795 sınıflandırma, ikinci aşamada %94,6 doğrulukla 37 sınıflandırma gerçekleştirilmektedir.

Sınıflandırma işlemlerinin %99'u birinci aşamada, %1'i ikinci aşamada gerçekleştirilmeltedir. Erken oy birliği ile kademeli sınıflandırma sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %98,9 azalma sağlanmaktadır.

Üçüncü kümede yapılan sınıflandırma işleminde; ilk aşamada %99,5 doğrulukla 20236 sınıflandırma, ikinci aşamada %75,6 doğrulukla 254 sınıflandırma gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %93,9 azalma sağlanmaktadır.

8.2. BODMAS Veri Seti ile Elde Edilen Sonuçlar

Önerilen tespit yaklaşımı kapsamında, BODMAS veri seti üzerinde elde edilen sonuçlar çizelge 8.6-10'da, en iyi tahmin modeli şekil 8.2'de sunulmaktadır.

Çizelge 8.6. Genel sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş	Tahmin Süresi (s)
KSVM	D : 0,9896 F1: 0,9828	D : 0,9957 F1: 0,9956	D : 0,9895 F1: 0,9887	D : 0,9915 F1: 0,9899	0,00368
KNN	D : 0,9821 F1: 0,9709	D : 0,9883 F1: 0,9879	D : 0,9904 F1: 0,9896	D : 0,9873 F1: 0,9851	0,00526
NB	D : 0,8673 F1: 0,7344	D : 0,9508 F1: 0,9480	D : 0,8623 F1: 0,8587	D : 0,8911 F1: 0,8686	1,11e-05
DT	D : 0,9848 F1: 0,9752	D : 0,9909 F1: 0,9906	D : 0,9897 F1: 0,9889	D : 0,9887 F1: 0,9837	6,84e-07
RF	D : 0,9948 F1: 0,9914	D : 0,9967 F1: 0,9966	D : 0,9952 F1: 0,9949	D : 0,9956 F1: 0,9948	2,61e-05
XGB	D : 0,9954 F1: 0,9925	D : 0,9978 F1: 0,9977	D : 0,9964 F1: 0,9961	D : 0,9965 F1: 0,9959	1,78e-05
CatB	D : 0,9959 F1: 0,9934	D : 0,9983 F1: 0,9982	D : 0,9966 F1: 0,9964	D : 0,9969 F1: 0,9964	2,04e-05
AdaB	D : 0,9844 F1: 0,9744	D : 0,9906 F1: 0,9902	D : 0,9870 F1: 0,9860	D : 0,9873 F1: 0,9851	8,52e-05
EXT	D : 0,9950 F1: 0,9918	D : 0,9971 F1: 0,9969	D : 0,9946 F1: 0,9942	D : 0,9955 F1: 0,9947	4,93e-05
GB	D : 0,9937 F1: 0,9898	D : 0,9973 F1: 0,9972	D : 0,9952 F1: 0,9949	D : 0,9954 F1: 0,9946	1,02e-05

Çizelge 8.6'da sunulan genel sınıflandırıcılar için tahmin performansları incelendiğinde, en

iyi tahmin performansının %99,69 doğruluk ve %99,64 f1 skoru ile CATB sınıflandırıcısı tarafından gerçekleştirildiği görülmektedir. En iyi tahmin süresi performansı ise örneklem başına 6,84e-07 saniye ile DT sınıflandırıcısı tarafından sağlanmaktadır. En kötü tahmin performansı %89,11 doğruluk ve %86,86 f1 skoru ile NB sınıflandırıcısı tarafından, en kötü tahmin süresi performansı ise örneklem başına 0,0052 saniye ile KNN sınıflandırıcısı tarafından gerçekleştirilmektedir.

Çizelge 8.7. Özelleşmiş sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş
KSVM	D : 0,9901 F1: 0,9837	D : 0,9937 F1: 0,9935	D : 0,9893 F1: 0,9885	D : 0,9909 F1: 0,9893
KNN	D : 0,9816 F1: 0,9701	D : 0,9963 F1: 0,9859	D : 0,9894 F1: 0,9886	D : 0,9866 F1: 0,9843
NB	D : 0,9297 F1: 0,8881	D : 0,9625 F1: 0,9602	D : 0,8682 F1: 0,8414	D : 0,9151 F1: 0,8948
DT	D : 0,9852 F1: 0,9759	D : 0,9912 F1: 0,9908	D : 0,9888 F1: 0,9880	D : 0,9885 F1: 0,9865
RF	D : 0,9939 F1: 0,9899	D : 0,9959 F1: 0,9957	D : 0,9940 F1: 0,9936	D : 0,9946 F1: 0,9936
XGB	D : 0,9864 F1: 0,9777	D : 0,9922 F1: 0,9920	D : 0,9903 F1: 0,9895	D : 0,9898 F1: 0,9879
CatB	D : 0,9959 F1: 0,9933	D : 0,9975 F1: 0,9975	D : 0,9964 F1: 0,9962	D : 0,9966 F1: 0,9960
AdaB	D : 0,9903 F1: 0,9840	D : 0,9945 F1: 0,9943	D : 0,9928 F1: 0,9922	D : 0,9918 F1: 0,9904
EXT	D : 0,9941 F1: 0,9903	D : 0,9966 F1: 0,9965	D : 0,9952 F1: 0,9948	D : 0,9959 F1: 0,9952
GB	D : 0,9949 F1: 0,9916	D : 0,9965 F1: 0,9963	D : 0,9952 F1: 0,9948	D : 0,9955 F1: 0,9947

Çizelge 8.7'de sunulan özelleşmiş sınıflandırıcı performansları incelendiğinde, her 3 küme için de en iyi tahmin performansının CATB sınıflandırıcıları tarafından gerçekleştirildiği görülmektedir. CATB algoritmasıyla, 0 numaralı küme için %99,59 doğruluk ve %99,33 f1 skoru, 1 numaralı küme için %99,75 doğruluk ve %99,75 f1 skoru, 2 numaralı küme için %99,64 doğruluk ve %99,62 f1 skoru performansı sağlanmaktadır. Her bir küme için ayrı ayrı eğitilmiş sınıflandırıcıların, birleştirilmiş tahmin performansları değerlendirildiğinde, CATB sınıflandırıcılarının %99,66 doğruluk ve %99,6 f1 skoru ile en iyi tahmin performansını gerçekleştirdiği görülmektedir. Ancak, özelleşmiş sınıflandırıcıların birleşik performansı, genel CATB sınıflandırıcısının performansından daha düşüktür.

Özelleşmiş sınıflandırıcıların birleştirilmiş performansları genel sınıflandırıcı performanslarıyla karşılaştırılmıştır. En belirgin iyileşme, NB sınıflandırıcısında gerçekleşmektedir. İlgili sınıflandırıcının doğruluk performansının %89,11'den %91,51'e, f1 skorunun %86,86'dan %89,48'e yükselmektedir. Belirgin artış gösteren bir diğer algoritma ADAB algoritmasıdır. ADAB algoritmasının doğruluk performansı %98,73'ten %99,18'e, f1 skoru %98,51'den %99,04'e yükselmektedir. Özelleşmiş sınıflandırma yaklaşımıyla tahmin performansı belirgin şekilde düşen tek algoritma XGB algoritmasıdır. XGB algoritmasının doğruluk performansı %99,65'ten %98,98'e, f1 skoru %99,59'dan %98,79'a düşmektedir.

Çizelge 8.8. Genel sınıflandırıcılar ile en iyi 5 kombinasyon (tüm veri seti için)

Kombinasyon	Doğruluk	F1 Skoru
RF+XGB+CATB	0,9973	0,9968
KSVM+XGB+CATB	0,9971	0,9966
XGB+CATB+EXT	0,9971	0,9966
KNN+XGB+CATB	0,9970	0,9965
RF+CATB+GB	0,9970	0,9965

Çizelge 8.8'de görüldüğü gibi, genel sınıflandırıcılar kullanılarak en iyi tahmin performansı sağlayan üçlü sınıflandırıcı kombinasyonu, %99,73 doğruluk ve %99,68 f1 skoruyla "RF-XGB-CATB" kombinasyonudur. Bu kombinasyonla, tüm genel ve özelleşmiş sınıflandırıcıların performanslarından daha iyi tahmin performansı elde edilmektedir.

Çizelge 8.9. Genel sınıflandırıcılar ile en iyi 5 kombinasyon (her küme için)

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	XGB + CATB + EXT	0,9968	0,9948
0	2	RF + XGB + CATB	0,9967	0,9946
0	3	KSVM + XGB + CATB	0,9966	0,9944
0	4	CATB + EXT + GB	0,9963	0,9940
0	5	KSVM + RF + CATB	0,9962	0,9938
1	1	KSVM + CATB + GB	0,9986	0,9986
1	1	CATB + EXT + GB	0,9986	0,9986
1	3	RF + XGB + CATB	0,9985	0,9985
1	4	RF + CATB + GB	0,9985	0,9985
1	5	KSVM + XGB + CATB	0,9984	0,9983
2	1	DT + XGB + CATB	0,9967	0,9965
2	1	KNN + XGB + CATB	0,9967	0,9965
2	1	RF + XGB + CATB	0,9967	0,9965
2	4	RF + CATB + GB	0,9965	0,9963
2	5	XGB + CATB + GB	0,9965	0,9963

Çizelge 8.9'da sunulan genel sınıflandırıcı kombinasyonlarının her bir küme için performansları incelendiğinde, 0 numaralı küme için "XGB-CATB-EXT" sınıflandırıcı kombinasyonu, 1 numaralı küme için "CATB-EXT-GB" sınıflandırıcı kombinasyonu ve 2 numaralı küme için "DT-XGB-CATB" sınıflandırıcı kombinasyonu en iyi tahmin performansını sağlamaktadır. Bu aşamada, aynı doğruluk ve f1 skoruna sahip kombinasyonlar arasında tahmin süresi performansı en iyi olan kombinasyon en iyi sınıflandırıcı kombinasyonu olarak kabul edilmektedir. Her bir küme için, en iyi sınıflandırıcı kombinasyonlarının tahmin performansının, ilgili kümede en iyi sınıflandırma performansına sahip genel ve özelleşmiş sınıflandırıcılardan daha iyi olduğu görülmektedir.

Çizelge 8.10. Özelleşmiş sınıflandırıcılar ile en iyi 5 kombinasyon

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	KSVM + CATB + GB	0,9965	0,9942
0	2	KSVM + DT + CATB	0,9963	0,9940
0	3	KNN + CATB + GB	0,9962	0,9938
0	4	KSVM + RF + CATB	0,9961	0,9935
0	5	KSVM + CATB + EXT	0,9961	0,9935
1	1	KSVM + CATB + GB	0,9977	0,9976
1	1	RF + CATB + ADAB	0,9977	0,9976
1	1	CATB + ADAB + EXT	0,9977	0,9976
1	4	CATB + EXT + GB	0,9975	0,9975
1	5	ADAB + EXT + GB	0,9975	0,9975
2	1	KNN + CATB + GB	0,9964	0,9962
2	2	CATB + EXT + GB	0,9963	0,9960
2	3	KSVM + CATB + GB	0,9962	0,9959
2	4	RF + CATB + GB	0,9962	0,9959
2	5	KNN + CATB + EXT	0,9960	0,9957

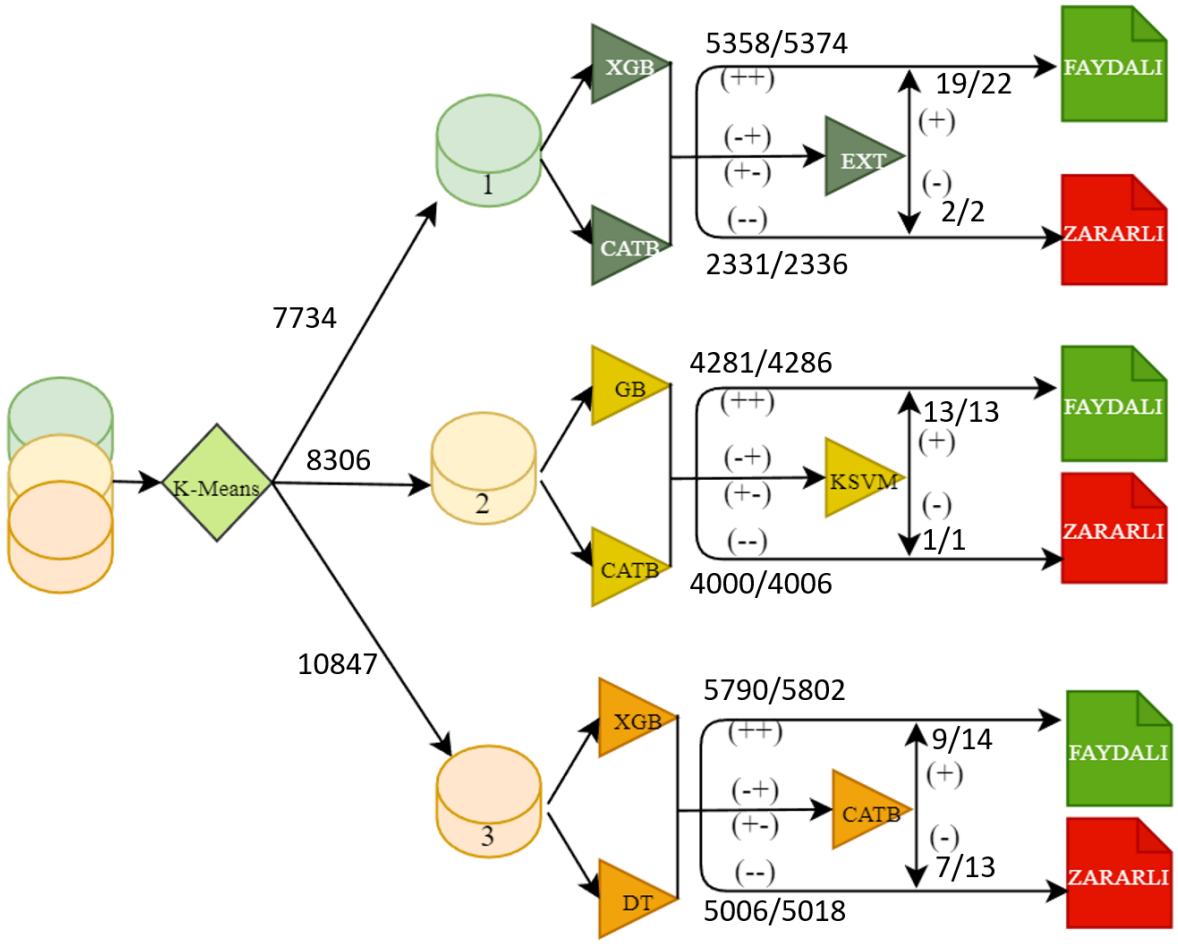
Çizelge 8.10'da sunulan özelleştirilmiş sınıflandırıcı kombinasyonlarının tahmin performansları incelendiğinde, en iyi sonuçların;

- 0 numaralı küme için "KSVM0-CATB0-GB0" kombinasyonu tarafından %99,65 doğruluk ve %99,42 f1 skoruyla,
- 1 numaralı küme için "KSVM1-CATB1-GB1" kombinasyonu tarafından %99,77 doğruluk ve %99,76 f1 skoruyla,
- 2 numaralı küme için "KNN2-CATB2-GB2" kombinasyonu tarafından %99,64 doğruluk ve %99,62 f1 skoruyla sağlandığı görülmektedir.

Elde edilen tüm veriler göz önünde bulundurulduğunda; en yüksek tahmin performanslarının;

- 0 numaralı küme için “XGB-CATB-EXT”,
- 1 numaralı küme için “KSVM-CATB-GB”,
- 2 numaralı küme için “DT-XGB-CATB” sınıflandırıcı kombinasyonları tarafından sağlandığı görülmektedir.

Yukarıda belirtilen sınıflandırıcı kombinasyonlarının kullanıldığı model, %99,74 doğruluk ve %99,77 f1 skoru ile hem genel ve özelleşmiş sınıflandırıcılardan hem de üçlü sınıflandırıcı kombinasyonlarından daha iyi tahmin performansı sağlamaktadır. BODMAS veri seti için oluşturulan tespit modeli Şekil 8.2’de sunulmaktadır.



Şekil 8.2. BODMAS veri seti için tahmin modeli

Şekil 8.2’de yer alan model incelendiğinde; 26887 (15499 faydalı, 11388 zararlı) örneklem, K-means kümeleme algoritması kullanılarak 3 ayrı kümeye ayrılmaktadır. İlk kümede 7734 (5382 faydalı, 2352 zararlı), ikinci kümede 8306 (4300 faydalı, 4006 zararlı) ve üçüncü kümede 10847 (5817 faydalı, 5030 zararlı) örneklem kümelenmektedir.

Birinci kümede yapılan sınıflandırma işleminde; ilk aşamada %99,73 doğrulukla 7710 sınıflandırma, ikinci aşamada %87,5 doğrulukla 24 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %99,68'i birinci aşamada %0,32'si ikinci aşamada gerçekleştirilmektedir. Erken oybirliği ile kademeli sınıflandırma yaklaşımı sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %56,3 azalma sağlanmaktadır.

İkinci kümede yapılan sınıflandırma işleminde; ilk aşamada %99,87 doğrulukla 8292 sınıflandırma, ikinci aşamada %100 doğrulukla 14 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %99,83'u birinci aşamada, %0,17'si ikinci aşamada gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %99,1 azalma sağlanmaktadır.

Üçüncü kümede yapılan sınıflandırma işleminde; ilk aşamada %99,77 doğrulukla 10820 sınıflandırma, ikinci aşamada %59,25 doğrulukla 27 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %99,75'i birinci aşamada, %0,25'i ikinci aşamada gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %72,69 azalma sağlanmaktadır.

Önerilen tespit yaklaşımının etkinliğinin BODMAS veri seti ile yapılan diğer çalışmaların sonuçlarıyla karşılaştırılması çizelge 8.11'de sunulmaktadır.

Çizelge 8.11. BODMAS veri seti kullanılan çalışmaların karşılaştırılması

Çalışma	Doğruluk	F1 Skoru
Ramadhan, F.H. ve diğerleri [24]	0,9962	0,9956
Qikai Lu [25]	0,97	0,9699
Önerilen Tespit Yaklaşımı	0,9974	0,9977

Önerilen tespit yaklaşımının performansı BODMAS veri seti kullanılarak yapılan diğer çalışmalarla karşılaştırıldığında (Çizelge 8.11), önerilen tespit yaklaşımının hem doğruluk hem de f1 skoru açısından, diğer çalışmalardan daha iyi tahmin performansı sergilediği görülmektedir.

8.3. EMBER 2018 Veri Seti ile Elde Edilen Sonuçlar

Önerilen tespit yaklaşımı kapsamında, EMBER veri seti üzerinde elde edilen sonuçlar çizelge 8.12-16'da, en iyi tahmin modeli şekil 8.3'te sunulmaktadır.

Çizelge 8.12. Genel sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş	Tahmin Süresi (s)
KSVM	D : 0,9806 F1: 0,9111	D : 0,8928 F1: 0,9218	D : 0,9072 F1: 0,8993	D : 0,9099 F1: 0,9113	0,07537
KNN	D : 0,9724 F1: 0,8777	D : 0,9143 F1: 0,9357	D : 0,8898 F1: 0,8801	D : 0,9086 F1: 0,9086	0,28473
NB	D : 0,8812 F1: 0,0589	D : 0,6546 F1: 0,7876	D : 0,6391 F1: 0,4649	D : 0,6724 F1: 0,6653	1,11e-05
DT	D : 0,9694 F1: 0,8696	D : 0,9081 F1: 0,9313	D : 0,8987 F1: 0,8899	D : 0,9103 F1: 0,9106	1,29e-06
RF	D : 0,9918 F1: 0,9613	D : 0,9473 F1: 0,9610	D : 0,9424 F1: 0,9366	D : 0,9499 F1: 0,9499	9,96e-05
XGB	D : 0,9921 F1: 0,9635	D : 0,9676 F1: 0,9758	D : 0,9543 F1: 0,9497	D : 0,9637 F1: 0,9636	1,44e-05
CatB	D : 0,9930 F1: 0,9676	D : 0,9674 F1: 0,9757	D : 0,9576 F1: 0,9534	D : 0,9654 F1: 0,9653	4,85e-06
AdaB	D : 0,9526 F1: 0,7960	D : 0,7903 F1: 0,8592	D : 0,8145 F1: 0,8047	D : 0,8208 F1: 0,8335	0,00015
EXT	D : 0,9874 F1: 0,9399	D : 0,9396 F1: 0,9552	D : 0,9323 F1: 0,9249	D : 0,9413 F1: 0,9410	0,0001
GB	D : 0,9809 F1: 0,9130	D : 0,9462 F1: 0,9602	D : 0,9340 F1: 0,9284	D : 0,9440 F1: 0,9445	1,28e-05

Çizelge 8.12'de sunulan genel sınıflandırıcılar için tahmin performansları incelendiğinde, en iyi tahmin performansının %96,54 doğruluk ve %96,53 f1 skoru ile CATB sınıflandırıcısı tarafından gerçekleştirildiği görülmektedir. En iyi tahmin süresi performansı ise örneklem başına 1,29e-06 saniye ile DT sınıflandırıcısı tarafından sağlanmaktadır. En kötü tahmin performansı %67,24 doğruluk ve %66,53 f1 skoru ile NB sınıflandırıcısı tarafından, en kötü tahmin süresi performansı ise örneklem başına 0,28473 saniye ile KNN sınıflandırıcısı tarafından gerçekleştirilmektedir.

Çizelge 8.13. Özelleşmiş sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş
KSVM	D : 0,9755 F1: 0,8809	D : 0,8895 F1: 0,9196	D : 0,9078 F1: 0,8984	D : 0,9084 F1: 0,9091

Çizelge 8.13. (devam) Özelleşmiş sınıflandırıcılar için tahmin performansları

Sınıflandırıcı	Küme-0	Küme-1	Küme-2	Birleştirilmiş
KNN	D : 0,9795 F1: 0,9073	D : 0,9201 F1: 0,9407	D : 0,9023 F1: 0,8944	D : 0,9179 F1: 0,9185
NB	D : 0,9048 F1: 0,5405	D : 0,4045 F1: 0,2381	D : 0,6987 F1: 0,5949	D : 0,6086 F1: 0,4392
DT	D : 0,9727 F1: 0,8801	D : 0,9006 F1: 0,9252	D : 0,8957 F1: 0,8853	D : 0,9063 F1: 0,9057
RF	D : 0,9855 F1: 0,9291	D : 0,9467 F1: 0,9607	D : 0,9411 F1: 0,9349	D : 0,9483 F1: 0,9482
XGB	D : 0,9851 F1: 0,9295	D : 0,9640 F1: 0,9731	D : 0,9578 F1: 0,9537	D : 0,9633 F1: 0,9632
CatB	D : 0,9848 F1: 0,9289	D : 0,9637 F1: 0,9729	D : 0,9535 F1: 0,9494	D : 0,9610 F1: 0,9610
AdaB	D : 0,9735 F1: 0,8765	D : 0,8929 F1: 0,9216	D : 0,8729 F1: 0,8628	D : 0,8919 F1: 0,8935
EXT	D : 0,9845 F1: 0,9243	D : 0,9394 F1: 0,9552	D : 0,9281 F1: 0,9198	D : 0,9389 F1: 0,9385
GB	D : 0,9822 F1: 0,9166	D : 0,9594 F1: 0,9697	D : 0,9508 F1: 0,9461	D : 0,9577 F1: 0,9576

Çizelge 8.13'te, özelleşmiş sınıflandırıcılar için tahmin performansları görülmektedir. 0 numaralı küme için, RF algoritması %98,55 doğruluk ve XGB algoritması %92,95 f1 skoru ile en iyi tahmin performansını sağlamaktadır. XGB algoritması, 1 numaralı küme için %96,40 doğruluk ve %97,31 f1 skoru, 2 numaralı küme için %95,78 doğruluk ve %95,38 f1 skoru ile en iyi tahmini gerçekleştirmektedir. Her bir küme için ayrı ayrı eğitilmiş sınıflandırıcıların, birleştirilmiş tahmin performansları değerlendirildiğinde, XGB sınıflandırıcılarının %96,33 doğruluk ve %96,32 f1 skoru ile en iyi tahmin performansını gerçekleştirdiği görülmektedir.

Özelleşmiş sınıflandırıcıların birleştirilmiş performansları genel sınıflandırıcı performanslarıyla karşılaştırılmıştır. Özelleşmiş sınıflandırıcı kullanımıyla ADAB ve GB sınıflandırıcılarında belirgin şekilde iyileşme görülmektedir. ADAB için doğruluk %82,08'den %89,19'a, f1 skoru %83,35'ten %89,35'e yükselmektedir. GB için doğruluk %94,40'tan %95,77'ye, f1 skoru %94,45'ten %95,76'ya yükselmektedir. En belirgin performans düşüşü NB için gerçekleşmiş olup, doğruluğun %67,24'ten %60,86'ya, f1 skorunun %66,53'ten %43,92'ye düştüğü görülmektedir.

Çizelge 8.14. Genel sınıflandırıcılar ile en iyi 5 kombinasyon (tüm veri seti için)

Kombinasyon	Doğruluk	F1 Skoru
KNN+XGB+CATB	0,9676	0,9676
DT+XGB+CATB	0,9655	0,9654
XGB+CATB+EXT	0,9653	0,9652
XGB+CATB+GB	0,9643	0,9643
KSVM+XGB+CATB	0,9639	0,9638

Çizelge 8.14'te görüldüğü gibi, genel sınıflandırıcılar kullanılarak en iyi tahmin performansı sağlayan üçlü sınıflandırıcı kombinasyonu, %96,76 doğruluk ve %96,76 f1 skoruyla "KNN-XGB-CATB" kombinasyonudur. Bu kombinasyonla, tüm genel sınıflandırıcıların performanslarından daha iyi tahmin performansı elde edilmektedir.

Çizelge 8.15. Genel sınıflandırıcılar ile en iyi 5 kombinasyon (her küme için)

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	XGB + CATB + EXT	0,9936	0,9703
0	2	NB + XGB + CATB	0,9935	0,97
0	3	RF + XGB + CATB	0,9935	0,9699
0	4	KSVM + XGB + CATB	0,9934	0,9697
0	5	DT + RF + CATB	0,9931	0,9684
1	1	KNN + XGB + CATB	0,9683	0,9763
1	2	DT + XGB + CATB	0,9674	0,9756
1	3	XGB + CATB + EXT	0,9669	0,9753
1	4	XGB + CATB + GB	0,9663	0,9749
1	5	KSVM + XGB + CATB	0,9662	0,9748
2	1	KNN + XGB + CATB	0,9615	0,9578
2	2	RF + XGB + CATB	0,9691	0,9550
2	3	XGB + CATB + EXT	0,9577	0,9535
2	4	XGB + CATB + GB	0,9565	0,9523
2	5	KSVM + XGB + CATB	0,9555	0,9511

Çizelge 8.15'te sunulan genel sınıflandırıcı kombinasyonlarının her bir küme için tahmin performansları incelendiğinde, en iyi sonuçların;

- 0 numaralı küme için "XGB-CATB-EXT" sınıflandırıcı kombinasyonu tarafından %99,36 doğruluk ve %97,03 f1 skoruyla,
- 1 numaralı küme için "KNN-XGB-CATB" sınıflandırıcı kombinasyonu tarafından %96,83 doğruluk ve %97,63 f1 skoruyla,
- 2 numaralı küme için "KNN-RF-CATB" sınıflandırıcı kombinasyonu tarafından %96,15 doğruluk ve %95,78 f1 skoruyla sağlandığı görülmektedir.

Çizelge 8.16. Özelleşmiş sınıflandırıcılar ile en iyi 5 kombinasyon

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	KNN0 + DT0 + CATB0	0,9883	0,9458
0	2	KNN0 + DT0 + RF0	0,9879	0,9430
0	3	KNN0 + DT0 + XGB0	0,9876	0,9422
0	4	KNN0 + DT0 + GB0	0,9871	0,9403
0	5	KNN0 + DT0 + EXT0	0,9873	0,94
1	1	NB1 + XGB1 + CATB1	0,9674	0,9756
1	2	KNN1 + XGB1 + CATB1	0,9663	0,9749
1	2	XGB1 + CATB1 + EXT1	0,9661	0,9747
1	4	RF1 + XGB1 + CATB1	0,9651	0,9740
1	5	NB1 + CATB1 + GB1	0,9648	0,9736
2	1	KNN2 + XGB2 + CATB2	0,9600	0,9564
2	2	XGB2 + CATB2 + EXT2	0,9582	0,9542
2	3	RF2 + XGB2 + CATB2	0,9570	0,9528
2	4	KNN2 + XGB2 + GB2	0,9568	0,9526
2	5	KNN2 + CATB2 + GB2	0,9566	0,9527

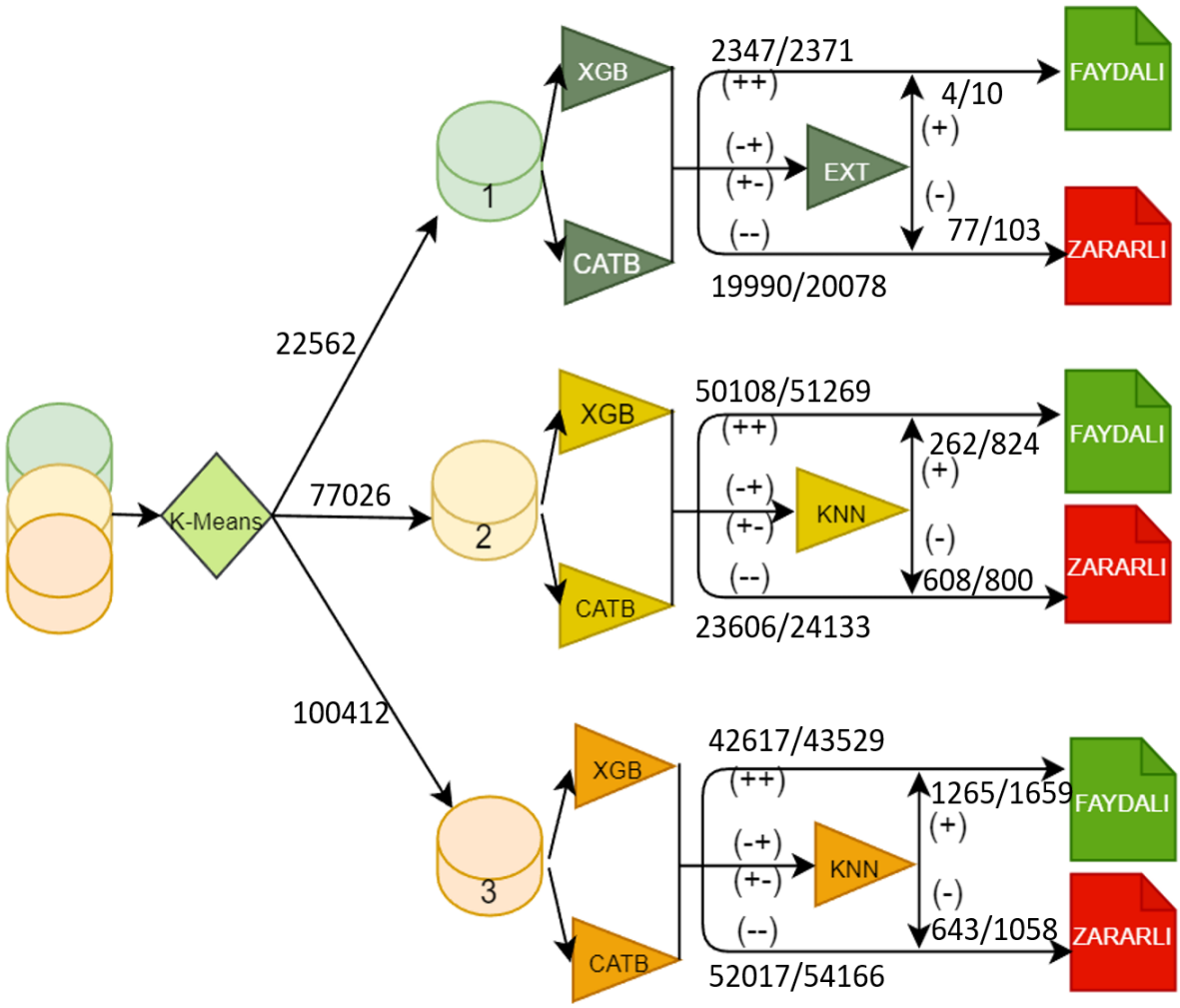
Çizelge 8.16'da sunulan özelleştirilmiş sınıflandırıcı kombinasyonlarının tahmin performansları incelendiğinde, en iyi sonuçların;

- 0 numaralı küme için "KNN0-DT0-CATB0" kombinasyonu tarafından %98,83 doğruluk ve %94,58 f1 skoruyla,
- 1 numaralı küme için "NB1-XGB1-CATB1" kombinasyonu tarafından %96,74 doğruluk ve %97,56 f1 skoruyla,
- 2 numaralı küme için "KNN2-XGB2-CATB2" kombinasyonu tarafından %96,00 doğruluk ve %95,64 f1 skoruyla sağlandığı görülmektedir.

Elde edilen tüm veriler göz önünde bulundurulduğunda; en iyi tahmin performanslarının;

- 0 numaralı küme için "XGB-CATB-EXT" sınıflandırıcı kombinasyonu tarafından %99,36 doğruluk ve %97,03 f1 skoruyla
- 1 numaralı küme için "KNN-XGB-CATB" kombinasyonu tarafından %96,83 doğruluk ve %97,63 f1 skoruyla,
- 2 numaralı küme için "KNN2-RF2-CATB2" kombinasyonu tarafından %96,15 doğruluk ve %95,78 f1 skoruyla sağlandığı görülmektedir.

Yukarıda belirtilen sınıflandırıcı kombinasyonlarının kullanıldığı model, %96,77 doğruluk ve %96,77 f1 skoru ile hem genel ve özelleşmiş sınıflandırıcılardan hem de üçlü sınıflandırıcı kombinasyonlarından daha iyi tahmin performansı sağlamaktadır. EMBER 2018 veri seti için oluşturulan tespit modeli Şekil 8.3'te sunulmaktadır.



Şekil 8.3. EMBER 2018 veri seti için tahmin modeli

Şekil 8.3'te yer alan model incelendiğinde; 200000 (100000 faydalı, 100000 zararlı) örneklem, K-means kümeleme algoritması kullanılarak 3 ayrı kümeye ayrılmaktadır. İlk kümede 22562 (2465 faydalı, 20097 zararlı), ikinci kümede 77026 (51089 faydalı, 25937 zararlı) ve üçüncü kümede 100412 (46446 faydalı, 53966 zararlı) örneklem kümelenmektedir.

Birinci kümede yapılan sınıflandırma işleminde; ilk aşamada %99,5 doğrulukla 22449 sınıflandırma, ikinci aşamada %71,68 doğrulukla 113 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %99,5'i birinci aşamada %0,5'ü ikinci aşamada gerçekleştirilmektedir. Erken oybirliği ile kademeli sınıflandırma yaklaşımı sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %83,76 azalma sağlanmaktadır.

İkinci kümede yapılan sınıflandırma işleminde; ilk aşamada %97,76 doğrulukla 75402 sınıflandırma, ikinci aşamada %53,57 doğrulukla 1624 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %97,89'u birinci aşamada, %2,11'i ikinci aşamada gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %97,88 azalma sağlanmaktadır.

Üçüncü kümede yapılan sınıflandırma işleminde; ilk aşamada %96,94 doğrulukla 97615 sınıflandırma, ikinci aşamada %70,22 doğrulukla 2717 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %97,21'i birinci aşamada, %2,79'u ikinci aşamada gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre tahmin süresinde %97,2 azalma sağlanmaktadır.

Önerilen tespit yaklaşımının etkinliğinin EMBER 2018 veri seti ile yapılan diğer çalışmaların sonuçlarıyla karşılaştırılması çizelge 8.17'de sunulmaktadır.

Çizelge 8.17. EMBER 2018 veri seti kullanılan çalışmaların karşılaştırılması

Çalışma	Doğruluk	F1 Skoru
Raff, E. ve diğerleri [26]	%93,29	Veri yok
Marais, B ve diğerleri [27]	%94	%93,4
Sumit, S.L. ve Adamuthe A.C. [28]	%94,09	%88,66
Nguyen, A.T. ve diğerleri [29]	%94,72	Veri yok
Thosar, K. ve diğerleri [30]	%96	%96
Önerilen Tespit Yaklaşımı	%96,77	%96,77

Çizelge 8.17'de sunulduğu üzere, önerilen tespit yaklaşımının performansı EMBER 2018 veri seti kullanılarak yapılan diğer çalışmalarla karşılaştırıldığında hem doğruluk hem de f1 skoru açısından, diğer çalışmalardan daha iyi tahmin performansı sergilediği görülmektedir.

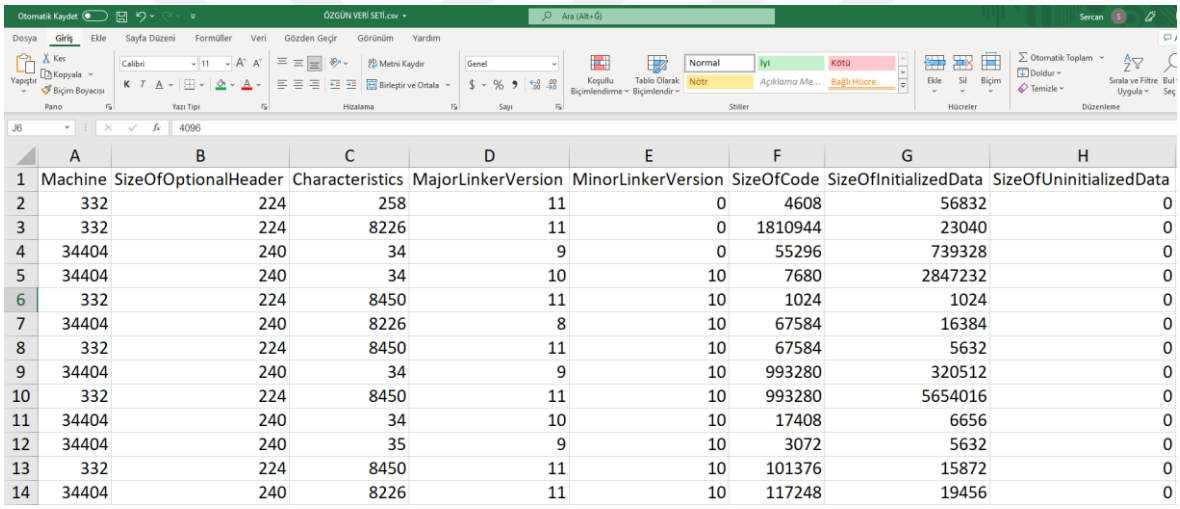
8.4. Modelin Canlı Ortamda Uygulanması ve Özgün Veri Seti ile Elde Edilen Sonuçlar

Bu bölümde, modelin canlı ortamda uygulanması ve özgün veri seti ile elde edilen sonuçlar sunulmaktadır. Modelin canlı ortamda uygulamasına yönelik çalışma üç safhada gerçekleştirilmektedir. Birinci safha dosyalardan özellik çıkarımı ve eğitim veri setinin

oluşturulması safhası, ikinci safha sınıflandırıcıların eğitilmesi ve dışa aktarılması safhası, son safha ise eğitilen modelin farklı bir ortamda çalıştırılması safhasıdır.

8.4.1. Özellik çıkarımı ve eğitim veri setinin oluşturulması

Çalışmanın bu safhasında, öncelikle VirusShare üzerinden indirilen zararlılar ile Windows 8.1 işletim sisteminin sistem dosyaları olmak üzere 977'si zararlı 5323'ü faydalı 6300 dosya toplanmaktadır. Listesi çizelge 5.6'da yer alan özellikler, Kali Linux işletim sistemine sahip bir sanal makinada, Python dili ile geliştirilen bir kod kullanılarak elde edilmektedir. Elde edilen bu özellikler özgün veri seti olarak kullanılmak üzere ".csv" formatında dışa aktarılmaktadır (Resim 8.1).



	A	B	C	D	E	F	G	H
1	Machine	SizeOfOptionalHeader	Characteristics	MajorLinkerVersion	MinorLinkerVersion	SizeOfCode	SizeOfInitializedData	SizeOfUninitializedData
2	332	224	258	11	0	4608	56832	0
3	332	224	8226	11	0	1810944	23040	0
4	34404	240	34	9	0	55296	739328	0
5	34404	240	34	10	10	7680	2847232	0
6	332	224	8450	11	10	1024	1024	0
7	34404	240	8226	8	10	67584	16384	0
8	332	224	8450	11	10	67584	5632	0
9	34404	240	34	9	10	993280	320512	0
10	332	224	8450	11	10	993280	5654016	0
11	34404	240	34	10	10	17408	6656	0
12	34404	240	35	9	10	3072	5632	0
13	332	224	8450	11	10	101376	15872	0
14	34404	240	8226	11	10	117248	19456	0

Resim 8.1. PE dosyalardan elde edilen veri seti

Resim 8.1'deki ekran görüntüsünde, özgün veri seti kapsamında çıkarılan özelliklerin bir kısmı yer almaktadır.

8.4.2. Sınıflandırıcıların eğitilmesi ve dışa aktarılması

Özellik Çıkarımı ve Eğitim Veri Setinin Oluşturulması aşamasında elde edilen veri seti, Google Colaboratory ortamında, önerilen tespit modeli için ihtiyaç duyulan sınıflandırıcıların eğitildiği koda girdi olarak verilmektedir. İlgili kodun çalıştırılması neticesinde Çizelge 8.18 ve Çizelge 8.19'teki sonuçlar elde edilmektedir. Bu aşamada, eğitilen ölçeklendirme, kümeleme ve sınıflama algoritmaları farklı ortamlarda kullanılmak üzere dışarı aktarılmaktadır.

Çizelge 8.18. Uygulama kapsamında genel sınıflandırıcılar için kümelere göre en iyi doğruluk ve f1 skorları

Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	NB+ADAB+EXT	0,9695	0,9782
1	1	NB+DT+GB	1,0	1,0
2	1	NB+DT+GB	1,0	1,0

Çizelge 8.18'de sunulan genel sınıflandırıcı kombinasyonlarının tahmin performansı incelenmiş olup:

- 0 numaralı küme için en iyi tahmin performansının NB-ADAB-EXT kombinasyonuyla %96,95 doğruluk ve %97,82 f1 skoruyla sağlandığı görülmektedir.
- 1 ve 2 numaralı kümelerde birden fazla sınıflandırıcı kombinasyonu %100 doğrulukla sınıflandırma yapmaktadır. Bu sebeple tahmin süresi en kısa olan sınıflandırıcı kombinasyonları (NB-DT-GB) ilgili küme için en iyi sınıflandırıcı kombinasyonu olarak kabul edilmektedir.

Çizelge 8.19. Uygulama kapsamında özel sınıflandırıcılar için kümelere göre en iyi doğruluk ve f1 skorları

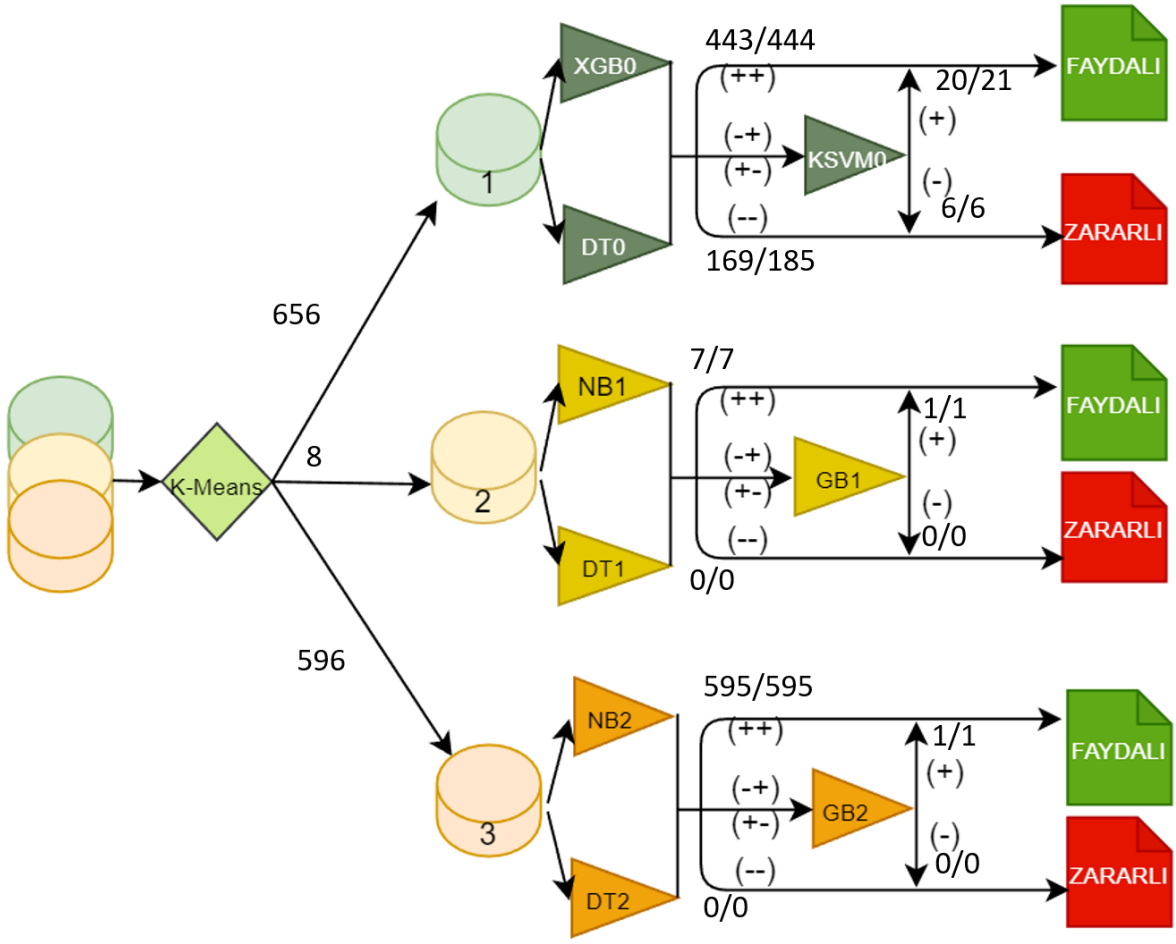
Küme	Sıra	Kombinasyon	Doğruluk	F1 Skoru
0	1	KSVM0+DT0+XGB0	0,9725	0,9803
1	1	NB1+DT1+GB1	1,0	1,0
2	1	NB2+DT2+GB2	1,0	1,0

Çizelge 8.19'te sunulan özelleşmiş sınıflandırıcıları kombinasyonlarının tahmin performansı incelenmiş olup:

- 0 numaralı küme için en iyi tahmin performansının KSVM0+DT0+XGB0 kombinasyonuyla %97,25 doğruluk ve %98,03 f1 skoruyla sağlandığı görülmektedir.
- 1 ve 2 numaralı kümelerde birden fazla sınıflandırıcı kombinasyonunun %100 doğrulukla sınıflandırma yapması sebebiyle tahmin süresi en kısa olan sınıflandırıcı kombinasyonları (NB1-DT1-GB1, NB2-DT2-GB2) ilgili küme için en iyi sınıflandırıcı kombinasyonu olarak kabul edilmektedir.

Tablolarda sunulan verilere ek olarak, en iyi genel sınıflandırıcı %98,33 doğruluk ve %99,01 f1 skoruyla ADAB, en iyi genel sınıflandırıcı kombinasyonu %98,41 doğruluk ve %99,05 f1 skoruyla NB-ADAB-EXT'dir.

Önerilen tespit yaklaşımında, 0 numaralı küme için KSVM0-DT0-XGB0, 1 numaralı küme için NB1-DT1-GB1 ve 2 numaralı küme için NB2-DT2-GB2 sınıflandırıcı kombinasyonlarının kullanıldığı durumda, %98,57 doğruluk ve %99,15 f1 skoru ile en iyi tahmin performansı sağlanmaktadır. Uygulama kapsamında oluşturulan tespit modeli Şekil 8.4'te sunulmaktadır.



Şekil 8.4. Özgün veri seti için tespit modeli

Şekil 8.4'te yer alan model incelendiğinde, 1260 (1083 faydalı, 177 zararlı) örneklem, K-means kümeleme algoritmasıyla 3 ayrı kümeye ayrılmaktadır. İlk kümede 656 (479 faydalı, 177 zararlı), ikinci kümede 8 (8 faydalı) ve üçüncü kümede 596 (596 faydalı) örneklem kümelenmektedir.

Birinci kümede yapılan sınıflandırma işleminde; ilk aşamada %97,29 doğrulukla 629 sınıflandırma, ikinci aşamada %96,3 doğrulukla 27 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %95,88'i birinci aşamada %4,12'si ikinci aşamada

gerçekleştirilmektedir. Erken oybirliği ile kademeli sınıflandırma yaklaşımı sayesinde, her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre, tahmin süresinde %83,09 azalma sağlanmaktadır.

İkinci kümede yapılan sınıflandırma işleminde; ilk aşamada %100 doğrulukla 8 sınıflandırma, ikinci aşamada %100 doğrulukla 1 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %88,89'u birinci aşamada, %11,11'i ikinci aşamada gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde, her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre, tahmin süresinde %49,45 azalma sağlanmaktadır.

Üçüncü kümede yapılan sınıflandırma işleminde; ilk aşamada %100 doğrulukla 595 sınıflandırma, ikinci aşamada %100 doğrulukla 1 sınıflandırma gerçekleştirilmektedir. Sınıflandırma işlemlerinin %99,83'ü birinci aşamada, %00,16'sı ikinci aşamada gerçekleştirilmektedir. Erken oy birliği ile kademeli sınıflandırma sayesinde, her üç sınıflandırıcının da aynı aşamada tahmin yaptığı modele göre, tahmin süresinde %50,44 azalma sağlanmaktadır.

8.4.3. Farklı bir ortamda tespit modelinin çalıştırılması

Bir önceki adımda, her küme için en iyi sonucu veren özelleştirilmiş sınıflandırıcı kombinasyonları dışa aktarılarak, Kali Linux üzerinde yer alan ve modelle özdeş tahmin gerçekleştiren Python koduna girdi olarak verilmektedir. Eğitim setine dahil edilmeyen 20 zararlı ve 80 faydalı dosya üzerinde uygulanan tahminde, oluşturulan model %100 doğrulukla tahmin gerçekleştirmektedir.

8.5. Genel Sonuçlar

8.1-8.4'üncü bölümlerde sunulan tablolar incelendiğinde, Kaggle zararlı yazılım tespiti veri setinde %99,04 doğruluk ve %98,63 f1 skoru, BODMAS veri setinde %99,74 doğruluk ve %99,77 f1 skoru, EMBER 2018 veri setinde %96,77 doğruluk ve %96,77 f1 skoru, özgün veri setinde ise %98,57 doğruluk ve %99,15 f1 skoru sağlanmaktadır.

Çizelge 8.20. Önerilen tespit yaklaşımının literatürdeki çalışmalarla karşılaştırılması

Veri Seti	Çalışma	Doğruluk	F1 Skoru
BODMAS	Ramadhan, F.H. ve diğerleri [24]	0,9962	0,9956
	Qikai Lu [25]	0,97	0,9699
	Önerilen Tespit Yaklaşımı	0,9974	0,9977
EMBER 2018	Raff, E. ve diğerleri [26]	%93,29	Veri yok
	Marais, B ve diğerleri [27]	%94	%93,4
	Sumit, S.L. ve Adamuthe A.C. [28]	%94,09	%88,66
	Nguyen, A.T. ve diğerleri [29]	%94,72	Veri yok
	Thosar, K. ve diğerleri [30]	%96	%96
	Önerilen Tespit Yaklaşımı	%96,77	%96,77

Çizelge 8.20'de sunulan tablo incelendiğinde, önerilen tespit yaklaşımının hem BODMAS hem de EMBER 2018 veri setleri kullanılarak gerçekleştirilen çalışmalardan daha iyi tespit performansı sergilediği görülmektedir. Literatürde Kaggle Zararlı Yazılım Tespiti veri seti kullanılarak gerçekleştirilen bir çalışmaya rastlanılmadığından, tabloda yer almamaktadır.

Modelin gerçek ortamda kullanılabilirliğinin değerlendirilmesi kapsamında, gerçek zararlı dosyaları ve faydalı sistem dosyalarından elde edilen özgün veri seti kullanılmaktadır. Gerçekleştirilen çalışmada, modelin gerçek ortamda etkin bir şekilde kullanılabildiği görülmektedir.

9. SONUÇ VE ÖNERİLER

Sürekli evrilen ve gelişen zararlı yazılımlarla mücadele, siber güvenlik araştırmacılarının üzerinde en çok çalışma yaptıkları alanlardan bir tanesidir. Özellikle, daha önce tespit edilmemiş zararlı yazılımların tespitinin gerçekleştirilmesi için, makine öğrenmesi destekli tespit yaklaşımlarının kullanımı öne çıkmaktadır. Zararlı yazılım tespiti kapsamında makine öğrenmesi yöntemlerinin kullanılması, imza veri tabanıyla eşleşmeyen ve uzman kişiler tarafından tanımlanmış kurallarla tespit edilemeyen zararlılar tespit edilmesine olanak sağlamaktadır.

Bu tez çalışmasında, son kullanıcılar tarafından yaygın olarak kullanılan Windows işletim sistemleri etkileyen çalıştırılabilir zararlı yazılımların, yüksek doğruluk ve f1 skoru ile tespit edilmesi amaçlanmaktadır. Bu amacın gerçekleştirilmesi maksadıyla, denetimli ve denetimsiz öğrenme metotlarının birlikte kullanıldığı bir toplu öğrenme modeli sunulmaktadır.

Modelin geliştirilmesi aşamasında, denetimsiz öğrenme algoritması olarak K-Means algoritması, denetimli öğrenme algoritmaları olarak KSVM, KNN, NB, DT, RF, XGB, CATB, ADAB, EXT ve GB algoritmaları kullanılmaktadır.

Sunulan tespit yaklaşımının etkinliğinin ortaya konulabilmesi maksadıyla Kaggle Zararlı Yazılım Tespiti veri seti, BODMAS veri seti, EMBER 2018 veri seti ve çalışma kapsamında hazırlanan özgün veri seti kullanılmaktadır.

Tespit yaklaşımının geliştirilmesi aşamasında, sınıflandırıcıların hızlı bir şekilde eğitilmesine imkân veren Kaggle Zararlı Yazılım Tespiti veri seti kullanılmıştır. Önerilen tespit yaklaşımı olgunlaştıktan sonra literatürde yer alan çalışmalarda kullanılan BODMAS ve EMBER 2018 veri setleriyle çalışmalar gerçekleştirilmiştir. Ayrıca, tespit yaklaşımının kullanılabilirliğinin ve etkinliğinin gerçek ortamda test edilmesi maksadıyla gerçek zararlılar da kullanılarak özgün veri seti oluşturulmuş ve tespit yaklaşımının gerçek ortamda zararlı yazılım tespiti gerçekleştirmesi sağlanmıştır.

Tespit yaklaşımının tahmin süresi performansını artırmak maksadıyla sınıflandırıcıların iki kademede konumlandırılarak sınıflandırma gerçekleştirmesi sağlanmaktadır. Bu yöntem sayesinde tahmin süresinde %56 ile %99 arasında azalma sağlandığı görülmektedir.

Önerilen tespit yaklaşımı kapsamında kullanılan özelleştirilmiş sınıflandırıcıların, bazı algoritmaların tahmin performansını arttırdığı, bazı algoritmaların tahmin performansını ise azalttığı görülmektedir. Özelleşmiş sınıflandırıcı kullanımı sayesinde performansı en çok artan sınıflandırma algoritması -EMBER 2018 veri seti için- ADAB olup, sınıflandırıcının doğruluğu %82,08'den %89,19'a, f1 skoru %83,35'ten %89,35'e yükselmektedir. En belirgin performans düşüşü ise yine EMBER 2018 veri setinde kullanılan NB algoritmasında görülmekte olup, sınıflandırıcının doğruluğu %67,24'ten %60,86'ya, f1 skoru %66,53'ten %43,92'ye düşmektedir.

Önerilen tespit yaklaşımının performansı BODMAS ve EMBER 2018 veri setleri kullanılarak yapılan diğer çalışmalarla [24-30] karşılaştırılmış olup, önerilen tespit yaklaşımının her iki veri seti için de en iyi tahmin performansını sağladığı görülmektedir.

Gelecek çalışmalar kapsamında;

- Kümeleme aşamasının öncesine, dosyaların hash değerini alarak, bir zararlı veri tabanı üzerinde imza karşılaştırması yapılarak bilinen zararlı yazılımlarının tespit edilmesi,
- Kümeleme aşamasında; DBScan, Birsh, spektral kümeleme, hiyerarşik kümeleme gibi kümeleme algoritmalarıyla yapılan kümelemenin sınıflandırıcıların tahmin performansı üzerine etkilerinin ortaya konulması,
- Kümeleme neticesinde tamamı aynı sınıfa ait örneklemlerden oluşan alt kümelerin sınıflandırmasının kümeleme esnasında gerçekleştirilmesinin etkilerinin ortaya konulması,
- Kademeli konumlandırmada, ikinci kademede konumlandırılmış sınıflandırıcıların tahmin performansının iyileştirilmesi için farklı yaklaşımların (bu sınıflandırıcıların sadece ilk kademe sınıflandırıcıların tahmin gerçekleştirmediği verilerle eğitilmesi) uygulanması ile daha iyi tahmin performansına sahip modellerin geliştirilebileceği değerlendirilmektedir.

Sonuç olarak hem son kullanıcıların hem de kurum ve kuruluşların sahip oldukları bilgi varlıklarının korunması için etkin ve efektif şekilde zararlı yazılım tespiti gerçekleştirilmesi bir zorunluluktur. Yüksek doğruluk, hassasiyet ve kesinlikle zararlı yazılım tespiti yapabilecek ve ilk kez kullanılan taktik ve teknikleri uygulayan zararlı yazılımları bilgi varlıklarına zarar vermeden engelleyebilecek yöntemlere duyulan ihtiyaç, bilgi varlıkları kullanımda olduğu sürece, baki kalacaktır. Bu tez çalışmasında, önerilen tespit yaklaşımının, duyulan bu güvenlik ihtiyacının karşılamasına yardımcı olabileceği ortaya konmuştur.

KAYNAKLAR

1. İnternet: Hane halkı bilişim teknolojileri kullanım araştırması. Web: https://data.tuik.gov.tr/Bulten/Index?p=Hanehalki-Bilisim-Teknolojileri-(BT)-Kullanim-Arastirmasi-2020-33679, Son Erişim Tarihi: 10.06.2022.

2. İnternet: Zararlı Yazılım İstatistikleri. Web: https://www.av-test.org/en/statistics/malware, Son Erişim Tarihi: 10.06.2022.

3. Gavrilut, D., Cimpoesu, M., Anton, D., and Ciortuz, L. (2009). *Malware detection using machine learning*. International Multiconference on Computer Science and Information Technology, 735-741.

4. Firdausi, A., Iim, C., Erwin,A., and Nugroho, A.S. (2010). *Analysis of machine learning techniques used in behavior-based malware detection*. Second International Conference on Advances in Computing, Control, and Telecommunication Technologies, 201-203.

5. Menahem, E., Shabtai, A., Rokach, L., and Elovici, Y. (2009). Improving malware detection by applying multi-inducer ensemble. *Computational Statistics & Data Analysis*, 53(4), 1483-1494.

6. Roseline, S. A. (2019). *Towards efficient malware detection and classification using multilayered random forest ensemble technique*. 2019 International Carnahan Conference on Security Technology (ICCST), 2019, 1-6.

7. Jahromi, N.A., Hashemi, S., Dehghantanha, A., Choo, K. K. R., Karimipour, H., Newton, D. E., and Parizi, R. M. (2020). An improved two-hidden-layer extreme learning machine for malware hunting. *Computers and Security*, 89.

8. Euh, S., Lee, H., Kim, D., and Hwang, D. (2020). Comparative analysis of low-dimensional features and tree-based ensembles for malware detection systems. *IEEE Access*, 8, 76796– 76808.

9. Cohen, A., Nissim, N., and Elovici, Y. (2020). Maljpeg: machine learning based solution for the detection of malicious jpeg images. *IEEE Access*, 8, 19997–20011.

10. Singh, J., and Singh, J. (2020). Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms. *Information and Software Technology*, 121.

11. Dai, Y., Li, H., Qian, Y., Yang, R., and Zheng, M. (2019). Smash: a malware detection method based on multi-feature ensemblelearning. *IEEE Access*, 7, 112588–112597.

12. Damaševičius, R., Venčkauskas, A., Toldinas, J., and Grigaliūnas, Š. (2021). Ensemble-based classification using neural networks and machine learning models for windows pe malware detection. *Electronics*, 10(4), 1–26.

13. Gupta, D., and Rani, R. (2020). Improving malware detection using big data and ensemble learning. *Computers and Electrical Engineering*, 86.

14. Verma, V., Muttoo, S. K., and Singh, V. B. (2020). Multiclass malware classification via first- and second-order texture statistics. *Computers and Security*, 97.

15. Ficco, M. (2021). Malware analysis by combining multiple detectors and observation windows. *IEEE Transactions on Computers*, 71(6), 1276-1290.

16. Gao, Y., Hasegawa, H., Yamaguchi, Y., and Shimada, H. (2022). Malware detection using lightgbm with a custom logistic loss function. *IEEE Access*, 10, 47792-47804.

17. Zhang, Y., Liu, Z., and Jiang, Y. (2022). The classification and detection of malware using soft relevance evaluation. *IEEE Transactions on Reliability*, 71(1), 309-320.

18. Kumar, R., and Subbiah, G. (2022). Zero-day malware detection and effective malware analysis using shapley ensemble boosting and bagging approach. *Sensors*, 22.

19. Dong, Y., Liu, Z., Yan, Y., Wang, Y., Peng, T., and Zhang, J. (2017). Machine Learning for Analyzing Malware. *Network and System Security*, 386-398.

20. Taha, A.A., and Malebary, S.J. (2021). Hybrid classification of android malware based on fuzzy clustering and gradient boosting machine. *Neural Computing and Applications*, 33, 6721-6732.

21. Badhani, S., and Mutto, S.K. (2019). Cendroid: a cluster-ensemble classifier for detecting malicious android applications. *Computer and Security*, 85, 25-40.

22. Yang, H., Li, S., Wu, X., Lu, H., and Han, H. (2019). A novel solutions for malicious code detection and family clustering based on machine learning. *IEEE Access*, Special Section on Security and Privacy in Emerging Decentralized Communication Environments, 148853– 148860.

23. Appice, A., Andresini, G., and Malerba, D. (2020). Clustering-aided multi-view classification: a case study on android malware detection. *Journal of Intellident Information Systems*, 55, 1-26.

24. Ramadhan, F.H., Suryani, V., and Mandala, S. (2021). Analysis study of malware classification portable executable using hybrid machine learning. *International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, 86-91.

25. Lu, Q. (2021). *An investigation on self-attentive models for malware classification*, (Yüksek Lisans Tezi, University of Alberta, 1990). University of Alberta Education and Research Archieve, 26-39.

26. Raff, E., Fleshman, W., Zak, R., Anderson, H.S., Filar, B., and McLean, M. (2020). Classifying sequences of extreme length with constant memory applied to malware detection. *Arxiv*, 2012.09390.

27. Marais, B., Quertier, T., and Chesneau, C. (2021). Malware analysis with artificial intelligence and a particular attention onresulrs interpretability. *Arxiv*, 2107.11100v1.

28. Lad, S.S., and Adamuthe, A.C. (2022). Improved deep learning model for static pe files malware detection and classification. *International Journal of Computer Network and Information Security*, 2, 14-26.

29. Nguyen, A.T., Raff, E., Nicholas, C., and Holt, J. (2021). Leveraging uncertainty for improved static malware detection under extreme false positive constraints. *Arxiv*, 2108.04081v1.

30. Thosar, K., Tiware, P., and Jyothula, R. (2021). *Effective malware detection using gradient boosting and convolutional neural network.* IEEE Bombay Section Signature Conference (IBSSC), 1-4.

31. Karresand, M. (2003). *Separating trojan horses, viruses, and worms: a proposed taxonomy of software weapons.* IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, 127–134

32. Wang, X., Yu, W., Champion, A., Fu, X., and Xuan, D. (2007). *Detecting worms via mining dynamic program execution.* Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks, SecureComm, 412–421.

33. Ye, Y., Li, T., Adjeroh, D., and Iyengar, S.S. (2017). A survey on malware detection using data mining techniques. *ACM Computing Surveys*, 50(3), 1–40.

34. Zaki, A., and Humphrey, B. (2014). *Unveiling the kernel: rootkit discovery using selective automated kernel memory differencing.* Virus Bulletin Conference, 239–256.

35. Scaife, N., Carter, H., Traynor, P., and Butler, K. R. B. (2016). *CryptoLock (and drop it): stopping ransomware attacks on user data.* In Proceedings of International Conference on Distributed Computing Systems, 303–312.

36. Mohamed, G. A. N., and Ithnin, N. B. (2017). Survey on representation techniques for malware detection system. *American Journal of Applied Sciences*, 14(11), 1049–1069.

37. Chowdhury, M., and Rahman, A. (2018). *Malware analysis and detection using data mining and machine learning classification.* International Conference on Applications and Techniques in Cyber Security and Intelligence, 580, 266–274.

38. Dusane, P.S., and Pavithra, Y. (2020). Logic bomb: an insider attack. *International Journal of Advanced Trends in Computer Science and Engineering*, 99(3), 3662-3665.

39. Canbek, G., ve Sağıroğlu, Ş. (2007). Bilgisayar sistemlerine yapılan saldırılar ve türleri: bir inceleme. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 23, 1-12.

40. Sudhakar, and Kumar, S. (2020). An emerging threat fileless malware: a survey and research challenges. *Cybersecurity*, 3(1), 1-12.

41. Komatwar, R., and Kokare, M. (2021). A survey on malware detection and classification. *Journal of Applied Security Research*, 16(3), 390–420.

42. İnternet: Loebenberger, D. Evolution From Creeper to Storm. Web: https://cosec.bit.uni-bonn.de/fileadmin/user_upload/teaching/07ws/malware/evolution_report.pdf, Son Erişim Tarihi: 10.06.2022.

43. Internet: Mueller, P. and Yadegari, B. The Stuxnet Worm. Web: https://www2.cs.arizona.edu/~collberg/Teaching/466566/2012/Resources/presentations/topic9-final/report.pdf, Son Erişim Tarihi: 10.06.2022.

44. Mohurle, S., and Patil, M. (2017). A brief study of wannacry threat: ransomware attack 2017. *International Journal of Advanced Research in Computer Science*, 8(5).

45. Etaher, N., Weir, G.R.S., Alazab, M. (2015). From zeus to zitmo: trends in banking malware. *IEEE Trustcom/BigDataSE/ISPA*, 1386-1391.

46. Alazab, M., Venkatraman, S., Watters, P., Alazab, M., and Alazab, A. (2011). Cybercrime: The case of obfuscated malware. In: Georgiadis, C.K., Jahankhani, H., Pimenidis, E., Bashroush, R., Al-Nemrat, A. (eds). *Global Security, Safety and Sustainability & e-Democracy*, Springer, Berlin, 99, 204-211.

47. Hajioff, S., and McKee, M. (2000). The 'I love you' virus and its implications for genodiversity. *Journal of the Royal Society of Medicine*, 93(8), 398–399.

48. Garber, L. Melissa Virus Creates a New Type of Threat. *Computer*, 32(6), 16-19.

49. Modlin A., Gregory A., Odebode I., Hodson D.D., and Grimaila M.R. (2021). CovidLock attack simulation. *Advances in Parallel & Distributed Processing, and Applications,* 10, 25-34.

50. Kuraku, S., and Kalla, D. (2020). Emotet malware: a banking credentials stealer. *International Organization of Scientific Research Journal*, 31-41.

51. Kyurkchiev, N., Iliev, A., Rahnev, A., and Terzieva, T. (2019). A new analysis of cryptolocker ransomware and welchia worm propagation behaviour. *Communications in Applied Analysis*, 23(2), 359-382.

52. Kim, D., Soh, W., and Kim, S. (2015). Design of quantification model for prevent of cryptolocker. *Indian Journal of Science and Technology*, 8(19).

53. Dunham, K. (2004). Mydoom miseries. *Information Systems Security*, 13(4), 2-5.

54. İnternet: MyDoom: do you get it yet. (2004, Şubat). *Network Security*, 2004, 2, 13-15. Web: https://www.sciencedirect.com/science/article/abs/pii/S1353485804000364 18 Ekim 2022'de alınmıştır.

55. Moore, D., Shannon, C., and Brown, J.. (2002). *Code-Red: a case study on the spread and victims of an internet worm.* Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment (IMW '02), 273–284.

56. Egele, M., Scholte, T., Kırda, E., and Kruegel, C. (2012). A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys*, 44(2), 1-42.

57. Ismail, S.J.I., Hendrawan, and Rahardjo, B. (2020). *A survey on malware detection technology and future trends.* 14th International Conference on Telecommunication Systems, Services, and Applications, 1-6.

58. Souri, A., and Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. *Human-centric Computing and Information Sciences*, 8(1), 1-22.

59. Bazrafshan, Z., Hashemi, H., Fard, S.M.H., and Hamzeh, A. (2013). *A survey on heuristic malware detection techniques.* The 5th Conference on Information and Knowledge Technology, 113-120.

60. Jonathan A.P., Marpaung, Sain, M., and Lee, H. (2012). *Survey on malware evasion techniques: state of the art and challenges.* 14th International Conference on Advanced Communication Technology (ICACT), 744-749.

61. Xu, R., and Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645-678.

62. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., and Wu, A.Y. (2002). An efficient K-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 881-892.

63. Amari, S., and Wu, S. (1999). Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*. 12(6), 783-789.

64. Patle, A. and Chouhan, D.S. (2013). *SVM kernel functions for classification.* 2013 International Conference on Advances in Technology and Engineering (ICATE), 1-9.

65. Kotsiantis, S. (2007). Supervised machine learning: a review of classification techniques. *Informatica*, 31, 249-268.

66. Rincy, T., and Gupta, R. (2020). *Ensemble learning techniques and its efficiency in machine learning: a survey.* 2nd International Conference on Data, Engineering and Applications (IDEA), 1-6.

67. Cao, Y., Miao, Q.G., Liu, J.C., and Gao, L. (2013). Advance and prospects of adaboost algorithm. *Acte Automatica Sinica*, 39(6), 745-758.

68. Friedman, J.H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4), 367-378.

69. Chen, T., and Guestrin, C. (2016). *XGBoost: a scalable tree boosting system*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 785–794.

70. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., and Gulin, A. (2018). *CatBoost: unbiased boosting with categorical features.* 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), 1-11.

71. Daho, M.E.H., Settouti, N., Lazouni, M.E.A., and Chikh, M.E.A. (2014). *Weighted vote for trees aggregation in Random Forest*. 2014 International Conference on Multimedia Computing and Systems (ICMCS), 438-443.

72. İnternet: Google Colaboratory. Web: https://colab.research.google.com/, Son Erişim Tarihi: 10.06.2022.

73. İnternet: Scikit-Learn. Web: https://scikit-learn.org/stable/, Son Erişim Tarihi: 10.06.2022.

74. İnternet: PeFile Kütüphanesi. Web: https://github.com/erocarrera/pefile, Son Erişim Tarihi: 10.06.2022.

75. İnternet: Virus Share web sitesi. Web: https://virusshare.com, Son Erişim Tarihi: 10.06.2022.

76. İnternet: PE Başlık Formatı. Web: https://docs.microsoft.com/en-us/windows/win32/ debug/ pe-format, Son Erişim Tarihi: 10.06.2022.

77. İnternet: Kaggle Zararlı Yazılım Tespiti Veri Seti. Web: https://www.kaggle.com/ c/malware-detection, Son Erişim Tarihi: 10.06.2022.

78. İnternet: BODMAS Veri Seti. Web: https://whyisyoung.github.io/BODMAS, Son Erişim Tarihi: 10.06.2022.

79. Yang, L., Ciptadi, A., Laziuk, I., Ahmadzadeh, A., and Wang, G. (2021). BODMAS: an open dataset for learning based temporal analysis of pe malware. *IEEE Security and Privacy Workshops (SPW)*, 78-84.

80. İnternet: Library to Instrument Executable Formats (LIEF). Web: https://lief-project. github.io, Son Erişim Tarihi: 10.06.2022.

81. İnternet: Sfone.b. Web: https://www.trendmicro.com/vinfo/hk/threat-encyclopedia/ malware/worm.win32.sfone.b/. Son Erişim Tarihi: 12.08.2022.

82. İnternet:Wacatac. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/trojan.win 32.wacatac.usxvpga19. Son Erişim Tarihi: 12.08.2022.

83. İnternet: Upatre. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/upatre. Son Erişim Tarihi: 12.08.2022.

84. İnternet: Wabot. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/trojan.win 32.wabot.disc/. Son Erişim Tarihi: 12.08.2022.

85. İnternet: Smal. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/TROJ_S WISYN.SMAL/. Son Erişim Tarihi: 12.08.2022.

86. İnternet: Smia. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/worm_ga nelp.smia. Son Erişim Tarihi: 12.08.2022.

87. İnternet: Dinwod. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/trojan.w in32.dinwod.b/. Son Erişim Tarihi: 12.08.2022.

88. İnternet: Mira. Web: https://www.pcrisk.com/removal-guides/14744-mira-ransomware. Son Erişim Tarihi: 12.08.2022.

89. İnternet: Berbew. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/bkdr_ber bew.i. Son Erişim Tarihi: 12.08.2022.

90. İnternet: Silly_cz. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/ malware/WORM_ SILLY.CZ/. Son Erişim Tarihi: 12.08.2022.

91. Anderson, H. A., and Roth, P. (2018). EMBER: an open dataset for training static pe malware machine learning models. *Arxiv*, 1804.04637v2.

92. İnternet: Elastic Malware Benchmark for Empowering Researchers. Web: https://github.com/elastic/ember. Son Erişim Tarihi: 04.08.2022.

93. İnternet: EMBER Improvements. Web: https://www.camlis.org/2019/talks/roth. Son Erişim Tarihi: 04.08.2022.

94. İnternet: Xtrat. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/XTRAT. Son Erişim Tarihi: 12.08.2022.

95. İnternet: Zbot. Web: https://www.malwarebytes.com/blog/detections/spyware-zbot. Son Erişim Tarihi: 12.08.2022.

96. İnternet: Ramnit. Web: https://www.f-secure.com/v-descs/virus_w32_ramnit.shtml. Son Erişim Tarihi: 12.08.2022.

97. İnternet: Install Monster. Web: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=SoftwareBundler:Win32/InstallMonster. Son Erişim Tarihi: 12.08.2022.

98. İnternet: Sality. Web: https://threats.kaspersky.com/en/threat/Virus.Win32.Sality/. Son Erişim Tarihi: 12.08.2022.

99. İnternet: Zusy. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/tspy_zb ot.zusy. Son Erişim Tarihi: 12.08.2022.

100. İnternet: Vtflooder. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/troj_vtflooder.t. Son Erişim Tarihi: 12.08.2022.

101. İnternet: Emotet Malware. Web: https://www.cisa.gov/uscert/ncas/alerts/aa20-280a. Son Erişim Tarihi: 12.08.2022.

102. İnternet: Fareit. Web: https://threats.kaspersky.com/en/threat/Trojan-PSW.Win32.Fareit/. Son Erişim Tarihi: 12.08.2022.

103. İnternet: Adposhel. Web: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/adware.w in32.adposhel.ab. Son Erişim Tarihi: 12.08.2022.

104. İnternet: Runouce. Web: https://www.f-secure.com/v-descs/runouce.shtml. Son Erişim Tarihi: 19.08.2022.

105. İnternet: Chir. Web: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?name=Win32%2FChir. Son Erişim Tarihi: 19.08.2022.

106. İnternet: Virut Botnet. Web: https://digital.nhs.uk/cyber-alerts/2018/cc-2829. Son Erişim Tarihi: 19.08.2022.

107. İnternet: Madangel. Web: https://securitynews.sonicwall.com/xmlpost/madangel-infects-executable-files-on-the-system-august-2-2013. Son Erişim Tarihi: 19.08.2022.

108. İnternet: Autoit. Web: https://threats.kaspersky.com/en/threat/Worm.Win32.Autoit. Son Erişim Tarihi: 19.08.2022.

109. İnternet: Cryptos. Web: https://threats.kaspersky.com/en/threat/Trojan.MSIL.Cryptos. Son Erişim Tarihi: 19.08.2022.

110. İnternet: Sirefef. Web: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Win32%2fSirefef. Son Erişim Tarihi: 19.08.2022.

111. İnternet: Nimda. Web: https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?name=Worm:Win32/Nimda. Son Erişim Tarihi: 19.08.2022.

112. İnternet: Lugia. Web: https://www.enigmasoftware.com/virusslugina-removal. Son Erişim Tarihi: 19.08.2022.

GAZİ GELECEKTİR...