Journal of the Faculty of Engineering and Architecture of Gazi University 38:3 (2023) 1781-1792

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Malware detection using image-based features and machine learning methods

Aslıhan Güngör[1]*, İbrahim Alper Doğru[2], Necaattin Barışçı[2], Sinan Toklu[2]

[1]Department of Computer Science, Institute of Informatics, Gazi University, 06680, Kavaklıdere, Ankara, Türkiye
[2]Department of Computer Engineering, Faculty of Technology, Gazi University, 06560, Yenimahalle, Ankara, Türkiye

Highlights:

- Extracting features from image files
- Classification using machine learning
- Detecting malware

Keywords:

- Malware
- Image Processing
- Feature extraction
- Machine Learning

Article Info:

Research Article
Received: 12.09.2021
Accepted: 03.02.2022

DOI:

10.17341/gazimmfd.994289

Correspondence:

Author: Aslıhan Güngör
e-mail: aslihan.gungor@gazi.edu.tr
phone: +90 545 746 9606

Graphical/Tabular Abstract

The use of machine learning methods has a potentially significant contribution to the detection of malware, where traditional methods are insufficient. API files of Android applications are converted to grayscale image file. Malware detection method has been developed by extracting global features from image files, creating feature matrices and classifying them with machine learning methods. The methodology followed in the study is shown in Figure A.

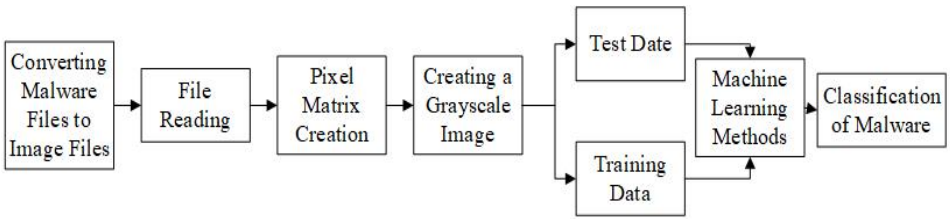


Figure A. Followed methodology in the study

Purpose:

Android devices have started to take place in every aspect of our lives and have become the target of malware. Developing technology, malicious software is marketed by hiding and traditional methods are insufficient. This study aims to detect malicious software by using image processing methods and machine learning techniques.

Theory and Methods:

API files have been converted to grayscale images for malware detection. using image processing methods. Feature matrices were created from these images and classified by machine learning methods. Malimg data set was used in the training and testing of the machine learning method. And its results are compared with previous studies on the same dataset.

Results:

In the study, global features were extracted on the malimg dataset and classified by machine learning methods. Accuracy and standard deviation of results with K-fold cross validation LR: 0.8422 (0.0310), LDA: 0.9111 (0.0198), KNN: 0.9672 (0.0122), CART: 0.9616 (0.0141), RF: 0.9744 (0.0093), NB: 0.9311 (0.0143), SVM: 0.6894 (0.0298). *When the developed model is compared with the previous studies on the same data set, a higher accuracy rate was obtained*

Conclusion:

In this study, hu moments were classified by machine learning methods in malware detection using image-based global features consisting of haralick texture and color diagrams. Higher accuracy was obtained with the applied method. In future studies, it is planned to investigate the effect of local features on the results, and to classify them with deep learning methods and compare them with machine learning methods.

Journal of the Faculty of Engineering and Architecture of Gazi University 38:3 (2023) 1781-1792

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Görüntü tabanlı özelliklerden ve makine öğrenmesi yöntemlerinden faydalanılarak kötücül yazılım tespiti

Aslıhan Güngör[1]*, İbrahim Alper Doğru[2], Necaattin Barışçı[2], Sinan Toklu[2]
[1]Gazi Üniversitesi, Bilişim Enstitüsü, Bilgisayar Bilimleri Bölümü, 06680 Kavaklıdere, Ankara, Türkiye
[2]Gazi Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 06560 Yenimahalle, Ankara, Türkiye

ÖNEÇIKANLAR

- Görüntü dosyalarından özellikleri çıkarma
- Makine öğrenimini kullanarak sınıflandırma
- Kötü amaçlı yazılım algılama

Makale Bilgileri
Araştırma Makalesi
Geliş: 12.09.2021
Kabul: 03.02.2022

DOI:
10.17341/gazimmfd.994289

Anahtar Kelimeler:
Kötü amaçlı yazılım, görüntü işleme, özellik çıkarılması, makine öğrenmesi

ÖZ

Android cihazların hayatın içinde daha çok yer alması kötü amaçlı yazılımların da hedefi haline gelmesine sebep olmuştur. Kötü amaçlı yazılımların tespit edilmesi ve bu yazılımlardan doğacak kayıpların ve zararların önlenmesi önem arz etmektedir. Bu amaçla kötü amaçlı yazılım tespitine yönelik çeşitli çalışmalar yapılmaktadır. Son zamanlarda görüntüye dayalı yöntemler ve makine öğrenmesi çalışmaları ön plana çıkmaktadır. Bu çalışmalarda statik ve dinamik analizde kullanılan ikili dosyalar görüntü dosyalarına çevrilmektedir. Görüntülerden çıkarılan global ve yerel özellikler çeşitli makine öğrenmesi metotları ile sınıflandırılmaktadır. Bu çalışmada malimg veri seti üzerinde global özellikler çıkarılarak (2000, 532) boyunda bir özellik matrisi elde edilmiştir. Elde edilen bu özellikler makine öğrenme yöntemleri (LR, LDA, K-NN, CART, RF, NB, SVM) kullanılarak sınıflandırılmıştır. Sonuçlar K-kat çaprazlama doğrulama yöntemi değerlendirilerek K-NN ile %96,72 RF ile en yüksek %97,44 doğruluk oranı elde edilmiştir. Bu çalışma aynı veri seti üzerinde yapılan diğer çalışmalarla kıyaslandığında daha yüksek bir doğruluk değerine ulaşarak literatüre katkı sağlamaktadır.

Malware detection using image-based features and machine learning methods

HIGHLIGHTS

- Extracting features from image files
- Classification using machine learning
- Detecting malware

Article Info
Research Article
Received: 12.09.2021
Accepted: 03.02.2022

DOI:
10.17341/gazimmfd.994289

Keywords:
Malware, image processing, feature extraction, machine learning

ABSTRACT

As Android devices occupy more of people's lives, they have also become a target of malicious software. It is important to detect malicious software and to prevent the losses and damages that may arise from this software. For this purpose, various studies are being carried out about malware detection. Recently, image-based methods and machine learning studies have come to the fore. In these studies, binary files used in static and dynamic analysis are converted into image files. Global and local features extracted from the images are classified by various machine learning methods. In this study, global features were extracted on the malimg dataset and a feature matrix (2000, 532) long was obtained. The obtained features were classified using machine learning methods (LR, LDA, K-NN, CART, RF, NB, SVM). The results were evaluated using the K-fold crossover validation method, and a highest accuracy rate of 96.72% was obtained with K-NN and 97.44% with RF. This study contributes to the literature by reaching a higher accuracy value compared to other studies on the same data set.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *aslihan.gungor@gazi.edu.tr, iadogru@gazi.edu.tr, nbarisci@gazi.edu.tr, stoklu@gazi.edu.tr / Tel: +90 545 746 9606

1. Giriş (Introduction)

İnternetin yaygın kullanılmaya başlamasıyla birlikte gelişen teknoloji ile mobil cihazlar hayatın içinde daha çok yer almaya başlamıştır. Birçok alanda mobil uygulamalar geliştirilerek çeşitli platformlar üzerinden kullanıcılara sunulmaktadır. Öğretimden ulaşıma, haberden eğlenceye, eğitimden sağlığa birçok mobil uygulama geliştirilmiş ve hizmete sunulmuştur. Özellikle sosyal mecralar çevrimiçi kullanıcı sayısının artmasında önemli rol oynayarak akıllı telefonların bir adım öne çıkmasını sağlamaktadır. Bu nedenle, güvenli mobil ve kablosuz ağ uygulamaları bütün bu organizasyonlar için önem arz etmektedir [1]. Son yıllarda hayatın her alanında yer almaya başlayan mobil cihazlar veriye erişmeyi ve veriyi paylaşmayı daha da kolay bir hale getirmiştir. İnsanlar günlük yaşamlarında sürekli olarak mobil iletişim araçlarını kullanmakta ve bu cihazlarda mesaj, fotoğraf, konum gibi kişisel verileri depolamaktadır. Şifre çalma, klavye dinleme, ağ izleme gibi işlemleri gerçekleştirmek maksadıyla kötü amaçlı yazılımlar geliştirilmektedir. Bu bilgiler cihazları veri hırsızlığı için cazip kılmaktadır. Bu da bu cihazları kötü amaçlı saldırıların hedefi haline getirmiştir. Bununla birlikte güvenlik sorunları gündeme gelmiştir. Kötü amaçlı yazılımlar beraberinde mobil cihazların güvenliği kavramını getirmektedir. Güvenlik kavramının gündeme gelmesine müteakip cihazların korunmasına yönelik çalışmalar başlamıştır [1].

2020 yılında işletmeyi kesintiye uğratan fidye yazılımı saldırıları neredeyse iki katına çıkmıştır [2]. E-posta Güvenlik Durumu Raporu'nda Mimecast, kuruluşların%51'inin iş operasyonlarında en azından kısmi bir kesintiye yol açan bir fidye yazılımı saldırısı yaşadığını tespit etmiştir [2]. Ocak 2021 itibariyle Google, haftada yaklaşık 600-800 kötü amaçlı yazılım bulaşmış site tespit etmiştir [2]. Google'ın Şeffaflık Raporu'na göre, 17 Ocak 2021 itibarıyla 2.195 milyon web sitesi "Güvenli Tarama Tarafından Tehlikeli Olarak Görülen Siteler" kategorisi listesinde yer almıştır [2]. Bunların büyük çoğunluğu (2,1 milyondan fazla) kimlik avı siteleridir [2]. “2021 Siber Güvenlik Durumu” raporunda işletmelerin ve devlet kurumlarının büyük çoğunluğunda siber güvenlik alanında uzman personelin yetersiz olduğu belirtilmiş [1], tehditlerin farkında olan kurum ve kişilerin koruyucu önlemler alması gerektiği ve personelini bu konuda bilinçlendirilerek eğitilmesinin bir ihtiyaç olduğu vurgulanmıştır [1].

Kötü amaçlı yazılımların sınıflandırılmasında iki temel yaklaşım mevcuttur. Statik analiz yaklaşımlarında yazılım kodu parçalara ayırma yoluyla yürütme mantığı ortaya çıkarılarak saldırı davranışlarını tetikleyen kalıplar bulunmaktadır [3]. Dinamik analiz yaklaşımında ise sanal bir ortamda çalıştırılarak saldırı davranış özelliklerini belirlemek için hareketler izlenerek bir rapor oluşturulmaktadır [3]. Bu rapor genellikle tek bir davranışı raporlamaktadır [3]. Statik analiz yönteminde değerlendirilen uygulama izinlerinin incelenmesi ve dinamik analiz yönteminde uygulanan uygulama davranışlarını takip yöntemlerine karşılık kötücül yazılım geliştiricileri de uygulamaların içine küçük kod parçaları ekleyerek gizleme yöntemlerine başvurmaktadır. Kötü amaçlı yazılımların tespit edilmesinde kullanılan statik ve dinamik analiz yöntemleri gizlenmiş yazılımları tespit etmekte yetersiz kalmaktadır [3]. Kod enjeksiyonu, anti-dinamik modifikasyon ve şifreleme yöntemleriyle imza ve dinamik analiz temelli antivirüs yazılımlarının büyük ölçüde atlatılabildiği, bununla birlikte farklı anti-tespit yöntemlerinin kullanılmasının gerekliliği ortaya konulmaktadır [4]. Makandar ve Patrot tarafından yapılan çalışmada görüntü işleme işlevlerinden faydalanılarak bir Destek Vektör Makinası(SVM) yöntemi önerilmektedir. Çoklu çözünürlük ve dalgacıklar, Gabor Wavelet, GIST ve Discrete wavelet Transform ve diğer özellikleri kullanarak etkili doku özelliği vektörü oluşturmaktadır. Önerilen yöntemde Malimg veri kümesinden toplam 12.470 örnek kullanılmıştır. Bu kapsamda 1610 örnek kullanılarak eğitilen yöntem, veri setinden rastgele seçilen 8 kötü amaçlı yazılım ailesi üzerinde 1710 örnek kullanılarak test edilmiş ve %89,11 doğruluk değeri elde edilmiştir. Bu, kötü amaçlı yazılım örneklerini mevcut çalışmaya kıyasla daha yetenekli bir şekilde algılamak için makine öğrenimi sınıflandırıcı teknikleriyle Wavelet Dönüşümünü kullanan verimli ve daha doğru bir kötü amaçlı yazılım algılama algoritmasıdır [5]. Yue tarafından yapılan çalışmada kötü amaçlı yazılım ailelerinin dengesizliği nedeniyle düşen performansı iyileştirmeye yönelik derin sinir ağı modelinin son katmanı olarak basit ama etkili softmax kaybı önerilmektedir. Orijinal softmax kaybı ağırlıklıdır ve bir ölçekleme parametresi yardımıyla ağırlık değeri sınıf mevcuduna göre belirlenebilmektedir. Bu parametrenin doğru seçimi çalışılmış ve ampirik bir seçenek verilmiştir. Ağırlıklı kayıp, uçtan uca öğrenme tarzında veri dengesizliğinin etkisini hafifletmeyi amaçlar. Malimg veri seti üzerinde yapılan çalışmalarda %97,32 doğruluk değerine erişilmektedir [6]. Yajamanam vd. tarafından yapılan çalışmada kötü amaçlı yazılım puanlamasının temeli olarak görüntü işleme teknikleri önerilmektedir. Bu tür puanlama tekniklerinde, kötü amaçlı yazılım ikili dosyalarını gri tonlamalı görüntüler olarak görselleştirilmekte ve kötü amaçlı yazılım örneklerini görüntü özelliklerine göre sınıflandırmaya çalışmaktadır. Malimg, Malicia veri setlerinin 320 temel özellik kullanılarak k = 1 ile K-NN derin öğrenme yöntemi ile sınıflandırmakta ve %93 doğruluk oranı elde edilmektedir. Ardından SVC kullanılarak özellik sayısı 60’a düşürülmektedir ve %92 doğruluk oranı elde edilmektedir [7]. Cui vd. tarafından kötü amaçlı yazılım türevlerinin tespitinde zayıf algılama doğruluğu ve düşük algılama hızını iyileştirmeye yönelik derin öğrenmeyi kullanan bir yöntem önerilmektedir. Uygulama kodları gri tonlamalı resimlere dönüştürülerek, görüntülerin özelliklerini otomatik olarak çıkarabilen bir evrişimsel sinir ağı (CNN) ile sınıflandırılmaktadır. Önerilen yöntemin test edilmesinde Vision Research Lab'den gelen kötü amaçlı yazılım görüntü verileri üzerinde bir dizi deney gerçekleştirilmiştir. Deneysel sonuçlar, modelin diğer kötü amaçlı yazılım algılama modelleriyle karşılaştırıldığında iyi bir doğruluk ve hız elde ettiğini gösterdi [8].

Bhodia vd. tarafından yapılan çalışmada görüntü analizine dayalı kötü amaçlı yazılım algılama ve sınıflandırma sorununu çözmek maksadıyla yürütülebilir dosyaları görüntülere dönüştürülmekte ve derin öğrenme (DL) modellerini kullanarak görüntü tanıma uygulanmaktadır. Bu modelleri eğitmek için, görüntü veri kümeleri üzerinde önceden eğitilmiş mevcut DL modellerine dayalı aktarım öğrenimi kullanılmakta ve performansı k-en yakın komşularla (K-NN) karşılaştırılmaktadır. Önerilen görüntü tabanlı DL tekniği deneylerde K-NN'den daha iyi performans göstermektedir. Simüle edilmiş sıfır gün deneylerinde K-NN(%89)'den daha iyi performans göstermeleri açısından DL(%94,80) modellerinin verileri daha iyi genelleştirebildiği gösterilmektedir [9].

Ünver ve Bakour tarafından yapılan görüntü tabanlı global ve yerel özelliklere dayalı Android kötü amaçlı yazılım tespiti çalışmasının makine öğrenimi tekniklerinden faydalanılan ilk çalışma olduğunu belirtmektedir. Her biri 9700 örnek (4850 iyi huylu örnek ve 4850 kötü amaçlı yazılım örneği) içeren üç gri tonlamalı görüntü veri kümesi, APK arşivlerinin içeriğinden farklı dosyalar temel alınarak oluşturulmaktadır. Android uygulamalarının Manifest.xml, DEX ve Resource.ARSC dosyaları gri tonlamalı bir görüntüye dönüştürülmektedir. SIFT, SURF, KAZE ve ORB dahil olmak üzere dört farklı görüntü tabanlı yerel özellik ve Color Histogram, Haralick Texture ve Hu Moments dahil olmak üzere üç farklı görüntü tabanlı global özellik çıkarılmakta ve birden fazla makine öğrenimi sınıflandırıcısını eğitmek için kullanılmaktadır. Birden fazla yerel özelliğin tanımlayıcı vektörlerinden bir özellik vektörü elde etmek amacıyla Görsel kelime torbası(BOVW) algoritmasından

faydalanılmaktadır ve bu özellik vektörü makine öğrenimi sınıflandırıcılarında kullanılmaktadır. Çıkarılan yerel ve global özellikler, Random forest, K-en yakın komşular, Karar Ağacı, Torbalama, AdaBoost ve Gradient Boost dahil olmak üzere altı farklı makine öğrenimi sınıflandırıcısını eğitmek için kullanılmaktadır. Önerilen yöntem daha önce yapılan çalışmalarla kıyaslandığında biraz daha fazla ek çalışma zamanı gerekmektedir ancak %98 doğruluk oranı ile daha yüksek bir doğruluk değeri elde etmektedir. Önerilen yöntem geneldir, her tür uygulama görüntülere dönüştürülebilmekte ve önerilen modeli eğitmek için kullanılabilmektedir [10].

Ünver ve Bakour tarafından yapılan VisDroid adı verdikleri jenerik görüntü tabanlı bir sınıflandırma yöntemi geliştirilmiştir. Bu çalışmada kullanılmak üzere farklı kaynaklardan beş farklı sınıfa ait gri tonlamalı görüntü veri seti oluşturulmuştur. Her sınıf 4850 adet veriden oluşmaktadır. Global ve yerel olmak üzere iki tür görüntü tabanlı özellik çıkarılmıştır. Global özellikler, Renk Histogramı, Hu Moments ve Haralick Dokusu iken yerel özellikler SIFT, SURF, ORB ve KAZE'dir. Bu özellikler Random Forest, K-en yakın komşu, Karar ağaçları, Torbalama, AdaBoost ve Gradient Boost makine öğrenimi sınıflandırıcıları ile sınıflandırılmıştır. Dahası hibrit bir topluluk oylama sınıflandırıcısı önerilmiştir. Artık Sinir Ağı ve Başlangıç-v3 derin öğrenme modelleri ile test edilmiştir. Önerilen model sınıflandırma doğruluğunun yaklaşık % 98,2'ye ulaştığı gösterilmiştir. Ayrıca, önerilen modelin sonuçları bazı son teknoloji çalışmaların sonuçlarıyla karşılaştırıldığında, önerilen modelin sınıflandırma doğruluğu, hesaplama süresi, genellik ve sınıflandırma modu açısından önceki modellere göre daha iyi performans gösterdiği ortaya çıkmıştır [11]. Ünver ve Bakour tarafından DeepVisDroid adı verilen görüntüye dayalı derin öğrenme yöntemleri kullanılarak kötü amaçlı yazılım sınıflandırması yapan bir yöntem önerilmektedir. Yapılan çalışmada dört gri tonlamalı görüntü veri seti kullanılmaktadır. Global ve yerel özellikler çıkarılarak görsel kelime çantası ile özellik vektörüne dönüştürülmekte ardından IB evrişimli katman sinir ağı modeli eğitilmektedir. Çalışma kapsamında katman sayıları değiştirilerek 3 evrişimli sinir ağı ile sınıflandırma yapılmakta en iyi sonucu veren 1D evrişimli sinir ağır modeli seçilmektedir. Önerilen DeepVisDroid modelinin sınıflandırma doğruluğu, her bir örnek için 0,11 ile 2,02 s arasında değişen çok verimli çalışma süresi ek yükü ile % 98'in üzerine çıkmaktadır (% 98,96) [12].

Venkatraman, Alazab ve Vinayakumar tarafından yapılan çalışmada görüntü tabanlı kötü amaçlı yazılım sınıflandırması için denetimli ve denetimsiz öğrenme modellerinin bir kombinasyonunu kullanan hibrit mimari önerilmektedir. Önerilen mimari, yalnızca bilinen kötü amaçlı yazılımları ve türlerini değil, bilinmeyen kötü amaçlı yazılımları da algılayabilen kendi kendine öğrenme sistemidir. Gizlenmiş kötü amaçlı yazılımların kötü amaçlı yazılım tespitini geliştirmek için, gerçek zamanlı sistemlerde kullanılabilmektedir. En önemlisi, önerilen maliyete duyarlı modeller, mevcut yöntemlere kıyasla F1 puanı için 0,0969'luk performans artışı göstermektedir. Önerilen mimari, mevcut yöntemlere kıyasla daha az sayıda parametre içermekte ve dolayısıyla hem eğitim hem de test aşamalarında hesaplama karmaşıklığını azaltabilmektedir [13].

Yuan vd. tarafından yapılan çalışmada gri görüntülere ve derin öğrenmeye (GDMC) dayalı mevcut yöntemin doğruluk oranının iyileştirilmesi gerektiği ve büyük ölçüde eğitim veri setinin miktarına bağlı olduğu belirtilmektedir. Doğruluğu artırmak için bu çalışmada, markov görüntülerine ve MDMC olarak adlandırılan derin öğrenmeye dayalı bayt düzeyinde bir kötü amaçlı yazılım sınıflandırma yöntemi önerilmektedir. MDMC'de ikili dosyalar görüntülerine dönüştürülerek derin evrişimli sinir ağı ile sınıflandırılmaktadır. Çalışma Microsoft veri kümesi ve Drebin veri kümesi üzerinde gerçekleştirilmektedir. MDMC'nin ortalama doğruluk oranları, iki veri kümesinde sırasıyla %99,264 ve%97,364'tür. Eğitim veri setinin ve test veri setinin farklı oranlarında yapılan diğer deneyler de MDMC'nin GDMC'den daha iyi performansa sahip olduğunu göstermektedir [14]. Iadarola vd. tarafından yapılan çalışmada özellik vektörü çıkarma yöntemlerinin ve makine öğrenimi modellerinin farklı kombinasyonu arasında bir karşılaştırma sunulmaktadır. Metodoloji, özellik çıkarıcıları ve denetimli makine öğrenimi algoritmalarını değerlendirmeyi amaçlamakta ve 10 farklı kötü amaçlı yazılım ailesinde gruplanmış 20 binden fazla kötü amaçlı yazılım görüntüsü üzerinde test edilmektedir. Test sonuçları GIST tanımlayıcıları ve Random Forest sınıflandırıcılarının bir kombinasyonu olan en iyi sınıflandırıcının, ortalama 0,97 doğruluk elde ettiğini göstermektedir [15].

Naeem vd. tarafından yapılan çalışmada IOT cihazların kötü amaçlı yazılımın derinlemesine analizi için, renkli görüntü görselleştirme ve derin evrişimli sinir ağına dayalı bir metodoloji önerilmektedir. Renkli görüntülere dönüştürülen APK dosyalarını DCNN modeline entegre eden bir MD-IIOT adı verilen yöntem geliştirilmektedir. Önerilen kötü amaçlı yazılım tespit modelinin IIOT veri setinde tespit doğruluğu sırasıyla %97,81 ve Windows veri seti %98,47'dir. Sonuçlar, önerilen yöntemin tahmin süresi ve algılama doğruluğunun önceki makine öğrenimi ve derin öğrenme yöntemlerinden daha yüksek olduğunu göstermektedir [16].

Angelo vd. tarafından yapılan çalışmada API çağrılarının dinamik ve statik özellikleri kullanılarak özellikler belirlenmekte ve yapay sinir ağlarına dayanan bir yöntem geliştirilerek sınıflandırılmaktadır. Çalışmada Minidump veri kümeleri, VirusShare, Playdrone veri seti, Google Play den indirilen 47500 veri kullanılmıştır. Önerilen iki boyutlu gösterimden ilgili bilgileri etkin bir şekilde çıkarma yeteneğine sahiptir. Ancak gürültülü veriler üzerinde çalışmalar yapılarak geliştirilmesi gerekmektedir [17].

Kötü amaçlı yazılımların tespit edilmesi maksadıyla önerdiğimiz yöntemde uygulamaların ikili dosyaları byte okunarak gri tonlamalı görüntülere çevrilmektedir. Bu görüntüler uygulamanın genel bir resmini oluşturduğundan daha kapsamlı bir bakış açısı sağlamaktadır. Görüntü işleme yöntemleri aracılığıyla bu görüntülerden uygulamanın genel dokusu, ani renk değişimleri, yoğunluk gibi değerleri belirlenerek özellik matrisleri oluşturulmakta ve makine öğrenmesi yöntemleri aracılığıyla sınıflandırılmaktadır. Uygulama izinleri gibi belirli kalıplara bağlı kalmaması açısından statik analiz yöntemine göre avantaj sağlarken, herhangi bir sanal makine üzerinde çalıştırılmasına gerek duyulmaması ile dinamik analiz yöntemine göre avantaj sağlamaktadır. Dinamik analiz yönteminde oluşturulan sanal ortama bağlı kalındığından her durum ve koşul için uygulama davranışının nasıl bir yol izleyeceği kestirilememektedir. Ancak önerilen yöntem daha geniş ve kapsamlı bir bakış sunmaktadır.

Bu çalışmanın giriş kısmında sıralı olarak literatürde yapılan çalışmalardan bahsedilmiştir. İkinci kısımda görüntü veri kümelerine ve makine öğrenmesi yöntemlerine değinilerek üçüncü kısımda deneysel sonuçlar verilmiştir ve dördüncü kısımda sonuçlar ifade edilmiştir.

2. Materyal ve Yöntem (Material and Method)

Solucanlar, Truva atları, kimlik avı siteleri gibi kötü amaçlı yazılımların tespitinde imza tabanlı ve davranış tabanlı yöntemler oldukça yaygın olarak kullanılmaktadır. Ancak bu yöntemler zamanla yerini veri madenciliği ve makine öğrenmesi gibi sezgisel yöntemlere bırakmaya başlamıştır. Bu çalışmada ise özniteliklerin belirlenmesi aşamasında görüntü işleme yöntemlerinden faydalanılarak makine öğrenmesi yöntemleri ile sınıflandırma işlemi yapılmaktadır. Eğitim ve test verileri olarak Adialer.C(çevirici) , Agent.FYI(arka kapı) , Allaple.A(solucan), Allaple.L(solucan), Alueron.gen!J(Truva atı),

Autorun.K(solucan), C2LOP.gen!g (Truva atı), C2LOP.P (Truva atı), Dialplatform.B (çevirici), Dontovo.A (dl), Fakerean (dolandırıcı), Instantaccess (çevirici), Lolyda.AA1 (pws), Lolyda.AA2 (pws), Lolyda.AA3 (pws), Lolyda.AT (pws), Malex.gen!J (Truva atı), Obfuscator.AD (dl), Rbot!gen (arkakapı) ,Skintrim.N (Truva atı), Swizzor.gen!E (dl), Swizzor.gen!I (dl), VB.AT (solucan), Wintrim.BX (dl), Yuner.A (solucan) olmak üzere 25 kötü amaçlı yazılım ailesine ait örnekler içeren Malimg veri seti kullanılmıştır. Malimg veri seti ikili dosyaların gri tonlamalı resme dönüştürülmesi ile elde edilen görüntülerden meydana gelmektedir. Bu çalışmada önerilen yönteme ait özellik matrisi oluşturulmasının iş adımları Şekil 1'de gösterilmektedir.

2.1. Görüntü Veri Kümelerinin Oluşturulması (Generating Image Datasets)

Veri kümelerini oluşturmak amacıyla PYTHON platformunda kod parçacıkları yazılmaktadır ve açık kaynak platformlardan bu kod parçacıkları aracılığıyla veri kümleri indirilmektedir [9]. Malimg veri kümesi gibi görüntülerden oluşan veri setleri herhangi bir ön işlem gerektirmemektedir, ancak Malicia gibi ikili dosyalardan meydana gelen veri setlerinin bir takım ön işlemlerden geçirilerek görüntülere dönüştürülmesi gerekmektedir [7]. Uygulamaların APK arşivinden okunan byte dizileri RGB ve gri tonlamalı görüntüdeki piksele yazılmaktadır [9, 15]. Gri tonlamada piksel değerleri 0-255 arasında değişmektedir [8].

2.1.1. APK arşiv yapısı (APK archive structure)

Manifest.xml dosyası, herhangi bir android uygulamasını çalıştırırken sistem tarafından okunan ilk dosya olmasından dolayı android uygulamasındaki en önemli dosyalardan biridir ve bu nedenle görüntüye çevrilecek dosyalar arasında ilk sırayı almaktadır [9, 18]. Manifest dosya boyutundaki sınırlama nedeniyle, bu dosya kullanılarak oluşturulan görüntünün genişliği 64 pikseldir ve yüksekliği dosya boyutuna göre değişkendir [9]. Android uygulama kodu genellikle Java programlama dili kullanılarak yazılmaktadır ve DEX koduna derlenmektedir. DEX (Dalvik yürütülebilir) kodu, Android uygulamalarını başlatmak ve çalıştırmak için kullanılan optimize edilmiş bir bayt kodudur [9, 19]. Dex kodu uygulamanın gerçek kodunu içerdiğinden, bu dosyalar uygulamanın davranışını belirtmek ve uygulama hakkında fazla bilgi toplamak için çok önemlidir [9, 19]. Bu nedenle Dex kodları da gri tonlamalı renk dönüştürülen dosyalar arasında yer almaktadır. Resource.arsc dosyası, UI düzenleri ve dize değerleri gibi kaynak referansları gibi uygulamanın ikili biçimde derlenmiş kaynaklarını içermektedir [9, 19]. İçerdiği bilgilerden dolayı gri tonlanan dosyalar arasında yerini almaktadır.

Lib / klasör genellikle, yerel kod kitaplıkları bu klasöre gömülü halde bulunmaktadır [10, 19].

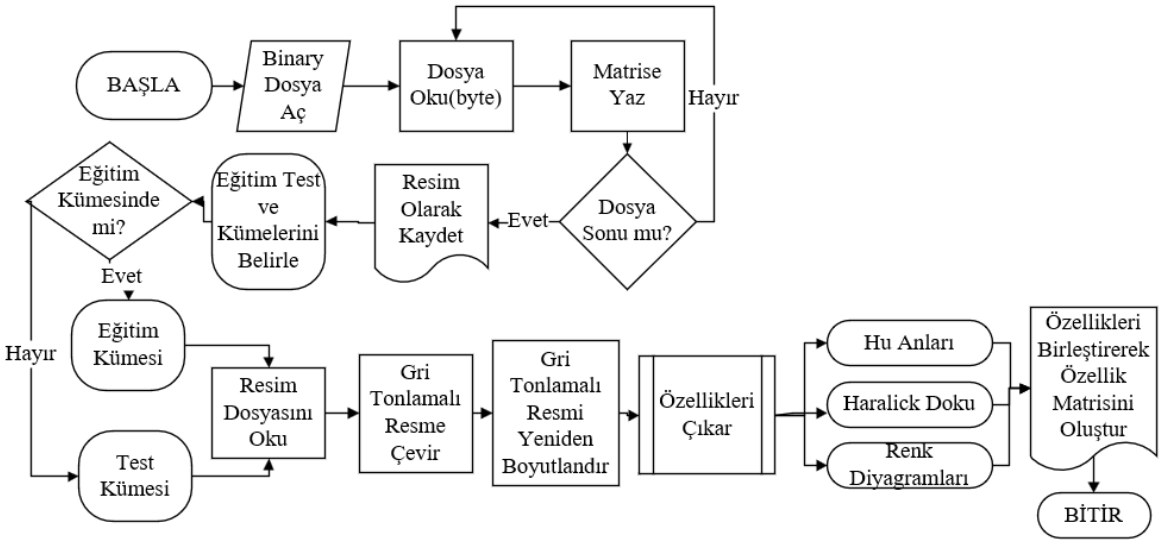
Diğer Bazı yapılandırılmış veri kümelerindeki görüntülere Jar dosyalarını (APK arşivinin içeriğinde varsa) eklenmesi önerilmektedir [10].

2.1.2. Görüntü tabanlı global özellikler (Image based global properties)

Görüntü bir bütün olarak tanımlanmaktadır, bu sebeple tüm görüntü tek bir özellik vektörü kullanılarak oluşturulmaktadır. Temel olarak 320 adet görüntü tabanlı global özellik bulunmaktadır [3].

Kullanılan global özelliklerden biri, histogramın görüntünün piksellerindeki (piksel yoğunluğu) renk dağılımına bağlı olarak hesaplandığı Renk Histogramıdır [9]. Gri tonlamalı görüntü temsili kullanılarak oluşturulan histogram 255 bölme içerir (çünkü gri tonlamalı görüntü pikselleri 0 ile 255 arasındaki yoğunluk değerlerini alır) [6]. 0 değeri siyah rengi ifade ederken 1 değeri beyaz rengi ifade etmektedir [9]. Önerilen veri setlerinde oluşturulan tüm görüntüler 256 piksel sabit genişliğe sahipken, uzunlukları dosya boyutuna göre değişmektedir [10]. Ancak, Manifest.xml ve Resources.arsc dosyalarının boyutundaki sınırlamalar nedeniyle, bu iki dosya kullanılarak oluşturulan görüntüler için 64 piksel genişlik kullanılmaktadır [10]. Oluşturulan bayt matrisindeki her bir bayt, 0 ile 255 arasında bir değer depolayabildiğinden, matristeki her bayt, son gri tonlamalı görüntüde bir piksele dönüştürülmektedir [10]. Bayt matrisini 0 ile 255 arasında bir değer matrisine dönüştürdükten sonra, son matris gri tonlamalı bir görüntü olarak kaydedilmektedir.

Kullanılan diğer bir global özellik görüntü anlarının görüntü piksel yoğunluğunun ağırlıklı ortalamalarından meydana geldiği Hu Moments'tır. Hu anları, merkezi anlar kullanılarak hesaplanan yedi sayıdan oluşan bir moment türüdür [9]. Hu anları çevirme, ölçekleme, yansıtma ve döndürme ile değişmemektedir.



Şekil 1. Özellik matrisi oluşturulması iş adımları (Work steps to create a feature matrix)

Global özellikler arasında yer alan bir yöntem de Haralick Doku'dur. Doku tanımlayıcı, pürüzsüzlük, kabalık ve düzenlilik gibi görüntü özelliklerinin ölçümlerini sağlamaktadır [9]. Haralick dokuları, görüntüde kullanılan en yaygın doku özelliklerinden biridir.

2.1.3. Görüntü tabanlı yerel özellikler (Image based native properties)

Global özelliklerin yanı sıra bir de görüntüdeki ilginç noktaları tespit ederek görüntünün nesnelerini tanımlamak amacıyla kullanılan yerel özellikler mevcuttur. Global özellikler görüntüye genel bir bakış sağlarken, yerel özellikler görüntüdeki her bir nesne hakkında ayrıntılı bilgi vermektedir [10]. Makine öğrenme algoritmaları girdi olarak n örnek n özellik vektörünü kabul ettiğinden, yerel özelliklerinin hepsinin kullanılması mümkün olmamaktadır [10]. Yaygın olarak kullanılanlar SIFT, SURF, KAZE ve ORB ve benzerleridir.

SIFT algoritması, Gauss LOG'unun Laplacian'ının farklı ölçek değerlerinde hesaplanmasına ve en iyi sonuçları veren ölçeğin seçilmesine dayanır [7]. SIFT algoritmasında ilginç noktalar, Laplacian of Gaussian'ın ölçek uzayı kullanılarak elde edilen yerel maksimum / yerel minimumdur. SURF, benzerlik değişmez gösterimi ve görüntülerin karşılaştırılması için hızlı ve sağlam bir algoritma olmasından dolayı SIFT algoritmasına bir alternatif oluşturmaktadır. Bu algoritma, kutu filtreleri kullanarak hızlı hesaplamasıyla karakterize edilir (LOG yaklaşımı için kutu filtreleri kullanır), bu nedenle izleme ve nesne tanıma gibi gerçek zamanlı uygulamalarda kullanılabilir [11]. KAZE detektörü, orijinal görüntü çözünürlüğüne dayalı olarak çalışan, doğrusal olmayan ölçek uzaylarında özelliklerin tespiti ve tanımlanması için 64 bit vektör boyundan oluşan bir yöntemdir. KAZE detektörü, birden fazla ölçek seviyesinde Hessian Matrix'in normalleştirilmiş determinantına dayanmaktadır [7]. Yerel minimum / maksimum (ekstremma) dikdörtgen bir pencereden seçilmektedir. ORB algoritması, görüntüdeki kilit noktaları tespit etmek için FAST algoritmasını kullanır, ardından tespit edilen anahtar noktalar arasında ilk N'yi bulmak için Harris köşesini kullanır. Dahası, ORB, çok ölçekli özellikler üretmek için bir piramit kullanır. ORB özellikleri, bir dizi ikili yoğunluk testi kullanılarak hesaplanan 32 bitlik BRIEF tabanlı tanımlayıcı kullanır. Buradaki parametrelerden biri düzleştirilmiş bir görüntü yamasıdır [11].

Yapılan çalışmalarda kullanılan global değişkenler, yerel değişkenler, veri seti sayıları ve makine öğrenmesi yöntemlerinin karşılaştırılması Tablo 1'de gösterilmektedir.

2.2. Makine Öğrenmesi (Machine Learning)

Makine öğrenmesi (ML), bir bilgisayarın doğrudan yönergeler olmadan öğrenmesine yardımcı olmak için matematiksel modelleri kullanma işlemidir [20]. Makine öğrenimi terimi 1959'da Amerikalı Arthur Samuel tarafından bilgisayar oyunları alanında ortaya konulmuştur [21]. Dama oyununu öğrenebilen bir sistem geliştirilmiştir [21]. 1960'lı yıllarda Nilsson tarafından örüntü sınıflandırmaya yönelik makine öğrenimi hakkında bir kitap yayınlanmıştır [22]. 1970'lı yıllarda Duda ve Hard örüntü tanıma ve sınıflandırma çalışmalarına devam etmiştir [23]. 1981'de, bir sinir ağı modeli geliştirilerek 26 harf, 10 rakam ve 4 özel sembolden oluşan 40 karakteri tanımayı öğrenmesi amacıyla öğretme stratejilerinin kullanımına ilişkin bir çalışma yapılmıştır [24].

Makine öğrenimi algoritmaları hedeflenen çıkış değerlerine göre gözetimli öğrenme [25, 26], gözetimsiz öğrenme [27-30], pekiştirmeli öğrenme [31] gibi çeşitli gruplara ayrılmaktadır. En sık kullanılan gözetimli öğrenme algoritmaları Karar Ağaçları, Lineer Regresyon, Destek Vektör Makineleri ve Lojistik Regresyondur [27].

Lojistik Regresyon Analizinde verilerin yapısındaki grup sayısı bilinmekte ve bu verilerden faydalanarak bir ayrımsama modeli elde edilmektedir [32]. Kurulan model yardımı ile veri kümesine yeni alınan gözlemlerin gruplara atanması yapılmaktadır [32]. İncelenmesinin kolay ve az karmaşık olmasının yanında doğrusal

Tablo 1. Yapılan çalışmalarda kullanılan değişkenler ve veri seti karşılaştırmaları
(Variables used in studies and data set comparisons)

	Global Değişkenler				Yerel Değişkenler					Bovw	Veriseti Sayısı
	Renk Diyagramlar	Haralick Doku	Hu Anları	Diğer	Sıft	Surf	Orb	Kaze	Diğer		
Makandar ve Patrot [5]	Çoklu çözünürlük ve dalgacıklar, Gabor Wavelet, GIST ve Discrete wavelet Transform ve diğer özellikleri									Belirtilmemiş	Malimg dataset: 12.470
Yajamanam vd. [7]	Var	Var	Var	Var	Yok	Yok	Yok	Yok	Yok	Yok	Malimg dataset: 9339 Malicia dataset: 11363
Bhodia vd. [9]	Belirtilmemiş			Var	Yok	Yok	Yok	Yok	Yok	Yok	Malimg dataset: 9339 Malicia dataset: 9,895
Ünver ve Bakour [10]	Var	Var	Var	Yok	Var	Var	Var	Var	Yok	Var	9700
Ünver ve Bakour [11]	Var	Var	Var		Var	Var	Var	Var	Yok	Var	5 aileden 4850 örnek toplam:24250
Ünver ve Bakour [12]	Var	Var	Var		Var	Var	Var	Var	Yok	Var	4 aileden 4850 örnek toplam:19400
Yuan vd. [14]	Sadece dex dosyalarının markov görüntülerine çevrildiği belirtilmiş										Microsoft veri set 10868 ve Drebin veri seti:4020
Iadarola vd. [15]	Sadece GIST kullanıldığı belirtilmiş (Gabor filte GIST identifier, Color Scheme filter and Auto Color Correlation filter uygulanmış)				Yok						20748
Naeem vd. [16]	GIST +512 özellik				Var	Belirtilmemiş			20 özellik	Belirtilmemiş	14.733 kötü amaç 2486 iyi huylu
Angelo vd. [17]	Sadece statik ve dinamik analiz yöntemleri ile görüntü elde edildiği belirtilmiş										47500

olmayan etkileri de işleyebilmesi bu yöntem için avantaj sağlarken ortalama tahmin değerinin düşük olması ve birçok uygun olmayan özelliği iyi idare edememesi dezavantaj sağlamaktadır [33]. Lojistik Regresyon genel formülü Eş. 1'de verildiği gibidir.

$$P = \frac{e^{a+bX}}{1+e^{a+bX}} \quad (1)$$

P;1'in olasılığı, e; doğal algoritma tabanı, a ve b; modelin parametreleridir. X değeri sıfır kabul edildiğinde a'nın değeri P'yi vermektedir, b ise X'in değerinin bir birim değiştirilmesi ile olasılığın değişim hızını ayarlamaktadır.

K-NN algoritması gözetimli öğrenme yöntemlerinden biridir. Eğitim kümesindeki verileri kullanarak eğitim işlemi gerçekleştirilir ve çeşitli parametreler belirlenir [23, 34]. Parametrelerin belirlenmesine müteakip test verileri, belirlenen bu parametreler kullanılarak sınıflandırılır [34]. K-NN algoritmasında ise ön eğitim işlemine ihtiyaç duyulmamaktadır [34]. Test verileri sınıflandırılırken her sınıflandırma işleminde eğitim kümesi kullanılarak sınıflandırılmaktadır [34-36]. K-NN algoritmasında ilk olarak veriler etiketlenerek bir eğitim kümesi oluşturulmaktadır [36]. K parametresi ve Öklid, Manhattan, Hamming ve Minkowski, gibi bir uzaklık fonksiyonu belirlenerek, yeni bir veri geldiğinde seçilen bu verinin eğitim kümesindeki verilere olan uzaklığı tek tek hesaplanmaktadır [35, 37]. Daha sonra uzaklığı en küçük olan k adet veri eğitim kümesinden seçilir ve sınıflama kümesi belirlenir [35]. Basit uygulanabilir olması ve yeni veriler ile düşük maliyet ile güncellenebilir olması avantaj sağlarken veri setinin eşit dağıtılmadığı durumlarda düşük performans göstermesi dezavantaj sağlamaktadır [37]. Destek Vektör Makinaları (SVM) algoritması, çoğunlukla veri sınıflandırma maksadıyla kullanılmaktadır [38-40]. Veri setini bir hiper düzlem oluşturarak farklı sınıflara ait kısımlar oluşturmaktadır [38]. İyi amaçlı ve kötü amaçlı olarak sınıflandırma yapılması durumda düzlem üzerindeki noktaları bir doğru ile ayırarak görselleştirebilmektedir [38]. Çizilen bu doğrunun iki sınıfa da en uzak mesafede olması istenen ve beklenen durumdur. SVM eğitim işlemlerinde en uygun çekirdek fonksiyonun seçilmesi sınıfların doğru olarak belirlenmesinde etkin rol oynamaktadır [38]. Yaygın olarak kullanılanlar ise Doğrusal, Radyal ve Çoklu çekirdek işlevleridir [38]. Yüksek doğruluk oranı sağlamasının yanı sıra yüksek boyutlu verilerde gösterdiği iyi performans ile avantaj sağlarken, ceza parametresi büyük olan verilerde eğitim süresinin çok yüksek değerlere ulaşması dezavantaj olarak görülebilmektedir.

Karar ağacı sınıflandırma yönteminde, her özelliğin bilgi kazancı hesaplanarak, en yüksek bilgi kazancına sahip olan özellik kök olarak belirlenmekte ve bir karar ağacı oluşturulmaktadır [41]. Diğer özellikler yaprak düğümlere yerleştirilmekte ve bu meydana gelen yapı sınıflandırma işlemlerinde tahminlerde bulunularak, karar vermekte kullanılmaktadır [41]. Yüksek boyutlu ve gürültülü verilerde yüksek performans göstererek avantaj sağlarken, az sayıda öznitelik belirlenmesi durumunda düşük performans göstererek dezavantaj sağlamaktadır [41]. Veri kümesindeki ufak değişikler ağacın yapısını etkilediğinden dolayı tutarsızlıklar oluşturabilmektedir [41]. Beyaz kutu olarak çalışması ile K-NN ve SVM yöntemlerinden ayrılmaktadır. Eğitilmiş verilerin yorumlanabilmesi ve modelin değerlendirilerek tekrar detaylandırılabilmesi imkânlarını sunmaktadır. Hızlı eğitilebilmesi de sağladığı avantajlardandır [41].

RF yöntemi, karar ağacı sınıflandırma yönteminden farklı olarak birden fazla karar ağacını birleştirerek bir grup torbalama makinesi gibi çalışmaktadır [42]. Her bir ağaç bağımsız olarak bir sonuç üretmektedir ve rasgele orman yönteminin sonucu her bir ağacın ürettiği sonuçlarla hesaplanmaktadır [40]. Birden fazla altkümeyi rasgele seçerek yüksek sınıflandırma doğruluğu elde etmekte ve fazla uyum riskini azaltarak avantaj sağlamaktadır [42]. Veri setinde meydana gelen değişimlerde iyi sonuçlar vermektedir. Karar ağaçlarının sayısının fazla olması modeli güçlü kılarken eğitim hızını yavaşlatmaktadır.

Naive Bayes(NB) yönteminin temelinde öznitelikler arasında güçlü bağımsızlık ilkesinin varsayıldığı Bayes teoremi yer almaktadır [40]. Her bir sınıfın olasılık değeri ve sınıflara ayrılmış her veri örneğinin koşullu olasılık değeri hesaplanmaktadır [43]. Hesaplanan bu koşullu olasılık değeri her bir veri örneğinin sınıf olasılık değeri ile çarpılarak kümülatif olasılık değeri elde edilmektedir [42]. İkili ve çoklu sınıflandırma problemlerinde kullanılabilmektedir [43, 44]. Basit uygulanabilir, kolay anlaşılabilir ve küçük boyutlu veri kümesi ile eğitilebilir olması avantaj sağlarken, eğitim veri kümesinin birbiri ile ilişkili verilerden oluşması dezavantaj sağlamaktadır [43]. Naive Bayes sınıflandırmasının karar teoremi Eş. 2'de verildiği gibidir.

$$P(Si)\prod_{k=1}^{L} P(xk|Si) > P(Sj)\prod_{k=1}^{L} P(xk|Sj) \quad (2)$$

$P(S_i)$ i sınıfının öncel olasılığını ve $P(S_j)$ ise j sınıfının öncel olasılığını temsil etmektedir. Eğer Eş. 2 sağlanıyorsa x sınıf S_i'ye aittir.

Yapılan çalışmalarda kullanılan yöntemler, avantajları, dezavantajları ve çalışmalarda kullanılan veri setleri Tablo 2'de gösterilmektedir.

3. Deneysel Sonuçlar (Experimental Results)

Bu çalışmada malimg veri seti üzerinde global özellikleri çıkarılarak çeşitli makine öğrenmesi yöntemleri ile sınıflandırılmıştır. Çıkarılan özellikler ile hu anlar, haralick doku ve renk diyagramlarından meydana gelen (2000, 532) boyunda bir öznitelik matrisi oluşturulmuştur. Çalışmanın yazılımsal kodlanması aşamalarında Python programla dili ve kütüphanelerinden faydalanılmıştır. Makine öğrenmesi yöntemleri için ise sklearn kütüphanesi kullanılarak LR, LDA, K-NN, CART, RF, NB, SVM yöntemleri ile sınıflandırma işlemi yapılmıştır. 25 sınıfın her birinden 2000 adet veri örneği alınarak veri seti oluşturulmuştur. 10 kat çapraz doğrulama ile sonuçların doğruluk oranı ve standart sapması hesaplanmıştır. Her bir iterasyon için algoritmaların doğruluk oranları Tablo 3'de gösterilmektedir. Algoritmaların önerilen yöntem ile elde edilen doğruluk oranları ve standart sapma değerleri Tablo 4'te gösterilmektedir.

En düşük doğruluk oranı %70,05 ile SVM yöntemi ile elde edilirken en yüksek doğruluk değeri %97,22 ile RF yöntemi ile elde edilmiştir. Çalışma ve sonuç üretme süreleri değerlendirildiğinde en hızlı algoritmanın 26,587 saniye ile RF, en yavaş olanın ise 32,089 saniye ile SVM algoritması olduğu görülmektedir. Makine öğrenmesi yöntemlerinin sonuçlarının grafiksel olarak karşılaştırılması Şekil 2'de gösterilmektedir.

Çapraz doğrulama yönteminde seed=5 olarak değiştirildiği takdirde LR, LDA ve SVM algoritmalarında doğruluk oranı düşerken K-NN, CART ve RF algoritmalarında doğruluk oranı yükselmektedir. Çalışma ve sonuç üretme süresi açısından değerlendirildiğinde 25,439 saniye ile en hızlı LDA algoritması, en yavaş ise 26,05 saniye ile LR algoritmasıdır. Seed=10 ve seed=5 iken hız değişimleri değerlendirildiğinde algoritmaların çalışma ve sonuç üretme sürelerinde genel bir düşüş olduğu görülmektedir. RF algoritmasındaki hızlanmanın ise en az olduğu görülmektedir. Tablo 5'te seed=5 olarak hesaplanan değerler görülmektedir. Seed=5 olarak uygulandığında önerilen makine uygulaması yöntemlerinin grafiksel kıyaslanması ise Şekil 3'te verilmektedir. Eğitim veri seti 10% azaltılarak test veri seti 10% arttırıldığında ise doğruluk oranları ve standart sapmaları Tablo 6'daki gibidir.

Tablo 2. Yapılan çalışmaların avantajları ve dezavantajları (Advantages and disadvantages of the studies performed)

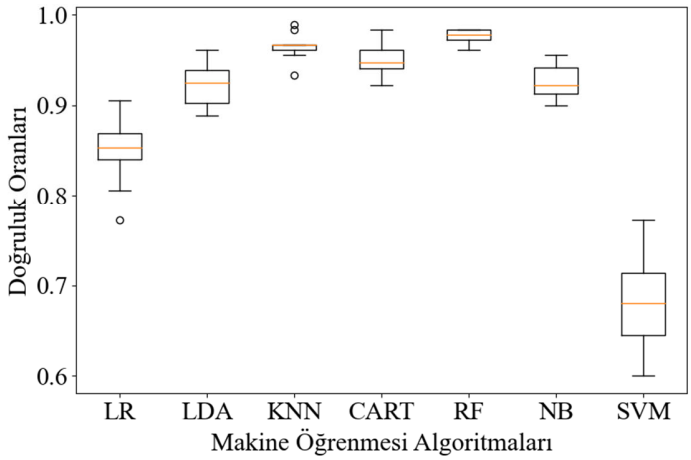
	Makine Öğrenmesi yöntemi	Avantajları	Dezavantajları	Veriseti
Makandar ve Patrot [5]	SVM	Daha yetenekli bir şekilde algılama ve daha yüksek doğruluk oranı	Belirtilmemiş	Malimg
Yue [6]	CNN	Ağırlık değerlerini etkin bir şekilde düzenleyerek veri dengesizliklerinin etkisi azaltmak	Belirtilmemiş	Malimg
Yajamanam vd. [7]	K-NN, SVC	Temel özellik sınıfından 320 özellik yerine 60 özellik kullanılarak hemen hemen aynı doğruluk oranına yakın sonuçlar elde edilebilmektedir	Sağlamlığı ile ilgili deneyler sınırlı ve sonuçsuz kalmıştır. İyi bir analiz çalışması yapılması gerekmektedir.	Malimg, Malicia
Cui vd. [8]	CNN	Daha yüksek doğruluk oranı ve hız	Belirtilmemiş	Vision Research Lab
Bhodia vd. [9]	K-NN	Benzeri görülmemiş derinlikteki sinir ağlarının eğitimini mümkün kılan "artık blokların" kullanılması	Denenen daha karmaşık ResNet varyantları önemli bir gelişme sağlanamamıştır. Bu yüzden ResNet 34 kullanılmaya devam edilmiştir.	Malimg, Malicia
Ünver ve Bakour [10]	RF, K-NN, Karar Ağacı, Torbalama, AdaBoost ve Gradient Boost	Yüksek doğruluk oranı	Ek çalışma süresi gerektirir. Kod gizleme, kod işlemeden etkilenebilir. Enjeksiyon saldırılarını tespit edemez.	Yazılan Ptyhon kodu ile çeşitli kaynaklardan toplanmıştır
Ünver ve Bakour [11]	RF, K-NN, Karar ağaçları, Torbalama, AdaBoost ve Gradient Boost	SIFT ve SURF yerel özellik kullanılarak yapılan çalışmada en yüksek doğruluk oranını vermiştir. Global özelliklerde daha yüksek doğruluk oranı.	KAZE ve ORB kullanılarak yapılan sınıflandırılmalarda fazla başarılı sonuçlar verememiştir. Kod gizleme ve manipülasyon tekniklerinden etkilenebilir	Drebin ve Malgenom kötü amaçlı yazılım veri kümesi
Ünver ve Bakour [12]	1B evrişimli katman sinir ağı modeli, ResNet, Inception V3	Yüksek doğruluk oranı. Uygun çalışma süresi	Kod gizleme ve manipülasyon tekniklerinden etkilenebilir	GooglePlay, Drebin ve Malgenom
Venkatraman, Alazab ve Vinayakumar [13]	CNN, LSTM	Daha az sayıda parametre içermekte, hem eğitim hem de test aşamalarında hesaplama karmaşıklığını azaltabilmektedir, gerçek zamanlı olarak eğitilerek kullanılabilir	Belirtilmemiş	Microsoft Kötü Amaçlı Yazılım Sınıflandırma Zorluğu veri seti, Malimg veri seti
Yuan vd. [14]	CNN	Eğitim sırasında zaman ve alan tüketimi VGG16'ya göre çok daha azdır. Transfer öğrenmeye dayalı yöntemlerle karşılaştırıldığında, önceden eğitilmiş modellere ihtiyaç duymaz.	Kötü amaçlı yazılım ailelerinin dağılımı dengesiz olduğu için, sınıflandırıcının her sınıf, özellikle de azınlık üzerindeki performansını doğruluk ve logloss olarak yansıtmak zordur.	Microsoft veri seti ve Drebin veri seti
Iadarola vd. [15]	K-NN, RF, Karar tablosu sınıflandırıcılar, C4.5 karar ağacı sınıflandırıcılar	Doğru şekilde sınıflandırmak için çeşitlilik ve zenginlik bilgileri sağlamak amacıyla uygun bir boyuta (960 öznitelik) sahip olduğundan, diğer özellik çıkarma yaklaşımlarından daha iyi performans gösterir. Görüntünün daha geniş temsilini sağlayan, oryantasyon, ölçekler ve gradyanlarla ilgili bilgileri özetler.	Zamanlama ve hesaplama açısından pahalı, gizli kodlar hesaba katılmamış	AMD dataset
Naeem vd. [16]	Derin Konvansiyonel Ağ (DCNN)	Geleneksel kötü amaçlı yazılım algılama tekniklerinin çoğu maliyet, bellek ve sınırlı işlem kapasitesi nedeniyle doğrudan IoT cihazları için kullanılamazken bu yöntem kullanılabilir. Kararlı, daha güvenilir ve daha az kaynak	Blok zincir için geliştirilmesi gerekiyor	Veri seti IKM Laboratuvarı 1'den toplanmıştır.
Angelo vd. [17]	API-images (CNN)	Önerilen iki boyutlu gösterimden ilgili bilgileri etkin bir şekilde çıkarma yeteneğine sahiptir	Sınırlı eğitim verisinin olması, gürültülü verilerde çalışılması gerekmektedir	Minidump veri kümeleri, VirusShare, Playdrone veri seti, Google Play

Tablo 3. Her iterasyon için algoritmaların doğruluk oranları (Accuracy rates of the algorithms for each iteration)

	LR	LDA	K-NN	CART	RF	NB	SVM
Fold 1	0,8111	0,8944	0,9333	0,9666	0,9555	0,8944	0,6888
Fold 2	0,7944	0,9	0,9777	0,9777	0,9888	0,9444	0,7166
Fold 3	0,8388	0,9388	0,9722	0,9388	0,9722	0,9166	0,6777
Fold 4	0,8388	0,9555	0,95	0,9388	0,9666	0,9055	0,7333
Fold 5	0,8888	0,9111	0,9777	0,95	0,9722	0,9	0,7055
Fold 6	0,8611	0,9277	0,9777	0,9388	0,9888	0,9388	0,7
Fold 7	0,8555	0,8944	0,9555	0,9333	0,9611	0,9166	0,7
Fold 8	0,8444	0,9611	0,9722	0,95	0,9666	0,9388	0,7166
Fold 9	0,8666	0,9555	0,9833	0,9777	0,9833	0,9333	0,7277
Fold 10	0,85	0,9	0,9666	0,9555	0,9666	0,95	0,6388

Tablo 4. Doğruluk oranları ve standart sapma değerleri (Accuracy rates and standard deviation values)

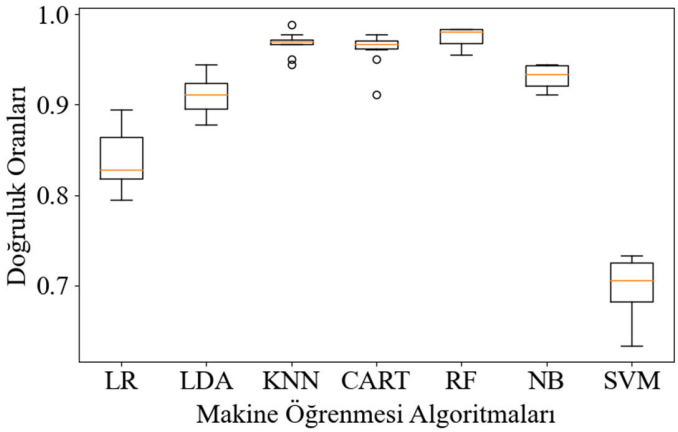
Yöntem	Doğruluk Oranı(%)	Standart Sapma(%)	Çalışma ve Sonuç Üretme Süresi(Saniye)
LR	0,8450	0,0256	30,128
LDA	0,9238	0,0258	32,017
K-NN	0,9666	0,0149	31,543
CART	0,9527	0,0155	31,623
RF	*0,9722*	0,0108	*26,587*
NB	0,9238	0,0187	30,661
SVM	0,7005	0,0261	32,089



Şekil 2. Makine öğrenmesi algoritmalarının karşılaştırılması (Machine learning algorithm comparison)

Tablo 5. Doğruluk oranları ve standart sapma değerleri (Accuracy rates and standard deviation values)

Yöntem	Doğruluk Oranı (%)	Standart Sapma (%)	Çalışma ve Sonuç Üretme Süresi (Saniye)
LR	0,8422	0,0310	26,05
LDA	0,9111	0,0198	*25,439*
K-NN	*0,9672*	0,0122	25,658
CART	*0,9616*	0,0141	25,488
RF	*0,9744*	0,0093	25,901
NB	*0,9311*	0,0143	25,571
SVM	0,6894	0,0298	25,449



Şekil 3. Makine öğrenmesi algoritmalarının karşılaştırılması (Machine learning algorithm comparison)

Tablo 6. Doğruluk oranları ve standart sapma değerleri (Accuracy rates and standard deviation values)

Yöntem	Doğruluk Oranı (%)	Standart sapma (%)	Çalışma ve Sonuç Üretme Süresi (Saniye)
LR	0,8207	0,0340	46,197
LDA	0,9157	0,0302	42,72
K-NN	0,9585	0,0134	49,914
CART	0,9478	0,0166	48,73
RF	0,9664	0,0153	43,594
NB	0,9257	0,0128	45,822
SVM	0,6950	0,0328	44,389

Tablo 7. Diğer çalışmalar kıyaslama (Other studies comparison)

	Kullanılan yöntem	Doğruluk Oranı (%)	Veri Seti
Makandar ve Patrot [5]	SVM	89,11	Malimg
Yue [6]	CNN	97,32	Malimg
Yajamanam vd. [7]	K-NN	93	Malimg
Cui vd. [8]	GIST+SVM	92,2	Malimg
Cui vd. [8]	GIST+K-NN	91,90	Malimg
Cui vd. [8]	GLCM+SVM	93,20	Malimg
Cui vd. [8]	GLCM+K-NN	92,50	Malimg
Bhodia vd. [9]	DL	94,80	Malimg
Ünver ve Bakour [12]	DeepVisDroid (3 Evrişimli Sinir Ağı Modeli)	98,96	Çeşitli kaynaklardan toplanarak elde edilen veri seti
Venkatraman vd. [13]	Hibrit Yöntem (Denetimli + denetimsiz Öğrenme yöntemleri)	98,6	Microsoft + Malimg
Yuan vd. [14]	MDMC (Derin Evrişimli Sinir Ağı)	99,264	Microsoft
Yuan vd. [14]	MDMC (Derin Evrişimli Sinir Ağı)	97,364	Drebin
Iadarola vd. [15]	GIST+RF	97	AMD dataset
Naeem vd. [16]	DCNN	97,81	Veri seti IKM Laboratuvarı 1'den toplanmıştır
Angelo vd. [17]	API-images(CNN)	95	Minidump veri kümeleri. VirusShare. Playdrone veri seti. Google Play
Önerilen yöntem	K-NN	96,72	Malimg
Önerilen yöntem	RF	97,44	Malimg

Tablo 6'dan da görüldüğü üzere her iki durumda da rasgele orman algoritması ile en yüksek doğruluk oranı elde edilmiştir. Ancak genel olarak doğruluk oranlarında düşüş olduğu tespit edilmiştir. Eğitim ve test verilerinde meydana gelen değişikliklerde algoritma çalışma ve sonuç üretme süreleri genel anlamda olumsuz yönde etkilenmektedir. Bu değişimler incelendiğinde LDA algoritmasının olumsuz yönde etkilenmesinin en az olduğu RF algoritmasının ise olumsuz etkilenmesine rağmen kabul edilebilir değerler olduğu görülmektedir.

Aynı veri seti üzerinde yapılan diğer çalışmalarda kullanılan yöntemler ile elde edilen doğruluk oranlarının önerilen yöntem ile kıyaslanması Tablo 7'de verilmiştir.

Yajamanam S., Selvin V.R.S. vd. [5] tarafından aynı veri seti üzerinde yapılan çalışmada K-NN ile %93 doğruluk değeri elde edilmiştir. 2019 yılında ise N. Bhodia. ve P. Prajapati K-NN [9] ile en yüksek %89,11 bulmuş DL ile doğruluk oranını %94,80'e çıkarmıştır. Makandar ve Patrot 2017 yılında SVM [5] yöntemi ile %89,11 doğruluk oranı değeri elde ettiklerini açıklamıştır. Cui vd. [8] yaptıkları çalışmada en yüksek GLCM+SVM uyguladıkları yöntem ile %93,20 değerine ulaşmıştır. Yue ise 2021 yılında CNN [6] uygulayarak %97,37 doğruluk değerine erişmiştir. Bu çalışmada önerilen yöntem ile K-NN ile %96,72, RF ile en yüksek %97,44 değeri elde edilmiştir. Deneysel sonuçlar değerlendirildiğinde farklı veri setleri üzerinde Ünver Bakour tarafından [12] %98,96, Yuan vd. tarafından [14] %99,264 Naem vd. [16] tarafından %97,81 doğruluk oranları ile derin öğrenme yöntemleri kullanılarak daha yüksek sonuçlara ulaşıldığı görülmektedir. Ancak önerilen yöntemin doğruluk oranı aynı veri seti üzerinde daha önce makine öğrenmesi yöntemleri ile yapılan çalışmalardan daha yüksek bir değere ulaşmıştır.

4. Sonuçlar (Conclusion)

Android cihazların yaygın kullanımı veriye erişmeyi, işlemeyi ve paylaşmayı kolay hale getirmiştir. Veri kıymetli olduğundan bu cihazlar kötü amaçlı kullanıcıların hedefi haline gelmiştir. Bu çalışmanın amacı kötü amaçlı yazılımların tespit edilerek veri hırsızlıklarının önüne geçmektir.

Çalışmamızda android APK dosyaları görüntüye çevrilerek gri tonlamalı görüntüler oluşturulmaktadır. Oluşturulan görüntüler analiz edilerek özellikler belirlenmekte ve makine öğrenmesi yöntemleri ile sınıflandırılmaktadır. Sonuçlar K-kat çapraz doğrulama yöntemi kullanılarak değerlendirilmektedir. Çapraz doğrulama yönteminde seed=5 olarak değiştirildiği takdirde doğruluk oranı LR algoritmasında %84,50'den %84,22'ye, LDA algoritmasında %92,38'den %91,11'e ve SVM algoritmasında %70,05'den %68,94'e düşerken K-NN algoritmasında %96,66'dan %96,72'ye, CART algoritmasında %95,27'den %96,16'ya, RF algoritmasında %97,22'den %97,44'a ve NB algoritmasında %92,38'den %93,11'e yükseldiği görülmektedir. Ayrıca eğitim veri kümesi %10 azaltılarak

test veri kümesi %10 artırılmıştır ve sonuçlara bakıldığında bütün yöntemlerin doğruluk oranının düştüğü görülmüştür. Önerdiğimiz yöntem kullanılarak K-NN ile %96,72, RF ile en yüksek %97,44 doğruluk değeri elde edilmiştir. Bu kapsamda aynı veri seti üzerinde yapılan diğer çalışmalar ile kıyaslandığında önerilen yöntem ile daha yüksek başarı oranı elde edilmesi ile literatüre katkı sağlanmaktadır. Uygulama izinleri gibi belirli kalıplara bağlı kalmadan ve bir sanal makineye ihtiyaç duyulmadan yüksek doğruluk oranına ulaşması ile statik ve dinamik analiz yöntemlerine kıyasla avantaj sağlamaktadır.

Bu çalışmanın APK dosyalarının görüntüye çevrilmesi, görüntünün özelliklerinin belirlenmesi ve makine öğrenmesi ile sınıflandırılmasının anlaşılmasına katkı sağlayacağı düşünülmektedir. Gelecek çalışmalarda veri seti geliştirilerek ve derin öğrenme yöntemleri ile sınıflandırma çalışmaları yapılarak başarı oranının artırılması hedeflenmektedir.

Kaynaklar (References)

1. Sağıroğlu Ş., Bulut H., An Analysıs of Informatıon and Telecommunication Security in Mobile Environments, Journal of the Faculty of Engineering and Architecture of Gazi University, 24 (3),499-507, 2009.
2. Cook S., Malware statistics and facts for 2021, Comparitech, https://www.comparitech.com/antivirus/malware-statistics-facts/,Güncelleme Tarihi Şubat 12. 2021, Erişim Tarihi Nisan 20.2021.
3. Mitsuhashi R., Shinagawa T., High-Accuracy Malware Classification with a Malware-Optimized Deep Learning Model, https://arxiv.org/pdf/2004.05258.pdf, Erişim Tarihi Kasım 15.2021.
4. Aygör D., Aktan E., The limitations of signature-based and dynamic analysis methods in detecting malwares: A case study, Journal of the Faculty of Engineering and Architecture of Gazi University, 37 (1), 305-315, 2022.
5. Makandar A., Patrot A., Malware class recognition using image processing techniques. In 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI*)*, 76-80, 2017.
6. Yue S., Imbalanced malware images classification: a cnn based approach, 2017. https://arxiv.org/abs/1708.08042, Erişim Tarihi Eylül 10.2021
7. Yajamanam S., Selvin V. R. S., Troia F. D. and Stamp M., Deep Learning versus Gist Descriptors for Image-based Malware Classification, In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Funchal-Madeira-Portuga, 553-561, 22-24 Ocak, 2018.
8. Cui Z., Xue F., Cai X., Cao Y., Wang G., Chen J., Detection of malicious code variants based on deep learning, IEEE Trans. Ind. Informatics, 14 (7), 3187-3196, Jul 2018.
9. Bhodia N., Prajapati P., Troia F. D. and Stamp M., Transfer Learning for Image-Based Malware Classification, https://arxiv.org/abs/1903.11551,Yayınlanma Tarihi Ocak 21. 2019, Erişim Tarihi Nisan 22.2021.
10. Ünver H. M., Bakour K., Android malware detection based on image-based features and machine learning techniques, SN Applied Sciences (2020) 2, 1299.2020, https://doi.org/10.1007/s42452-020-3132-2
11. Bakour K., Ünver., H. M., VisDroid: Android malware classification based on local and global image features, bag of visual words and machine learning techniques, Neural Computing and Applications (2021) 33, 3133–3153, 2021.
12. Bakour K., Ünver H.M., DeepVisDroid: android malware detection by hybridizing image-based features with deep learning techniques, Neural Computing and Applications,2021.
13. Venkatraman S., Alazab M., Vinayakumar R., A hybrid deep learning image-based analysis for effective malware detection, Journal of Information Security and Applications 47 (2019) 377–389, 2019.
14. Yuan B., Wang J., Liu D., Guo W., Wu P., Bao X., Byte-level malware classification based on markov images and deep learning, Computers & Security 92 (101740), 2020.
15. Iadarola G., Martinelli F., Mercaldo F. and Santone A., Image-based Malware Family Detection: An Assessment between Feature Extraction and Classification Techniques, In Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020), Online Streaming, 499-506, 2020.
16. Naeem H., Ullah F., Naeem M., Khalid S. vd., Malware detection in industrial internet of things based on hybrid image visualization and deep learning model, Ad Hoc Networks 105 (102154), 2020.
17. Angelo G. D., Ficco M., Palmieri F., Malware detection in mobile environments based on Autoencoders and API-images, Journal of Parallel and Distributed Computing 137, 26–33, 2020.
18. Kabakuş A. T., Doğru İ. A., Çetin A., APK Denetçisi: İzin Tabanlı Android Kötü Amaçlı Yazılım Algılama Sistemi, Digital Investigation 13, 1-14, 2015.
19. Arslan R. S., AndroAnalyzer: android malicious software detection based on deep learning, Peerj Computer Scıence, 2021
20. Anonim, Makine Öğrenmesi Nedir?, Microsoft.,https://azure.microsoft.com/tr-tr/overview/what-is-machine-learning-platform/, Erişim Tarihi Temmuz 05. 2021.
21. SamuelA. L., Some Studies in Machine Learning Using the Game of Checkers. IBM Journal of Research and Development, 3 (3), 210-229, 1959. CiteSeerX 10.1.1.368.2254 *$2.* doi:10.1147/rd.33.0210, Erişim Tarihi Temmuz 03.2021.
22. Nilsson N., Learning Machines, McGraw Hill, 1965.
23. Duda R. and Hart P., Pattern Recognition and Scene Analysis, Wiley Interscience, 1973.
24. Bozinovski S., Teaching space: A representation concept for adaptive pattern classification, University of Massachusetts at Amherst, Computer and Information Science Department. MA., COINS Technical Report No. 81-28.1981, https://web.cs.umass.edu/publication/docs/1981/UM-CS-1981-028.pdf
25. Alpaydın E., Introduction to Machine Learning, Londra: The MIT Press. s. 8. ISBN 978-0-262-01243-0, 2010.
26. Kutlugün M.A., Gözetimli makine öğrenmesi yoluyla türe göre metinden ses sentezleme İstanbul Sabahattin Zaim Üniversitesi, Fen Bilimleri Enstitüsü, YL, 2017.
27. Çalışkan E., Makine Öğrenmesinde Gözetimli ve Gözetimsiz Öğrenme, Databulls, Erişim tarihi Temmuz 04. 2021.
28. Koptur M., Gözetimsiz Öğrenme (K-Merkezli Öbekleme), Makine Öğrenimi, Yayınlanma tarihi 2017, Erişim tarihi Temmuz, 2021.
29. Malık Z. M. M., Al-Shehabı S., Dökeroğlu T., Gözetimsiz Makine Öğrenme Teknikleri ile Miktara Dayalı Negatif Birliktelik Kural Madenciliği, Düzce Üniversitesi Bilim ve Teknoloji Dergisi, 6 (2018) 1119-1138, 2018.
30. Bektaş O., Uçuş Aşamalarının Bölümlendirilmesi: Havacılık Verilerinde Gözetimsiz Öğrenme Uygulaması, AKU J. Sci. Eng, 20 (6),1178-1186, 2020.
31. Bölük N., Uçar Ö., İnner A. B., Mobil Robotlarda Navigasyon Problemi için Pekiştirmeli Öğrenme, Türkiye Robotbilim Konferansı 2019, İstanbul,40-44, 26- 29 Haziran 2019.
32. Başarır G., Çok Değişkenli Verilerde Ayrımsama Sorunu ve Lojistik Regresyon Analizi, Hacettepe Üniversitesi, Uygulamalı istatistik doktora tezi, 1-36, Ankara, 1990.
33. Mohaisen A., Alrawi O., Mohaisen M., AMAL: high-fidelity. behavior-based automated malware analysis and classification, Comput, Secur, 52, 251-256, 2015.
34. Taşcı E., Onan A., K-En Yakın Komşu Algoritması Parametrelerinin Sınıflandırma Performansı Üzerine Etkisinin İncelenmesi, https://ab.org.tr/ab16/bildiri/102.pdf, Erişim Tarihi Temuz 14.2021.
35. Mitchell T., Machine Learning, McGraw Hill, New York, 1997.
36. Han J. and Kamber M., Data mining: concepts and techniques, Morgan Kaufmann Publishers, Burlington, 2006.
37. Nagano Y., Static analysis with paragraph vector for malware detection, IMCOM '17: Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication January, Japan-Beppu, 80, 1–7, 2017.
38. Huda S., Miah S., Hassan M.M., Islam R., Yearwood J., Alrubaian M., Almogren A., Defending unknown attacks on cyber-physical systems by semisupervised approach and available unlabeled data, Inform, Sci, 379, 211–228, 2017.
39. Sahs J. And Khan L., A machine learning approach to android malware detection, In: 2012 European intelligence and security informatics conference (EISIC), Denmark-Odense, 141–7, 22-24 Ağustos, 2012.
40. Milosevic N., Dehghantanha A., Choo K.K. R., Machine learning aided Android malware classification, Computers and Electrical Engineering,61 (2017), 266–274, 2017.

41. Mira F., Brown A., Huang W., Novel malware detection methods by using LCS and LCSS, in: 2016 22nd International Conference on Automation and Computing, ICAC 2016: Tackling the New Challenges in Automation and Computing,U.K-Colchester, 554–559, 07-08 Semtember, 2016.

42. Damodaran A., Troia F.D., Visaggio C.A., Austin T.H., Stamp M., A comparison of static, dynamic and hybrid analysis for malware detection, J. Comput, Virol, Hacking Tech, 13 (1), 1–24, 2017.

43. Markel Z., Bilzor M., Building a machine learning classifier for malware detection, in: 2014 Second Workshop on Anti-malware Testing Research (WATeR), 1-4, 2014.

44. Utku A., Doğru İ.A., Permission based detection system for android malware, Journal of the Faculty of Engineering and Architecture of Gazi University, 32 (4), 1015-1024, 2017.