Teknik Bilimleri Dergisi
Cilt 11, Sayı 2, 41-46, 2021

Journal of Technical Sciences
Volume 11, Issue 2, 41-46, 2021

Android Zararlı Yazılımlarının Derin Öğrenme ile Kategorilerine ve Ailelerine Göre Sınıflandırılması

Mahmut TOKMAK[*1]

[1]Isparta Uygulamalı Bilimler Üniversitesi, Gelendost Meslek Yüksekokulu, Finans-Bankacılık ve Sigortacılık Bölümü, Isparta, Türkiye

(Alınış / Received: 07.06.2021, Kabul / Accepted: 02.07.2021)

Araştırma Makalesi

Anahtar Kelimeler
Android zararlı yazılımları
Zararlı yazılım kategorileri
Zararlı yazılım aileleri
Derin Öğrenme

Özet: En yaygın kullanılan mobil platform olan Android, mobil zararlı yazılımların da hedefi haline gelmiştir. Günden güne de Android zararlı yazılım sayısı ve çeşidi artmaktadır. Bu durum göz önüne alındığında, kötü amaçlı yazılım kategorilerini ve ailelerini tespit etmek, zararlı yazılım analistlerinin işlerini kolaylaştıracaktır. Analistler, benzer davranışlar sergileyen zararlı yazılımları incelemek yerine motivasyonlarını yeni örnekleri incelemeye odaklayacaklardır. Bu çalışmada, ICInvesAndMal2019 Android zararlı yazılım veri setinin dinamik analiz yöntemi ile elde edilen özellikleri barındıran kısmı kullanılmıştır. Kullanılan veri seti ile Android zararlı yazılımları kategorilerine ve ailelerine göre sınıflandırılmıştır. Sınıflandırmada Derin Sinir Ağları (DSA) kullanılmıştır. Kurulan model ile yapılan sınıflandırma sonucunda Android zararlı yazılımların kategorilerine göre sınıflandırmasında %85 doğruluk değerine, Android zararlı yazılımların ailelerine göre sınıflandırılmasında %62 doğruluk değerine erişilmiştir.

Classification of Android Malware by Categories and Families with Deep Learning

Keywords
Android malware
Malware category
Malware family
Deep learning

Abstract: Android, the most widely used mobile platform, has also become the target of mobile malware. The number and variety of Android malware is increasing day by day. Given this situation, detecting malware categories and families will make it easier for malware analysts. Instead of examining malware that exhibits similar behavior, analysts will focus their motivation on examining new examples. In this study, the part of the ICInvesAndMal2019 Android malware dataset containing the features obtained by the dynamic analysis method was used. With the data set used, Android malware is classified by category and family. Deep Neural Network (DNN) was used for classification. As a result of the classification made with the established model, 85% accuracy was achieved in the classification of Android malware by category, and 62% accuracy in classification of Android malware according to family.

1. Giriş

Günümüzde akıllı telefonlar eskiden olduğu gibi sadece telefon görüşmesi yapmak için değil, kişisel bilgileri saklama, sağlık hizmetlerinin takibi, bankacılık uygulamaları, sosyal medya, oyun ve daha birçok e-servis için kullanılan bir araç haline gelmiş ve kullanıcıların vazgeçilmezi olmuştur [17]. International Data Corporation (IDC) firmasının 2021 yılının ilk çeyreğinde güncellediği verilere göre; dünya akıllı telefon pazarı işletim sistemlerinin oranları incelendiğinde Android'in bu pazardaki yeri %83,8 'dir ve firmanın tahminine göre 2023 yılında %84,5 oranında olması beklenmektedir [16].

Android açık kaynak kodlu ve dünya çapında tercih edilen bir işletim sistemi olarak, zararlı yazılımların tehditi altındadır. Bu zararlı yazılımlar kullanıcıdan habersiz özel ücretli numaralara metin mesajları gönderme, kişisel bilgilere erişme ve hatta kullanıcı aygıtına ek zararlı yazılım indirebilme ve yürütebilme özelliklerine sahiptir [3, 4]. Son yıllarda, Android platformunu hedefleyen zararlı yazılım örneklerinin sayısında önemli ölçüde artış gerçekleşmiştir.

McAfee'nin Nisan 2021 tarihli raporuna göre, 2020'nin ilk çeyreğinde 1,5 milyonun altında tespit edilen yeni Android zararlı yazılım sayısı 2020'nin dördüncü çeyreğinde yaklaşık 3,5 milyona kadar çıkmıştır [3, 24].

Android'in yaygın bir şekilde kullanımı ve zararlı yazılımların sayısında görülen artışlar, zararlı yazılımların tespit edilme noktasındaki çalışmaları da gerekli hale getirmiştir. Google Play zararlı yazılımları tespit etmek amacıyla 2012'de Bouncer adını verdikleri sistemi tanıtmışlardır. Bouncer sistemi ile analiz edilmek istenen uygulama, sanal bir ortamda beş dakika boyunca koşturulup, çeşitli testlere tabi tutularak, analiz edilen uygulamadaki zararlı davranışlar tespit edilmeye çalışılmıştır. Ancak Bouncer tespit mekanizmasının tespit edemediği zararlı yazılımların olduğu gösterilmiştir [3]. Bununla birlikte Google, 2017 yılında sürekli çalışan ve cihazı otomatik olarak tarayarak mobil güvenliği sağlamayı amaçlayan Google Play Protect'i duyurmuştur [13]. Ancak, McAfee'nin yayınladığı rapora göre, yayınladığı tarih itibariyle son üç ay içinde tespit edilen zararlı yazılımlar ile test edilmesi sonucunda Google Play Protect hizmetinin başarısız kaldığını bildirmişlerdir [25]. Bu bağlamda, zararlı uygulama sayılarının bu derece artması, zararlı yazılımları tespit etme noktasında ki çeşitli yaklaşım ve çalışmaları zorunlu hale getirmiştir.

Android zararlı yazılımlarını tespiti amacıyla önerilen yöntemler statik analiz, dinamik analiz ve hibrit analiz başlıkları altında kategorize edilmektedir. Statik analiz; uygulamayı çalıştırmadan, yazılmış olan kodun analizi esasına dayanmaktadır. Ancak yazılmış kodların anlaşılmasını zorlaştırmak amacıyla kullanılan kod gizleme gibi şaşırtma tekniklerine karşı çok etkili değillerdir. Diğer taraftan çalışma zamanında oluşturulan ağ trafiğinin izlerini takip edememektedirler [23, 30]. Dinamik analiz, uygulamanın sanal bir ortamda veya gerçek bir cihazda çalıştırılarak uygulamanın davranışlarını belirleme esasına dayanmaktadır. Statik analiz yöntemindeki bu dezavantajlar dinamik analiz yöntemi ile aşılmaya çalışılmaktadır [23]. Hibrit analiz yöntemi ise statik ve dinamik analiz yönteminin birlikte kullanılması ilkesine dayanmaktadır [8].

Zararlı yazılım analizine ilişkin literatürdeki çalışmaların çoğu zararlı yazılımları zararlı olmayan yazılımlardan ayırt etmeye odaklanmışlardır [9]. Bu çalışmalar farklı makine öğrenme yöntemleri ve farklı öznitelikler kullanarak statik [6, 12, 14, 15, 20, 26], dinamik [7, 10, 33] ve hibrit analiz [2, 3, 27, 34] tekniklerini kullanmışlardır.

Android yazılım örneklerinin zararlı veya zararlı olmayan olarak sınıflandırılmasına ek olarak, kategorilerine ve ailelerine göre sınıflandırılma yöntemleri üzerine çalışmalar ortaya konmaktadır.

Aynı aileye ait örnekler benzer davranışlar sergilemekte, aynı güvenlik açıklarından yararlanmakta ve aynı hedeflere sahiplerdir. Bu nedenle bir uygulamanın bilinen zararlı yazılımın bir çeşidi olup olmadığını hızlı bir şekilde kontrol etmek için aile sınıflandırması kullanılabilir. Bu sınıflandırma yöntemi, analistlere incelenmek üzere gelen örneklerden bazılarının bilinen bir zararlı ailesine ait olması durumunda, analistin bu örnekleri incelemek için zaman ve çaba harcamak yerine yeni zararlı yazılımlara odaklanması noktasında büyük fayda getirecektir. Bu durum da zararlı uygulamaların kategori ve ailelerine göre sınıflandırılması konusunu önemli hale getirmektedir [23].

Android zararlı yazılımlarının ailelerini belirlemede önerilen çalışma ve teknikler literatürde mevcuttur. Bu alanda yapılan çalışmalardan bazıları ise şu şekildedir;

Jiang vd., DREBIN veri setinden, hassas opkod dizilerini çıkararak zararlı uygulamaları ailelerine göre K-En Yakın Komşuluk algoritması (KNN) kullanarak sınıflandırmışlardır [19]. Fang vd., Android Malware Dataset (AMD) veri setini kullanmışlar ve dex dosyasını RGB resim formatına dönüştürmüşler ve KNN, Destek Vektör Makineleri (SVM) ve Rastgele Ormanlar (RF) sınıflandırma yöntemleri ile ailelerine göre sınıflandırmışlardır [11].
Chakraborty vd. DREBIN veri setini kullanmışlar ve topluluk öğrenme metotları ile kümeleme ve sınıflandırma yapmışlardır [9]. Kim vd., Andro-Simnet veri setini kullanmışlar ve API çağrılarını kullanarak benzerlik tabanlı kümeleme yapmışlardır [21]. Lashkari vd., CICAndMal2017 veri setini oluşturmuşlar ve ağ trafiğine göre zararlı yazılımları tespit etmeye çalışmışlardır. RF, KNN, ve Karar Ağacı (DT) kullanarak ikili sınıflandırma ve ailelerine göre sınıflandırma yapmışlardır. [22]. Xu vd., DREBIN, Marvin, VirusShare, Contagio-Dump veri setlerini kullanmışlar ve kontrol akış grafları ve veri akış grafları kullanarak sınıflandırma yapmışlardır [32]. Sun vd., DREBIN veri setini kullanmışlardır. Zararlı uygulamaların ikili bayt-kodlarını resim formatına çevirip derin öğrenme yöntemlerinden evrişimsel sinir ağları ile sınıflandırma yapmışlardır [28]. Türker ve Can, AMD ve DREBIN veri seti kullanarak zararlı yazılımların ailelerine göre sınıflandırılması ile ilgili bir çalışma yapmıştır. İzinler ile API çağrılarını öznitelik olarak kullanmışlar ve SVM, DT, Lojistik Regresyon (LR), KNN, RF AdaBoost, and Çok Katmanlı Perseptron (MLP) teknikleri ile sınıflandırma yapmışlardır [31]. Mahmoud ve Abuthawabeh, CICAndMal2017 veri setini kullanmışlardır. Kategorilerine ve ailelerine göre Ekstra Ağaç (ET), RF, DT algoritmaları kullanarak sınıflandırma yapmışlardır [1]. Imtiaz vd., CICInvesAndMal2019 veri setini kullanmışlardır. Zararlı yazılımları kategorilerine ve ailelerine göre J48, Navie Bayes, SMO, MLP, Derin Öğrenme yöntemleri ile sınıflandırmışlardır [17].

Bu çalışmada, Android zararlı yazılımlarının ait oldukları kategori ve aile tespit edilmeye çalışılmıştır. CICInvesAndMal2019 veri seti kullanılarak kurulan DSA modeli ile eğitimler ve testler yapılmıştır. Deneysel çalışmalar, zararlı yazılımların kategorilerine ve ailelerine göre sınıflandırması olmak üzere iki ayrı çalışma olarak yapılmış ve elde edilen sonuçlar çalışmada sunulmuştur.

2. Materyal ve Metot

2.1. Veri seti

Çalışmada yakın zamanda yayınlanan Android Kötü Amaçlı Yazılım Veri Kümesi (CICInvesAndMal2019) kullanılmıştır [18]. Kullanılan veri seti zararlı ve zararsız Android uygulamalarının gerçek akıllı cihazlarda test edildiği CICAndMal2017 [5] veri kümesinin devamı niteliğindedir. Bu veri kümesinde iki farklı yöntemle çıkarılmış eğitim seti ve test seti bulunmaktadır. Veri setinin statik analiz yöntemi ile çıkarılmış bölümünde izinler, amaçlar, API çağrıları gibi öznitelikler yer almaktadır. Dinamik analiz ile çıkardıkları bölümünde ise API çağrıları ve ağ trafiği öznitelikleri bulunmaktadır. Bu çalışmada dinamik analiz ile elde edilen veri seti kullanılmıştır. Bu veri kümesindeki zararlı yazılım örnekleri dört kategoride toplanmıştır. Bunlar: Reklam Yazılımı (Adware), Fidye Yazılımı (Ransomware), Aldatma Yazılımı (Scareware) ve SMS.Zararlı Yazılımıdır (SMS Malware). Ayrıca bu zararlı yazılım kategorilerinin aileleri de Tablo 1'de gösterilmiştir.

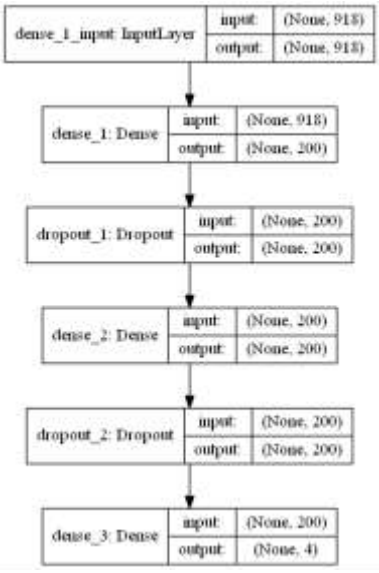
Tablo 1. Veri seti zararlı yazılım kategori ve aileleri [1]

Kategori	Aile Tipi		
Adware	Ewind	koodous	Kemoge
	Dowgin	Mobidash	Youmi
	Feiwo	Selfmite	Shuanet
		Gooligan	
Ransomware	Charger	Pletor	LockerPin
	Jisut	PornDroid	Svpeng
	Koler	RansomBO	WannaLocker
		Simplocker	
Scareware	Android Defender	FakeAV	FakeApp.AL
	Android Spy.277	FakeJobOffer	FakeAV
	AV for Android	FakeTaoBao	FakeApp
	AVpass	Penetho	VirusShield
SMS Malware	BeanBot	Jifake	FakeNotify
	Biige	Mazarbot	SMSsniffer
	FakeInst	Nandrobox	FakeMart
		Plankton	

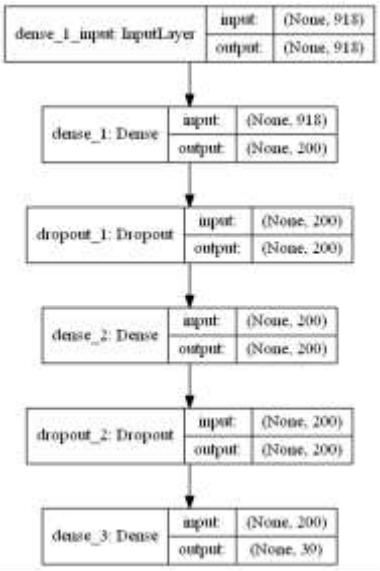
2.2. Derin sinir ağları

Derin Sinir Ağları (DSA): çok katmanlı tam bağlantılı sinir ağlarıdır ve bir girdi katmanı, birden çok gizli katman ve bir çıktı katmanından oluşur. Bir katmandaki her düğüm, bir sonraki katmandaki diğer düğümlere bağlanır. Gizli katman sayısını artırarak ağı derinleşmesi sağlanmaktadır. DSA'lar, yüksek performansları nedeniyle günümüzde birçok gerçek dünya uygulamasında kullanılmaktadır. DSA'lar görüntü analizi, örüntü tanıma, nesne algılama, doğal dil işleme ve sürücüsüz arabalar gibi birçok alanda etkileyici sonuçlar elde edilerek kullanılmaktadır [17].

Bu çalışmada veri setleri DSA ile eğitilmiş ve teste tabi tutulmuştur. DSA modeli Python 3.7 kullanılarak oluşturulmuş ve scipy, numpy, matplotlib, sklearn, keras, pandas kütüphaneleri kullanılmıştır. Zararlı yazılımları kategorilerine ve ailelerine göre sınıflandırmak için oluşturulan DSA modelleri bir adet giriş katmanı, 1 adet gizli katman, giriş ve gizli katmandan sonra ağın ezberlemesini (overfitting) önlemek amacıyla 1'er adet Dropout katmanı ve 1 adet çıkış katmanı içermektedir. Gizli katman 200 adet düğümden oluşmaktadır. DSA'da ezberlemeyi önleme amacıyla Dropout rate 0,5 olarak belirlenmiştir. Giriş katmanı ve gizli katmanlarda aktivasyon fonksiyonu olarak Rectifier Lienar Units (RELU) kullanılmıştır. Çıkış katmanında kullanılan aktivasyon fonksiyonu ise softmax'tır. Modeli eğitmeden önce öğrenme sürecinin yapılandırması işlemi compile fonksiyonu yardımı ile gerçekleştirilmiştir. Bu fonksiyona öğrenme hızını ayarlayan fonksiyonu belirlemek için gerekli olan optimizer parametresi olarak adam, model çoklu bir sınıflandırma içerdiği için loss parametresi categorical_crossentropy ve modelin performansını değerlendirmek için kullanılacak fonksiyonu belirlemek için verilen metrics parametresine accuracy değerleri verilmiştir. Son olarak model fit fonksiyonu kullanılarak eğitilmiştir. Fit fonksiyonuna modeli eğitmek için kaç defa çalışacağını belirleyen epoch parametresi 150 olarak verilmiştir. Zararlı yazılımları kategorilerine göre sınıflandırmak için kurulan DSA modeli Şekil 1'de, ailelerine göre sınıflandırmak için kurulan DSA modeli Şekil 2'de gösterilmiştir.



Şekil 1. Kategori sınıflandırması için kurulan DSA modeli



Şekil 2. Aile sınıflandırması için kurulan DSA modeli

Kurulan modelin başarımının değerlendirilebilmesi için kullanılan; doğru sınıflandırılan sınıf örneklerinin oranı olan doğruluk değeri Eşitlik 1'de, pozitif olarak tahmin edilen örneklerin gerçekte ne kadarının pozitif olduğunu ifade eden kesinlik (precision) değeri Eşitlik 2'de, gerçek pozitif değerlerin ne kadarının doğru olduğunu ifade eden duyarlılık (recall) değeri Eşitlik 3'te, kesinlik ve duyarlılık değerlerinin harmonik ortalaması olan F1-skoru (F1 score) Eşitlik 4'te gösterilmiştir. Ayrıca test için ayrılan örnek sayısı, destek (Support) ibaresi ile ifade edilmiş ve Tablo 1 ve Tablo 2'nin ilgili sütunlarında gösterilmiştir.

$$\text{Doğruluk}=\frac{TP+TN}{TP+FN+TN+FP} \quad (1)$$

$$\text{Kesinlik}=\frac{TP}{TP+FP} \quad (2)$$

$$\text{Duyarlılık}=\frac{TP}{TP+FN} \quad (3)$$

$$F1\text{-Skor}=\frac{2x\text{Duyarlılık}x\text{Kesinlik}}{\text{Duyarlılık}+\text{Kesinlik}} \quad (4)$$

3. Bulgular

Çalışmadaki tüm deneysel çalışmalar, 2.50 GHz Intel Core i5 işlemci ve 16 GB belleğe sahip Microsoft Windows 10 (64-bit) sürümü üzerinde Python 3.7 kullanılarak gerçekleştirilmiştir. Çalışmada önerilen DSA modeli ile öncelikle Android zararlı yazılımları kategorilerine göre sınıflandırılmıştır. Yapılan test sonucunda %85 doğruluk değerine, %86 kesinlik değerine, %85 duyarlılık değerine, %85 F1-skoru değerine erişilmiştir ve elde edilen sonuçlar Tablo 2'de gösterilmiştir. Aynı zamanda testin sonucunda doğru ve yanlış olarak sınıflandırılan zararlı yazılım kategorilerini gösteren karmaşıklık matrisi (confusion matrix) Şekil 3'te gösterilmiştir.

Şekil 3. Karmaşıklık matrisi

Tablo 2. Kategori sınıflandırma performans ölçütleri

	Precision	Recall	F1-score	Support
Ransomware	0,81	0,74	0,77	23
Adware	0,80	1,00	0,89	24
Scareware	0,79	0,84	0,82	32
Sms Malware	1,00	0,84	0,91	37
Average	**0,86**	**0,85**	**0,85**	**116**
Accuracy			**0,85**	**116**

Android zararlı yazılımları ailelerine göre sınıflandırılmak için kurulan DSA modeli ile yapılan test sonucunda %62 doğruluk değerine, %65 kesinlik değerine, %62 duyarlılık değerine, %59 F1-skoru değerine erişilmiştir ve elde edilen sonuçlar Tablo 3'te gösterilmiştir.

Tablo 3 Aile sınıflandırma performans ölçütleri

	Precision	Recall	F1-score	Support
AndroidDefender	0,75	1,00	0,86	6
AvForAndroid	0,25	1,00	0,40	2
FakeTaoBao	1,00	0,50	0,67	2
RansomBO	0,75	0,75	0,75	4
android.spy.277	1,00	1,00	1,00	1
avpass	1,00	0,75	0,86	4
beanbot	0,60	0,60	0,60	5
biige	0,00	0,00	0,00	1
charger	1,00	1,00	1,00	4
dowgin	1,00	0,67	0,80	3
ewind	0,67	1,00	0,80	4
fakeapp	0,00	0,00	0,00	2
fakeav	1,00	0,75	0,86	4
fakeinst	0,00	0,00	0,00	4
fakejoboffer	1,00	1,00	1,00	3
fakemart	0,30	1,00	0,46	3
feiwo	0,00	0,00	0,00	4
gooligan	0,33	0,50	0,40	2
jifake	0,50	0,50	0,50	4
jisut	1,00	0,67	0,80	3
kemoge	0,67	0,67	0,67	3
koler	0,00	0,00	0,00	1
lockerpin	0,00	0,00	0,00	1
mazarbot	1,00	0,20	0,33	5
mobidash	0,40	1,00	0,57	2
nandrobox	0,50	0,75	0,60	4
penetho	0,80	1,00	0,89	4
plankton	1,00	0,33	0,50	3
pletor	1,00	1,00	1,00	3
porndroid	1,00	1,00	1,00	2
selfmite	0,00	0,00	0,00	1
shuanet	1,00	0,50	0,67	2
simplocker	1,00	0,50	0,67	2

smssniffer	0,67	1,00	0,80	4
svpeng	0,00	0,00	0,00	3
virusshield	0,50	0,25	0,33	4
wannalocker	0,17	1,00	0,29	1
youmi	0,00	0,00	0,00	2
zsone	1,00	0,75	0,86	4
Average	**0,65**	**0,62**	**0,59**	**116**
Accuracy			**0,62**	**116**

4. Tartışma ve Sonuç

Android zararlı yazılımlarının sayısal ve çeşitlilik anlamında arttığı, kullanıcıları tehdit ettiği günümüz teknoloji dünyasında mobil güvenlik tedbirlerinin alınması, zararlı yazılımların tespit edilerek verebilecekleri zararların önlenmesi önemli bir çalışma alanı olarak karşımıza çıkmaktadır. Android yazılımlarının, zararlı ya da zararsız olarak tespitinin yanında, bu yazılımların kategorilerinin ve ailelerinin tespiti üzerine yapılan analizler ise bu alandaki en önemli konulardan biridir.

Bu çalışmada, Android zararlı yazılımlarının kategori ve aile tespiti amacıyla, CICAndMal2017 veri setinin yakın zamanda güncellenmiş CICInvesAndMal2019 versiyonu kullanılmıştır. Veri setinin dinamik analiz ile çıkarılmış olan kısmı içindeki öznitelikler kullanılmıştır. Kullanılan bu öznitelikler kurulan DSA modeli ile eğitilip test edilmiştir. Deneysel çalışmalar, zararlı yazılımların kategorilerine ve ailelerine göre sınıflandırması olmak üzere iki ayrı çalışma olarak yapılmıştır. Kategoriye göre yapılan test sonucunda sırasıyla %85 doğruluk, %86 kesinlik, %85 duyarlılık, %85 F1-skoru değerleri elde edilmiştir. Aileye göre yapılan test sonucunda sırasıyla %62 doğruluk, %65 kesinlik, %62 duyarlılık, %59 F1-skoru değerleri elde edilmiştir.

Literatürde görüldüğü kadarıyla CICAndMal2017 veri seti ile yapılan çalışmalar; çeşitli makine yöntemleri uygulayarak elde ettiği kategori sınıflandırmasında %50-%81 arasında doğruluk değeri elde etmişlerdir Aile sınıflandırmasında ise %30-%61 doğruluk değeri elde etmişlerdir [1, 22, 29]. Imtiaz vd. CICInvesAndMal2019 veri seti ve DSA kullanılarak yaptığı çalışmada kategori sınıflandırmasında %80,3 doğruluk, %82,2 kesinlik, %80,3 duyarlılık, %80,5 F1-skoru değeri elde etmişlerdir. Aile sınıflandırmasında %55,7 doğruluk, %59,1 kesinlik, %55 duyarlılık değeri, %58,1 F1-skoru değeri elde etmişlerdir. Navie Bayes yöntemini kullandıkları aile sınıflandırmasında ise %59 doğruluk, %65 kesinlik, %59 duyarlılık değeri, %58,1 F1-skoru değeri elde etmişlerdir [17]. Imtiaz vd. yaptığı çalışma da paylaştığı grafikler incelendiğinde; belirli bir çevrimden sonra ağın öğrenmeyi ezberlediği düşünüldüğü için önerilen çalışmada kurulan DSA modelinde, ağdaki ezberlemenin önlenmesi amacıyla parametre değerleri regülize edilmiş, giriş ve gizli katmanından sonra Dropout metodu kullanılarak mevcut çalışmalardaki performans ölçütleri daha yukarı çekilmiştir.

İlerleyen çalışmalarda, mevcut veri setlerine ek olarak kendi veri setimizi oluşturarak, çeşitli makine öğrenme yöntemleri ile analizler yapılması elde edilen performans ölçütlerinin daha da yukarılara ilerletilmesi hedeflenmektedir.

Kaynakça

[1] Abuthawabeh, M. and Mahmoud, K. 2020. Enhanced Android Malware Detection and Family Classification, using Conversation-level Network Traffic Features. The International Arab Journal of Information Technology, 17, 4A, 607–614.

[2] Alshahrani, H., Mansourt, H., Thorn, S., Alshehri, A., Alzahrani, A. and Fu, H. 2018. DDefender: Android application threat detection using static and dynamic analysis. 2018 IEEE International Conference on Consumer Electronics (ICCE), 1–6.

[3] Alzaylaee, M.K., Yerima, S.Y. and Sezer, S. 2020. DL-Droid: Deep learning based android malware detection using real devices. Computers & Security, 89, 101663.

[4] Anagnostopoulos, M., Kambourakis, G. and Gritzalis, S. 2016. New facets of mobile botnet: architecture and evaluation. International Journal of Information Security, 15, 5, 455–473.

[5] Android Malware Dataset, 2021. https://www.unb.ca/cic/datasets/andmal2017.html (Erişim Tarihi: 10.4.2021).

[6] Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K. and Siemens, C. 2014. Drebin: Effective and explainable detection of android malware in your pocket. Ndss, 23–26.

[7] Bhatia, T. and Kaushal, R. 2017. Malware detection in android based on dynamic analysis. 2017 International Conference on Cyber Security And Protection Of Digital Services (Cyber Security), 1–6.

[8] Cam, N.T., Pham, V.-H. and Nguyen, T. 2019. Detecting sensitive data leakage via inter-applications on Android using a hybrid analysis technique. Cluster Computing, 22, 1, 1055–1064.

[9] Chakraborty, T., Pierazzi, F. and Subrahmanian, V.S. 2020. EC2: Ensemble Clustering and Classification for Predicting Android Malware Families. IEEE Transactions on Dependable and Secure Computing, 17, 2, 262–277.

[10] De Lorenzo, A., Martinelli, F., Medvet, E., Mercaldo, F. and Santone, A. 2020. Visualizing the outcome of dynamic analysis of Android malware with VizMal. Journal of Information Security and Applications, 50, 102423.

[11] Fang, Y., Gao, Y., Jing, F. and Zhang, L. 2020. Android Malware Familial Classification Based on DEX File Section Features. IEEE Access, 8, 10614–10627.

[12] Feizollah, A., Anuar, N.B., Salleh, R., Suarez-Tangil, G. and Furnell, S. 2017. Androdialysis: Analysis of android intent effectiveness in malware detection. computers & security, 65, 121–134.

[13] Google Play, 2021. https://www.android.com/play-protect/ (Erişim Tarihi: 10.4.2021).

[14] Hou, S., Saas, A., Chen, L. and Ye, Y. 2016. Deep4maldroid: A deep learning framework for android malware detection based on linux kernel system call graphs. 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), 104–111.

[15] Hou, S., Saas, A., Chen, L., Ye, Y. and Bourlai, T. 2017. Deep neural networks for automatic android malware detection. Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, 803–810.

[16] IDC, 2021. https://www.idc.com/promo-/smartphone-market-share (Erişim Tarihi: 28.5.2021).

[17] Imtiaz, S.I., Rehman, S. ur, Javed, A.R., Jalil, Z., Liu, X. and Alnumay, W.S. 2021. DeepAMD: Detection and identification of Android malware using high-efficient Deep Artificial Neural Network. Future Generation Computer Systems, 115, 844–856.

[18] Investigation of the android malware (cicinvesandmal2019), 2021. https: //www.unb.ca/cic/datasets/invesandmal2019. html (Erişim Tarihi: 10.4.2021).

[19] Jiang, J., Li, S., Yu, M., Li, G., Liu, C., Chen, K., Liu, H. and Huang, W. 2019. Android Malware Family Classification Based on Sensitive Opcode Sequence. 2019 IEEE Symposium on Computers and Communications (ISCC), June , Barcelona, Spain, 1–7.

[20] Karbab, E.B., Debbabi, M., Derhab, A. and Mouheb, D. 2018. MalDozer: Automatic framework for android malware detection using deep learning. Digital Investigation, 24, S48–S59.

[21] Kim, H.M., Song, H.M., Seo, J.W. and Kim, H.K. 2018. Andro-Simnet: Android Malware Family Classification using Social Network Analysis. 2018 16th Annual Conference on Privacy, Security and Trust (PST), August , Belfast, 1–8.

[22] Lashkari, A.H., Kadir, A.F.A., Taheri, L. and Ghorbani, A.A. 2018. Toward developing a systematic approach to generate benchmark android malware datasets and classification. 2018 International Carnahan Conference on Security Technology (ICCST), 1–7.

[23] Massarelli, L., Aniello, L., Ciccotelli, C., Querzoni, L., Ucci, D. and Baldoni, R. 2017. Android malware family classification based on resource consumption over time. 2017 12th International Conference on Malicious and Unwanted Software (MALWARE), October , Fajardo, 31–38.

[24] McAfee, 2021. https://www.mcafee.com-/enterprise/en-us/assets/reports/rpquarterly-threats-apr-2021.pdf (Erişim Tarihi: 30.5.2021).

[25] McAfee, 2018. https://www.mcafee.com/-enterprise/en-us/assets/reports/rp-mobile-threat-report-2018.pdf (Erişim Tarihi: 11.10.2019).

[26] Milosevic, N., Dehghantanha, A. and Choo, K.-K.R. 2017. Machine learning aided Android malware classification. Computers & Electrical Engineering, 61, 266–274.

[27] Sugunan, K., Kumar, T.G. and Dhanya, K.A. 2018. Static and dynamic analysis for android malware detection. Advances in Big Data and Cloud Computing. Springer. 147–155.

[28] Sun, Y., Chen, Y., Pan, Y. and Wu, L. 2019. Android malware family classification based on deep learning of code images. IAENG International Journal of Computer Science, 46, 4, 524–533.

[29] Taheri, L., Kadir, A.F.A. and Lashkari, A.H. 2019. Extensible Android Malware Detection and Family Classification Using Network-Flows and API-Calls. 2019 International Carnahan Conference on Security Technology (ICCST), October , CHENNAI, India, 1–8.

[30] Tam, K., Feizollah, A., Anuar, N.B., Salleh, R. and Cavallaro, L. 2017. The evolution of android malware and android analysis techniques. ACM Computing Surveys (CSUR), 49, 4, 1–41.

[31] Turker, S. and Can, A.B. 2019. AndMFC: Android Malware Family Classification Framework. 2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops), September , Istanbul, Turkey, 1–6.

[32] Xu, Z., Ren, K. and Song, F. 2019. Android Malware Family Classification and Characterization Using CFG and DFG. 2019 International Symposium on Theoretical Aspects of Software Engineering (TASE), July , Guilin, China, 49–56.

[33] Yang, Y., Wei, Z., Xu, Y., He, H. and Wang, W. 2018. Droidward: an effective dynamic analysis method for vetting android applications. Cluster Computing, 21, 1, 265–275.

[34] Yuan, Z., Lu, Y. and Xue, Y. 2016. Droiddetector: android malware characterization and detection using deep learning. Tsinghua Science and Technology, 21, 1, 114–123.