www.dergipark.gov.tr
ISSN:2148-3736
El-Cezerî Fen ve Mühendislik Dergisi
Cilt: 8, No: 3, 2021 (1536-1549)

El-Cezerî Journal of Science and Engineering
Vol: 8, No: 3, 2021 (1536-1549)
DOI :10.31202/ecjse.967919

ECJSE

Makale / Research Paper

Zararlı Yazılım Tespiti Amacıyla Kullanılan Makine Öğrenmesi Algoritmalarının Büyük Veri Platformlarındaki Performanslarının İncelenmesi

Sercan GÜLBURUN[a], Murat DENER[b]

[a,b]Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgi Güvenliği Mühendisliği, Ankara/TÜRKİYE
sercan.gulburun@gazi.edu.tr, muratdener@gazi.edu.tr

Received/Geliş: 08.07.2021 **Accepted/Kabul:** 31.08.2021

Öz: Bilgi teknolojileri varlıklarının hem bireylerin günlük hayatlarındaki hem de kurum ve kuruluşların işleyişindeki yeri son çeyrek asırda hızlı bir artış göstermiş, bu artışa paralel olarak bilgi varlıklarına yönelik tehditler de artmıştır. Zararlı yazılımlar, bilgi varlıklarına yönelik başlıca tehditlerden biridir. Sürekli olarak kendini yenileyen zararlı yazılımlara karşı geleneksel tespit yaklaşımlarının yetersiz kalması sebebiyle, makine öğrenmesi modelleri kullanan tespit yaklaşımları geliştirilmiştir. Bu çalışmada, zararlı yazılım tespiti maksadıyla kullanılan farklı makine öğrenme algoritmalarının çeşitli büyük veri teknolojileri ve platformları üzerinde ortaya koydukları performanslar incelendi. Modeller, Kaggle Zararlı Yazılım Tespiti veri seti kullanılarak eğitildi. En iyi doğruluk (%98.8), kesinlik (%98.5), f1 skoru (%98.2) ve yanlış pozitif oranı (%2) performansları Google Colaboratory ortamında Sci-Kit Learn kütüphanesi ile çalıştırılan rastgele orman modeli ile elde edildi.

Anahtar Kelimeler: Büyük Veri, Makine Öğrenmesi, Zararlı Yazılım Tespiti, Bilgi Güvenliği

Analyzing Performance of Machine Learning Algorithms for Malware Detection in Big Data Platforms

Abstract: The place of information technology assets in both the daily lives of individuals and the functioning of institutions and organizations has increased rapidly in the last quarter century and in parallel, threats to information assets have also increased. One of the main threats to these assets is malware. Detection approaches using machine learning models have been developed due to the inadequacy of traditional detection approaches against constantly regenerating malware. In this study, the performances of different machine learning algorithms used for malware detection on various big data technologies and platforms were examined. Models were trained using the Kaggle Malware Detection dataset. The best accuracy (98.8%), precision (98.5%), f1 score (98.2%) and false positive rate (2%) performances were obtained with the random forest model run with the Sci-Kit Learn library in the Google Colaboratory environment.

Keywords: Big Data, Machine Learning, Malware Detection, Information Security

1. Giriş

Bilgi teknolojileri, eğitimden yönetime, sağlıktan ekonomiye insan hayatını etkileyen birçok alanda aktif şekilde yer bulmaktadır. Bilgi varlıkları hayatı kolaylaştıran araçlar olmasının yanında, büyük ekonomik değeri ve etkisi olan enstrümanlar haline gelmiştir [1-4]. Bu gelişime paralel olarak, bilgi varlıklarına yönelik tehditlerin boyutu ve karmaşıklığı da artmıştır. Özellikle son yıllarda, kötücül yazılımlar kullanılarak yapılan bazı saldırılar ciddi maddi kayıplara neden olmuştur [5].

Bu makaleye atıf yapmak için
Gülburun, S., Dener, M., "Büyük Veri Ortamlarında Zararlı Yazılım Tespiti Kapsamında Makine Öğrenmesi Algoritmalarının Performansının İncelenmesi" El-Cezerî Fen ve Mühendislik Dergisi 2021, 8 (3); 1536-1549.

How to cite this article
Gülburun, S., Dener, M., "*Analyzing Performance of Machine Learning Algorithms in Big Data Environment in terms of Malware Detection*" El-Cezerî Journal of Science and Engineering, 2021, 8 (3); 1536-1549.
ORCID ID: [a]0000-0001-5272-3911; [b]0000-0001-5746-6141

Zararlı yazılımların tespiti için imza tabanlı, tanımlama tabanlı ve anomali tabanlı tespit yaklaşımları bulunmaktadır. İmza tabanlı yaklaşımlar, bilinen zararlı yazılımların kendilerine özgü davranışlarına dayanarak tespit yaparken, anomali tabanlı yaklaşımlar, normal dışı görülen durumların tespit ederek zararlı yazılım tespiti yaparlar. Tanımlama tabanlı tespit yaklaşımı ise anomali tabanlı tespit yaklaşımının bir çeşidi olup, yüksek yanlış pozitif durumları sergileyen anomalilere çözüm olarak kullanılırlar. [6]

Zararlı yazılımların analizi için, yazılımların statik ve dinamik özelliklerinin kullanımıyla tespitler gerçekleştirilebilir. Statik analizde kullanılan tespit desenleri, imzaları, bit dizilerini, kütüphane çağrılarını, kontrol akış grafiklerini ve operasyonel işlem frekanslarının dağılımı gibi hususları içerir. Bu tür analiz özellikle karartılmış kodların tespiti konusunda yetersizdir. Dinamik analiz, zararlı kodun kontrollü bir ortamda çalıştırılarak analiz edilmesi işlemidir. Çeşitli araç ve ortamlar kullanılarak, fonksiyon çağrısı izleme, fonksiyon parametreleri analizi, veri akışı takibi gibi tekniklerle dinamik analiz gerçekleştirilebilir [7].

Büyük veri kavramı, çok çeşitli ve karmaşık yapıları olan, depolama, analiz ve görselleştirme işlemleri açısından zorluklara sahip çok büyük veri setleri için kullanılan bir terimdir. Hacim, çeşitlilik, hız karakteristikleri ile tanımlanan büyük verilerin işlenmesi için farklı yaklaşımlar (MapReduce) ve teknolojiler (Hadoop, Yüksek Performanslı Hesaplama Kümesi - HPCC) geliştirilmiştir [8]. Büyük veri, gizlilik ve bilgi güvenliği açısından kritik problemler ortaya çıkarırken, diğer taraftan büyük veri analitiği ise iç ve dış güvenlik verilerinin korelasyonu ile siber saldırıların önlenmesi ve tespiti için önemli fırsatlar sunmaktadır [9].

Çalışmada, farklı büyük veri platformlarında zararlı yazılım tespiti amacıyla kullanılan makine öğrenmesi algoritmalarının etkinliğinin ortaya konması amaçlanmış olup, rastgele orman (Random Forest - RF), karar ağaçları (Decision Trees – DT) ve gradyan yükseltme ağaçları (Gradient Boosting Trees – GBT) algoritmalarının performansları Google Colaboratory, Azure HDInsight, Amazon EMR ve Google Dataproc ortamlarında karşılaştırılmış, ayrıca Sci-Kit kütüphanesi kullanılarak elde edilen sonuçlar da kullanılarak değerlendirme gerçekleştirildi.

Çalışmanın ikinci bölümünde, özellikle büyük veri ortamına uygun zararlı yazılım tespiti için yapılan güncel bazı çalışmalar ele alındı, üçüncü bölümünde kullanılan veri seti tanıtılmış ve ilgili algoritmaların farklı ortamlardaki performansları ortaya konulmuş, son bölümde ise genel değerlendirmeler ile gelecekte yapılabilecek geliştirmelere yer verildi.

2. İlgili Çalışmalar

Büyük veriye uyumlu, yinelemeli çok katmanlı bir toplu öğrenme metodunun sunulduğu çalışmada [10], farklı toplu öğrenme meta sınıflandırıcıların belirli sayıda katmana dahil edilip otomatik olarak oluşturulmuş yinelemeli bir sistemde birleştirilmesi suretiyle toplu meta sınıflandırıcıların yüksek katmanlardaki toplu meta sınıflandırıcıların birer parçası olarak yer alması sağlanmıştır. Sunulan yöntemin her ne kadar büyük veriye uygun olduğu belirtilse de çalışma ortamında büyük veri platformlarından faydalanılmamış, Weka Platformu üzerinde uygulama gerçekleştirilmiştir. Temel sınıflandırıcı olarak RF ve toplu öğrenici olarak ise süsleme yöntemi kullanılarak en iyi sonuçlar elde edilmiştir.

Davranışsal bir zararlı yazılım sınıflandırıcısı olan Zararlı Yazılım Tespiti için Çok Katmanlı Grafik (Multilayer Graph for Malware Detection - MAGMA) metodunun sunulduğu çalışmada [11], ağ aktivitelerinden çıkarılan desenlerin zaman, uzay ve ağ protokolü kapsamında korelasyonu yapılarak öncelikle Ağ Bağlantı Grafiği oluşturulmuş ve daha sonra bu grafikten özellik çıkarımı yapılarak bir denetimli sınıflandırıcı tasarlanmıştır. Farklı sınıflandırıcılar kullanılarak

gerçekleştirilen sınıflandırmalar neticesinde en iyi sonuçların RF metoduyla alındığı görülmüş ancak çalışmanın gerçekleştirilme ortamına değinilmemiştir.

Apache Spark üzerine kurulu, büyük ölçekli bir zararlı yazılım analiz metodunun sunulduğu çalışmada [12], makine öğrenmesi algoritmaları kullanılarak sıfırıncı gün saldırılarına karşı da etkili olan bir sistem geliştirilmiştir. Apache Spark platformu kullanılarak gerçekleştirilen çalışmada farklı sınıflandırıcılar ile tahmin gerçekleştirilmiş olup en iyi sonuç RF metoduyla sağlanmıştır.

Yazılımların statik ve dinamik özelliklerinin birlikte kullanılmasıyla büyük veri ortamında tespit gerçekleştirilmesi amaçlanan çalışmada [13], tahmin için biri ağırlıklı oylamaya diğeri istiflemeye dayalı iki metot sunulmuştur. Apache Spark platformu üzerinde gerçekleştirilen çalışmada Naive Bayes, K en yakın komşu, DT, destek vektör makinesi ve RF sınıflandırıcıları ile tahminler gerçekleştirilmiş olup en iyi sonuç RF yöntemi ile elde edilmiştir.

Zararlı yazılım tespiti kapsamında kullanılan mevcut toplu öğrenme yaklaşımlarının boyut, bellek ve hesaplama gereksinimlerinin yüksek olması probleminin temel alındığı çalışmada [14], HPC (Hybrid Consensus Pruning) yaklaşımı ile temel sınıflandırıcı sayısının azaltılması yaklaşımı izlenmiştir. Çalışmada sunulan metodun büyük veri için uygun olduğu belirtilse de çalışma bir büyük veri platformu üzerinde değil Weka uygulaması üzerinde gerçekleştirilmiştir.

Zararlı yazılım ve siber saldırı tespiti kapsamında kullanılan başarılı tekniklerden olan IP itibar sistemlerinin yüksek yönetim maliyeti, hatalı pozitif oranları, tüketim süresi gibi sorunlarının aşılmasını hedefleyen çalışmada [15], dinamik zararlı yazılım analizi, siber tehdit istihbaratı, makine öğrenmesi ve veri adli analizine dayalı hibrit bir yaklaşım önerilmiştir. Farklı makine öğrenme yöntemleriyle gerçekleştirilen çalışmalarda en iyi sınıflandırmanın DT ile gerçekleştirildiği görülmüştür.

Sahoo ve diğerleri [16] tarafından yapılan çalışmada, Hadoop dağıtık dosya sistemi ortamında, Map-Reduce yaklaşımı ile yapısal olmayan verilerden elde edilen anahtar-değer çifti kullanılarak zararlı yazılım tespiti yapılmıştır. Suhasini ve diğerleri [17], HDFS ve MapReduce Parser gibi büyük veri için özelleştirilmiş araçlar kullanılarak gerçekleştirilen çalışmada, ağ trafiği ve olay kayıtları incelenmesi, anomalilerin ve şüpheli davranışların ortaya konması suretiyle sanallaştırma ortamlarında zararlı yazılım tespiti yapılması için bir model tasarlamıştır. Statik, dinamik ve imaj işleme için görselleştirme ve derin öğrenme mimarilerinin birleştirildiği çalışmada [18], gerçek zamanlı zararlı yazılım tespiti gerçekleştirilmesi için büyük veri ortamına uygun ölçeklenebilir bir mimari önerilmiş, farklı makine öğrenmesi ve derin öğrenme yöntemleri ile gerçekleştirilen tahminler neticesinde en iyi sonucun CNN ve LSTM'in birlikte kullanıldığı durumda alındığı görülmüştür.

Cevap süresi ve tespit performansı arasında bir ödünleşme yapılabilmesi için statik ve dinamik analiz yöntemlerinin birlikte kullanıldığı çalışmada [19], istemciler tarafından sağlanan veriler ile sürekli öğrenme sürecinin sağlandığı bir zararlı yazılım tespit modeli model sunulmuştur. Zararlı yazılımların statik ve dinamik özelliklerinden faydalanılan çalışmada [20], derin öğrenme ve büyük veri analitiğinin kullanıldığı gerçek zamanlı bir izleme, analiz ve tespit yaklaşımı önerilmiştir. Yousefi-azar ve diğerleri [21] tarafından yapılan çalışmada, özellik çıkarımı, benzerlik ölçümü ve sınıflandırma safhalarından oluşan, işletim sistemi ve araç bağımsız bir zararlı yazılım tespit şeması olan Malytics ile Android ve Windows ortamlarında etkin şekilde zararlı yazılım tespiti hedeflenmiştir. Mao ve diğerleri [22] tarafından, uç kullanıcı bilgisayarlarındaki çalıştırılabilir dosyaların loglarının toplandığı bir bulut tabanlı güvenlik hizmetindeki kayıtlardan faydalanılan çalışmada, zararlı yazılım tespiti için uzay-zamansal özelliklerin dikkate alındığı, grafik tabanlı yarı denetimli öğrenen bir algoritma tasarlanmıştır. Sıfırıncı gün zararlı yazılımlarının yüksek

doğrulukla tespiti ve sınıflandırılmasını amaçlayan çalışmada [23], statik ve dinamik tekniklerin birleştirilerek etkin şekilde kullanıldığı bir büyük veri çerçevesi önerilmiştir, en iyi sınıflandırma destek vektör makineleri ile sağlanmıştır. Libri ve diğerleri [24] tarafından veri merkezlerinin ve süper bilgisayarların güvenliği için, bant dışı IoT tabanlı izleme sistemleri üzerinde çalışan, gerçek zamanlı zararlı yazılım tespiti yapabilen, hafif ve ölçeklenebilir bir yaklaşım olan pAElla sunulmuş ve anomali tespitine uygun bir sinir ağı olan otokodlayıcı ile en iyi tespit performansı elde edilmiştir.

Makine öğrenmesi, özellik çıkarımı, emulasyon gibi tekniklerin kullanıldığı çalışmada [25], Android cihazlarda zararlı yazılımların tespiti için dinamik bir zararlı yazılım analizi çerçevesi olan DroidDolphin sunulmuştur. Çalışan Android uygulamalarının tanımlanması ve zararlı yazılım tespitine yönelik yapılan çalışmada [26], her iki kapsamda ayrı ayrı özellik çıkarımı yapılmış ve düşük boyutlu özelliklerin kullanıldığı durumlarda dahi DT metoduyla yüksek duyarlılıkla tespit gerçekleştirilmiştir. Memon ve diğerleri [27] tarafından, 17 düğümlü bir Apache Spark kümesi kullanılarak farklı makine öğrenmesi yöntemlerinin etkinliğinin karşılaştırıldığı çalışmada, zararlı yazılım tespiti kapsamında en iyi performansının GBT algoritmasıyla, en iyi ikinci ve üçüncü performansların ise sırasıyla RF ve DT algoritmaları ile sağlandığı görülmüştür.

IoT ortamında sıfırıncı gün zararlı yazılım tespiti için görselleştirme tekniklerinin kullanıldığı hibrit bir zararlı yazılım tespiti yaklaşımının önerildiği çalışmada [28], evrişimli sinir ağları ve t-SNE ile görselleştirilen zararlı yazılım dosyaları destek vektör makineleri kullanılarak sınıflandırılmıştır. IoT ağlarında, IoT'nin doğasına uygun dağıtık ve merkezi olmayan uç hesaplama yaklaşımı ile zararlı yazılım tespiti ve sınıflandırmasının amaçlandığı çalışmada [29], işlem kodları vektörlere dönüştürülmüş, tespit ve sınıflandırma için bulanık ve hızlı bulanık desen ağaçları yöntemleri kullanılmıştır.

Büyük veri kapsamında zararlı yazılım tespitine yönelik yapılan çalışmalar incelendiğinde, bu çalışmaların temelini büyük verinin ihtiyaç duyduğu kaynakların azaltılmasına ya da büyük verinin işlenebilmesine yönelik yaklaşımların sunulduğu görülmektedir. Her ne kadar çalışmaların büyük bir kısmında sunulan yöntemlerin büyük veriye uygun olduğu belirtilse de sunulan yöntemlerin çok azı büyük veri ortamlarında uygulanmıştır. Yapılan çalışmayı literatürdeki çalışmalardan farklı kılan hususlar şu şekildedir:

- Aynı veri seti ve algoritmalar kullanılarak farklı büyük veri ortamlarının tahmin ve çalışma süresi performanslarının karşılaştırılması
- Aynı veri seti ve algoritmalar için büyük veri teknolojilerine yönelik geliştirilen PySpark ve makine öğrenmesi kapsamında yaygın olarak kullanılan Sci-kit Learn kütüphaneleri ile elde edilen tahmin ve çalışma süresi performanslarının karşılaştırılması

3. Makina Öğrenmesi Algoritmalarının Performanslarının Ölçülmesi

Çalışmanın bu bölümünde, büyük veri platformlarında zararlı yazılım tespiti için faydalanılan makine öğrenmesi algoritmalarının incelenmesi için kullanılan veri seti, bu veri setinin büyük veri ortamında makine öğrenmesi algoritmaları tarafından işlenebilir hale getirilmesi maksadıyla izlenen adımlar ve değerlendirme metrikleri anlatıldı.

3.1. Veri Seti

Çalışmada, Kaggle Zararlı Yazılım Tespiti veri seti [30] kullanıldı. Hem faydalı hem de zararlı yazılımlar için veriler barındıran veri seti 2018 yılında hazırlanmıştır. Veri seti hazırlanırken statik analiz yaklaşımı kullanılmış olup, PE özelliklerinin çıkarılması ve dosyaların farklı bölümlerindeki entropinin hesaplanması ile veri seti oluşturulmuştur. Kullanılan veri setinde 75503 faydalı, 140849

zararlı örnek bulunmaktadır. Veri setinde 53 adet bağımsız özellik ve 1 adet bağımlı özellik bulunmaktadır.

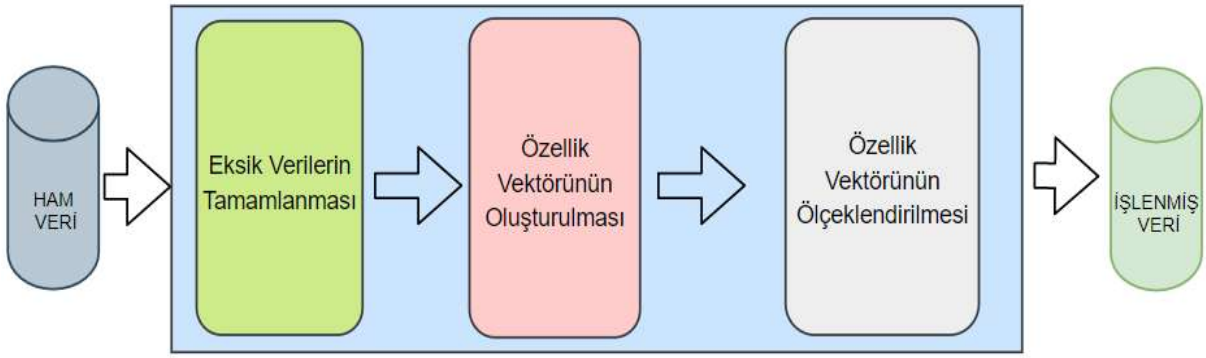
Kullanılan veri seti, Kaggle tarafından eğitim ve test veri setlerine ayrılmış olup, test veri setlerine yönelik tahminlerin kontrolü, tahminlerin çevrimiçi olarak yüklenmesiyle mümkündür. Bu sebeple, tespit yaklaşımının oluşturulması sürecinde hazırlanan eğitim seti ikiye bölünerek hem eğitim hem de test seti olarak kullanıldı. Çalışmada, 151439 örneklem eğitim verisi, 64913 örneklem ise test verisi olarak kullanıldı.

3.2. Performans Ölçüm Modeli

Büyük veri ortamında zararlı yazılım tespiti kapsamında makine öğrenmesi algoritmalarının performansının incelenmesi için, veri önişleme, sınıflandırıcıların eğitilmesi ve performans verilerinin ortaya konması adımlarından oluşan bir model kullanıldı.

3.2.1. Veri Önişleme

Veri önişleme aşamasında, veri setinin makine öğrenmesi algoritmaları tarafından işlenebilir hale getirilmesi için eksik ve/veya hatalı verilerin düzeltilmesini içeren "eksik verilerin tamamlanması", verinin Pyspark kütüphanesinde bulunan makine öğrenme algoritmaları tarafından işlenebilmesi için "özellik vektörünün oluşturulması" ve özellik değerlerinin varyanslarının azaltılarak model performansının iyileştirilmesi için "özellik vektörünün ölçeklendirilmesi" adımlarından oluşmaktadır (Şekil 1). Veri önişleme sonucunda, makine öğrenmesi algoritmalarının eğitilebileceği ve yine bu algoritmalar tarafından üzerinde tahmin gerçekleştirilebilecek işlenmiş veriler oluşmaktadır.



Şekil 1. Veri Önişleme Adımları

Eksik Verilerin Tamamlanması

Bir veri setinde eksik bir veri bulunması durumunda üç farklı yöntem izlenebilir. Bunlardan birincisi eksik verilerin olduğu gibi bırakılması, ikincisi ilgili örneklemin veri setinden çıkarılması, üçüncüsü ise eksik veri yerine yeni bir veri atanmasıdır [31]. Çalışmada, eksik veri yerine yeni veri atama yaklaşımı izlendi ve ilgili sütunun ortalaması alınarak eksik veriler tamamlandı.

Özellik Vektörünün Oluşturulması

Büyük verilerin işlenebilmesi için, ham verinin işlenebilir hale getirilmesi gerekmektedir. Apache Spark veri çerçevesi (dataframe) ve dayanıklı dağıtık veri seti (resilient distributed dataset - rdd) formatlarını desteklemektedir. Her ne kadar rdd formatı daha uzun süredir kullanılıyor olsa da

Spark 3.0 ile veri çerçevesi formatı Spark tarafından desteklenen temel format olarak kabul edilmiştir. Bu sebeple, çalışmada kullanılan veri seti veri çerçevesi olarak işlendi. Pyspark'ın ilgili metotlarından yararlanılarak verinin işlenebilmesi için özellik vektörü oluşturuldu.

Özellik Vektörünün Ölçeklendirilmesi

Özellik ölçeklendirme, verilerdeki öznitelikleri veya özellik aralıklarını birleştirme yöntemidir. Orijinal verilerde değişkenlerin aralığının çok farklı olması sorununu ortadan kaldırmak için veri önişleme adımında özellik ölçeklendirme yapılır [32]. Özellik vektörünün ölçeklendirilmesi için Pyspark kütüphanesinde StandardScaler metodundan faydalanıldı ve özellikler 0 ile 1 arasında ölçeklendirildi.

3.2.2. Sınıflandırıcıların Eğitilmesi

Büyük veri işleme teknolojilerinin en yaygın olarak kullanılanlarından biri olan Apache Spark'ın en güncel sürümünde (3.1.1) lojistik regresyon, DT, RF, GBT çok katmanlı algılayıcı, lineer destek vektör makinesi, naive bayes ve faktorizasyon makineleri sınıflandırıcıları bulunmakta olup bunlardan DT, RF ve GBT tarafından denetimli öğrenme gerçekleştirilebilmektedir [33]. Bu sebeple çalışmada DT, RF ve GBT algoritmaları sınıflandırıcı olarak kullanıldı.

Veri önişleme aşamasında Pyspark kütüphanesinde bulunan sınıflandırıcılar tarafından işlenebilir hale getirilen eğitim verisi kullanılarak ilgili sınıflandırıcılar eğitildi.

3.3. Performans Metrikleri

İkili sınıflandırma problemlerinde en iyi modelin tespitinde temel karmaşıklık matrisi temel alınabilir. Karmaşıklık matrisinde doğru tahminleri belirten doğru pozitif (dp), doğru negatif (dn) değerleri ile hatalı sınıflandırılan negatif örnekler için yanlış pozitif (yp) ve hatalı sınıflandırılan pozitif örnekler için yanlış negatif (yn) değerleri bulunmaktadır [34]. Bu değerler kullanılarak farklı metrikler elde edilebilir. Çalışmada kullanılan modellerin değerlendirilmesi için Tablo 1'de yer alan metriklerden faydalanıldı.

Tablo 1. Çalışmada kullanılan metrikler [34]

Metrik	Formül	Açıklama
Doğruluk	(dp + dn) / (dp + dn + yp + yn)	Yapılan doğru tahmin sayısının toplam tahmin sayısına oranı
Kesinlik (p)	dp / (dp+yp)	Doğru yapılan pozitif tahminlerin tüm pozitif tahminlere oranı
Duyarlılık (r)	dp /(dp + yn)	Pozitif örneklerin doğru tahmin edilme oranı
F1 Skoru	2 * p * r / (p + r)	Kesinlik ve duyarlılık metriklerinin harmonik ortalaması
Yanlış Pozitif Oranı	yp / (yp + dn)	Yanlış pozitif tahmin (hatalı alarm) oranı

Doğruluk, yapılan doğru tahminlerin yapılan tüm tahminlere oranını ifade etmekte olup, ikili ve çoklu sınıflandırmalarda en sık kullanılan metrik olmakta birlikte açıklayıcı olma açısından kısıtlıdır. Kesinlik, doğru yapılan pozitif tahminlerin yapılan tüm pozitif tahminlere oranını ifade etmektedir. Duyarlılık, pozitif örneklerin ne kadarının doğru şekilde sınıflandırıldığını ölçmek için kullanılır [34]. İkili sınıflandırma en iyi modelin tespiti için iyi bir metrik olarak görülen f1 skoru, duyarlılık ve kesinliğin harmonik ortalaması alınarak elde edilir [35]. Yanlış tahmin edilen pozitif

örneklerin yapılan toplam negatif tahmin sayısına oranını belirten, hatalı alarm olarak da bilinen yanlış pozitif oranı, model seçimi için önemli bir metrik olup, bu oranın düşürülmesine yönelik çeşitli çalışmalar mevcuttur [36-38].

4. Deneysel Sonuçlar

Çalışmanın bu aşamasında, DT, RF ve GBT algoritmalarının performans verileri Google Colaboratory'de, Azure HDInsight'ta, Amazon EMR'de ve Google Dataproc'ta Pyspark [39] kütüphanesi yardımıyla elde edildi. Ayrıca, algoritmaların performans verileri Sci-Kit Learn [40] kütüphanesi kullanılarak ölçüldü.

4.1. Google Colaboratory Ortamında PySpark Kütüphanesinden Faydalanılarak Algoritma Performanslarının Ölçülmesi

Google Colaboratory [41], Google'ın makine öğrenmesi ve büyük veri çalışmaları için ücretsiz olarak sunduğu bir servistir. Servis, kullanıcılara farklı zamanlarda farklı boyutlarda bellek tahsis etmekte olup, çalışmanın yapıldığı farklı zamanlarda 14-16GB bellek tahsisi yapıldı. Çalışma boyunca bellek kullanım miktarı %50'nin üzerine çıkmadı. Modelin çalıştırılması sonucu elde edilen tahmin performansı verileri Tablo 2'de ve çalışma performansı verileri Tablo 3'te sunuldu.

Tablo 2. Google Colaboratory ortamında tahmin performansı verileri

Metrik	DT	RF	GBT
Doğruluk	0.9661	0.9585	0.9726
Kesinlik	0.9720	0.9663	0.9760
Duyarlılık	0.9757	0.9697	0.9820
F1 Skoru	0.9739	0.9680	0.9790
Yanlış Pozitif Oranı	0.0448	0.0559	0.0447

Tablo 3. Google Colaboratory ortamında çalışma performansı verileri

Metrik	DT	RF	GBT
Eğitim Süresi (ES)	23.521811	24.621717	32.305181
Birim Başına ES	0.000155	0.000162	0.000213
Tahmin Süresi (TS)	0.245908	0.131061	0.097544
Birim Başına TS	3.788 e-06	2.019 e-06	1.502 e-06

Çizelgelerde gösterilen verilere ek olarak, her üç modelin aynı anda çalıştırıldığı durumda, toplam eğitim süresi 80.44 s, toplam tahmin süresi 0.47s olarak hesaplandı.

4.2. Azure HDInsight Ortamında PySpark Kütüphanesinden Faydalanılarak Algoritma Performanslarının Ölçülmesi

Bu bölümde, Azure HDInsight [42] üzerinde Spark kümesi oluşturularak ilgili makine öğrenmesi algoritmalarının performansları incelendi.

Tablo 4. Azure HDInsight ortamında tahmin performansı verileri

Metrik	DT	RF	GBT
Doğruluk	0.9656	0.9639	0.9705
Kesinlik	0.9751	0.9728	0.9757
Duyarlılık	0.9721	0.9725	0.9793
F1 Skoru	0.9736	0.9724	0.9775
Yanlış Pozitif Oranı	0.0466	0.0517	0.0459

Spark kümesi oluşturulurken, 2 adet D12v2 (4 çekirdek, 28 GB RAM) düğümü ana düğüm, 3 adet A2v2 (2 çekirdek, 4 GB RAM) düğüm Zookeeper ve 2 adet D12v2 (4 çekirdek, 28 GB RAM) düğüm işçi düğüm olarak yapılandırıldı. Modelin çalıştırılması sonucu elde edilen tahmin performansı verileri Tablo 4'te ve çalışma performansı verileri Tablo 5'de sunuldu.

Tablo 5. Azure HDInsight ortamında çalışma performansı verileri

Metrik	DT	RF	GBT
Eğitim Süresi (ES)	12.165558	9.783118	15.237745
Birim Başına ES	0.000080	0.000064	0.000100
Tahmin Süresi (TS)	0.099560	0.065763	0.078780
Birim Başına TS	1.541e-06	1.017e-06	1.219e-06

Çizelgelerde gösterilen verilere ek olarak, her üç modelin aynı anda çalıştırıldığı durumda, toplam eğitim süresi 39.18 s, toplam tahmin süresi 0.24s olarak hesaplandı.

4.3. Amazon EMR Ortamında PySpark Kütüphanesinden Faydalanılarak Algoritma Performanslarının Ölçülmesi

Bu bölümde, Amazon EMR [43] ortamında Spark kümesi oluşturularak ilgili makine öğrenmesi algoritmalarının performansları incelendi. Spark kümesi oluşturulurken, 1 adet m5.xlarge (4 çekirdek, 16 GB RAM) tipi sanal makine ana düğüm, 2 adet m5.xlarge tipi sanal makine ise işçi düğüm olarak yapılandırıldı. Modelin çalıştırılması sonucu elde edilen tahmin performansı verileri Tablo 6'da ve çalışma performansı verileri Tablo 7'de sunuldu.

Tablo 6. Amazon EMR ortamında tahmin performansı verileri

Metrik	DT	RF	GBT
Doğruluk	0.9661	0.9628	0.9715
Kesinlik	0.9758	0.9735	0.9770
Duyarlılık	0.9720	0.9692	0.9791
F1 Skoru	0.9739	0.9713	0.9781
Yanlış Pozitif Oranı	0.0448	0.0490	0.0427

Tablo 7. Amazon EMR ortamında çalışma performansı verileri

Metrik	DT	RF	GBT
Eğitim Süresi (ES)	12.490172	10.922188	14.353764
Birim Başına ES	0.000082	0.000072	0.000095
Tahmin Süresi (TS)	0.070333	0.069403	0.067036
Birim Başına TS	1.083507e-06	1.069175e-06	1.032711e-06

Çizelgelerde gösterilen verilere ek olarak, her üç modelin aynı anda çalıştırıldığı durumda, toplam eğitim süresi 37.76 s, toplam tahmin süresi 0.21s olarak hesaplandı.

4.4. Google Dataproc Ortamında PySpark Kütüphanesinden Faydalanılarak Algoritma Performanslarının Ölçülmesi

Bu bölümde, Google Dataproc [44] ortamında Spark kümesi oluşturularak ilgili makine öğrenmesi algoritmalarının performansları incelendi. Spark kümesi oluşturulurken, 1 adet n1-standard-4 (4 çekirdek, 15 GB RAM) tipi sanal makine ana düğüm, 2 adet n1-standard-4 tipi sanal makine ise işçi düğüm olarak yapılandırıldı. Modelin çalıştırılması sonucu elde edilen tahmin performansı verileri Tablo 8'de ve çalışma performansı verileri Tablo 9'da sunuldu.

Tablo 8. Google Dataproc ortamında tahmin performansı verileri

Metrik	DT	RF	GBT
Doğruluk	0.9663	0.9607	0.9721
Kesinlik	0.9759	0.9696	0.9769
Duyarlılık	0.9724	0.9700	0.9803
F1 Skoru	0.9741	0.9698	0.9786
Yanlış Pozitif Oranı	0.0514	0.0558	0.0430

Tablo 9. Google Dataproc ortamında çalışma performansı verileri

Metrik	DT	RF	GBT
Eğitim Süresi (ES)	16.287233	14.394784	19.637797
Birim Başına ES	0.000107	9.505e-05	0.000129
Tahmin Süresi (TS)	0.191430	0.118620	0.109122
Birim Başına TS	2.949025e-06	1.827375e-09	1.681047e-06

Çizelgelerde gösterilen verilere ek olarak, her üç modelin aynı anda çalıştırıldığı durumda, toplam eğitim süresi 50.32 s, toplam tahmin süresi 0.42s olarak hesaplandı.

4.5. Google Colaboratory Ortamında Sci-Kit Learn Kütüphanesinden Faydalanılarak Algoritma Performanslarının Ölçülmesi

Büyük veri işlemede yaygın olarak kullanılan PySpark kütüphanesinin etkinliğinin değerlendirilmesi için makine öğrenmesinde yaygın olarak kullanılan Sci-Kit (sklearn) kütüphanesi kullanılarak, aynı hiper değişkenlerle, DT, RF ve GBT algoritmalarının performansı ölçüldü. Modelin çalıştırılması sonucu elde edilen tahmin performansı verileri Tablo 10'da ve çalışma performansı verileri Tablo 11'de sunuldu.

Tablo 10. Google Colaboratory ortamında tahmin performansı verileri

Metrik	DT	RF	GBT
Doğruluk	0.9827	0.9878	0.9713
Kesinlik	0.9747	0.9856	0.9625
Duyarlılık	0.9758	0.9794	0.9549
F1 Skoru	0.9752	0.9824	0.9587
Yanlış Pozitif Oranı	0.0241	0.0205	0.0450

Tablo 11. Google Colaboratory ortamında çalışma performansı verileri

Metrik	DT	RF	GBT
Eğitim Süresi (ES)	4.233333	4.425238	43.032803
Birim Başına ES	0.000028	0.000029	0.000284
Tahmin Süresi (TS)	0.020946	0.107535	0.131214
Birim Başına TS	3.227 e-07	1.656 e-06	2.021 e-06

Çizelgelerde gösterilen verilere ek olarak, her üç modelin aynı anda çalıştırıldığı durumda, toplam eğitim süresi 51.69 s, toplam tahmin süresi 0.26s olarak hesaplandı.

5. Değerlendirme

Bu bölümde, önceki bölümlerde elde edilen veriler kullanılarak hem makine öğrenme algoritmalarının hem de çalışma ortamlarının değerlendirmesi yapıldı.

5.1. Tahmin Performanslarının Karşılaştırılması

Çalışmada kullanılan makine öğrenme algoritmalarına ait farklı ortamlardaki tahmin performansı verilerinin birleştirilmiş hali Tablo 12'de sunuldu. Veriler incelendiğinde:

- Aynı küyüphane kullanılarak çalıştırılan algoritmaların farklı büyük veri ortamlarında benzer tahmin performansları gösterdiği,
- Özellikle DT ve RF algoritmalarında Sci-Kit Learn kütüphanesinin kullanıldığı durumlarda diğer ortamlarda PySpark kütüphanesi kullanılarak elde edilen sonuçlardan daha iyi sonuçlar elde edildiği,
- En iyi doğruluk (0.987), kesinlik (0.985), f1 skorunu (0.982) ve yanlış pozitif oranının (0.02) RF algoritması tarafından Google Colaboratory ortamında Sci-Kit kütüphanesi kullanılarak elde edildiği,
- En iyi duyarlılığın (0.981) ise GBT algoritması tarafından Google Colaboratory ortamında PySpark kütüphanesi kullanılarak elde edildiği,
- Bulut servis sağlayıcılar tarafından büyük veri hizmeti olarak sunulan Azure HDInsight, Amazon EMR ve Google Dataproc ortamlarının her üç algoritma ve beş metrik için de birbirleriyle tutarlı sonuçlar verdiği görüldü.

Tablo 12. Tahmin performansı verileri

Algoritma	Metrik	GC	AH	AE	GD	GC-S
DT	Doğruluk	0.966	0.965	0.966	0.966	**0.983**
	Kesinlik	0.972	**0.975**	**0.975**	**0.975**	**0.975**
	Duyarlılık	0.975	0.972	0.972	0.972	**0.976**
	F1-Skoru	0.973	0.973	0.973	0.974	**0.975**
	Yanlış Pozitif Oranı	0.045	0.046	0.044	0.051	**0.024**
RF	Doğruluk	0.958	0.963	0.962	0.961	**0.988**
	Kesinlik	0.966	0.972	0.973	0.970	**0.985**
	Duyarlılık	0.969	0.972	0.969	0.970	**0.979**
	F1-Skoru	0.968	0.972	0.971	0.970	**0.982**
	Yanlış Pozitif Oranı	0.055	0.051	0.049	0.055	**0.020**
GBT	Doğruluk	**0.972**	0.970	0.971	**0.972**	0.971
	Kesinlik	0.976	0.975	**0.977**	**0.977**	0.962
	Duyarlılık	**0.982**	0.979	0.979	0.980	0.955
	F1-Skoru	**0.979**	0.977	0.978	0.978	0.959
	Yanlış Pozitif Oranı	0.044	0.045	**0.042**	0.043	0.045

Google Colaboratory - GC, Azure HDInsight - AH, Amazpn EMR - AE, Google DataProc - GD, Google Colaboratory (Sci-Kit) - GC-S

5.2. Çalışma Süresi Performanslarının Karşılaştırılması

Çalışmada kullanılan makine öğrenme algoritmalarına ait farklı ortamlardaki çalışma performansı verilerinin birleştirilmiş hali Tablo 13'te ve ortamlar için algoritmaların birleştirilmiş çalışma süresi performansı Tablo 14'te sunulmuştur. Çalışma performansı verileri incelendiğinde:

- Farklı algoritmalar için en iyi ortalama eğitim ve tahmin zamanı sunan algoritmaların ve ortamların değişiklik gösterdiği,
- En iyi eğitim sürelerinin DT (4.2333s) ve RF (4.4252s) için Google Colaboratory ortamında Sci-Kit Learn kütüphanesi, GBT (15.2377s) için Amazon EMR ortamında PySpark kütüphanesi kullanılarak elde edildiği,

- En iyi birim başına eğitim süresinin Google Colaboratory ortamında Sci-Kit kütüphanesi kullanılarak DT (3.22e-07) algoritması ile elde edildiği, RF (1.01-e06) için Azure HDInsight ve GBT (1.21e-06) için Amazon EMR ortamlarında PySpark kütüphanesi kullanılarak en iyi sonuçlara ulaşıldığı görüldü.

Tablo 13. Çalışma süresi performansı

Algoritma	Metrik	GC	AH	AE	GD	GC-S
DT	ES	23.5218	12.1655	12.4901	16.2872	**4.2333**
	BBES	0.000155	0.000080	0.000082	0.000107	**0.000028**
	TS	0.245908	0.099560	0.070333	0.191430	**0.020946**
	BBTS	3.78e-06	1.54e-06	1.08e-06	2.94e-06	**3.22e-07**
RF	ES	24.6217	9.7831	10.9221	14.3947	**4.4252**
	BBES	0.000162	0.000064	0.000072	0.000095	**0.000029**
	TS	0.131061	**0.065763**	0.069403	0.118620	0.107535
	BBTS	2.01e-06	**1.01e-06**	1.06e-06	1.82e-06	1.65e-06
GBT	ES	32.3051	15.2377	**14.3537**	19.6377	43.0328
	BBES	0.000213	0.000100	**0.000095**	0.000129	0.000284
	TS	0.097544	0.078780	**0.067036**	0.109122	0.131214
	BBTS	1.50e-06	1.21e-06	**1.03e-06**	1.68e-06	2.02e-06

Eğitim Süresi – ES, Birim Başına Eğitim Süresi – BBES, Tahmin Süresi – TS, Birim Başına Tahmin Süresi -BBTS

Birleştirilmiş çalışma süresi performans verileri incelendiğinde:

- En iyi eğitim süresi performansının Amazon EMR ortamında sağlandığı, Amazon EMR kümesi ile benzer kaynaklara sahip ortamlardan Azure HDInsight ortamında Amazon EMR kümesinin performansına yakın performans elde edildiği ancak Google Dataproc ortamında eğitim süresinin yaklaşık %33 daha fazla olduğu,
- Google Colaboratory ortamında Sci-Kit kütüphanesi PySpark kütüphanesiyle karşılaştırıldığında eğitim süresi performansının ortalama %36, tahmin süresi performansının ise ortalama %44 daha iyi olduğu,
- PySpark kütüphanesi kullanılarak elde edilen en iyi (Amazon EMR) ve en kötü (Google Colaboratory) performanslar incelendiğinde eğitim süresinde %53, tahmin süresinde %55 iyileşme olduğu görüldü.

Tablo 14. Birleştirilmiş çalışma süresi performansı

Metrik	GC	AH	AE	GD	GC-S
Toplam Eğitim Süresi	80.44	39.18	**37.76**	50.32	51.69
Toplam Tahmin Süresi	0.47	0.24	**0.21**	0.42	0.26

5. Sonuç

Çalışmada, büyük veri ortamında zararlı yazılımların tespit edilmesi kapsamında makine öğrenmesi algoritmalarının etkinliği incelendi. Google Colaboraty, Azure HDInsight, Amazon EMR ve Google Dataproc ortamlarında yapılan çalışmada, Apache Spark 3.0'da bulunan ve ikili sınıflandırma yapabilen RF, DT ve GBT makine öğrenme metotları kullanılarak Kaggle Zararlı Yazılım Tespiti Veri Seti üzerinde modellerin etkinliği test edildi. Farklı kütüphaneler kullanıldığı durumlarda algoritmalarının tahmin performanslarının değişkenlik gösterebileceği görüldü. PySpark kütüphanesi kullanılarak daha fazla kaynağa sahip bir Spark kümesinde çalıştırılan modellerin eğitim ve tahmin sürelerinde yarı yarıya azaltma sağlanabildiği ortaya konuldu.

Aynı veri seti ile farklı makine öğrenmesi algoritmaları kullanılarak gerçekleştirilen çalışmalarda daha iyi tahmin performanslarına erişilebildiği görülmüş olup, mevcut algoritmalar dışında ikili sınıflandırma yeteneğine sahip makine öğrenmesi algoritmalarının (Extra Trees, CatBoosting vb.) büyük veri ortamlarında kullanılabilir hale getirilmesine ihtiyaç duyulduğu değerlendirilmektedir.

Yazarların Katkıları

SG çalışmanın temel halini oluşturdu ve deneysel çalışmaları gerçekleştirdi. MD çalışmayı kontrol ederek gerekli düzeltme ve düzenlemelerin yapılmasını sağladı.

Her iki yazar da makalenin son halini okudu ve onayladı.

Çıkar Çatışması

Yazarlar, çıkar çatışması olmadığını beyan eder.

Kaynaklar

[1]. Beraa, D. and Mallikb, S., “Importance of information product for economic development”, Library Philosophy and Practice, 2021:1–11.
[2]. Luftman, J., Lyytinen K. and Zvi, T.B., “Enhancing the measurement of information technology (IT) business alignment and its influence on company performance”, Journal of Information Technology, 2017, 32(1):26-46.
[3]. Ellitan, L., “The importance of entrepreneurship and information technology for SMEs strategic planning”, International Journal of Trend in Scientific Research and Developmen, 2021, 5 (4):1003-1009.
[4]. Khan, S. A. R., Golpîra, H. and YU, Z., “The importance of advanced information technology and green vehicles in supply chain management”, International Conference on Computer, Communications and Mechatronics Engineering, 2018, 351–355.
[5]. FireEye M-Trends 2021, https://content.fireeye.com/m-trends/rpt-m-trends-2021, Son erişim: 26 Ağustos 2021.
[6]. Idika, N. and Mathur, A.P., “A survey on malware detection techniques”, Purdue University, 2007, 48.
[7]. Gandotra, E., Bansal, D. and Sofat, S., “Malware analysis and classification: A survey”, Journal of Information Security, 2014, 05(02):56–64.
[8]. S. Sagiroglu and D. Sinanc, "Big data: A review", International Conference on Collaboration Technologies and Systems (CTS), 2013, 42-47.
[9]. Alguliyev, R. and Imamverdiyev, Y., “Big data: Big promises for information security”, 8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014, 13–16.
[10]. Abawajy, J. H., and Kelarev, A., “Large iterative multitier ensemble classifiers for security of big data”, IEEE Trans. Emerg. Top. Comput., 2014, 2(3):352–363.
[11]. Bocchi, E., Grimaudo, L., Mellia, M., Baralis, E., Saha, S., Miskovic, S., Modelo-Howard, G. and Lee, S.J., “MAGMA network behavior classifier for malware traffic” Comput. Networks2016, 109:142–156.
[12]. Gupta, D., and Rani, R., “Big data framework for zero-day malware detection” Cybern. Syst.,2018, 49(2):103–121.
[13]. Gupta, D., and Rani, R., “Improving malware detection using big data and ensemble learning,” Comput. Electr. Eng., 2020, 86.
[14]. Abawajy, J. H., Chowdhury, M., and Kelarev, A., “Hybrid consensus pruning of ensemble classifiers for big data malware detection”, IEEE Trans. Cloud Comput., 2020, 8(2):398–407.

[15]. Usman, N., Usman, S., Khan, F., Jan, M.A., Saj,d, A., Alazab, M. and Watters, P., "Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics", Futur. Gener. Comput. Syst., 2021, 118:124–141.
[16]. Sahoo, A. K., Sahoo, K. S., and Tiwary, M., "Signature based malware detection for unstructured data in Hadoop", 2014 Int. Conf. Adv. Electron. Comput. Commun. (ICAECC) 2014.
[17]. Suhasini, N. S., Hirwarkar, T., and Ashok, J., "Big data analytics for malware detection in a virtulaized framework", 2020, 7(14):3184-3191.
[18]. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., and Venkatraman, S., "Robust intelligent malware detection using deep learning", IEEE Access, 2019, 7:46717-46738.
[19]. De Paola, A., Gaglio, S., Lo Re, G., and Morana, M., "A hybrid system for malware detection on big data", INFOCOM 2018 - IEEE Conf. Comput. Commun. Work., 2018, 45–50.
[20]. Masabo, E., Kaawaase, K. S., and Sansa-Otim, J., "Big data: Deep learning for detecting malware", Proc. - Int. Conf. Softw. Eng., 2018, 20-26.
[21]. Yousefi-azar, M., Hamey, L. G. C., Varadharajan, V., and Chen, S., "Malytics : A malware detection scheme", IEEE Access, 2018, 6:49418-49431.
[22]. Mao, W., Cai, Z., Yang, Y., and Shi, X., "From big data to knowledge : A spatio- temporal approach to malware detection", Comput. Secur., 2018, 74:167-183.
[23]. Niveditha, V. R., Ananthan, T. V. , Amudha, S., Sam, D., and Srinidhi, S., "Detect and classify zero day malware efficiently in big data platform", Int. J. Adv. Sci. Technol., 2020, 29(4):1947-1954.
[24]. Libri, A., Bartolini, A., and Benini, L., "pAElla: Edge AI-based real-time malware detection in data centers", IEEE Internet Things J., 2020, 7(10):9589-9599.
[25]. Wu, W. C., and Hung, S. H., "DroidDolphin: A dynamic android malware detection framework using big data and machine learning", Proc. 2014 Res. Adapt. Converg. Syst. (RACS 2014), 2014, 247-252.
[26]. Wassermann, S., and Casas, P., "BIGMOMAL - Big data analytics for mobile malware detection", Proc. 2018 Work. Traffic Meas. Cybersecurity, Part SIGCOMM 2018, 2018, 33–39, 2018
[27]. Memon, L. U., Bawany, N. Z., and Shamsi, J. A., "A comparison of machine learning techniques for android malware detection using apache spark", J. Eng. Sci. Technol., 2019, 14(3):1572-1586.
[28]. Venkatraman, S., and Alazab, M., "Use of Data Visualisation for Zero-Day Malware Detection", Secur. Commun. Networks, 2018.
[29]. Modiri, E., Azmoodeh, A., Dehghantanha, A., Ellis, D., Parizi, R. M., and Karimipour, H., "Fuzzy pattern tree for edge malware detection and categorization in IoT", J. Syst. Archit. Comput., 2018, 97:1-7.
[30]. Kaggle Zararlı Yazılım Tespiti veri seti, https://www.kaggle.com/c/malware-detection, son erişim: 26 Ağustos 2021
[31]. Sessa, J. and Syed, D., "Techniques to deal with missing data", 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), 2016, 1-4.
[32]. Wan, X., "Influence of feature scaling on convergence of gradient iterative algorithm", J. Phys.: Conf. Ser., 2019, 1213(3).
[33]. Apache Spark, İkili Sınıflandırıcılar ve Regresyon Analizi, https://spark.apache.org/docs/latest/ml-classification-regression.html, Son erişim: 27 Ağustos 2021.
[34]. Hossin, M., and Sulaiman M.N., "A review on evaluation metrics for data classification evaluations", International Journal of Data Mining & Knowledge Management Process, 2015, 5(2):01–11.

[35]. Joshi, M.V., “On evaluating perfomrance of classifiers for rare classes”, in Porceedings of the 2002 IEE Int. Conference on Data Mining, 2002, 641-644.
[36]. Landress, A. D., "A hybrid approach to reducing the false positive rate in unsupervised machine learning intrusion detection", SoutheastCon 2016, 2016, pp. 1-6.
[37]. Kong, S., Shen, W., Zheng, Y., Zhang, A., Pu, J. and Wang, J., “False positive rate control for positive unlabled learning”, Neurocomputing, 2019, 367:13-19.
[38]. Jallad, K. A., Aljnidi, M., and Desouki, M. S., “Anomaly detection optimization using big data and deep learning to reduce false-positive”, Journal of Big Data, 2020, 7(68).
[39]. PySpark, https://spark.apache.org/docs/latest/api/python/, Son erişim: 27 Ağustos 2021.
[40]. Scikit-learn, https://scikit-learn.org/stable/, Son erişim: 27 Ağustos 2021.
[41]. Google Colaboratory, https://research.google.com/colaboratory/, Son erişim: 27 Ağustos 2021.
[42]. Azure HDInsight, https://azure.microsoft.com/en-us/services/hdinsight/, Son erişim: 27 Ağustos 2021.
[43]. Amazon EMR, https://aws.amazon.com/emr, Son erişim: 27 Ağustos 2021.
[44]. Google Dataproc, https://cloud.google.com/dataproc, Son erişim: 27 Ağustos 2021.