T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ZARARLI YAZILIM TESPİTİ

Şeyma GÜLEŞ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ağustos-2020
KONYA

TEZ KABUL VE ONAYI

Şeyma GÜLEŞ tarafından hazırlanan "MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ZARARLI YAZILIM TESPİTİ" adlı tez çalışması 27/08/2020 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Dr. Öğr. Üyesi Sait Ali UYMAZ	……..
Danışman Doç. Dr. Barış KOÇER	……..
Üye Doç. Dr. Mehmet HACIBEYOĞLU	……..
Üye Dr. Öğr. Üyesi Sedat KORKMAZ	……..

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Şeyma GÜLEŞ

Tarih:

ÖZET

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE ZARARLI YAZILIM TESPİTİ

Şeyma GÜLEŞ

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Barış KOÇER

2020, 62 Sayfa

Jüri
Doç. Dr. Barış KOÇER
Doç. Dr. Mehmet HACIBEYOĞLU
Dr. Öğr. Üyesi Sait Ali UYMAZ
Dr. Öğr. Üyesi Sedat KORKMAZ

Teknolojinin gelişmesi ile birlikte hayatımızla ayrılmaz bir parça haline gelen bilgisayar sistemlerinin kullanımı giderek artmaktadır. Bu artış birçok siber güvenlik probleminin ortaya çıkmasına sebep olmuştur. Siber güvenlik açıklarının sonucunda kullanıcıların bilgisayar sistemlerine giren kötü amaçlı yazılımlar birçok zarara sebep olmaktadır. Bu zararları engelleyebilmek amacıyla kötü amaçlı yazılım tespit sistemleri geliştirilmiştir. Kötü amaçlı yazılımların herhangi bir zarara sebep olmadan önce tespit edilebilmesi bilgi sistemleri için hayati önem taşımaktadır. Bu çalışmada makine öğrenmesi yöntemleri ile bilgisayar sistemlerinin telemetri bilgileri kullanılarak kötü amaçlı bir yazılımın sistem üzerinde var olup olmadığının tahmini yapılmıştır. Çalışmada kullanılan veri seti Microsoft tarafından bilgisayarların telemetri bilgileri toplanarak oluşturulmuştur.

Veri setinin ilk bir milyon veri satırı Bilgi Kazancı, Ki-Kare özellik seçme yöntemleri ile birlikte Naive Bayes, Karar Ağacı, Random Forest, Adaboost, LightGBM sınıflandırma algoritmaları ile 10 çapraz doğrulama tekniği kullanılarak test edilmiştir. Ayrıca Microsoft Kötü Amaçlı Yazılım Tahmini veri setini çalışmasında kullanan Lin'in (2019) veri ön işleme adımlarından sonra elde ettiği veri seti Naive Bayes, Karar Ağacı, Random Forest, Adaboost, LightGBM sınıflandırma algoritmaları ile test edilmiştir.

Anahtar Kelimeler: Bilgi Güvenliği, Kötü Amaçlı Yazılımlar, Makine Öğrenmesi Yöntemleri, Siber Güvenlik, Telemetri Bilgisi

ABSTRACT

MS THESIS

MALWARE DETECTION WITH MACHINE LEARNING METHODS

Şeyma GÜLEŞ

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Assoc. Prof. Dr. Barış KOÇER

2020, 62 Pages

Jury
Assoc. Prof. Dr. Barış KOÇER
Assoc. Prof. Dr. Mehmet HACIBEYOĞLU
Asst. Prof. Dr. Sait Ali UYMAZ
Asst. Prof. Dr. Sedat KORKMAZ

With the development of technology, the use of computer systems, which have become an integral part of our lives, is gradually increased. This increase has been caused many cyber security problems. Malicious software entering users' computer systems as a result of cyber security vulnerabilities are caused many damages. In order to prevent these damages, malware detection systems have been developed. It is vital for information systems that malware can to detected before it causes any damage. In this study, it has been estimated whether a malware exists on the system by using machine learning methods and telemetry information of the computer system. The data set used in the study was created by Microsoft by collecting telemetry information of computers.

The first one million data rows of the data set were tested using 10 cross validation techniques with Naive Bayes, Decision Tree, Random Forest, Adaboost, LightGBM classification algorithms along with Knowledge Gain, Chi-Square feature selection methods. In addition, the data set obtained after data preprocessing steps by Lin (2019), who used the Microsoft Malware Prediction dataset in his study, was tested with Naive Bayes, Decision Tree, Random Forest, Adaboost, LightGBM classification algorithms after data preprocessing steps.

Keywords: Information security, Malware, Machine Learning Methods, Cyber security, Telemetry Information

ÖNSÖZ

Yüksek Lisans eğitimim ve tez çalışmamın büyük bir kısmını kendisinin destekleri ile tamamladığım ancak bugün aramızda olmayan saygı değer hocam Doç. Dr. Barış KOÇER'e teşekkürlerimi bir borç bilirim.

Gösterdiği yollarla çalışmama farklı bir boyut kazandıran ve her türlü desteğini benden esirgemeyen kıymetli hocam Doç. Dr. Mehmet HACIBEYOĞLU'na minnettarlığımı ifade etmek isterim.

Değerli vakitlerini bana ayırarak çalışmama destek veren Öğr. Gör. Mustafa GÖKMEN, Doç. Dr. Mesut GÜNDÜZ, Dr. Öğr. Üyesi Alper KILIÇ, Dr. Öğr. Üyesi Nurdan BAYKAN, Dr. Öğr. Üyesi Ömer Kaan BAYKAN, Dr. Öğr. Üyesi Ersin KAYA' ya teşekkür ederim.

Ayrıca gösterdikleri sabırlardan ve teşviklerinden dolayı arkadaşlarım Arş. Gör. Ferda Nur ARICI, Fatma AKACAN, Fatma Seda ÖZYURT'a ve her zaman yanımda olan Aileme teşekkür ederim.

Şeyma GÜLEŞ
KONYA-2020

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Kısaltmalar

DVM:	Destek Vektör Makinesi
FEP:	Forefront Endpoint Protection
GBM:	Gradient Boosting Machine
LightGBM:	Light Gradient Boosted Machine
LR:	Logistik Regresyon
META:	Ortadoğu, Türkiye ve Afrika bölgesindeki ülkeler
MSE:	Microsoft Security Essentials
Mseprerelease:	Microsoft Security Essentials Pre-Release
MV:	Majority Vote
ROC:	Reciever Operator Characteristics Curve
SCEP:	System Center Endpoint Protection
XGBoost:	Extreme Gradient Boosting
YSA:	Yapay Sinir Ağları

1. GİRİŞ

Gelişen teknoloji insan yaşamı üzerindeki etkisini her geçen gün arttırmış ve yaşamımızın ayrılmaz bir parçası haline gelmiştir. Yaşamımızı kolaylaştıran teknolojik gelişmeler internet kullanım oranını da attırmıştır. WeAreSocial ve Hootsuite tarafından hazırlanan 2020 yılının ilk raporunda 4.54 milyar kişi yani dünya genelinde %59'u internet kullanmaktadır. Aynı rapora göre internet kullanıcıları ortalama olarak günde 7 saat internet kullanmaktadırlar (WeAreSocial ve Hootsuite, 2020).

İnternet kullanımının artması çeşitli bilgi güvenliği zafiyetlerini beraberinde getirmektedir. Cyber Security Weekend 2019 etkinliğinde sadece 2019'un ilk çeyreğinde Kaspersky Lab tarafından 150 milyondan fazla zararlı yazılım (malware) tespit edilmiştir. Zararlı yazılımların 2018'in ilk çeyreğine göre %8,2 oranında artmış olduğunu raporlanmıştır. Aynı araştırmaya göre META (Ortadoğu, Türkiye ve Afrika) bölgesindeki ülkeler bazındaki karşılaştırmada Türkiye, kimlik avı saldırısı (1.24 milyon), zararlı yazılım (39 milyon) ve mobil zararlı yazılım (87 bin) kategorilerinde en çok saldırıya uğrayan ülke olarak birinci sırada yer almıştır (KasperskyLab, 2019). CyberMag'in (2020) haberine göre IBM Güvenlik İş Birimi, 2020 yıllık araştırmasında veri ihlallerinin kuruluşlar üzerindeki maliyetini incelemiştir. Yapılan incelemelerde Türkiye'deki veri ihlallerinin ortalama maliyeti 12,3 milyon TL'ye ulaşmıştır (CyberMag, 2020).

Yapılan saldırılarda elde edilen bilgiler kurum, kuruluş ya da toplumların imajını zedelemek, bilgisayarlı sistemlerin zarar görmesini sağlamak, hizmet kesintilerine sebep olmak, maddi kayıplara neden olmak gibi amaçlarla kullanılmaktadır. Ayrıca saldırganlar bilgisayarları bir botnet üyesi haline getirip kendi amaçlarına hizmet edecek şekilde yapılandırabilmektedirler. Bilgi güvenliği zafiyetlerinden yararlanan saldırganlar çeşitli yollar deneyerek kullanıcılara manevi zararlar da verebilmektedir.

Bilişim sistemlerindeki bilgi güvenliğinin sağlanması amacıyla zararlı yazılım tespit sistemleri, saldırı tespit sistemleri, şifreleme yöntemleri gibi yazılımlar ayrıca donanımsal cihazlar geliştirilmiştir ve geliştirilmektedir. Zararlı yazılım tespiti için genellikle program koduna ya da programın sistem içerisindeki davranışı incelenmektedir. Ayrıca Microsoft firması Windows işletim sistemlerinde tanılama verileri (diagnostic data) yani Windows telemetri verileri ile kritik güvenlik ve güvenilirlik sorunlarına hızlı bir çözüm bulunması amaçlanmaktadır (Microsoft, 2020a).

Bu tezde Windows işletim sistemine sahip bir bilgisayarın telemetri bilgilerine dayanılarak makine öğrenmesi yöntemlerinden bazı sınıflandırma metotları kullanılarak

sistemde kötü amaçlı bir yazılımın var olup olmadığının tahmini yapılmıştır. Microsoft firması tarafından oluşturulan kötü amaçlı yazılım tahmini veri seti, üzerinde makine öğrenmesi algoritmalarından Naive Bayes, Karar Ağacı, RandomForest, Adaboost ve LightGBM sınıflandırma algoritmaları ile test edilmiştir. Yapılan test sonuçları karşılaştırılmıştır.

Tez çalışmasının ikinci bölümünde kaynak araştırmalarına yer verilmiştir. Kaynak araştırmasında Microsoft kötü amaçlı yazılım tahmini veri seti kullanılarak yapılan çalışmalar incelenmiştir. Üçüncü bölüm olan materyal ve yöntem bölümünde bilgi güvenliği kavramından, kötü amaçlı yazılımların nasıl tespit edilebildiğinden, telemetri kavramından, kullanılan veri setinden, sınıflandırma için kullanılan yöntemlerden ve özellik seçmek için kullanılan yöntemlerden bahsedilmiştir.

Dördüncü bölümde araştırma sonuçları verilmiştir. Son bölümde ise sonuçlar ve önerilere yer verilmiştir.

2. KAYNAK ARAŞTIRMASI

1971'de Bob Thomas tarafından yazılan program ilk bilinen kötü amaçlı yazılım olarak Creeper kabul edilmektedir. Creeper aslında bilgisayarlara zarar vermeyen sadece bilgisayar arasında kendini kopyalayarak ekrana mesaj çıktısı veren basit bir program olmakla birlikte genellikle kendini kopyalayan sürümü ilk bilgisayar virüsü olarak kabul edilmektedir (Wikipedia, 2020). Creeper'den günümüze kadar birçok kötü amaçlı yazılım ortaya çıkmış ve güvenlik açıkları meydana gelmiştir. Güvenlik açıklarından yararlanılarak siber saldırılar yapılmıştır. Bu sebeple bilgi güvenliğinin sağlanmasına yönelik geliştirilen yazılımsal ürünler ve donanımsal cihazlar önem kazanmıştır.

Microsoft firması kullanıcıların bilgi güvenliğinin sağlamak amacıyla 2018 yılında Windows makinelerinin telemetri verilerinden yararlanarak bir veri seti oluşturmuştur. Veri seti Kaggle platformunda "Microsoft Malware Prediction" isimli yarışmada kullanılmıştır (Microsoft, 2018). Amacı kötü amaçlı yazılımların tahmin edilebilmesi sağlamaktır. Microsoft Kötü Amaçlı Yazılım Tahmini veri seti kullanılarak farklı çalışmalar yapılmıştır.

Yeboah-Ofori ve Boachie'in (2019) yaptığı çalışmada Microsoft Kötü Amaçlı Yazılım Tahmini veri setinin 20000 verisini ve 54 özelliğini kullanarak Karar Ağacı, Logistik Regresyon (LR), Destek Vektör Makinesi (DVM) ve bu üç algoritmanın kombinasyonu şeklinde kullanılan Topluluk Oylaması (Majority Voiting) algoritması kullanarak sınıflandırma işlemi yapmışlardır. Belirlenen veri setinde 10 turlu çapraz doğrulama (K-Fold) uygulayarak test sonuçlarını kaydetmişlerdir. Elde edilen sonuçlarda Logistik Regresyon'dan 0.65, Karar Ağacı'ndan 0.59, DVM'den 0.66, Topluluk Oylaması algoritmasından ise 0.65 oranında başarı elde etmişlerdir (Yeboah-Ofori ve Boachie, 2019).

Lin'in (2019) yaptığı çalışmada Microsoft Kötü Amaçlı Yazılım Tahmini veri setinin 71110 verisini eğitim için 22638 verisini test için kullanmıştır. Gradient Boosting Machine (GBM) kullanarak 20 özellik seçmiş ve GBM ile sınıflandırma yapmıştır. Oluşturulan ilk sınıflandırma deneyinde 0.6369 oranında başarı elde edilirken Doğuran Çekirdekli Hilbert Uzayı (Reproducing Kernel Hilbert Space) yöntemi kullanılarak üç özellik kaldırılarak tekrar GBM ile sınıflandırma yapmıştır. Oluşturulan ikinci deneyde başarı 0.6476 olarak belirlemiştir (Lin, 2019).

Çayır ve Ark.'nın (2019) yaptıkları çalışmada Microsoft Kötü Amaçlı Yazılım Tahmini veri setinin tamamını kullanmışlar ve Kaggle platformunda gerçekleştirilen

yarışmada 2426 ekip arasından 5. olmuşlardır. Yaptıkları çalışmalarında öncelikle %70'den fazla kayıp olan özellikler elenmiş, %70 den az olan özellikleri tamamlamışlardır. Daha sonrasında veri ön işleme adımlarını tamamlamışlar ve veri setinin 30 özelliği kullanarak Extreme Gradient Boosting (XGBoost) yöntemi ile sınıflandırma yapmışlardır. Sınıflandırma sonucunda public ve private tablolarındaki eğri altındaki alan (ROC-AUC) skorları sırası ile 0.6781 ve 0.6640 olarak belirlenmiştir (Çayır ve ark., 2019).

Patel ve Ark.(2020) tarafından Microsoft Kötü Amaçlı Yazılım Tahmini veri seti, kullanılarak yapılan çalışmalarında öncelikle veri boyutunun yüksek olduğu durumlarda kötü amaçlı yazılımları tahmin etmek için gradyan artırıcı metotların kullanılması gerektiğini önermişlerdir. Yaptıkları bu çalışmada Light Gradient Boosted Machine (LightGBM) ve XGBoost yöntemlerini karşılaştırmışlardır. LightGBM ile yapılan deney sonucunda %73.89 oranında bir başarı sağlarken XGBoost yöntemi ile %70.48 oranında bir başarı elde etmişlerdir. Ayrıca LightGBM algoritmasının XGBoost yönteminden daha hızlı olduğunu gözlemlemişlerdir.

Asad ve Ark.(2020) tarafından yapılan çalışmada Microsoft Kötü Amaçlı Yazılım Tahmini veri seti ile danışmanlı makine öğrenme algoritmaları ve gradyan artırma algoritmaları kullanılarak kötü amaçlı yazılım tahmini çalışması yapmışlardır. Yapılan çalışmada LightGBM, Yapay Sinir Ağları (YSA) ve Karar Ağacı kullanılarak deneyler gerçekleştirilmişlerdir. Yapılan deneyler sonucunda 0.73926'lık bir doğruluk oranı ile en iyi model LightGBM algoritması olduğunu gözlemlemişlerdir. En kötü sonucu ise 0.63923'lük bir doğruluk oranı ile Karar Ağacı modeli olduğunu belirtmişlerdir.

Bu tez çalışmasında kullanılan Microsoft Kötü Amaçlı Yazılım Tahmini veri seti (2018) yeni oluşturulduğu için veri setini kullanarak yapılan çalışma sayısının sınırlı olduğu görülmüştür.

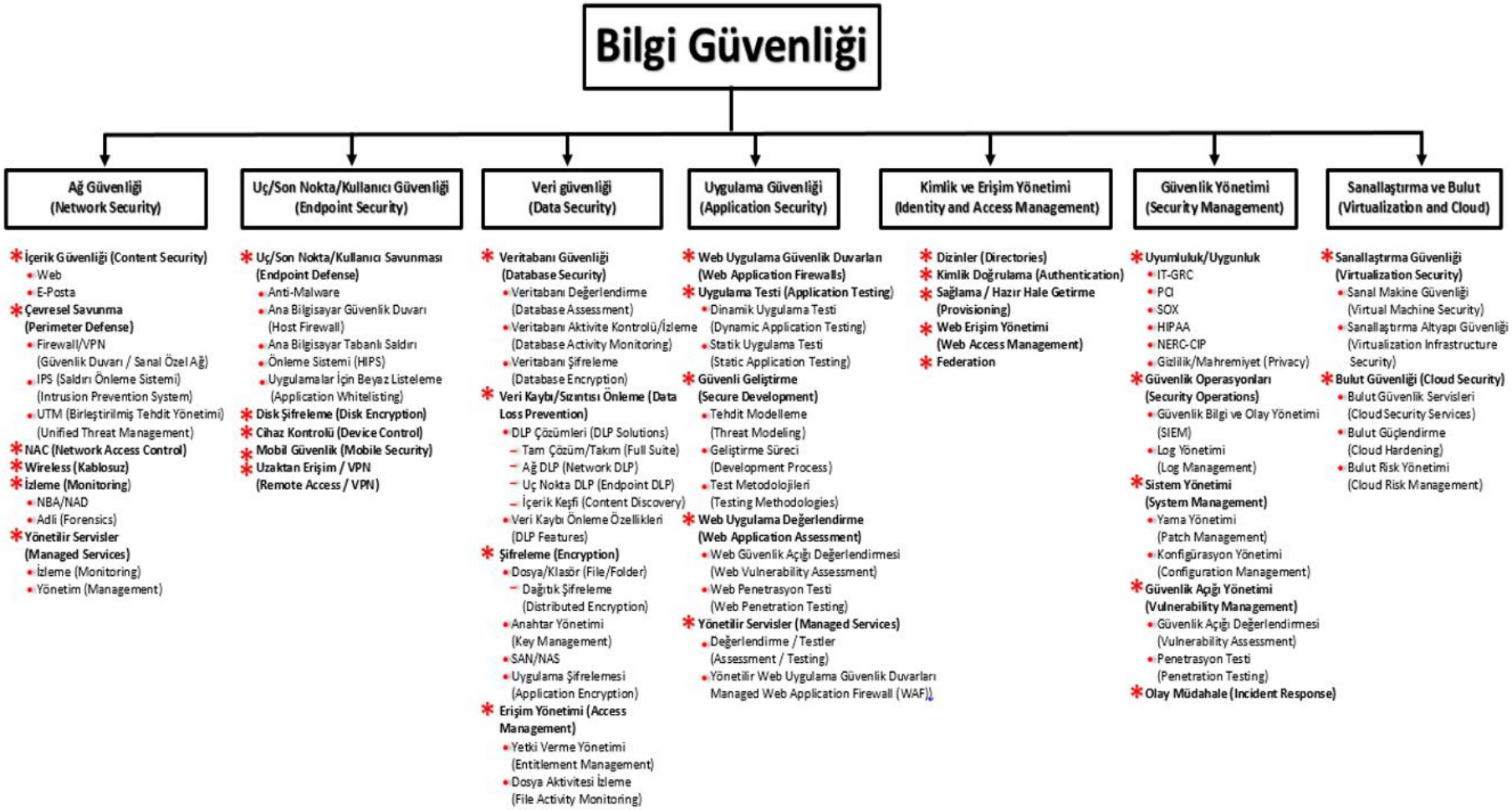
3. MATERYAL VE YÖNTEM

3.1. Bilgi Güvenliği

Verinin anlamlandırılmış, değerlendirilmiş ve düzenlenerek belirli bir hale dönüştürülmüş şekline bilgi (information) denilmektedir (Sağıroğlu, 2018). Bilgiler fiziksel veya elektronik ortamlarda depolanabilmektedir. Fiziksel veya elektronik bir ortamda saklanan bilginin, izinsiz veya yetkisiz bir biçimde erişme, kullanma, değiştirme, hasar verme, ortadan kaldırma gibi girişimlerden korunmasına bilgi güvenliği denilmektedir (Sağıroğlu, 2018). Bilgi güvenliğinin temel unsurları Şekil 3.1'de gösterilmiş olup gizlilik (Confidentiality), erişilebilirlik (Availability) ve bütünlükten (Integrity) oluşmaktadır (KurtKaya, 2017). Gizlilik bilginin istenmeyen kişilerden saklanması, erişilebilirlik bilginin istenilen kişilerce ihtiyaç duyulduğunda kullanılabilir, ulaşılabilir durumda olması iken, bütünlük ise bilginin sadece yetki sahibi olan kişilerce değiştirilebilmesidir (NormaTürk, 2016).

Şekil 3.1. Bilgi güvenliği temel unsurları (Singh ve ark., 2014)

Bilgi güvenliğinin temel unsurlarının zarar görmesi sonucunda güvenlik zafiyetleri meydana gelmektedir. Diğer bir tanımıyla zafiyet bir varlığı tehlikelere karşı korumasız hale getirebilecek unsur (sistemlerin yanlış veya eksik organize edilmesi, güvenlik politikaları, insan faktörü) şeklinde tanımlanabilmektedir (Erol ve Sağıroğlu, 2018). Bu zafiyetlerin bulunması ve bu sorunların giderilebilmesine yönelik çeşitli uygulamalar yapılmaktadır. Her ne kadar bilgi güvenliği alanında yapılan çalışmalar birbiri içerisine geçmiş olsa da Securosis firmasının yaptığı araştırmalar sonucunda bilgi güvenliği alanında yapılan çalışmalar sınıflandırılmıştır (Pesen, 2015). Yapılan bu sınıflandırma Şekil 3.2'de verilmiştir.



Şekil 3.2. Bilgi güvenliği sınıflandırması (Pesen, 2015)

Bilgi güvenliği, Şekil 3.2'de yapılan sınıflandırmaya göre ağ güvenliği (network security), uç(son nokta) kullanıcı güvenliği (endpoint security), veri güvenliği (data security), uygulama güvenliği (application security), kimlik ve erişim yönetimi (identity and access management), güvenlik yönetimi (security management), sanallaştırma ve bulut (virtualization and cloud) olarak yedi genel kategoride sınıflandırırken bu kategoriler toplamda otuz iki alt başlığa bölünerek çok geniş bir alanı kapsadığı gösterilmiştir.

3.1.1. Kötü Amaçlı Yazılım ve Analizi

Zararlı yazılım, kötü amaçlı yazılım, bazı kaynaklarda kötücül yazılımlar veya malware (malicious software) olarak geçen yazılımlar sistemler (bilgisayar, mobil cihazlar, televizyon, buzdolabı vb.) üzerinde zarar vermek, işleyişini bozmak, kritik bilgileri toplamak, erişim sağlayabilmek, istenmeyen reklamları göstermek amacıyla kullanılan yazılımlardır (Wikipedia, 2019).

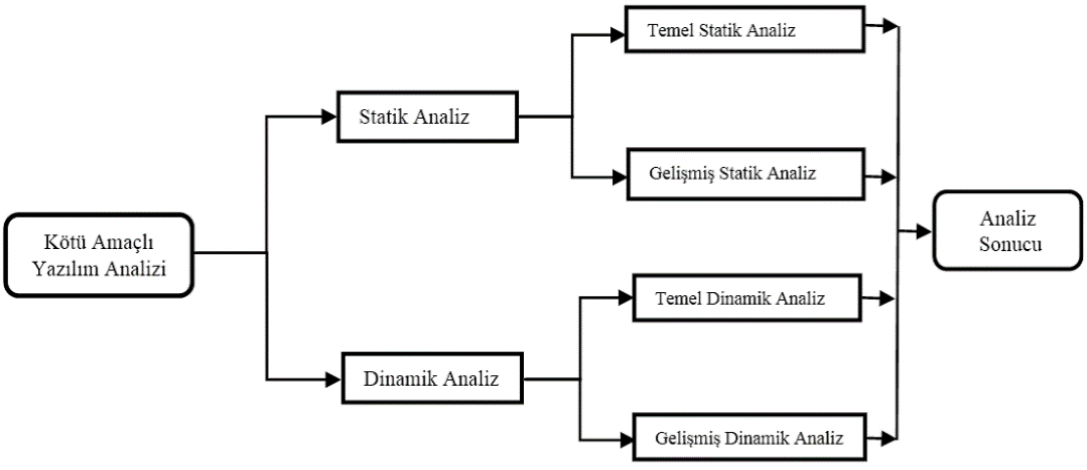
Kötü amaçlı yazılımlar başlangıçta basit ve küçük amaçlar için yazılmış olsalar da zaman içerisinde gerek çekirdek modda çalışabilen gerekse kullanıcı modunda çalışabilen büyük amaçlara hizmet eden yeni nesil kötü amaçlı yazılımlar geliştirilmiştir. Kendilerini sürekli olarak yenileyebilen yeni nesil kötü amaçlı yazılımlar ile geleneksel kötü amaçlı yazılımların karşılaştırılması Çizelge 3.1.'de verilmiştir. Yeni teknik ve yöntemlerin kullanılmasına rağmen bütün kötü amaçlı yazılımları tespit edebilmek mümkün görülememektedir (Samet ve Aslan, 2018).

Çizelge 3.1. Geleneksel ve yeni nesil kötü amaçlı yazılım özelliklerinin karşılaştırılması (Samet ve Aslan, 2018)

Geleneksel Kötü Amaçlı Yazılım Özellikleri	Yeni Nesil Kötü Amaçlı Yazılım Özellikleri
Genelde 1 işlemden oluşmaktadır.	1'den fazla işlemden oluşmaktadır.
Sınırlı sayıda işlemle iletişimde bulunmaktadır.	Var olan işlemleri etkilemektedir.
Sistemi bir sefer etkilemektedir.	Sistemde kalıcı hale gelmektedir.
Gizlenme ihtiyacı duymamaktadır.	Gizlenme teknikleri kullanmaktadır.
Genel saldırılar başlatmaktadır.	Hedef odaklı saldırılar başlatmaktadır.
Genelde ".exe" dosyaları yoluyla yayılmaktadır.	Genelde ".dll" dosyaları yoluyla yayılmaktadır.
Her kopyanın aynı ya da benzerdir.	Her kopyanın farklıdır.

Amaçlarına göre değişiklik gösteren kötü amaçlı yazılımları tespit edebilmek, bulaştıkları sistemdeki çalışmalarını görebilmek için ve verdikleri zararı önlemek amacıyla kötü amaçlı yazılım analizi çalışmaları yapılmaktadır. Yapılan analizlerle kötü amaçlı yazılımların etkilediği makineler veya programlar belirlenebilir, sistemdeki hangi güvenlik zafiyetlerini kullanarak sisteme eriştiğini veya hangi tür verilere ne gibi zararlar verdiği konusunda birçok bilgi elde edilebilmektedir. Analiz sonuçlarından elde edilen bilgiler ışığında gelecekte karşılaşılacak olan saldırılar önlenmeye çalışılmaktadır. Kötü amaçlı yazılım analizinde tersine mühendislik teknikleri kullanılmaktadır. Tersine mühendislik (reverse engineering), mekanik, elektronik sistemlerin veya bir yazılımın çalışma yapısını ortaya çıkarmak, kopyalamak veya üzerinde değişiklik yapılabilmesi için uygulanan mühendislik teknikleri olarak tanımlanmaktadır (Şirincan, 2016). Tersine mühendislik teknikleri ile kötü amaçlı yazılımların program yapısı ve sistemle veya sistemlerle olan ilişkisi belirlenmeye çalışılır. Bu çalışmalar sırasında sistem izleme araçları, paket ayırıcılar (Diassemblers), hata ayıklayıcılar (Debuggers) gibi araçlar kullanılmaktadır.

Kötü amaçlı yazılım analizi dinamik ve statik analiz olmak üzere iki ana gruba ayrılmaktadır (Sikorski ve Honig, 2012). Şekil 3.3.'de Statik analiz kendi içerisinde temel statik analiz, ileri düzey statik şeklinde ayrılırken dinamik analizde temel dinamik analiz ve ileri düzey dinamik olarak ayrılır ve toplamda dört gruba ayrıldığı görülmektedir (Samet ve Aslan, 2018).



Şekil 3.3. Kötü amaçlı yazılım analiz yöntemleri (Samet ve Aslan, 2018)

Statik analiz kötü amaçlı yazılımın sistem üzerinde çalıştırılmadan (execute edilmeden) dosya bilgilerine bakılarak incelenmesi işlemidir. Kodlar incelenerek dosya adı, dosya uzantısı, dosya boyutu, hash değeri, kötü amaçlı yazılım imzası, hangi kütüphaneler kullanıldığı gibi bilgiler elde edilmeye çalışılır. Bu tür bilgileri elde edebilmek için MD5deep, PEview, Hew-Rays Decompiler gibi birçok hazır araç bulunmaktadır. Statik analiz temel statik analiz ve gelişmiş statik analiz olmak üzere ikiye ayrılır.

- **Temel (basit) statik analiz** kötü amaçlı yazılımlar hakkında genel bilgiler elde edebilmek amacıyla yapılmaktadır. Bu amaçla kötü amaçlı yazılım dosyasının string dizilimleri, dosya başlıkları, hash değerleri, kullanılan metotlar, dosya başlıkları, dosya üzerine paketleme yapılıp yapılmadığı ve virüstotal gibi siteler yardımı ile antivirüslerin verdiği tepkiler kullanılarak yazılım hakkında bilgi edinilmeye çalışılır.
- **Gelişmiş (ileri düzey) statik analiz** sırasında program kodları tersine mühendislik (reverse engineering) metotları ve makine kodlarının assembly diline çevrilerek (diassember) derinlemesine incelemeler yapılır.

Dinamik analiz kötü amaçlı yazılımın sistem üzerinde çalışmasının incelenerek tespitlerin yapıldığı analiz yöntemidir. Analiz sırasında kötü amaçlı yazılımın hangi program ya da dosyaları etkilediği, verilen zararın ne boyutta olduğu, sistemdeki hangi güvenlik zafiyetlerinin kullanıldığı, etkilenen program dosya ya da sistemin eski haline nasıl döndürülebileceği gibi konular incelenmektedir (Samet ve Aslan, 2018). Kötü amaçlı yazılımlar çalıştırılmadan önce yalıtılmış bir çalıştırma ortamının sağlanması diğer sistemlerin güvenliğinin sağlanması için gerekli görülmektedir (Sikorski ve Honig, 2012).

- **Temel (basit) dinamik analizde** çalıştırılan uygulamanın ağ üzerindeki davranışları, bellek üzerindeki davranışları, dosya ve kayıt defteri üzerindeki yaptığı değişiklikler, oluşturulan processlerin incelendiği kısaca davranışsal analizin yapıldığı dinamik analiz çeşididir (Ceylan, 2018).
- **İleri (gelişmiş) dinamik analiz** sırasında kötü amaçlı yazılım çalışır durumdayken işlemci komutları, register (kayıt defteri) üzerindeki değişimler, fonksiyon çağırımları, fonksiyon parametreleri gibi veriler kullanılarak inceleme yapılır (Ceylan, 2018).

Statik ve dinamik analiz yöntemlerinin birbirlerine göre avantaj ve dezavantajları bulunmaktadır. Çizelge 3.2.'de karşılaştırılması görülmektedir. Bu karşılaştırmaya göre statik analizde zaman ve kaynak tüketiminin az olması, kararlı ve tekrarlanabilir olması ve bilinen kötü amaçlı yazılımlar için etkili sonuçlar vermesi gibi avantajlarının olmasına rağmen gizleme tekniği (Obfuscation, packed vb) kullanılmış kötü amaçlı yazılımlar için etkili değildir. Dinamik analiz kod gizleme tekniklerine karşı güçlü olması, yeni nesil kötü amaçlı yazılımları daha iyi tespit edebiliyor olmasına rağmen bu yönteminde zaman ve kaynak tüketiminin fazla olması gibi çeşitli dezavantajları bulunmaktadır.

Tez çalışmasında kullanılan Microsoft Kötü Amaçlı Yazılım Tahmini veri seti, Microsoft tarafından statik ve dinamik analiz yöntemleri beraber kullanılarak oluşturulmuştur.

Çizelge 3.2. Statik ve dinamik analiz avantaj ve dezavantajları

	Avantajlar	**Dezavantajlar**
Statik Analiz	Genel bir bakış açısı sunar (Multiple path execution)	Tersine mühendislik teknikleri bazı kısıtlamalar içerir
	Zaman ve kaynak tüketimi azdır	“Obfuscation” ve Polymorphic” tekniklerine karşı savunmasızdır
	Kararlı ve tekrarlanabilir	Zor ve karmaşıktır
	Gerçek makine zarar görmez	Yeni nesil kötü amaçlı yazılımların analizde etkisizdir
Dinamik Analiz	Basit ve kesin sonuçlar üretir	Sınırlı görünüm (Single path execution) sunar
	Gerçek zamanlı davranış bilgileri elde edilir	Çalışma durumuna ve zamanına göre farklı davranışlar gösterir
	Kod obfuscation tekniklerine karşı güçlüdür	Analiz otomatikleştirildiğinde etkileşimli davranış eksik kalır
	Yeni nesil kötü amaçlı yazılımlar tespit edilebilir	Zaman ve kaynak tüketimi fazladır
	-	Sanal ortamlarda bazen bütün davranışlar belirlenemez

Aslan'ın (2017) yaptığı çalışmada imza tabanlı algılama araçlarından biri olan antivirüs programları ile statik analiz araçlarını bilinen ve bilinmeyen kötü amaçlı yazılımların tespit oranlarını ve doğruluk oranlarını karşılaştırmıştır. Çalışmada antivirüs programları otomatik olarak çalışabildiğinden bilinen kötü amaçlı yazılımları tespit etmede hızlı ve etkili olduğu gösterilirken bilinmeyen kötü amaçlı yazılımlar için ne statik araçlar ne de antivirüs uygulamalarının etkili olmadığı belirtilmiştir. Ayrıca statik araçların oluşturduğu sonuçları yorumlamak için çok fazla insan gücünün gerekliliğinden de bahsedilmiştir. Çizelge 3.3.'de statik araçlar ile antivirüs tarayıcılarının bilinen ve bilinmeyen kötü amaçlı yazılımları tespit edebilme oranları ve doğruluk oranları verilmiştir (Aslan, 2017).

Çizelge 3.3. Bilinen ve bilinmeyen kötü amaçlı yazılımlar üzerinde antivirüs tarayıcısı ve statik analiz araçlarının performans değerlendirmesi (Aslan, 2017)

Kötü Amaçlı Yazılım Analizi	**Araç İsimleri**	**Bilinen Kötü Amaçlı Yazılımlar**		**Bilinmeyen Kötü Amaç Yazılımlar**	
		Tespit Oranı	**Doğruluk**	**Tespit Oranı**	**Doğruluk**
Statik Analiz Araçları	**1.** BinText, Dependency Walker	%66.2	%71.2	%54.2	%56.4
	2. PEiD, Dependency Walker	%66.4	%72.4	%55.4	%56.2
	3. Dependency Walker, PeiD, PE Explorer	%67.6	%75.9	%58.8	%58.6
	4.PEiD, BinText, PEview, IDA Pro	%74.2	%78.2	%60.7	%61.9
	5. UPX, IDA Pro, BinText MD5deep, Resource Hacker	%83.2	%84.2	%66.3	%68.2
Antivirüs Tarayıcı	**1.** Norton	%78.2	%79.2	%52.2	%53.2
	2. Bitdefender	%77.2	%77.8	%53.6	%53.7
	3. Avira, Kaspersky	%77.3.2	%80.5	%54.2	%55.2
	4. ClamAV, McAfee Avast	%79.6	%80.2	%56.2	%56.8
	5. Norton, ClamAV, Kaspersky, Bitdefender	%84.2.	%85.7	% 59.2	%58.9

Aslan (2017)'ın yaptığı aynı araştırmaya göre statik analiz araçları ile yapılan analizlerin kötü amaçlı bazı yazılımların karmaşıklığına göre birkaç gün sürebileceği göz önüne alınarak analiz çalışmalarının otomatik hale getirilmesi yönünde talep olduğundan söz edilmiştir (Aslan, 2017).

3.2. Telemetri Bilgisi

Telemetri kelimesi Tele (uzak) metry (ölçüm) kelimelerinin bir araya gelmesinden meydana gelmiş olup sözlük anlamı olarak uzaktan izleme anlamına gelmektedir (TÜBİTAK-SAGE, 2020). Bir sistem ya da tesisten uzaktan erişim yöntemleri ile gelen verileri ifade etmektedir. Uygulamalar ilk kez 150 yıl öncesinde Amerika'da görülmüştür (TÜBİTAK-SAGE, 2020) ve günümüzde yaygın bir şekilde askeri ve sivil alanlarda kullanılmaktadır. Genellikle uygulamaların kullanıcılara daha iyi hizmet vermesi amacı ile kullanılmaktadır.

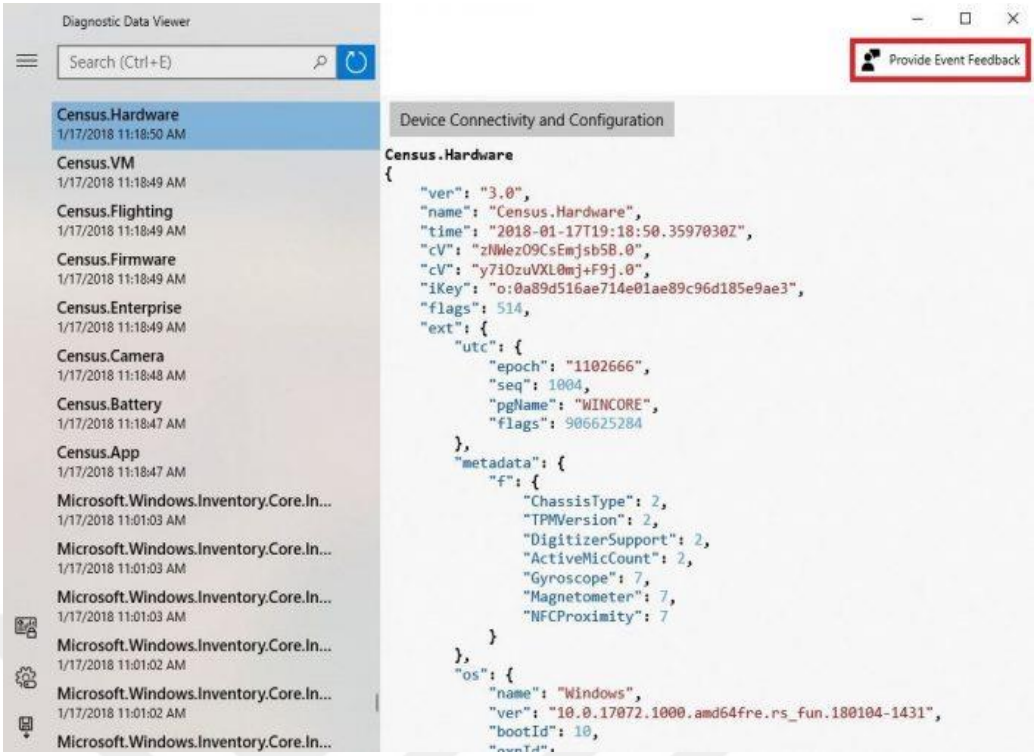
Bilgisayar sistemlerinde de sıklıkla kullanılan telemetri verileri Windows işletim sistemlerinin birçoğunda Windows telemetri verileri toplanmaktadır. Otomatik olarak toplanan bu veriler Microsoft kontrol noktasına gönderilmektedir. DiagTrack servis hizmetine sahip Windows'lar her on beş dakikada bir telemetri bilgilerini göndermekle beraber güç seçenekleri ve ağ ayarlarına göre bu süre değişebilmektedir. Servisin sürekli olarak sistem üzerinden veri toplaması bilgisayar sisteminin kaynaklarının bazı durumlarda verimsiz olarak kullanılmasına sebep olabilmektedir. Varsayılan olarak Windows kullanıcılarına açık olarak gelen bu hizmet Han'a göre arzu edildiği takdirde kapılabilmesi mümkünken (Han ve ark., 2020) kapatılması halinde Windows üzerinde otomatik olarak çalışan bazı uygulamaların çalışmadığı görülebilmektedir.

Windows telemetri verileri sisteme özel bilgileri, özellikleri, uygulamalar ile ilgili bilgileri, sistem dosyalarını ve henüz açıklanmayan daha fazla veriyi içerebilir (Chu, 2015). Gönderilecek telemetri verilerini kullanıcılar belirleyebilir ancak Chu'e (2015) göre bu tam olarak mümkün değildir (Chu, 2015). Microsoft, telemetri verilerini diğer bir ifade ile tanılama verilerinin (diagnostic data) işletim sisteminin türüne göre değişmekle birlikte genel olarak dört katmana bölerek incelemektedir ve bunlar Güvenlik (Security), Temel (Basic), Gelişmiş (Enhanced), Tam (Full) katmanlarıdır (Microsoft, 2020a).

- **Güvenlik (Diagnostic data off (Security))**: Bu ayar yapılandırıldığında cihazdan Windows telemetri verisi gönderilmemektedir. Bu ayar yalızca Windows Server, Windows 10 Enterprise ve Windows 10 Education sürümlerinde bulunmaktadır (Microsoft, 2020a).

- **Temel (Required diagnostic data(Basic)):** Bu ayar sınırlı sayıda veri toplamaktadır. Bu veriler toplanarak donanım veya yazılım konfigürasyonunda ortaya çıkabilecek sorunların belirlenmesinde yardımcı olmaktadır. Örneğin sürücü sayısı, türü ve boyutu gibi depolama özellikleri, cihazların güncellemeden sonra çalışıp çalışmadığı, kamera ve ekran çözünürlüğü gibi cihaz özellikleri, Microsoft Store'un nasıl performans gösterdiğine dair bilgiler gibi bilgileri toplanmaktadır (Microsoft, 2020a).
- **Gelişmiş (Enhanced diagnostic data):** Bu ayar seçildiğinde temel teşhis verine ek olarak ziyaret edilen siteler, Windows uygulamalarının nasıl kullanıldığı ve nasıl çalıştığı gibi Windows uygulamalarından gelen ek veriler toplanmaktadır. Ayrıca ağ iletişimi, dosya sistemleri, işletim sistemi olayları gibi birçok farklı veri toplanmaktadır (Microsoft, 2020a).
- **Tam (Optional diagnostic data (Full)):** Bu ayar tercih edildiğinde temel teşhis verilerine ek olarak cihaz özellikleri, ziyaret edilen siteler, işletim sistemi ve diğer sistemlerin durumu hakkında bilgiler, programların çalışma süreleri, hata raporları gibi bilgiler toplanmaktadır (Microsoft, 2020a).

Bazı Windows sürümlerinde sistem üzerinden toplanan verileri Windows Tanılama Veri Görüntüleyicisi'ni (Diagnostic Data Viewer) kullanarak Microsoft'ta iletilen tanılama verileri görüntülenebilmektedir. Şekil 3.4’de sistemden toplanan verilerin Windows Tanılama Veri Görüntüleyicisi üzerinde veri içerikleri gösterilmektedir



Şekil 3.4. Windows Tanılama Veri Görüntüleyicisi'nin gösterdiği veri örneği (Chu, 2015)

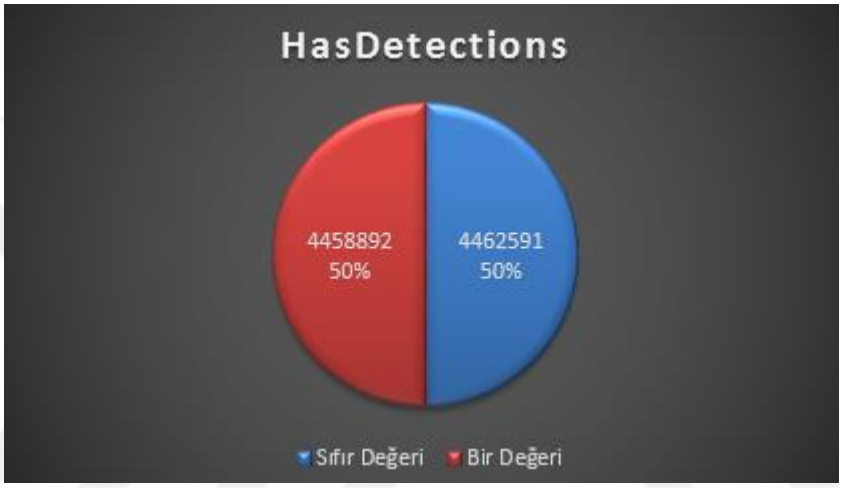
3.3. Microsoft Kötü Amaçlı Yazılım Tahmini Veri Seti

Kötü amaçlı yazılım endüstrisi, geleneksel güvenlik yöntemlerini aşmaya çalışan bu yolda her türlü faaliyeti sürdüren iyi organize edilen ve finanse edilen bir pazar olmaya devam etmektedir. Bir bilgisayar sistemine bulaşmış kötü amaçlı yazılım tüketicilere ve işletmelere birçok şekilde zarar vermektedir.

Microsoft firması kötü amaçlı yazılımların üstesinden gelebilmek amacıyla 2018 yılında makine öğrenmeni ve veri bilimi topluluğu olan Kaggle platformunda "Microsoft Malware Prediction" (2018) isimli bir yarışma düzenlenmiştir (Microsoft, 2018). Yarışmaya 2426 ekip katılmış ve 2019 yılının Mart ayında son bulmuştur. Yarışmanın liderlik tablosu (private leaderboard) test verilerinin yaklaşık %37'si ile hesaplanmış olup ROC eğrisi altında kalan alana göre sıralanmıştır. Tablodaki en yüksek ROC_AUC değeri 0.67585 olarak kaydedilmiştir. Düzenlenen yarışmada kullanılan veri seti makine enfeksiyonlarını içeren telemetri bilgileri, Windows Defender tarafından toplanan veriler ve tehdit raporlarının birleştirilmesi ile üretilen açık erişimli bir veri setidir.

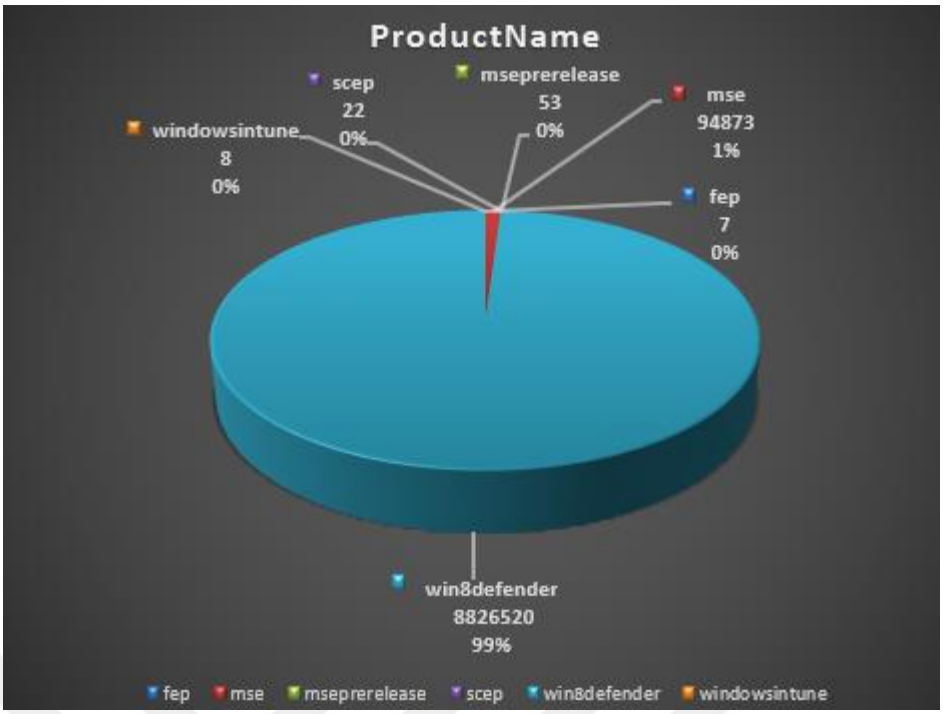
Veri kümesi içerisindeki her bir veri satırı, MachineIdentifier ile tanımlanan benzersiz bir makine numarasına karşılık gelmektedir. Örneğin ilk makine "0000028988387b115f69f31a3bf04f09" şeklinde numaralandırmıştır. Veri seti içerisindeki son özellik olan "HasDetections" özelliği kötü amaçlı bir yazılımın tespit

edilip edilmediğini göstermektedir. Bu özellik sınıf etiketini oluşturmaktadır ve sınıf etiketinde kayıp veri bulunmamaktadır. Veri kümesi test ve eğitim veri seti olarak iki kısımdan oluşmaktadır. Test veri seti henüz doğrulanmış yani HasDetections özelliği olmadan paylaşılmıştır ve 7.853.253 satır veriden (3.53 GB) meydana gelmektedir. Eğitim veri seti 8.921.483 satır veriden (4.08 GB) oluşmaktadır. Test veri seti HasDetections özelliği olmadığı için 82 özellikten, eğitim veri seti ise HasDetections özelliği ile birlikte 83 özellikten meydana gelmektedir. HasDetections özelliğinin aldığı değerlerin dağılımı Şekil 3.5'de gösterilmiştir.



Şekil 3.5. HasDetections özelliğinin aldığı değerler dağılımı

Veri setinin ikinci özelliği olan "ProductName" Defender durumlarını göstermektedir ve "fep" (Forefront Endpoint Protection), "mse" (Microsoft Security Essentials), "mseprerelease" (Microsoft Security Essentials Pre-Release), "scep" (System Center Endpoint Protection), win8defender, windowsintune gibi altı farklı değer alabilmektedir. Bu değerler arasında en fazla %98.936 ile windows8defender değeri ön plana çıkmaktadır. "ProductName" özelliğinin alabileceği değerlerin dağılımı Şekil 3.6'de gösterilmiştir.



Şekil 3.6. ProductName özelliğinin aldığı değerler dağılımı

Veri seti içerisinde yer alan diğer özelliklerin isimleri ve özellikler ile ilgili açıklamalar Çizelge 3.4.'de yer verilmiştir.

Çizelge 3.4. Microsoft kötü amaçlı yazılım tahmini veri seti özelik isimleri ve tanımlamaları (Betancourt, 2019).

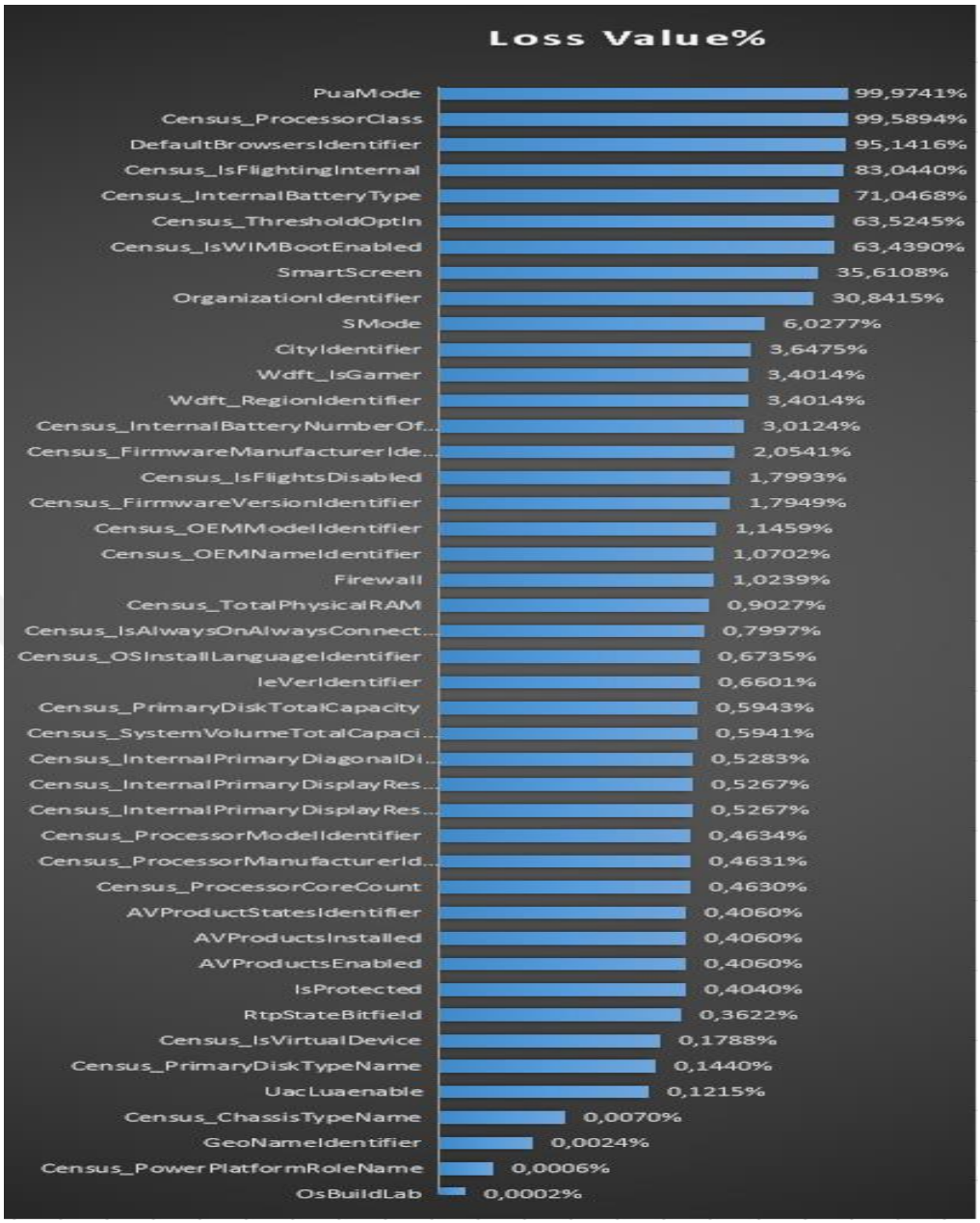
No	Özellik	Tanım
1	MachineIdentifier	Makinenin benzersiz kimliği
2	ProductName	Defender durum bilgisi Ör: win8defender
3	EngineVersion	Defender durum bilgisi Ör: 1.1.12603.0
4	AppVersion	Defender durum bilgisi Ör: 4.9.10586.0
5	AvSigVersion	Defender durum bilgisi Ör: 1.217.1014.0
6	IsBeta	Defender durum bilgisi Ör: False
7	RtpStateBitfield	NA
8	IsSxsPassiveMode	NA
9	DefaultBrowsersIdentifier	Makinenin varsayılan tarayıcı kimliği
10	AVProductStatesIdentifier	Kullanıcının virüsten koruma yazılımının belirli yapılandırmasının kimliği
11	AVProductsInstalled	NA
12	AVProductsEnabled	NA
13	HasTpm	Makinede TPM varsa doğru

14	CountryIdentifier	Makinenin bulunduğu ülkenin kimliği
15	CityIdentifier	Makinenin bulunduğu şehrin kimliği
16	OrganizationIdentifier	Makinenin ait olduğu kuruluşun kimliği
17	GeoNameIdentifier	Makinenin bulunduğu coğrafi bölgenin kimliği
18	LocaleEnglishNameIdentifier	Geçerli kullanıcının yerel ayar kimliğinin ingilizce adı
19	Platform	Platform adının hesaplanması
20	Processor	Kurulu işletim sisteminin mimarisi
21	OsVer	Geçerli işletim sisteminin sürümü
22	OsBuild	Mevcut işletim sisteminin yapısı
23	OsSuite	Geçerli işletim sistemi için ürün paketi maskesi
24	OsPlatformSubRelease	İşletim sisteminin platform alt sürüm numarası
25	OsBuildLab	İşletim Sistemini üreten lab.
26	SkuEdition	’SKU'suna eşlemek için MSDN'de tanımlanan ürün türü
27	IsProtected	Spynet Raporunun AV ürünlerinden türetilen hesaplanan alan
28	AutoSampleOptIn	Hizmetten geçirilen SubmitSamplesConsent değeri
29	PuaMode	Pua etkin Modunu gösterir
30	SMode	Cihazın 'S Modu'nda ise bu alan true olarak ayarlanır
31	IeVerIdentifier	NA
32	SmartScreen	Kayıt defterindeki SmartScreen etkin dize değeri
33	Firewall	Güvenlik duvarı etkin ise 1 değerini alır
34	UacLuaenable	"Yönetici Onay Modunda yönetici" etkin olup olmadığı gösterir
35	Census_MDC2FormFactor	Aygıt Sayımı düzeyi HW özelliklerine dayalı bir gruplama.
36	Census_DeviceFamily	Aygıt sınıfını gösterir.
37	Census_OEMNameIdentifier	NA
38	Census_OEMModelIdentifier	NA
39	Census_ProcessorCoreCount	İşlemcideki mantıksal çekirdek sayısı
40	Census_ProcessorManufacturerIdentifier	NA
41	Census_ProcessorModelIdentifier	NA

42	Census_ProcessorClass	İşlemcilerin yüksek / orta / düşük gibi sınıflandırılması.
43	Census_PrimaryDiskTotalCapacity	Makinenin birincil diskindeki disk alanı miktarı.
44	Census_PrimaryDiskTypeName	Birincil disk türünün kısa adı
45	Census_SystemVolumeTotalCapacity	Sistemin yüklü olduğu bölümün boyutu
46	Census_HasOpticalDiskDrive	Makinenin optik disk sürücüsünün olup olmadığını gösterir.
47	Census_TotalPhysicalRAM	Fiziksel Ram'i MB cinsinden gösterir.
48	Census_ChassisTypeName	Makinenin ne tür chasis sahip olduğunun sayısal gösterimi
49	Census_InternalPrimaryDiagonalDisplaySizeInInches	Birincil ekranın inç cinsinden fiziksel diyagonal uzunluğu
50	Census_InternalPrimaryDisplayResolutionHorizontal	Dâhili ekranın yatay yönündeki piksel sayısı
51	Census_InternalPrimaryDisplayResolutionVertical	Dâhili ekranın dikey yönündeki piksel sayısı
52	Census_PowerPlatformRoleName	OEM'in tercih ettiği güç yönetim profilini belirtir.
53	Census_InternalBatteryType	NA
54	Census_InternalBatteryNumberOfCharges	NA
55	Census_OSVersion	Sayısal işletim sistemi sürümü
50	Census_InternalPrimaryDisplayResolutionHorizontal	Dâhili ekranın yatay yönündeki piksel sayısı
51	Census_InternalPrimaryDisplayResolutionVertical	Dâhili ekranın dikey yönündeki piksel sayısı
52	Census_PowerPlatformRoleName	OEM'in tercih ettiği güç yönetim profilini belirtir.
53	Census_InternalBatteryType	NA
54	Census_InternalBatteryNumberOfCharges	NA
55	Census_OSVersion	Sayısal işletim sistemi sürümü
56	Census_OSArchitecture	İşletim sistemi mimarisi
57	Census_OSBranch	OsVerisyonFull'dan çıkarılan işletim sisteminin şubesi
58	Census_OSBuildNumber	OSVersionFull'dan çıkarılan derleme numarası
59	Census_OSBuildRevision	OSVersiyonFull'dan çıkarılan revizyon derlemesi
60	Census_OSEdition	Geçerli işletim sisteminin sürümü
61	Census_OSSkuName	İşletim sisteminin sürümü kısa adı
62	Census_OSInstallTypeName	Makinede hangi kurulumun kullanıldığı gösteren kısa tanım
63	Census_OSInstallLanguageIdentifier	NA
64	Census_OSUILocaleIdentifier	NA

65	Census_OSWUAutoUpdateOptionsName	WindowsUpdate otomatik güncellemelerin ayarlarının kısa adı
66	Census_IsPortableOperatingSystem	İşletim sisteminin Windows üzerinden başlatılıp başlatılmadığını ve çalışıp çalışmadığını gösterir
67	Census_GenuineStateName	OSGenuineStateID ögesinin kısa adı.
68	Census_ActivationChannel	Bir makine için perakende lisans anahtarı veya Toplu lisans anahtarı.
69	Census_IsFlightingInternal	NA
70	Census_IsFlightsDisabled	Makinenin flighting'a katılıp katılmadığını gösterir
71	Census_FlightRing	Cihaz kullanıcısının uçuş almak istediği zil.
72	Census_ThresholdOptIn	NA
73	Census_FirmwareManufacturerIdentifier	NA
74	Census_FirmwareVersionIdentifier	NA
75	Census_IsSecureBootEnabled	Güvenli Önyükleme modunun etkin olup olmadığını gösterir.
76	Census_IsWIMBootEnabled	NA
77	Census_IsVirtualDevice	Sanal makineyi tanımlar
78	Census_IsTouchEnabled	Dokunmatik bir cihaz mı?
79	Census_IsPenCapable	Cihaz kalem girişi yapabilir mi?
80	Census_IsAlwaysOnAlwaysConnectedCapable	Pilin cihazın AlwaysConnected olmasını sağlayıp sağlamadığı
81	Wdft_IsGamer	Cihazın HW'ye dayalı bir oyun cihazı olup olmadığı.
82	Wdft_RegionIdentifier	NA
83	HasDetections	Makinede kötü amaçlı yazılım algılanıp algılanmadığını gösterir.

Eğitim veri seti içerisinde %99 ile %0.12 arasında değişen 44 farklı özellikte kayıp değer bulunmaktadır. Şekil 3.7’de kayıp özelliklerin isimleri ve kayıp oranları gösterilmiştir.

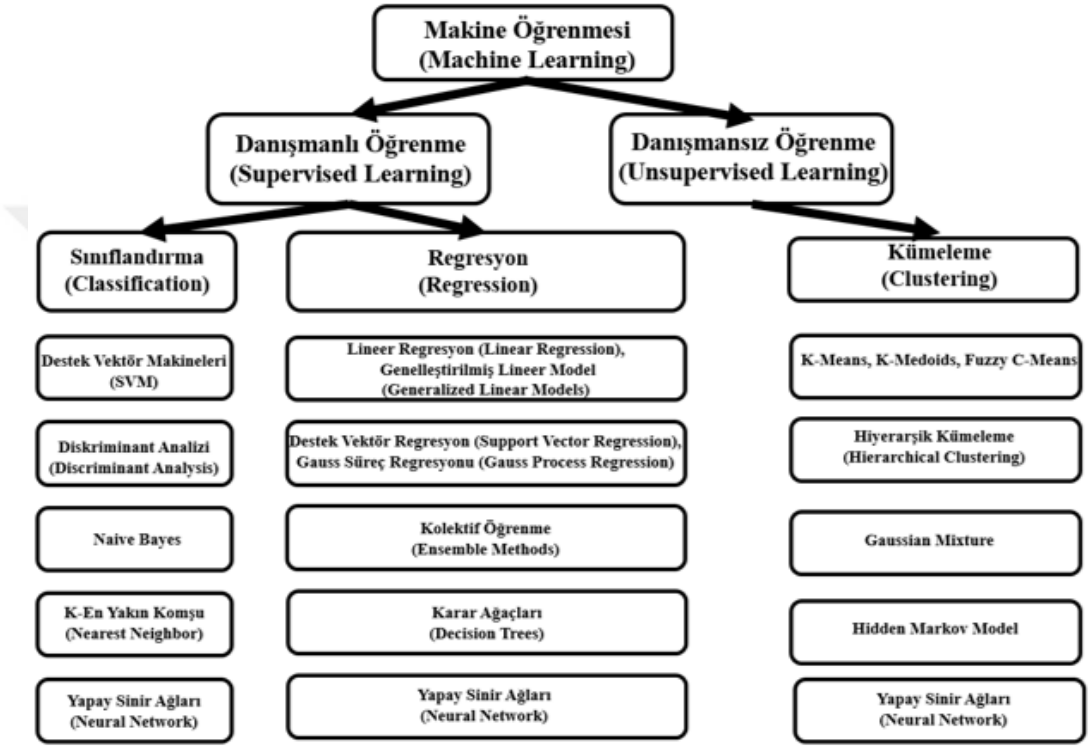


Şekil 3.7. Kayıp değeri bulunan özellikler ve kayıp veri oranı

3.4. Makine Öğrenmesi Yöntemleri

Makine öğrenmesi (Machine Learning) son zamanlarda sıklıkla kullanılan terimler arasında yer almaktadır. Yapay zekânın alt dallarından olan makine öğrenimi özellikle büyük verilerin karşımıza çıkmasından sonra popülerliğini daha da artırmıştır. Makine öğrenmesi terimi Amerikan bilgisayar bilimcisi Arthur Samuel tarafından 1959'da "bilgisayarın açıkça programlanmadan öğrenme yeteneği" olarak tanımlamaktadır (Samuel, 1959). Daha açık bir tanım yapmak gerekirse makine öğrenmesi çözülmek istenen bir problemin bu probleme ait ortamlardan elde edilen verilere göre modelleme işlemi yapan bilgisayar algoritmalarının genel adı olarak

tanımlanabilmektedir. Makine öğrenmesi saldırı tespit sistemlerinde, kötü amaçlı yazılım tanımada, kanserli hücre tespitinde, nesne ve kişi tanıma işlemleri gibi birçok alanda kullanılmaktadır. Verinin yapısına göre makine öğrenmesi algoritmaları danışmanlı (supervised) öğrenme ve danışmansız öğrenme (unsupervised) öğrenme olmak üzere ikiye ayrılmaktadır (VanderPlas, 2017). Şekil 3.8'de makine öğrenmesi teknikleri gösterilmiştir.



Şekil 3.8. Makine öğrenmesi teknikleri (MathWorks, 2020)

Danışmanlı (supervised) Öğrenme: Sınıf etiketi bilinen veri seti kullanılarak, seçilen algoritma eğitilir. Sonrasında eğitimi tamamlana algoritma, tahmini yapılmaya çalışılan veri setinin sonuçlarını belirlenmeye çalışılır. Örneğin bir bölgenin geçmiş yıllarda yağan yağış miktarlarını göz önüne alarak bu sene ne kadar yağış olacağının tahmin edilmesi danışmanlı öğrenmeye örnek gösterilebilir.

Danışmansız (gözetimsiz unsupervised) Öğrenme: Herhangi bir sınıf etiketine başvurmaksızın veri kümesinin özelliklerine göre modelleme yapar. Örneğin bir marketten her bir müşterinin neler aldığı kullanılarak hangi reyonların yan yana olması gerektiğine karar verilebilir (Yıldırım, 2018).

3.4.1. Naive Bayes Sınıflandırma Algoritması

Naive Bayes algoritması İngiliz matematikçi Thomas tarafından geliştirilmiş olan bir sınıflandırma algoritmasıdır. Koşullu olasılık ilkelerine dayanarak daha önceden sınıfları bilinen veriler ile eğitim tamamlanır ve daha sonra sınıflandırılmak istenen veriler sınıflandırılmaya çalışılır. Naive Bayes sınıflandırma yapısı bayes Denklem 3.1'de gösterilen bayes teoremine dayanmaktadır.

$$P(A|B)=\frac{P(B|A)\,P(A)}{P(B)} \tag{3.1}$$

Denklem 3.1'deki A ve B olayları rasgele olaylar olmak üzere;

P(A) : A olayının gerçekleşme olasılığı

P(B) : B olayının gerçekleşme olasılığı

P(A|B) : B olayı gerçekleştiğinde A olayının gerçekleşeme olasılığı

P(B|A) : A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı

Naive Bayes metodu birçok farklı alanda kullanılabilmektedir. Çizelge 3.5.'de verilen veri seti Naive Bayes metodunun uygulamasını gösterebilmek amacıyla Microsoft Malware Prediction (2018) veri setinden yararlanılarak oluşturulmuştur (Yumak, 2011), (Microsoft, 2018). Veri kümesinde bir bilgisayarın OsVer, PuaMode ve SmartScreen özeliklerinden yararlanılarak makinede kötü amaçlı yazılım tespit edilip edilmediği (HasDetect) gösterilmiştir.

Çizelge 3.5. Bilgisayarlarda kötü amaçlı yazılım tespit durumu veri kümesi

Bilgisayar İsmi	OsVer	PuaMode	SmartScreen	HasDetect.
1.Bilgisayar	Win8	1	2	1
2.Bilgisayar	Win8.1	1	1	0
3.Bilgisayar	Win10	0	0	0
4.Bilgisayar	Win8	1	0	1
5.Bilgisayar	Win8.1	1	0	1
6.Bilgisayar	Win10	0	2	1
7.Bilgisayar	Win8.1	0	1	0
8.Bilgisayar	Win8	0	0	1
9.Bilgisayar	Win10	0	0	?

Çizelgedeki 3.5.'deki veri kümesinden yararlanılarak bu verilerin frekans özellikleri Çizelge 3.6.'de verilmiştir. Bu olasılıklar Naive Bayes metodu üzerinde uygulanarak dokuzuncu bilgisayarın sınıfı bulunmaya çalışılmıştır. $P(1_{evet}|Win10,0,0)$ için Denklem 3.2 kullanılmış ve $P(0_{hayır}|Win10,0,0)$ için Denklem 3.3 kullanılmıştır. Her iki denklem için ortak olan P(Win10,0,0) göz ardı edilmiştir.

Çizelge 3.6. Özelliklerin frekanslarına göre olasılıkları

	Değerler	**HasDetect (1)**	**HasDetect (0)**		
OsVer	Win8	3	0	P(Win8|1)=3/5	P(Win8|0)=0/3
	Win8.1.	1	2	P(Win8.1|1)=1/5	P(Win8.1|0)=2/3
	Win10	1	1	P(Win10|1)1/5	P(Win10|0)1/3
PuaMode	1(etkin)	3	1	$P(1_{etkin}\|1)=3/5$	$P(1_{etkin}\|0)=1/3$
	0(pasif)	2	2	$P(0_{pasif}\|1)=2/5$	$P(0_{pasif}\|0)=2/3$
SmartScreen	0	3	1	$P(0_{Smart}\|1)=3/5$	$P(0_{Smart}\|0)=1/3$
	1	0	2	$P(1_{Smart}\|1)=0/5$	$P(1_{Smart}\|0)=2/3$
	2	2	0	$P(2_{Smart}\|1)=2/5$	$P(2_{Smart}\|0)=0/3$

$$P(1_{evet}|Win10,0,0) = 1/5*2/5*3/5*5/8 = 0.03 \quad (3.2)$$

$$P(0_{hayır}|Win10,0,0) = 1/3*2/3*1/3*3/8 = 0.028 \quad (3.3)$$

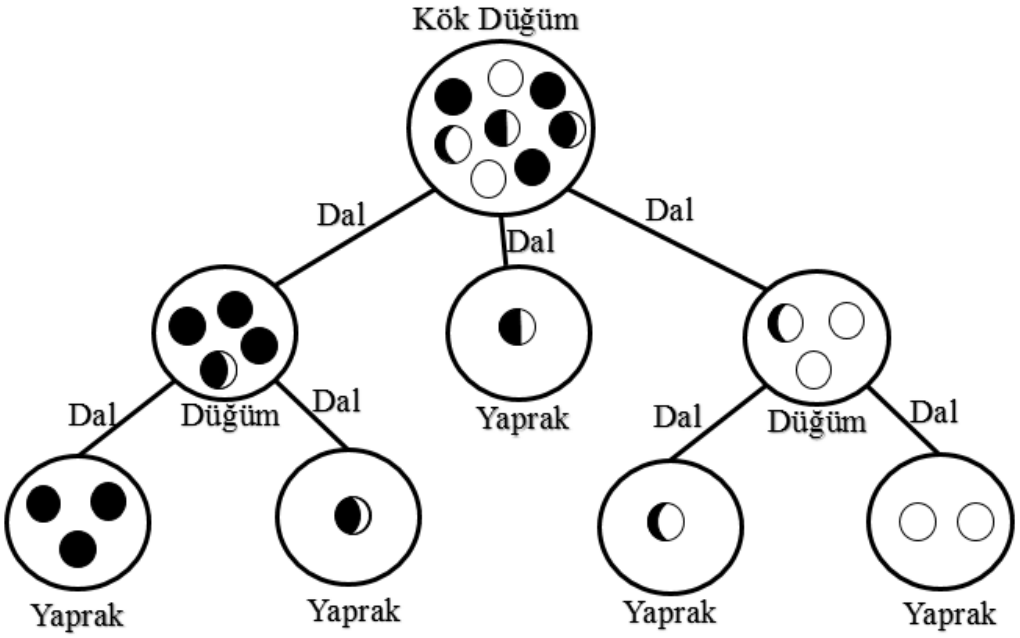
$P(1_{evet}|Win10,0,0)$ değeri, $P(0_{hayır}|Win10,0,0)$ değerinden daha büyük olduğu için 9. Bilgisayar için HasDetect değeri 1 yani bilgisayarda kötü amaçlı yazılım var olduğu kabul edilmiştir.

Naive Bayes'in yapılan uygulama alanına göre birçok farklı türü bulunmaktadır. Genellikle özelliklerin sürekli değer alıyorsa Gaussian Naive Bayes kullanılmaktadır. Çok sınıflı bir kategorize işlemi yapılmak isteniyorsa örneğin ticaret sitelerini spor, giyim, kozmetik gibi sınıflarda kategorize edilmek isteniyorsa Multinominal Naive Bayes kullanılmaktadır. Multinominal Naive Bayes'e benzeyen ancak sadece boolean (ikili)

sınıflandırma için kullanılan Bernoulli Naive Bayes yöntemi, kategorik değeri Evet/Hayır, Spam/Spam Değil, 1/0 gibi değerleri alarak yapmaktadır.

3.4.2. Karar Ağacı

Karar ağaçları (decision tree) sınıflandırma problemleri için kolay yapılandırılabilmesi, anlaşılması ve yorumlanmasının basit olması sebebi ile çok sık tercih edilen bir yöntemdir. Danışmanlı bir öğrenme yöntemi olan karar ağaçları önceden tanımlanmış etiketli verilerin öğrenilmesi ile oluşturulmaktadır. Karar ağaçları düğüm, dal ve yapraklardan oluşurken ağacın tüm eğitim örneklerini içeren en üstteki düğümüne kök denilmektedir (Krzysztof Cios ve ark., 2007). Kök düğüm ile düğümler arasında kalan yapıya dal, en sonda oluşan yapılara yaprak denilmektedir. Şekil 3.9.'da basit bir karar ağacı yapısı gösterilmiştir. Karar ağaçlarının oluşumunda dallanmanın nasıl olacağı, hangi kritere göre belirleneceği ya da hangi özelliklere göre ağaç yapısının belirleneceği konusunda farklı yaklaşımlar bulunmaktadır. Bunlardan en önemlileri bilgi kazancı, Gini İndeksi, Twoing kuralı, Ki-Kare olasılık tablo istatistiği gibi yaklaşımlardır (Kavzoğlu ve Çölkesen, 2010). Karar ağaçlarındaki dallanmanın amacı eğitim veri kümesini mümkün olduğunca benzer özellikleri gösteren alt veri kümelerine bölmektir (Aggarwal, 2015). Bu bölme işlemi kesme kuralı şartı sağlanana kadar devam etmektedir.



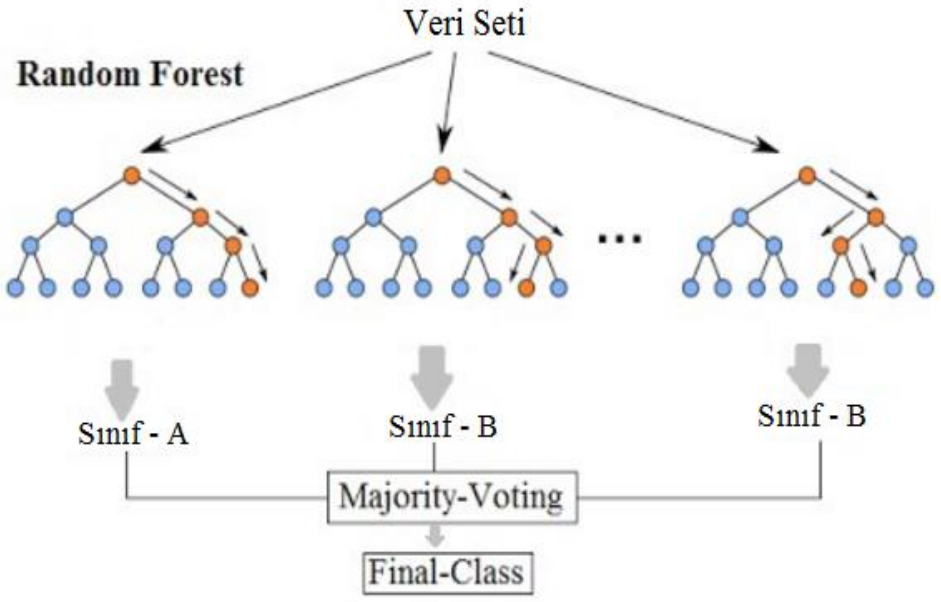
Şekil 3.9. Basit bir karar ağacı yapısı

Gini indeksi bütün değişkenlerin sürekli olduğunu varsayarak işlem yapmaktadır (Akçetin ve Çelik, 2014). Sürekliliği bozan en küçük gini indeksine sahip olanı seçerek dallanma işlemini gerçekleştirmektedir. T düğümü için gini indeksi Denklem 3.4.'e göre hesaplanmaktadır. Eşitlikte verilen p(j|T), T düğümündeki j sınıfına ait bağlı olasılığı göstermektedir.

$$\text{Gini(T)} = 1 - \sum_j [p(j|T)]^2 \qquad (3.4)$$

3.4.3. Random Forest

Random Forest algoritması Bagging algoritmasının temellerine dayanmaktadır. 1996'da Breiman tarafından geliştirilen Bagging algoritması eğitim setini kullanarak farklı alt veri kümeleri oluşturur ve bu alt veri kümelerinden oluşan ağaçlar ile sınıflandırma işlemi gerçekleştirilmektedir (Breiman, 1996). Birden fazla sayıda ağaç oluşturulur ve oluşturulacak ağaç sayısı kullanıcıdan alınmaktadır. Ağaç sayısının fazla olması sınıflandırma işleminin başarısını artırmaktadır. Oluşturulan ağaçlar maksimum boyutta olacak şekilde budama işlemi olmadan geliştirilir. Bunun için CART algoritması kullanılmaktadır (Breiman, 2001). Ağaç üretim işlemi tamamlandıktan sonra yeni gelen bir veri tüm ağaçlarda sınıflandırma işlemi uygulanır ve en çok oyu alan (Majority Voting) sınıfa dâhil edilir. Şekil 3.10'da Random Forest algoritmasının Majority Voting işlemine göre sınıflandırılması gösterilmiştir.



Şekil 3.10. Random Forest algoritması sınıflandırma yapısı (Kumar, 2016)

3.4.4. Adaboost (Adaptive Boosting)

Boosting kelimesinin Türkçe anlamı yükseltme anlamına gelmektedir. AdaBoost yönteminde de Boosting kelimesinin anlamında olduğu gibi zayıf olan sınıflandırıcıların güçlendirilerek sınıflandırma başarısının artırılması amaçlanmaktadır. Topluluk öğrenme (Ensemble Learning) yöntemlerinin temellerine dayanmaktadır ve sınıflandırma problemi için birden fazla öğrenici ile eğitim gerçekleştirilir (Zhou, 2012). AdaBoost yönteminde başlangıçta tüm verilere eşit ağırlık verilir. Daha sonra oluşturulan sınıflandırıcılar arasından zayıf sınıflandırıcı seçilir (Aydın ve Aslan, 2017). Zayıf sınıflandırıcı her çalışmasında veri ağırlıklarını güncellemektedir. Yani bir önceki sınıflandırıcıda yanlış olarak sınıflandırılmış verilerin ağırlıkları arttırılarak bir sonraki sınıflandırıcının oluşmasında gerekli olan veri kümesinde seçilme olasılığı artar. Böylece doğru sınıflandırma yapılabilmesi amaçlanmaktadır. Daha sonra oluşturulan tüm sınıflandırıcılar birleştirilerek sınıflandırma tamamlanır (Zhou, 2012).

3.4.5. Light Gradyan Artırma (Light Gradient Boosting, LightGBM)

LightGBM topluluk öğrenme (ensemble learning) yöntemlerindendir. Verilerin son yıllarda hızla artması algoritmaların daha hızlı çalışmasını ve donanımsal ihtiyaçlarının daha aza indirilmesi amacıyla Microsoft araştırmacıları tarafından 2017 yılında öne sürülen ağaç tabanlı bir makine öğrenme algoritmasıdır (Ke; ve ark., 2017). Birçok karar ağacı tabanlı algoritma seviye odaklı (level-wise) büyüme strateji kullanırken, LightGBM karar ağacı tabanlı olmasına rağmen yaprak odaklı (leaf-wise) büyüme gerçekleştirerek eğitimi tamamlamaktadır. Ayrıca karar ağaçlarının eğitimi sırasında LightGBM iki özgün teknik kullanmaktadır. Bunlar radyan tabanlı tek yönlü örnekleme (GOSS) ve ayrıcalıklı özellik desteleme (EFB) algoritmalarıdır. Gradyan tabanlı tek yönlü örnekleme algoritmasının kullanılmasındaki amaç veri kümesinin tamamının kullanılması yerine veri sayısının azaltılarak alt veri kümelerinin kullanılmasını sağlamaktır. Ayrıcalıklı özellik desteleme algoritması ile karmaşıklığı azaltabilmek için seyrek yapıdaki özellikleri birleştirerek daha az sayıda yoğun özellikler oluşturmaktadır (Üstüner ve Şanlı, 2019). LightGBM'de kullandığı algoritmalar ve benimsediği yöntemlerden dolayı algoritmanın hızlı çalışması amaçlanmaktadır (Ke; ve ark., 2017).

3.5. Özellik Seçme

Özellik seçimi (feature selection) veri seti içerisinde bulunan her bir özelliği belirlenen puanlama sistemine göre puanladıktan sonra en iyi k tane özelliğin seçilmesi işlemine özellik seçme denilmektedir (Forman, 2003). Diğer bir tanımıyla veri setinin orijinal halini, en iyi temsil edebilecek alt kümelerin belirlenmesi olarak tanımlanmaktadır ve özellik seçime işleminin temel amacı öğrenme sürecine katkı sağlayacak en yararlı özelliklerin seçilmesidir (Budak, 2018). Bir veri kümesi içerinde N adet özellik bulunuyorsa ayrıntılı arama yaklaşımıyla (exhaustive search approach) 2^N-1 tane alt küme elde edilebilmektedir (Hacıbeyoğlu, 2012). En iyi alt kümeyi belirlemek için tüm alt kümelerin tek tek denenmesi zaman, bellek karmaşıklığının artası gibi sebeplerden dolayı gerçek hayat problemleri için çoğu zaman mümkün görülememektedir. Bu nedenle özellik seçme yöntemleri sıklıkla tercih edilmektedir. Özellik seçme işlemi öğrenme sürecinde bazı avantajlar sağlamaktadır. Bunlar:

- Veri setinin saklanabilmesi için gerekli olan bellek miktarını azaltmaktadır (Ladha ve Deepa, 2011).
- Veri setindeki özelliklerin sayısının azalması algoritma hızını artırmaktadır (Ladha ve Deepa, 2011).
- Veri seti içerisindeki gerek duyulmayan, ilgili olmayan veya gürültülü olan verilerin kaldırılmasını sağlar (Ladha ve Deepa, 2011).
- Veri setinin kalitesinin artmasına yardımcı olur (Ladha ve Deepa, 2011).
- Öğrenme algoritmalarının hızını artırmaktadır (Ladha ve Deepa, 2011).
- Öğrenme modellerinin doğruluk oranını artırmaktadır (Ladha ve Deepa, 2011).

Özellik seçme işleminin sağladığı avantajlardan dolayı birçok alanda kullanılmaktadır. Örneğin Saldırı tespit sistemleri, görüntü işleme, kötü amaçlı yazılım tespit sistemleri vs. dir. Şekil 3.11.'de sıklıkla tercih edilen özellik seçme yöntemleri görülmektedir.

Şekil 3.11. Sıklıkla tercih edilen özellik seçme yöntemleri (Saeys ve ark., 2007)

Filtreleme (filter) yöntemleri kullanılan eski bir özellik seçme yöntemlerindendir. Yöntemde bilgi, tutarlılık, uzaklık gibi istatistiksel yöntemler kullanılmaktadır. Veri seti içerisindeki her bir özelliğe seçilen filtreleme yöntemi uygulanır ve bir değer (skor) elde edilmektedir. En fazla değere sahip özellikler en iyi özellik alt kümesini oluşturmaktadır (Budak, 2018). Filtreleme yöntemleri sınıflandırma algoritmasına bağımlı değildir ve genellikle hızlı bir şekilde işlem yapmaktadır.

Sarmal (wrapper) yöntemler, farklı öğrenme algoritmalarının kullanılarak gerçekleştirilen yöntemlerdir. Oluşturulan alt nitelik grupları sınıflandırma algoritmaları üzerinde denenerek en iyi özellikleri bulma yöntemidir (Gümüşçü ve ark., 2016). Sarmal yöntemler, filtreleme yöntemlerine göre daha iyi alt kümeyi belirleyebilirken hesaplama maliyetini arttırmaktadır (Budak, 2018).

Gömülü (embedded) yöntemler, yapısı itibari ile içerisinde hem sınıflandırma hem de özellik seçme işlemini eş zamanlı bir şekilde gerçekleştirmektedir. Gömülü yöntemlerde, filtreleme yöntemlerine göre daha çok hesaplama maliyeti gerektirmektedir (Budak, 2018).

Bu tez çalışmasında özellik seçme yöntemi olarak filtreleme yöntemlerinden olan Ki-Kare Testi ve Bilgi Kazancı yöntemleri kullanılmıştır.

3.5.1. Ki-Kare Testi

Ki-Kare özellik seçme yöntemi istatistiksel yöntem temelli olup yaygın bir şekilde kullanılmaktadır. Ki-Kare formülü Denklem 3.5.'de verilmiştir.

$$X^2 = \sum_{i=1}^{n} \frac{(O_i - e_i)^2}{e_i} \quad (3.5)$$

n: Veri kümesindeki özellik sayısı

O_i : i'inci özellik için gözlenen frekans değeri

e_i : i'inci özellik için beklenen frekans değeri

Veri seti içerisinden Ki-Kare ile özellik seçimi yapılmak istendiğinde her bir özellik için hesaplanan X^2 değerine göre büyükten küçüğe doğru sıralama gerçekleştirilir ve en üstten başlanarak belirlenen sayıda özellik seçilmektedir (Budak, 2018).

3.5.1. Bilgi Kazancı

Bilgi kazancı (Information Gain) özellik seçim yöntemi entropi kavramının temellerine dayandırılmaktadır. Sistemdeki düzensizliğin veya belirsizliğin ölçüsüne entropi denilmektedir. Entropi değeri sıfır ile bir arasında değer almaktadır. Sıfır değeri sistemin düzenli olduğunu ifade ederken, değer 1'e doğru yaklaştıkça sistem belirsizliği artmaktadır. Belirsizlik yani entropi değeri yüksekse bilgi miktarı fazladır denilmektedir. Denklem 3.6.'da entropinin formülü verilmiştir.

$$H(Y) = -\sum_{y \in Y} p(y) log_2 (p(y)) \quad (3.6)$$

$H(Y)$: Gruba ait entropi değerini belirtmektedir.

$p(y)$: Belirli bir sınıfa ait oranı belirtmektedir.

Bilgi kazancı yöntemi veri seti içerisindeki en belirleyici özelliklerin tespit edilmesinde kullanılmaktadır. Bilgi kazancının formülü Denklem 3.7.'de ve Denklem 3.8.'de gösterilmiştir.

$$H(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y/x) \, log_2(p(y|x)) \quad (3.7)$$

$$Bilgi\ Kazancı = H(Y) - H(Y|X) \quad (3.8)$$

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez çalışmasında Windows telemetri bilgileri ile kötü amaçlı yazılım tahmini yapılmıştır. Bu amaçla Windows telemetri bilgileri içeren Microsoft Kötü Amaçlı Yazılım Tahmini veri seti kullanılmıştır. Bu veri seti üzerinde Ki-Kare ve Bilgi Kazancı yöntemleri kullanılarak özellik seçme işlemi uygulanmış daha sonra ise Naive Bayes, Karar Ağacı, Random Forest, Adaboost ve LightGBM sınıflandırma algoritmaları test edilmiştir. Ayrıca bu sınıflandırma algoritmaları Lin (2019) çalışmasında kullandığı veri kümesi üzerinde test edilmiştir.

Bu bölümde testin gerçekleştiği ortam, veri ön işlemede kullanılan yöntemler, algoritmaların parametreleri, performans değerlendirme metrikleri, seçilen K-Turlu Çapraz Doğrulama yöntemi, seçilen yöntem mimarisi ve sonuçlara yer verilmiştir.

4.1. Testin Gerçekleştirildiği Ortam ve Özellikleri

Veri setlerine uygulan işlemler pandas, numpy, lightgbm, sklearn, time gibi kütüphaneler kullanılarak python (version 3.6.3) programlama dili ile gerçekleştirilmiştir. Python'nın ücretsiz ve geniş bir kütüphanelere sahip olması sebebi ile python programlama dili tercih edilmiştir. Editör olarak Visual Studio Code (version 1.47.3) kullanılmıştır. Testin gerçekleştirildiği sistem özellikleri Çizelge 4.1'de verilmiştir.

Çizelge 4.1. Sistem özellikleri

İşlemci	Intel(R) Xeon(R) CPU E5-2630 v4@ 2.20 GHz (2 işlemci)
İşlemci Tabanı	64 bit işletim sistemi, x64 tabanlı işlemci
İşlemci Max Performans	%100
RAM	24GB
İşletim Sistemi	Windows 10 Pro

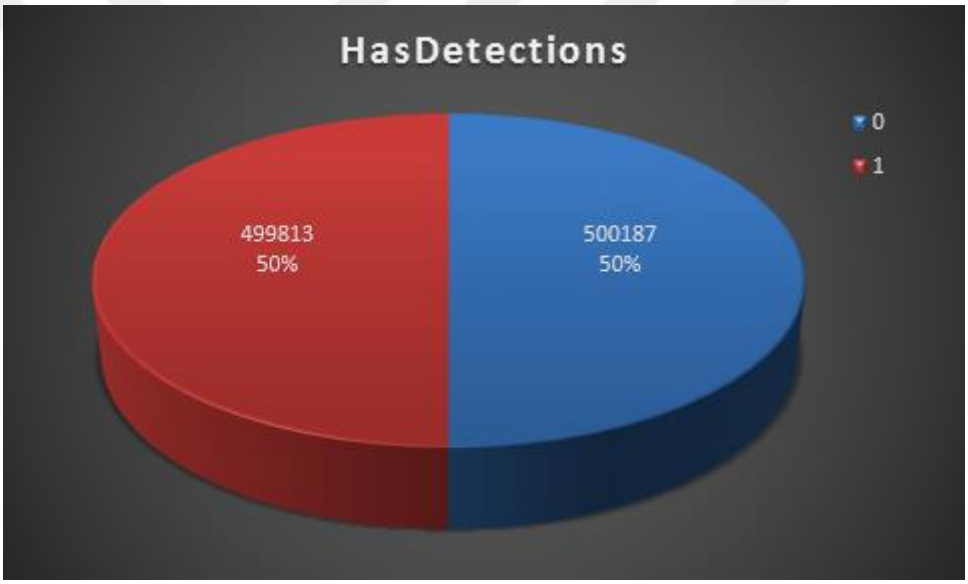
4.2. Veri Ön İşleme

Veri setinin bir sınıflandırıcı model ile sınıflandırılabilmesi için veri setinin uygun olması gerekmektedir. Bu sebeple sınıflandırma işlemine geçmeden önce veri setine bir takım işlemler uygulanmıştır.

Microsoft Kötü Amaçlı Yazılım Tahmini veri seti içerisinde 44 farklı özellikte kayıp değer bulunmaktadır. Kayıp Özelliklerin tamamlanabilmesi için özellik içerisinde en çok tekrar eden (Mod) değer bulunarak eksik olan değerler tamamlamıştır. Daha

sonrasında veri seti içerinde bulanan sayısal olmayan değerler, sayısal hale dönüştürülmüş yani encode edilmiştir. Ayrıca farklı değer aralığında olan değerleri aynı aralığa toplayabilmek amacıyla Min-Max normalizasyon işlemi uygulanmıştır.

Microsoft Kötü Amaçlı Yazılım Tahmini veri seti eğitim ve test olarak iki veri kümesinden oluşmaktadır. Veri setinin, test veri kümesindeki son özellik olan HasDetections özelliği belirtilmediğinden yani sınıf etiketi olmaksızın paylaşıldığından dolayı gerçekleştirilen tüm deneyler eğitim veri kümesi üzerinde yapılmıştır. Deneylerin yapıldığı sistem kısıtlamalarından dolayı eğitim veri kümesi içeride bulunan yaklaşık 9 milyon veri satırından başından başlanarak 1 milyon veri satırı belirlenmiştir. Oluşturulan veri seti üzerinde test işlemleri gerçekleştirilmiştir. Oluşturulan veri setinin sınıf dağılımı Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Bir milyon veri içeren veri kümesinin sınıf etiketi dağılımı oranı

Veri setinin sınıflandırma başarısını artırabilmek, sınıflandırma algoritmasının çalışma süresini kısaltabilmek amacıyla Bilgi Kazancı ve Ki-Kare özellik seçme yöntemleri uygulanmıştır. Her bir özellik seçme yöntemi ile 70,60 ve 50 özellik seçilerek sınıflandırma algoritmaları ile test edilmiştir.

4.3. Algoritma Parametreleri

Microsoft Kötü Amaçlı Yazılım Tahmini veri seti Naive Bayes, Karar Ağacı, Random Forest, Adaboost, LightGBM sınıflandırma algoritmaları ile test edilmiştir. Python Scikit-learn kütüphanesinden faydalanılarak Naive Bayes, Karar Ağacı, Random

Forest, Adaboost algoritmaları ile sınıflandırma işlemi yapılmıştır. Gerçekleştirilen testlerde algoritmaların parametre seçimi kullanım sayfaları dikkate alınarak çoğunlukla varsayılan değerleri kullanılmıştır.

LightGBM algoritmasının kendine özgü çekirdek kontrol parametreleri, öğrenme kontrol parametreleri gibi yüzden fazla kontrol parametresi bulunmaktadır. Microsoft tarafından ücretsiz olarak sağlanan Python paketi kullanılmıştır. LightGBM ait parametre seçimi LightGBM belgelerine dayanılarak varsayılan parametreleri kullanılmıştır (Microsoft, 2020b). Çizelge 4.2.'de kullanılan algoritmaların parametreleri gösterilmiştir. Belirtilmeyen parametre değerleri için varsayılan değerleri kullanılmıştır.

Çizelge 4.2. Kullanılan algoritma parametreleri

Algoritmanın İsmi	**Parametreleri**
Naive Bayes	BernoulliNB türü kullanılmıştır.
Karar Ağacı	criterion=gini (scikit-learn, CART algoritmasının optimize edilmiş sürümü)
Random Forest	n_estimators = 150, random_state=0
Adaboost	n_estimators = 150, random_state=0
LightGBM	n_estimators=150, learning_rate=0.1, num_leaves=31, boosting_type= "gbrt"

4.4. Performans Değerlendirme Metrikleri ve K-Turlu Çapraz Doğrulama

Sınıflandırma algoritmalarının, sınıflandırma performansını değerlendirebilmek amacıyla karışıklık matrisi (Confusion Matrix) diğer bir adıyla hata matrisi kullanılmaktadır (Gürmen, 2020). Karışıklık matrisinin anlaşılabilir ve kolay bir yöntem olması sebebi ile kötü amaçlı yazılım tespitinde sıklıkla tercih edilmektedir. Bu çalışma içerisinde kullanılan algoritmaların performansını değerlendirebilmek amacıyla karışıklık matrisinden yararlanılmıştır. Karışıklık matrisi sınıflandırıcı tarafından bulunan sonuçlar ile gerçek veriler arasındaki ilişkiyi inceleyerek sınıflandırıcı performansının değerlendirilmesine yardımcı olmaktadır. Sınıflandırma etiketinin iki kategorili olduğu bir veri kümesinin karışıklık matrisi Şekil 4.2'deki gibi gösterilmektedir.

	Tahmin Değeri: Hayır (0)	Tahmin Değeri: Evet (1)
Gerçek Değeri : Hayır (0)	TN	FP
Gerçek Değeri : Evet (1)	FN	TP

Şekil 4.2. İki kategorili karışıklık matrisi

TP (Gerçek Pozitifler): Sınıflandırılacak verinin gerçek değerinin 1 (evet) iken seçilen sınıflandırma modeli veriyi 1 (evet) olarak tahmin etmesidir. Doğru bir değerlendirme olarak kabul edilmektedir.

TN (Gerçek Negatifler): Sınıflandırılacak verinin gerçek değerinin 0 (hayır) yani kötü amaçlı değil veya saldırı değil iken sınıflandırma modelinin de veriyi doğru tahmin edilerek 0 (hayır) olarak sınıflandırmasıdır. Doğru bir değerlendirme olarak kabul edilmektedir.

FP (Yanlış Pozitifler): Sınıflandırılacak verinin gerçek değerinin 0 (hayır) iken seçilen sınıflandırma modeli veriyi 1 (evet) olarak tahmin etmesidir. Yanlış bir değerlendirme olarak kabul edilmektedir.

FN (Yanlış Negatifler): Sınıflandırılacak verinin gerçek değerinin 1 (evet) yani kötü amaçlı veya saldırı olan verinin, sınıflandırma modeli tarafından yanlış bir tahminde bulunularak kötü amaçlı değil veya saldırı değil yani 0 (hayır) olarak sınıflandırılmasıdır. Yanlış bir değerlendirme olarak kabul edilmektedir.

Karışıklık matrisinin temel özelliklerinden faydalanılarak yapılan tüm denemelerde Doğruluk (Accuracy), Hassasiyet (Precision), Duyarlılık (Recall), F-Ölçütü (F1-Score), ROC eğrisi altında kalan alan (ROC Area Under the Curve-Roc_Auc) değerlendirme kriterleri hesaplanmıştır.

Doğruluk (Accuracy): Seçilen sınıflandırıcı modelinin verileri hangi oranda doğru sınıflandırdığını gösterir. Doğruluk yani sınıflandırma başarısı Denklem 4.1.'de gösterilmiştir.

$$\text{Doğruluk (Accuracy)} = (TP+TN)/(TP+TN+FP+FN) \quad (4.1)$$

Hassasiyet (Precision): Doğru evet olarak tahmin edilen verilerin, toplamdaki evet olarak tahmin edilenlerin oranı hassasiyeti göstermektedir. Hassasiyet Denklemi 4.2.'de gösterilmiştir.

Hassasiyet (Precision) = (TP)/(TP+FP) (4.2)

Duyarlılık (Recall): Seçilen sınıflandırıcının gerçekte evet olanlarının hangi oranda tespit edebildiğini göstermektedir. Karışıklık matrisindeki yanlış negatifler (FN), yanlış pozitif (FP) oranından daha kritik bir öneme sahiptir (Şimşek, 2018). Örneğin Korona virüs sonucu pozitif olan bir kişinin negatif olarak tanı konulması yerine, gerçek sonucu negatif olan bir kişiye yanlış tahminde bulunularak önlem alınmasını sağlamak daha makul bir davranış olacaktır. Testler için önemli bir oran olan duyarlılık, Denklem 4.3'deki gibi hesaplanabilmektedir.

Duyarlılık (Recall) = (TP)/(TP+FN) (4.3)

F-Ölçütü (F1-Score): Seçilen sınıflandırma modelinin hassasiyet ve duyarlılık değerlerinin harmonik ortalaması F-ölçütünü oluşturmaktadır. F-Ölçütünün hesaplanması Denklem 4.4.'te gösterilmiştir.

F Ölçütü = (2*Precision*Recall) / (Precision+Recall) (4.4)

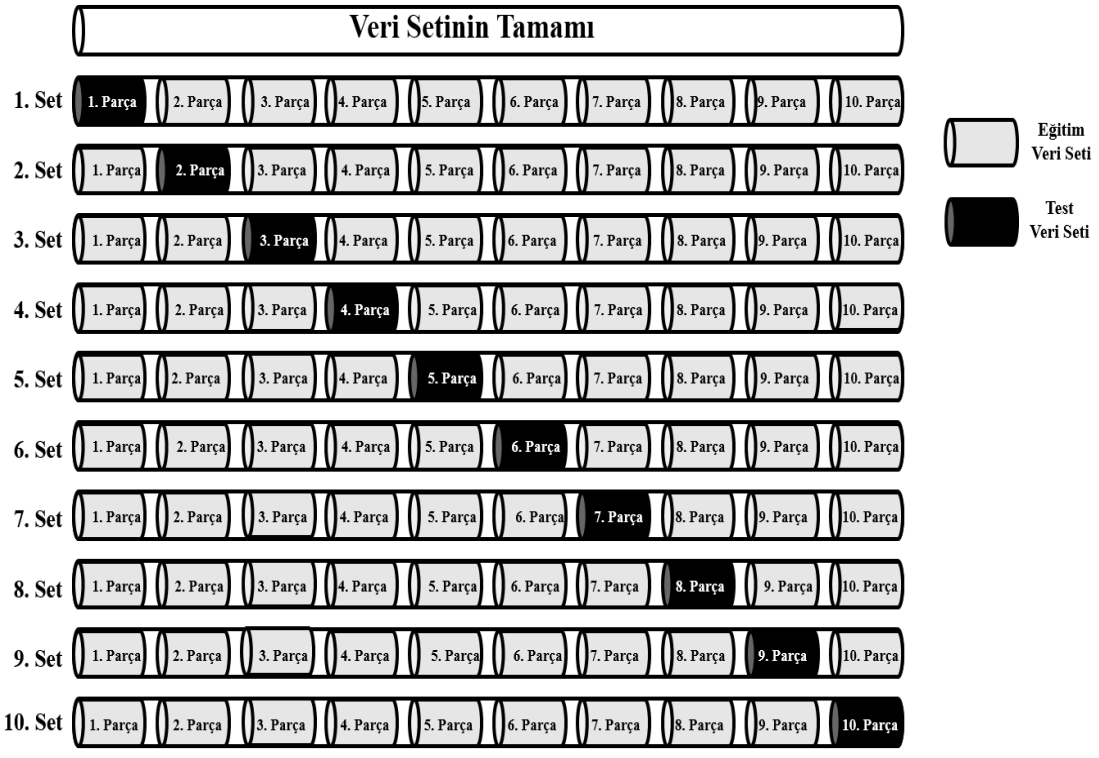
ROC eğrisi altında kalan alan (ROC Area Under the Curve-Roc_Auc): Yapılan testlerin performansı ROC (Receiver Operating Characteristic) eğrisi altında kalan alanın miktarı kullanılarak değerlendirilmektedir. ROC eğrisi iki boyutlu bir grafiktir. ROC eğrisi y-ekseni olan gerçek pozitif oranının, x-ekseni olan yanlış pozitif oranına karşı çizilen grafik olarak tanımlanabilmektedir. Denklem 4.5. y-ekseni ve Denklem 4.6'de x-ekseninin hesaplaması gösterilmiştir.

Doğru pozitif (TPR - y ekseni) = TP / (TP + FN) (4.5)
Yanlış pozitif (FPR – x ekseni) = FP / (TN + FP) (4.6)

Makine öğrenmesi yöntemlerinin test sırasındaki hataları tespit edebilmek için veri setinin test ve eğitim kümesi olarak bölünmesi gerekmektedir. Bu bölme işleminin

gerçekleşmesi için birçok farklı yöntem bulunmaktadır. Bu çalışmada Python Scikit-learn kütüphanesinin sağladığı K-turlu çapraz doğrulama (Stratified K-Fold) kullanılmıştır. Literatürde 10 turlu çapraz doğrulama sıklıkla tercih edilmesi sebebi ile bu çalışmada da 10 turlu çapraz doğrulama kullanılmıştır (Krzysztof J. Cios ve ark., 2007).

K-turlu çapraz doğrulama belirlenen k sayısına eşit olarak veri setini eşit parçalara bölmektedir. İlk turda 1. veri kümesi test için kullanılırken geriye kalan k-1 parça veri kümesi eğitim için kullanılmaktadır. İkinci turda 2. veri kümesi test için kullanılırken geriye kalan kümeler eğitim için kullanılmaktadır. Bu işlem belirlenen k sayısınca tekrar etmektedir. Şekil 4.3.'de 10 turlu bir çapraz doğrulamanın çalışma yapısı gösterilmiştir. Son kısımda ise çıkan sonuçların ortalaması alınarak sınıflandırıcının performansı belirlenmektedir. K-turlu çapraz doğrulama kullanılarak sınıflandırıcıların performansları hakkında tutarlı sonuçların elde edilmesi sağlanmaktadır. Stratified K-Fold yöntemi de K-turlu çapraz doğrulama yöntemi gibi çalışmaktadır. Ancak K-turlu çapraz doğrulama veri setindeki veri sınıf dağılımını dikkate almazken Stratified K-Fold yöntemi veri setinin genel sınıf dağılımı dikkate alarak veri setlerini oluşturmaktadır. Stratified K-Fold yönteminin n_splits=10 olarak belirlenirken geriye kalan parametreleri için varsayılan değerleri kullanılmıştır.



Şekil 4.3. K-Turlu çapraz doğrulmanın genel yapısı

4.5. Test İşlemleri

Telemetri bilgisi ile kötü amaçlı yazılım tahmini için Microsoft Kötü Amaçlı Yazılım Tahmini veri seti kullanılmıştır. Veri seti içerisinde kayıp değeri bulunan özellikler tespit edilmiştir. Kayıp değeri bulunan 44 farklı özelliğin kayıp oranı belirlenmiştir. Kayıp değer oranı %70'den, %50'den ve %30'dan fazla olan özelliklerin isimleri belirlenmiştir. Çizelge 4.3.'de kayıp oranlarına göre veri ön işleme adımından sonra elenecek özellik isimleri gösterilmiştir.

Çizelge 4.3. Kayıp oranı %70, %50 ve %30'dan fazla olan özellik isimleri

Kayıp Oranı %70'den Fazla Olan Özellik İsimleri	Kayıp Oranı %50'den Fazla Olan Özellik İsimleri	Kayıp Oranı %30'den Fazla Olan Özellik İsimleri
PuaMode	PuaMode	PuaMode
Census_ProcessorClass	Census_ProcessorClass	Census_ProcessorClass
DefaultBrowsersIdentifier	DefaultBrowsersIdentifier	DefaultBrowsersIdentifier
Census_IsFlightingInternal	Census_IsFlightingInternal	Census_IsFlightingInternal
Census_InternalBatteryType	Census_InternalBatteryType	Census_InternalBatteryType
	Census_ThresholdOptIn	Census_ThresholdOptIn
	Census_IsWIMBootEnabled	Census_IsWIMBootEnabled
		SmartScreen
		OrganizationIdentifier

Microsoft Kötü Amaçlı Yazılım Tahmini veri seti kayıp oranlarına göre özellikler çıkarılmadan (drop edilmeden) önce eksik değere sahip tüm özellikler tamamlanmıştır. Özelliklerin tamamlanması, özellikteki en çok tekrar eden değere (mod işlemine) göre gerçekleştirilmiştir. Eksik değere sahip özelliklerin, tamamlandığı değerler Çizelge 4.4.'de gösterilmiş.

Çizelge 4.4. Eksik değere sahip özelliklerin, tamamlandığı değerler

Özelliğin Adı	Özelliğin Frekansı (Mod)
PuaMode	on
Census_ProcessorClass	mid
DefaultBrowsersIdentifier	239
Census_IsFlightingInternal	0
Census_InternalBatteryType	lion
Census_ThresholdOptIn	0
Census_IsWIMBootEnabled	0
SmartScreen	RequireAdmin
OrganizationIdentifier	27
SMode	0
CityIdentifier	130775

Wdft_IsGamer	0
Wdft_RegionIdentifier	10
Census_InternalBatteryNumberOfCharges	0
Census_FirmwareManufacturerIdentifier	142
Census_IsFlightsDisabled	0
Census_FirmwareVersionIdentifier	33105
Census_OEMModelIdentifier	313586
Census_OEMNameIdentifier	2668
Firewall	1
Census_TotalPhysicalRAM	4096
Census_IsAlwaysOnAlwaysConnectedCapable	0
Census_OSInstallLanguageIdentifier	8
IeVerIdentifier	137
Census_PrimaryDiskTotalCapacity	476940
Census_SystemVolumeTotalCapacity	28542
Census_InternalPrimaryDiagonalDisplaySizeInInches	15,5
Census_InternalPrimaryDisplayResolutionHorizontal	1366
Census_InternalPrimaryDisplayResolutionVertical	768
Census_ProcessorModelIdentifier	2697
Census_ProcessorManufacturerIdentifier	5
Census_ProcessorCoreCount	4
AVProductStatesIdentifier	53447
AVProductsInstalled	1
AVProductsEnabled	1
IsProtected	1
RtpStateBitfield	7
Census_IsVirtualDevice	0
Census_PrimaryDiskTypeName	HDD
UacLuaenable	1
Census_ChassisTypeName	Notebook
GeoNameIdentifier	277
Census_PowerPlatformRoleName	Mobile
OsBuildLab	17134.1.amd64fre.rs4_release.180410-1804

Veri setindeki eksik verilerin tamamlanmasının ardından veriler encode edilmiştir. Daha sonrasında ise Min-Max normalizasyon işlemi uygulanmıştır. Başlangıçta belirlenen kayıp oranlarına göre (Çizelge 4.3.) veri setlerinin özellikleri kayıp oranı %70'den fazla olan veri setinde (Miss70) 77 özellik, kayıp oranı %50'den fazla olan veri setinde (Miss50) 75 özellik ve kayıp oranı %30'dan fazla olan veri setinde (Miss30) 73 özellik bulunacak şekilde belirlenmiştir. Her bir veri setinde dolayı 1 milyon veri satırı kullanılmıştır. Her bir veri seti Naive Bayes, Karar Ağacı, Random Forest, Adaboost, LightGBM sınıflandırma algoritmaları ve 10 turlu çapraz doğrulama kullanılarak test

edilmiştir. 10 turlu çapraz doğrulamanın ortalama değerleri ve algoritmaların çalışma zamanlarının sonuçları Çizelge 4.5'de verilmiştir. Sonuçların detayları Ek1'de verilmiştir. Çizelge 4.5'de göre sınıflandırma algoritmaları içerisinde en başarılı algoritma %65.39 oran ile LightGBM algoritması olmuştur. En başarısız sınıflandırmayı yapan sınıflandırma algoritması ise Karar Ağacı olmuştur. En hızlı çalışabilen algoritma BernoulliNB sınıflandırma algoritması olmasına rağmen başarı oranı LightGBM algoritmasına göre düşük olduğu görülmüştür. Eksik verileri içeren özellikler elenirken belli bir orana kadar başarı oranında değişme olmazken özellik sayısı azaltıldığında başarı oranı düştüğü görülmüştür.

Çizelge 4.5. Eksik değer oranına göre veri setlerinin test sonuçları

%70'den Fazla Kayıp Bulunan özellikleri atılması ile oluşan veri seti (Miss70) 10 Turlu Çapraz Doğrulama Ortalama Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6539	0,6557	0,6476	0,6516	0,7169	468,6
AdaBoost	0,6360	0,6325	0,6486	0,6405	0,6937	12431,5
RandomForest	0,6474	0,6532	0,6279	0,6403	0,7072	19698,4
DecisionTree	0,5710	0,5707	0,5721	0,5714	0,5710	830,7
BernoulliNB	0,5887	0,5775	0,6599	0,6160	0,6119	169,6
%50'den Fazla Kayıp Bulunan özellikleri atılması ile oluşan veri seti (Miss50) 10 Turlu Çapraz Doğrulama Ortalama Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6539	0,6556	0,6480	0,6517	0,7169	664,3
AdaBoost	0,6360	0,6325	0,6486	0,6405	0,6937	13293,3
RandomForest	0,6470	0,6527	0,6277	0,6400	0,7072	21649,0
DecisionTree	0,5716	0,5713	0,5730	0,5721	0,5716	848,4
BernoulliNB	0,5887	0,5775	0,6599	0,6160	0,6119	166,8
%30'den Fazla Kayıp Bulunan özellikleri atılması ile oluşan veri seti (Miss30) 10 Turlu Çapraz Doğrulama Ortalama Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6354	0,6276	0,6653	0,6459	0,6878	482,0
AdaBoost	0,6169	0,6035	0,6809	0,6399	0,6604	13045,5
RandomForest	0,6271	0,6243	0,6379	0,6310	0,6762	21755,7
DecisionTree	0,5527	0,5524	0,5537	0,5531	0,5527	805,7
BernoulliNB	0,5887	0,5775	0,6599	0,6160	0,6119	163,3

Microsoft Kötü Amaçlı Yazılım Tahmini veri setinde %70'den fazla kayıp değeri bulunan özellikler çıkartılarak oluşturulan veri setinin (Miss70) her bir özelliğine Bilgi Kazancı ve Ki-Kare işlemine göre skorlama yapılarak 70,60 ve 50 özellik seçilerek tekrar

test işlemi yapılmıştır. Çizelge 4.6'da 10 turlu çapraz doğrulamanın ortalama değerleri ve algoritmaların çalışma zamanlarının sonuçları verilmiştir.

Çizelge 4.6. %70'den fazla kayıp bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) bilgi kazancı ve ki-kare skoruna göre test sonuçları

%70'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Bilgi Kazancına Göre 70 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6539	0,6556	0,6479	0,6518	0,7169	710
AdaBoost	0,6360	0,6325	0,6488	0,6405	0,6936	12546
RandomForest	0,6471	0,6532	0,6268	0,6397	0,7070	21234
DecisionTree	0,5713	0,5709	0,5726	0,5718	0,5713	807
BernoulliNB	0,5888	0,5775	0,6605	0,6162	0,6121	158
%70'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Bilgi Kazancına Göre 60 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6540	0,6558	0,6476	0,6517	0,7168	1006
AdaBoost	0,6357	0,6322	0,6484	0,6402	0,6935	11917
RandomForest	0,6469	0,6532	0,6260	0,6393	0,7070	20741
DecisionTree	0,5714	0,5710	0,5728	0,5719	0,5714	766
BernoulliNB	0,5887	0,5777	0,6587	0,6155	0,6116	134
%70'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Bilgi Kazancına Göre 50 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6533	0,6553	0,6465	0,6509	0,7161	676
AdaBoost	0,6354	0,6315	0,6496	0,6404	0,6931	10922
RandomForest	0,6457	0,6520	0,6244	0,6379	0,7055	21319
DecisionTree	0,5711	0,5707	0,5724	0,5715	0,5711	694
BernoulliNB	0,5895	0,5770	0,6692	0,6197	0,6103	112
%70'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Ki-Kare Skoruna Göre 70 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6540	0,6560	0,6471	0,6515	0,7169	1015
AdaBoost	0,6357	0,6322	0,6484	0,6402	0,6935	12455
RandomForest	0,6475	0,6535	0,6273	0,6402	0,7071	20962
DecisionTree	0,5717	0,5714	0,5726	0,5720	0,5717	808
BernoulliNB	0,5887	0,5775	0,6599	0,6160	0,6119	159
%70'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Ki-Kare Skoruna Göre 60 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6533	0,6555	0,6455	0,6505	0,7163	707
AdaBoost	0,6357	0,6317	0,6502	0,6408	0,6933	11484
RandomForest	0,6462	0,6528	0,6239	0,6380	0,7061	20027
DecisionTree	0,5706	0,5701	0,5729	0,5715	0,5706	725
BernoulliNB	0,5886	0,5774	0,6597	0,6158	0,6116	137
%70'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Ki-Kare Skoruna Göre 50 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6524	0,6544	0,6453	0,6499	0,7150	832
AdaBoost	0,6352	0,6315	0,6488	0,6400	0,6927	9352
RandomForest	0,6446	0,6514	0,6218	0,6362	0,7038	19079
DecisionTree	0,5713	0,5709	0,5724	0,5717	0,5713	570
BernoulliNB	0,5886	0,5777	0,6581	0,6153	0,6114	122

Microsoft Kötü Amaçlı Yazılım Tahmini veri setinde %70'den fazla kayıp değeri bulunan özellikler çıkartılarak oluşturulan veri setinde Bilgi Kazancı yönteminde 60 özellik Ki-Kare skorlama yönteminde 70 özellik seçildiğinde ve LightGBM algoritması ile sınıflandırıldığında en yüksek başarı oranı olan %64.40 değerini vermektedir.

Microsoft Kötü Amaçlı Yazılım Tahmini veri setinde %50'den fazla kayıp değeri bulunan özellikler çıkartılarak oluşturulan veri setinin (Miss50) her bir özelliğine Bilgi Kazancı ve Ki-Kare işlemine göre skorlama yapılarak 70,60 ve 50 özellik seçilerek test işlemi yapılmıştır. Çizelge 4.7'da 10 turlu çapraz doğrulamanın ortalama değerleri ve algoritmaların çalışma zamanlarının sonuçları verilmiştir.

Çizelge 4.7. %50'den fazla kayıp bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) bilgi kazancı ve ki-kare skoruna göre test sonuçları

%50'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) Bilgi Kazancına Göre 70 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6539	0,6556	0,6479	0,6518	0,7169	445
AdaBoost	0,6360	0,6325	0,6488	0,6405	0,6936	12489
RandomForest	0,6471	0,6532	0,6268	0,6397	0,7070	21660
DecisionTree	0,5707	0,5703	0,5718	0,5711	0,5707	829
BernoulliNB	0,5888	0,5775	0,6605	0,6162	0,6121	158
%50'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) Bilgi Kazancına Göre 60 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6540	0,6558	0,6476	0,6517	0,7168	463
AdaBoost	0,6357	0,6322	0,6484	0,6402	0,6935	11864
RandomForest	0,6469	0,6532	0,6260	0,6393	0,7070	21136
DecisionTree	0,5713	0,5708	0,5729	0,5719	0,5713	773
BernoulliNB	0,5887	0,5777	0,6587	0,6155	0,6116	136
%50'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) Bilgi Kazancına Göre 50 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6533	0,6553	0,6465	0,6509	0,7161	479
AdaBoost	0,6354	0,6315	0,6496	0,6404	0,6931	10897
RandomForest	0,6457	0,6520	0,6244	0,6379	0,7055	21792
DecisionTree	0,5705	0,5701	0,5714	0,5708	0,5705	693
BernoulliNB	0,5895	0,5770	0,6692	0,6197	0,6103	112
%50'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) Ki-Kare Skoruna Göre 70 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6539	0,6557	0,6477	0,6517	0,7170	675
AdaBoost	0,6360	0,6325	0,6486	0,6405	0,6937	12932
RandomForest	0,6475	0,6535	0,6271	0,6401	0,7074	21756
DecisionTree	0,5712	0,5709	0,5725	0,5717	0,5712	966
BernoulliNB	0,5888	0,5775	0,6599	0,6160	0,6119	292
%50'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) Ki-Kare Skoruna Göre 60 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6533	0,6555	0,6455	0,6505	0,7163	1313
AdaBoost	0,6357	0,6317	0,6502	0,6408	0,6933	11879
RandomForest	0,6462	0,6528	0,6239	0,6380	0,7061	20705

DecisionTree	0,5708	0,5704	0,5725	0,5715	0,5708	731
BernoulliNB	0,5886	0,5774	0,6597	0,6158	0,6116	137
%50'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss50) Ki-Kare Skoruna Göre 50 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6524	0,6544	0,6453	0,6499	0,7150	1314
AdaBoost	0,6352	0,6315	0,6488	0,6400	0,6927	9686
RandomForest	0,6446	0,6514	0,6218	0,6362	0,7038	20041
DecisionTree	0,5713	0,5711	0,5720	0,5715	0,5713	580
BernoulliNB	0,5886	0,5777	0,6581	0,6153	0,6114	123

Microsoft Kötü Amaçlı Yazılım Tahmini veri setinde %50'den fazla kayıp değeri bulunan özellikler çıkartılarak oluşturulan veri setinde, en yüksek başarı oranı Çizelge 4.7'ye göre Bilgi Kazancı yöntem ile 60 özellik seçildiğinde %65.40 doğruluk oranı olmuştur. Aynı çizelgeye göre LightGBM algoritmasından sonra en iyi sınıflandırıcı Random Forest algoritması olmuştur. Bu test sonuçlarına bakılarak Random Forest sınıflandırma algoritmasının çalışabilmesi için daha çok süreye ihtiyaç duyulduğu görülmektedir.

Microsoft Kötü Amaçlı Yazılım Tahmini veri setinde %30'den fazla kayıp değeri bulunan özellikler çıkartılarak oluşturulan veri setinin (Miss30) her bir özelliğine Bilgi Kazancı ve Ki-Kare işlemine göre skorlama yapılarak 70,60 ve 50 özellik seçilerek test işlemi yapılmıştır. Çizelge 4.8'de 10 turlu çapraz doğrulamanın ortalama değerleri ve algoritmaların çalışma zamanlarının sonuçları verilmiştir.

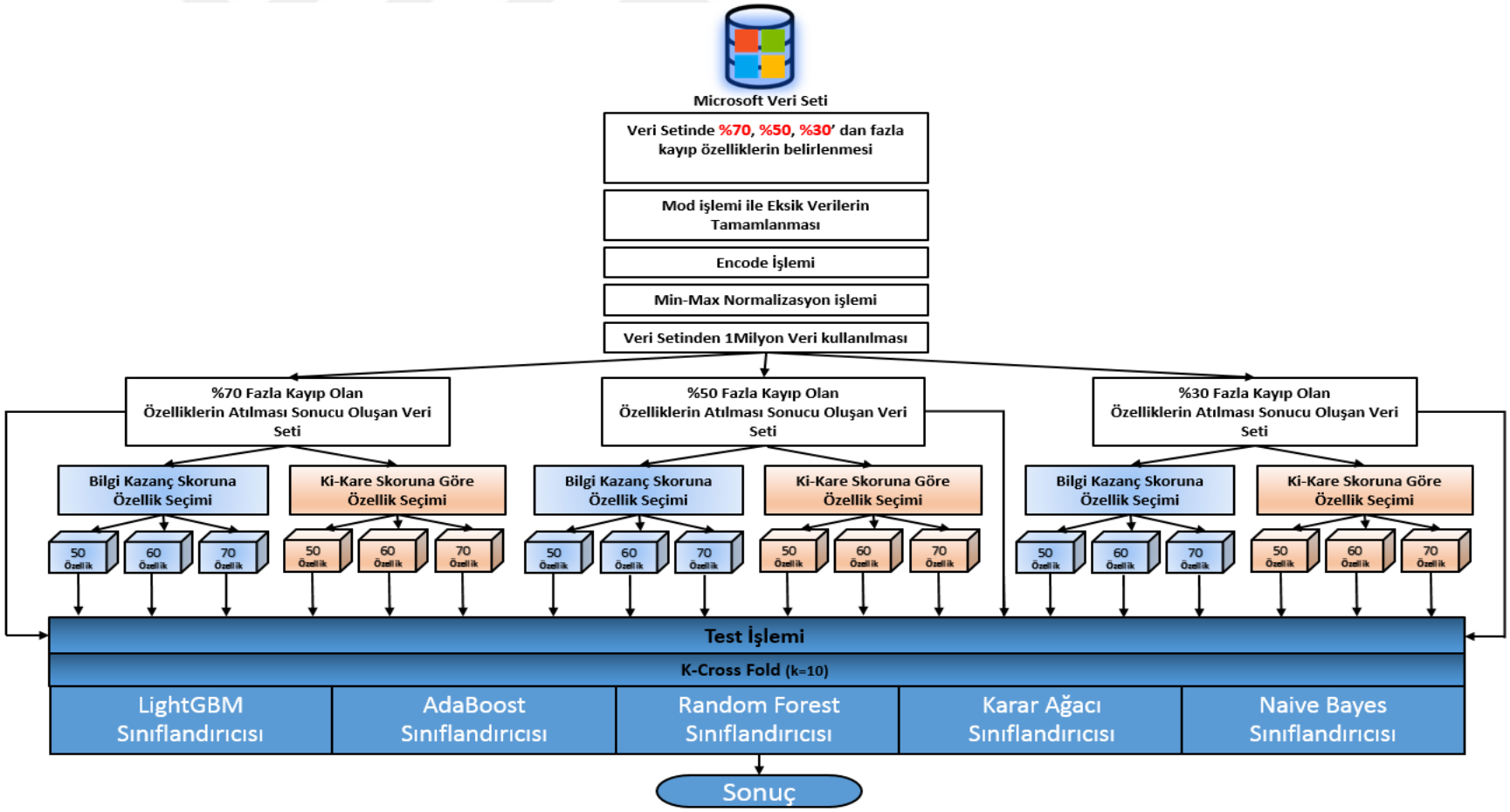
Çizelge 4.8. %30'den fazla kayıp bulunan özelliklerin atılması ile oluşan veri setinin (Miss30) bilgi kazancı ve ki-kare skoruna göre test sonuçları

%30'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Bilgi Kazancına Göre 70 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6354	0,6276	0,6653	0,6459	0,6878	971
AdaBoost	0,6169	0,6035	0,6809	0,6399	0,6604	13608
RandomForest	0,6275	0,6247	0,6379	0,6312	0,6765	21668
DecisionTree	0,5529	0,5526	0,5536	0,5531	0,5529	802
BernoulliNB	0,5887	0,5775	0,6599	0,6160	0,6119	160
%30'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Bilgi Kazancına Göre 60 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6354	0,6277	0,6652	0,6459	0,6877	514
AdaBoost	0,6169	0,6035	0,6809	0,6399	0,6604	12113
RandomForest	0,6273	0,6248	0,6368	0,6307	0,6763	22579
DecisionTree	0,5532	0,5530	0,5536	0,5533	0,5532	741
BernoulliNB	0,5887	0,5777	0,6588	0,6155	0,6117	137
%30'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss70) Bilgi Kazancına Göre 50 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6344	0,6268	0,6636	0,6447	0,6865	799
AdaBoost	0,6165	0,6026	0,6832	0,6404	0,6598	11738

RandomForest	0,6267	0,6242	0,6357	0,6299	0,6752	23253
DecisionTree	0,5526	0,5524	0,5532	0,5528	0,5526	1737
BernoulliNB	0,5894	0,5772	0,6676	0,6191	0,6102	114
%30'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss30) Ki-Kare Skoruna Göre 70 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6353	0,6275	0,6653	0,6459	0,6877	1427
AdaBoost	0,6169	0,6035	0,6809	0,6399	0,6604	12882
RandomForest	0,6278	0,6249	0,6387	0,6318	0,6766	30438
DecisionTree	0,5524	0,5521	0,5537	0,5529	0,5524	775
BernoulliNB	0,5887	0,5775	0,6599	0,6160	0,6119	160
%30'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss30) Ki-Kare Skoruna Göre 60 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6345	0,6271	0,6628	0,6445	0,6867	1458
AdaBoost	0,6167	0,6031	0,6819	0,6401	0,6601	12026
RandomForest	0,6268	0,6243	0,6365	0,6303	0,6754	24249
DecisionTree	0,5519	0,5516	0,5531	0,5523	0,5519	718
BernoulliNB	0,5886	0,5774	0,6597	0,6158	0,6116	143
%30'den Fazla Kayıp Bulunan özelliklerin atılması ile oluşan veri setinin (Miss30) Ki-Kare Skoruna Göre 50 özelliğinin Seçilerek Yapılan Test Sonuçları						
Algoritma	**Accuracy**	**Precision**	**Recall**	**F1-Skor**	**Roc_Auc**	**Time (sn)**
LightGBM	0,6340	0,6265	0,6630	0,6442	0,6856	1213
AdaBoost	0,6161	0,6025	0,6816	0,6396	0,6596	9646
RandomForest	0,6246	0,6230	0,6307	0,6268	0,6724	20017
DecisionTree	0,5539	0,5536	0,5548	0,5542	0,5539	579
BernoulliNB	0,5886	0,5776	0,6582	0,6153	0,6114	125

Microsoft Kötü Amaçlı Yazılım Tahmini veri setinde %30'den fazla kayıp değeri bulunan özellikler çıkartılarak oluşturulan veri seti, diğer veri setlerine göre daha az başarı elde ediyor olması elenen özelliklerin sınıflandırma başarısı için önemli olduğu görülmektedir. Bu veri seti için yapılan test işlemlerin en yüksek başarı bilgi kazancı ile 60 özellik seçilip LightGBM algoritması ile test edildiğinde %63.54 oranında başarı elde edilmektedir.

Microsoft Kötü Amaçlı Yazılım Tahmini veri setine uygulanan işlemlerin genel şeması Şekil 4.4'de gösterilmiştir.



Şekil 4.4. Uygulanan test işlemlerinin genel yapısı

Lin'in (2019) yaptığı çalışmada Microsoft Kötü Amaçlı Yazılım Tahmini veri setinin 71110 verisini eğitim için 22638 verisini test için ve veri setinden GBM ile önce 20 özellik seçmiş daha sonrasında ise aynı algoritma ile sınıflandırma işlemi uygulamıştır. Yapılan deneyde %63.69 oranında bir başarı elde etmiştir (Lin, 2019).

Lin'in (2019) yaptığı çalışmada kullandığı veri seti bu tez çalışmasında kullanılan Naive Bayes, Karar Ağacı, Random Forest, Adaboost ve LightGBM test edilmiştir. Test sonuçları Çizelge 4.9'da gösterilmiştir.

Çizelge 4.9. Lin'in çalışmasında (Lin, 2019) kullandığı veri seti gerçekleştirilen test sonuçları

Algoritma	Accuracy	Precision	Recall	F1-Skor	Roc_Auc	Time (sn)
LightGBM	0,65571	0,65571	0,65571	0,65571	0,57804	0,875205
AdaBoost	0,65651	0,65651	0,65651	0,65651	0,58351	12,062656
RandomForest	0,65063	0,65063	0,65063	0,65063	0,58235	31,109360
DecisionTree	0,55168	0,55168	0,55168	0,55168	0,52299	1,327910
BernoulliNB	0,39169	0,39169	0,39169	0,39169	0,50000	0,125210

Yapılan bu deneyde elde edilen maksimum başarı oranı %65.65 ile Adaboost algoritması olurken hemen ardından %65.57 başarı oranı ile LightGBM gelmektedir. Bu iki algoritmanın aralarındaki başarı oranın az olması ve LightGBM algoritmasının AdaBoost algoritmasına göre daha hızlı çalışması sebebi ile büyük verilerin söz konusu olduğu durumlarda LightGBM algoritmasının daha verimli olacağı düşünülmektedir. Ayrıca LightGBM, Adaboost ve Random Forest algoritmalarının GBM'den daha iyi sınıflandırma yapabildiği görülmektedir.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Teknolojinin sürekli olarak gelişmesi internet üzerinde var olan cihaz sayısında sürekli olarak artırmaktadır. İnternet kullanımının yayınlaşması birçok bilgi güvenliği açığının oluşmasına sebebiyet vermektedir. Oluşan bilgi güvenliği açıklarından faydalanan saldırganlar sistemleri kötü amaçları için kullanmaktadırlar. Saldırganların kullandığı güvenlik açıkları bazen kullanıcılardan kaynaklanırken bazen de sistemlerin güncellenmesinden kaynaklanabilmektedir. Sonuç olarak bu güvenlik açıklarından faydalanan kişiler kurumlara, kuruluşlara, insanlara ya da toplumlara maddi ve manevi zararlar vermektedir.

Bu tez çalışmasında kötü amaçlı yazılım tahmini yapılmıştır. Amaç Windows işletim sistemine sahip bir sistemin, sadece telemetri bilgilerinden faydalanılarak sistemde kötü amaçlı yazılım var olup olmadığının tahmin edilmesidir. Kullanılan veri seti Microsoft firması tarafından oluşturulan Microsoft Kötü Amaçlı Yazılım Tahmini veri setidir. Kötü amaçlı yazılım tahmini için Bilgi Kazancı ve Ki-Kare yöntemleri ile özellik seçme işlemleri uygulanmış ve Naive Bayes, Karar Ağacı, Random Forest, Adaboost, LightGBM yöntemleri olmak üzere beş farklı sınıflandırma algoritması sınıflandırılmıştır.

Çalışmada kullanılan özellik seçme metotlarından olan Bilgi Kazancı ve Ki-Kare yöntemleri aynı özellik sayısına sahip veri setlerinde test edilirken doğruluk oranları bakımından neredeyse aynı sonuçlar verdiği gözlemlenmiştir. Ancak seçilen özellik sayısının azaltılması özelikle de %30'dan fazla kayıp bulunan özelliklerin atılması sırasında "SmartScreen" özelliği de atıldığı için oluşan veri setinin test edilmesinde başarı oranının düştüğü gözlemlenmiştir.

Yapılan test işlemlerinde kullanılan sınıflandırma algoritmaları arasında genel olarak en hızlı çalışan algoritma Naive Bayes algoritması olmasına karşın başarı oranı LightGBM algoritmasına göre daha az olduğu görülmüştür. Topluluk öğrenme algoritmalarının daha iyi sonuç verdiği ve bu algoritmalar arasındaki en hızlı ve başarılı algoritmanın LightGBM algoritması olduğu gözlemlenmiştir.

Lin'in (2019) çalışmasında (Lin, 2019) kullandığı veri seti ile yapılan çalışmalar da LightGBM, Random Forest ve AdaBoost yöntemlerinin GBM'den daha başarılı olduğu görülmüştür.

İnternete bağlanan cihaz sayısının artması kötü amaçlı yazılım tespiti için taranması gereken cihaz sayısını arttırmıştır. Sistemlerin telemetri bilgisi ile kötü amaçlı yazılım tahmini yapılabilir ve taranması gereken bilgisayar sayısı azaltılabilir. Böylece bilgi güvenliği daha hızlı sağlanmış olacaktır. Gelecek çalışmalar için Microsoft kötü amaçlı yazılım tahmini veri seti derin öğrenme kullanılarak sınıflandırılabilir.

KAYNAKLAR

Aggarwal, C. C., 2015, Data mining: the textbook, Springer, p.

Akçetin, E. ve Çelik, U., 2014, İstenmeyen Elektronik Posta (Spam) Tespitinde Karar Ağacı Algoritmalarının Performans Kıyaslaması, *Journal of Internet Applications & Management/İnternet Uygulamaları ve Yönetimi Dergisi*, 5 (2).

Aslan, Ö., 2017, Performance comparison of static malware analysis tools versus antivirus scanners to detect malware, *International Multidisciplinary Studies Congress (IMSC)*.

Aydın, F. ve Aslan, Z., 2017, Yapay öğrenme yöntemleri ve dalgacık dönüşümü kullanılarak nöro dejeneratif hastalıkların teşhisi, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 32 (3), 749-766.

Betancourt, S. E., 2019, Microsoft Malware Prediction Challenge in the Cloud.

Breiman, L., 1996, Bagging predictors, *Machine learning*, 24 (2), 123-140.

Breiman, L., 2001, Random forests, *Machine learning*, 45 (1), 5-32.

Budak, H., 2018, Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım, *Journal of Natural & Applied Sciences*.

Ceylan, M., 2018, Zararlı yazılım analizi için lab ortamı hazırlamak, https://www.bgasecurity.com/makale/zararli-yazilim-analizi-icin-lab-ortami-hazirlamak/: [01 Mart 2020].

Chu, W., 2015, Should you disable Windows 10 telemetry?, https://www.neweggbusiness.com/smartbuyer/windows/should-you-disable-windows-10-telemetry/: [18 Mart 2020].

CyberMag, 2020, CyberMag, 2020 IBM araştırması: türkiyede bir veri ihlalinin ortalama maliyeti 12,3 milyon tl.

Çayır, A., Ünal, U., Yenidoğan, I. ve Dağ, H., 2019, Use Case Study: Data Science Application for Microsoft Malware Prediction Competition on Kaggle, *Proceedings Book*, 98.

Erol, S. E. ve Sağıroğlu, Ş., 2018, Siber güvenlik farkındalığı,farkındalık ölçüm yöntem ve modelleri. Siber Güvenlik ve Savunma FARKINDALIK ve CAYDIRICILIK. Sağıroğlu, Ş. ve Alkan, M. Ankara. 1**:** 105-141.

Forman, G., 2003, An extensive empirical study of feature selection metrics for text classification, *Journal of machine learning research*, 3 (Mar), 1289-1305.

Gümüşçü, A., Aydilek, İ. B. ve RamazanTaşaltın, 2016, 3 farklı filtre modelli öznitelik seçme algoritmalarının kombine edilerek iyileştirilmesi, *AfyonKocatepeÜniversitesiFenveMühendislikBilimleriDergisi*, 16, 31-35.

Gürmen, C., 2020, Saldırı tespit sistemleri için makine öğrenme yöntemlerinin performans karşılaştırması, Yüksek Lisans, *Harran Üniversitesi*, Şanlıurfa.

Hacıbeyoğlu, M., 2012, Bilgi sistemlerinde fark fonksiyonu tabanlı özellik seçme yöntemlerinin geliştirilmesi, Doktora Tezi, *Selçuk Üniversitesi*, Konya.

Han, J., Park, J., Chung, H. ve Lee, S., 2020, Forensic analysis of the Windows telemetry for diagnostics, *arXiv preprint arXiv:2002.12506*.

KasperskyLab, 2019, Dijital tehlike alanı: kaspersky lab; ortadoğu, türkiye ve afrika'daki siber güvenlik trendlerine ışık tuttu, https://www.kaspersky.com.tr/about/press-releases/2019_digital-hazard-area-kaspersky-lab-middle-east-sheds-light-on-cyber-security-trends-in-turkey-and-africa: [04.Ağustos.2020].

Kavzoğlu, T. ve Çölkesen, İ., 2010, Karar ağaçları ile uydu görüntülerinin sınıflandırılması: Kocaeli örneği, *Harita Teknolojileri Elektronik Dergisi*, 2 (1), 36-45.

Ke;, G., Meng;, Q., Finley;, T., Wang;, T., Chen;, W., Ma;, W., Ye;, Q. ve Liu;, T.-Y., 2017, Lightgbm: A highly efficient gradient boosting decision tree, *Advances in neural information processing systems*, 3146-3154.

Krzysztof Cios, Witold Pedrycz, Roman Swiniarski ve Lukasz Andrzej Kurgan, 2007, Data mining: a knowledge discovery approach, Springer Science & Business Media, p. 388.

Krzysztof J. Cios, Witold Pedrycz, Roman W Swiniarski ve Lukasz Andrzej Kurgan, 2007, Data mining: a knowledge discovery approach, Springer Science & Business Media, p. 388.

Kumar, N. R., 2016, Random Forest based Classification, https://www.youtube.com/watch?v=ajTc5y3OqSQ: [20 Ağustos 2020].

KurtKaya, G. D., 2017, Bilgi güvenliği ve siber güvenlik kapsamında bakanlık uygulamaları için güvenli yazılım geliştirme metodolojisi önerisi, Uzmanlık Tezi, *Çevre ve Şehircilik Bakanlığı*, Ankara, 4-8.

Ladha, L. ve Deepa, T., 2011, Feature selection methods and algorithms, *International journal on computer science and engineering*, 3 (5), 1787-1797.

Lin, C., 2019, Naive Transfer Learning Approaches for Suspicious Event Prediction, *2019 IEEE International Conference on Big Data (Big Data)*, 5897-5901.

MathWorks, 2020, Machine learning in MATLAB, https://www.mathworks.com/help/stats/machine-learning-in-matlab.html: [2 Mayıs 2020].

Microsoft, 2018, Microsoft Malware Prediction.

Microsoft, 2020a, Configure Windows diagnostic data in your organization, https://docs.microsoft.com/tr-tr/windows/privacy/configure-windows-diagnostic-data-in-your-organization: [8 Ağustos 2020].

Microsoft, 2020b, LightGBM's documentation-parameters, https://lightgbm.readthedocs.io/en/latest/Parameters.html: [25 Haziran 2020].

NormaTürk, 2016, Bilgi güvenliği nedir? ne işe yarar?, http://blog.normaturk.com/bilgi-guvenligi-nedir/: [21 Şubat 2020].

Pesen, M. M., 2015, Bilgi güvenliği nedir ve nasıl sınıflandırılır, http://www.sibergah.com/genel/bilgi-guvenligi-nedir-ve-nasil-siniflandirilir/: [23 Şubat 2020].

Saeys, Y., Inza, I. ve Larrañaga, P., 2007, A review of feature selection techniques in bioinformatics, *bioinformatics*, 23 (19), 2507-2517.

Sağıroğlu, Ş., 2018, Siber güvenlik ve savunma: önem, tanımlar, unsurlar ve önlemler, In: Siber Güvenlik ve Savunma FARKINDALIK ve CAYDIRICILIK, Eds: SAĞIROĞLU, Ş. ve ALKAN, M., 1, *Ankara*: Grafiker Yayıncılık, p. 21-46.

Samet, R. ve Aslan, Ö., 2018, Kötü amaçlı yzılımlar ve analizi, In: Siber Güvenlik ve Savunma FARKINDALIK ve CAYDIRICILIK, Eds: Sağıroğlu, Ş. ve Alkan, M., 1, *Ankara*: Grafiker Yayınları, p. 223-256.

Samuel, A. L., 1959, Some studies in machine learning using the game of checkers, *IBM Journal of research and development*, 3 (3), 210-229.

Sikorski, M. ve Honig, A., 2012, Practical malware analysis: the hands-on guide to dissecting malicious software, no starch press, p. 1-5.

Singh, A., Vaish, A. ve Keserwani, P. K., 2014, Information security: components and techniques, *International Journal*, 4 (1).

Şimşek, H. K., 2018, Sınıflandırma modellerinde başarı kriterleri, https://medium.com/data-science-tr/s%C4%B1n%C4%B1fland%C4%B1rma-modellerinde-ba%C5%9Far%C4%B1-kriterleri-2d86488799c6: [27 Haziran 2020].

Şirincan, M., 2016, Tersine mühendislik, https://msirincan.wordpress.com/2016/04/06/tersine-muhendislik/: [26 Şubat 2020].

TÜBİTAK-SAGE, 2020, TÜBİTAK SAGE Mobil telemetri sistemi http://www.sage.tubitak.gov.tr/tr/hizmetlerimiz/mobil-telemetri-test-hizmetleri: [13 Mart 2020].

Üstüner, M. ve Şanlı, F. B., 2019, Çok zamanlı polarimetrik SAR verileri ile tarımsal ürünlerin sınıflandırılması, *Jeodezi ve Jeoinformasyon Dergisi*, 1-10.

VanderPlas, J., 2017, Python data science handbook, oreilly, p. 331-333.

WeAreSocial ve Hootsuite, 2020, Digital 2020 global digital overview

Wikipedia, 2019, Malware, https://tr.wikipedia.org/wiki/Malware: [25 Şubat 2020].

Wikipedia, 2020, Creeper (program), https://en.wikipedia.org/wiki/Creeper_(program): [9 Ağustos 2020].

Yeboah-Ofori, A. ve Boachie, C., 2019, Malware Attack Predictive Analytics in a Cyber Supply Chain Context Using Machine Learning, *2019 International Conference on Cyber Security and Internet of Things (ICSIoT)*, 66-73.

Yıldırım, Y., 2018, Denetimsiz öğrenme (Unsupervised learning), https://yavuz.github.io/denetimsiz-ogrenme/: [01 Mayıs 2020].

Yumak, B., 2011, Elektronik postaların ayrıştırılmasında naive bayesian ve bulanık mantık yöntemlerinin karşılaştırıması, Yüksek Lisans, *Gazi Üniversite*, 21-27.

Zhou, Z.-H., 2012, Ensemble methods: foundations and algorithms, CRC press, p.

EK1

%70'den Fazla Kayıp Bulunan Özellikleri Atılması İle Oluşan Veri Setinin (Miss70) Test Sonuçları

%70'den Fazla Kayıp Bulunan özellikleri atılması ile oluşan veri seti (Miss70) Sonuçları											
Test_Accuracy											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6514	0,6558	0,6550	0,6521	0,6533	0,6555	0,6540	0,6550	0,6554	0,6517	0,6539
AdaBoost	0,6334	0,6371	0,6370	0,6339	0,6363	0,6385	0,6381	0,6355	0,6357	0,6348	0,6360
RandomForest	0,6456	0,6475	0,6483	0,6455	0,6477	0,6487	0,6479	0,6470	0,6492	0,6466	0,6474
DecisionTree	0,5708	0,5717	0,5692	0,5707	0,5692	0,5737	0,5728	0,5688	0,5725	0,5707	0,5710
BernoulliNB	0,5863	0,5914	0,5895	0,5878	0,5872	0,5880	0,5904	0,5892	0,5900	0,5876	0,5887
Test_Precision											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6524	0,6578	0,6566	0,6543	0,6545	0,6573	0,6569	0,6568	0,6570	0,6537	0,6557
AdaBoost	0,6288	0,6345	0,6330	0,6309	0,6320	0,6355	0,6352	0,6316	0,6326	0,6312	0,6325
RandomForest	0,6503	0,6530	0,6543	0,6514	0,6535	0,6551	0,6540	0,6526	0,6549	0,6526	0,6532
DecisionTree	0,5701	0,5711	0,5695	0,5703	0,5689	0,5736	0,5722	0,5685	0,5723	0,5704	0,5707
BernoulliNB	0,5751	0,5799	0,5778	0,5768	0,5762	0,5770	0,5793	0,5778	0,5786	0,5768	0,5775
Test_Recall											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6475	0,6488	0,6495	0,6445	0,6487	0,6493	0,6441	0,6488	0,6498	0,6447	0,6476
AdaBoost	0,6509	0,6462	0,6512	0,6447	0,6518	0,6487	0,6481	0,6499	0,6464	0,6482	0,6486
RandomForest	0,6292	0,6287	0,6284	0,6254	0,6281	0,6274	0,6276	0,6282	0,6300	0,6265	0,6279
DecisionTree	0,5745	0,5743	0,5661	0,5722	0,5704	0,5734	0,5757	0,5697	0,5724	0,5720	0,5721
BernoulliNB	0,6595	0,6625	0,6631	0,6580	0,6584	0,6587	0,6589	0,6612	0,6614	0,6570	0,6599
Test_F1											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6499	0,6533	0,6530	0,6494	0,6516	0,6533	0,6504	0,6528	0,6534	0,6492	0,6516
AdaBoost	0,6396	0,6403	0,6420	0,6378	0,6418	0,6420	0,6416	0,6406	0,6394	0,6396	0,6405
RandomForest	0,6396	0,6406	0,6411	0,6381	0,6406	0,6410	0,6405	0,6402	0,6422	0,6393	0,6403
DecisionTree	0,5723	0,5727	0,5678	0,5713	0,5696	0,5735	0,5739	0,5691	0,5724	0,5712	0,5714
BernoulliNB	0,6144	0,6185	0,6175	0,6147	0,6145	0,6151	0,6166	0,6167	0,6172	0,6143	0,6160
Test_Roc_Auc											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,7149	0,7183	0,7178	0,7136	0,7166	0,7185	0,7179	0,7177	0,7189	0,7144	0,7169
AdaBoost	0,6913	0,6947	0,6950	0,6904	0,6929	0,6958	0,6946	0,6944	0,6955	0,6921	0,6937
RandomForest	0,7059	0,7079	0,7074	0,7045	0,7073	0,7083	0,7088	0,7078	0,7092	0,7051	0,7072
DecisionTree	0,5708	0,5717	0,5692	0,5707	0,5692	0,5737	0,5728	0,5688	0,5725	0,5707	0,5710
BernoulliNB	0,6080	0,6151	0,6134	0,6112	0,6099	0,6126	0,6148	0,6110	0,6128	0,6106	0,6119

Time	
Algoritma	**sn**
LightGBM	468,6
AdaBoost	12431,5
RandomForest	19698,4
DecisionTree	830,7
BernoulliNB	169,6

%50'den Fazla Kayıp Bulunan Özellikleri Atılması İle Oluşan Veri Setinin (Miss50) Test Sonuçları

%50'den Fazla Kayıp Bulunan özellikleri atılması ile oluşan veri seti (Miss50) Sonuçları											
Test_Accuracy											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6517	0,6555	0,6552	0,6516	0,6530	0,6555	0,6540	0,6550	0,6553	0,6520	0,6539
AdaBoost	0,6334	0,6371	0,6370	0,6339	0,6363	0,6385	0,6381	0,6355	0,6357	0,6348	0,6360
RandomForest	0,6445	0,6468	0,6493	0,6447	0,6463	0,6482	0,6475	0,6484	0,6471	0,6475	0,6470
DecisionTree	0,5721	0,5730	0,5702	0,5713	0,5715	0,5735	0,5732	0,5684	0,5731	0,5701	0,5716
BernoulliNB	0,5863	0,5914	0,5895	0,5878	0,5872	0,5880	0,5904	0,5892	0,5900	0,5876	0,5887
Test_Precision											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6530	0,6575	0,6565	0,6539	0,6543	0,6573	0,6569	0,6568	0,6561	0,6534	0,6556
AdaBoost	0,6288	0,6345	0,6330	0,6309	0,6320	0,6355	0,6352	0,6316	0,6326	0,6312	0,6325
RandomForest	0,6490	0,6530	0,6552	0,6507	0,6521	0,6545	0,6539	0,6537	0,6523	0,6531	0,6527
DecisionTree	0,5715	0,5725	0,5703	0,5710	0,5712	0,5730	0,5725	0,5679	0,5729	0,5697	0,5713
BernoulliNB	0,5751	0,5799	0,5778	0,5768	0,5762	0,5769	0,5793	0,5778	0,5786	0,5768	0,5775
Test_Recall											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6472	0,6488	0,6504	0,6436	0,6484	0,6493	0,6441	0,6488	0,6522	0,6470	0,6480
AdaBoost	0,6509	0,6462	0,6512	0,6447	0,6518	0,6487	0,6481	0,6499	0,6464	0,6482	0,6486
RandomForest	0,6287	0,6259	0,6299	0,6241	0,6266	0,6271	0,6260	0,6305	0,6293	0,6288	0,6277
DecisionTree	0,5756	0,5750	0,5680	0,5720	0,5717	0,5754	0,5767	0,5702	0,5733	0,5723	0,5730
BernoulliNB	0,6596	0,6625	0,6632	0,6580	0,6584	0,6587	0,6589	0,6613	0,6614	0,6570	0,6599
Test_F1											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6501	0,6531	0,6534	0,6487	0,6513	0,6533	0,6504	0,6528	0,6541	0,6502	0,6517
AdaBoost	0,6396	0,6403	0,6420	0,6378	0,6418	0,6420	0,6416	0,6406	0,6394	0,6396	0,6405
RandomForest	0,6387	0,6392	0,6423	0,6371	0,6391	0,6405	0,6396	0,6419	0,6406	0,6407	0,6400
DecisionTree	0,5735	0,5738	0,5692	0,5715	0,5715	0,5742	0,5746	0,5691	0,5731	0,5710	0,5721
BernoulliNB	0,6144	0,6185	0,6176	0,6147	0,6145	0,6151	0,6166	0,6167	0,6172	0,6143	0,6160
Test_Roc_Auc											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,7150	0,7181	0,7180	0,7133	0,7165	0,7185	0,7179	0,7177	0,7189	0,7150	0,7169
AdaBoost	0,6913	0,6947	0,6950	0,6904	0,6929	0,6958	0,6946	0,6944	0,6955	0,6921	0,6937
RandomForest	0,7057	0,7074	0,7082	0,7048	0,7070	0,7084	0,7079	0,7082	0,7089	0,7055	0,7072
DecisionTree	0,5721	0,5730	0,5702	0,5713	0,5715	0,5735	0,5732	0,5684	0,5731	0,5701	0,5716
BernoulliNB	0,6080	0,6151	0,6134	0,6112	0,6099	0,6126	0,6148	0,6110	0,6128	0,6106	0,6119

Time	
Algoritma	**sn**
LightGBM	664,3
AdaBoost	13293,3
RandomForest	21649,0
DecisionTree	848,4
BernoulliNB	166,8

%30'den Fazla Kayıp Bulunan Özellikleri Atılması İle Oluşan Veri Setinin (Miss30) Test Sonuçları

%30'den Fazla Kayıp Bulunan özellikleri atılması ile oluşan veri seti (Miss30) Sonuçları											
Test_Accuracy											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6332	0,6379	0,6361	0,6332	0,6372	0,6359	0,6378	0,6334	0,6347	0,6343	0,6354
AdaBoost	0,6161	0,6197	0,6163	0,6145	0,6168	0,6174	0,6188	0,6164	0,6168	0,6162	0,6169
RandomForest	0,6258	0,6296	0,6286	0,6269	0,6262	0,6273	0,6283	0,6255	0,6270	0,6262	0,6271
DecisionTree	0,5529	0,5500	0,5537	0,5540	0,5534	0,5545	0,5520	0,5517	0,5526	0,5524	0,5527
BernoulliNB	0,5863	0,5914	0,5895	0,5878	0,5872	0,5880	0,5904	0,5892	0,5900	0,5876	0,5887
Test_Precision											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6248	0,6301	0,6282	0,6259	0,6293	0,6283	0,6306	0,6257	0,6264	0,6261	0,6276
AdaBoost	0,6018	0,6062	0,6028	0,6014	0,6032	0,6043	0,6060	0,6030	0,6039	0,6024	0,6035
RandomForest	0,6226	0,6268	0,6259	0,6239	0,6235	0,6244	0,6260	0,6227	0,6241	0,6232	0,6243
DecisionTree	0,5523	0,5496	0,5535	0,5539	0,5531	0,5542	0,5517	0,5516	0,5521	0,5522	0,5524
BernoulliNB	0,5751	0,5799	0,5778	0,5768	0,5762	0,5769	0,5794	0,5778	0,5786	0,5768	0,5775
Test_Recall											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6662	0,6670	0,6665	0,6614	0,6669	0,6647	0,6644	0,6633	0,6666	0,6657	0,6653
AdaBoost	0,6854	0,6826	0,6813	0,6781	0,6817	0,6796	0,6786	0,6806	0,6784	0,6829	0,6809
RandomForest	0,6383	0,6397	0,6388	0,6385	0,6366	0,6384	0,6366	0,6361	0,6382	0,6378	0,6379
DecisionTree	0,5566	0,5528	0,5537	0,5533	0,5546	0,5555	0,5526	0,5500	0,5552	0,5525	0,5537
BernoulliNB	0,6596	0,6626	0,6632	0,6580	0,6584	0,6587	0,6589	0,6613	0,6614	0,6570	0,6599
Test_F1											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6448	0,6480	0,6468	0,6431	0,6476	0,6460	0,6471	0,6440	0,6459	0,6453	0,6459
AdaBoost	0,6409	0,6421	0,6396	0,6375	0,6401	0,6398	0,6402	0,6395	0,6390	0,6401	0,6399
RandomForest	0,6303	0,6332	0,6323	0,6311	0,6300	0,6313	0,6313	0,6293	0,6310	0,6304	0,6310
DecisionTree	0,5545	0,5512	0,5536	0,5536	0,5538	0,5549	0,5521	0,5508	0,5536	0,5524	0,5531
BernoulliNB	0,6144	0,6185	0,6176	0,6147	0,6145	0,6151	0,6166	0,6167	0,6172	0,6143	0,6160
Test_Roc_Auc											
Algoritma	**1.Tur**	**2.Tur**	**3. Tur**	**4. Tur**	**5. Tur**	**6. Tur**	**7. Tur**	**8. Tur**	**9. Tur**	**10. Tur**	**Ortalama**
LightGBM	0,6852	0,6897	0,6889	0,6842	0,6890	0,6881	0,6903	0,6861	0,6894	0,6867	0,6878
AdaBoost	0,6575	0,6631	0,6622	0,6571	0,6595	0,6610	0,6628	0,6586	0,6623	0,6595	0,6604
RandomForest	0,6741	0,6776	0,6776	0,6741	0,6765	0,6764	0,6776	0,6763	0,6777	0,6740	0,6762
DecisionTree	0,5529	0,5500	0,5537	0,5540	0,5534	0,5545	0,5520	0,5517	0,5526	0,5524	0,5527
BernoulliNB	0,6080	0,6151	0,6134	0,6112	0,6099	0,6126	0,6148	0,6110	0,6128	0,6106	0,6119

Time	
Algoritma	**sn**
LightGBM	482,0
AdaBoost	13045,5
RandomForest	21755,7
DecisionTree	805,7
BernoulliNB	163,3

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**
Uyruğu	**:**
Doğum Yeri ve Tarihi	**:**
Telefon	**:**
Faks	**:**
E-Posta	**:**

EĞİTİM

Derece	**Adı**	**İlçe**	**İl**	**Bitirme Yılı**
Lise :				
Üniversite :				
Yüksek Lisans :				
Doktora :				

YABANCI DİLLER