
GAZİ ÜNİVERSİTESİ
Gazi
1926

ZARARLI YAZILIMLARIN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TESPİT EDİLMESİ

Fırat GÖKKİS

YÜKSEK LİSANS TEZİ
BİLGİSAYAR BİLİMLERİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

HAZİRAN 2022

Fırat GÖKKİS tarafından hazırlanan "ZARARLI YAZILIMLARIN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TESPİT EDİLMESİ" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Bilimleri Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Doç.Dr. Tahsin ÇETİNYOKUŞ
Endüstri Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ...……

Başkan: Doç.Dr. Recep BENZER
Emekli Öğretim Üyesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

……...

Üye: Doç.Dr. Murat DENER
Bilgi Güvenliği Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

……...

Tez Savunma Tarihi: 03/06/2022

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……..…….

Prof. Dr. Aslıhan TÜFEKCİ
Bilişim Enstitüsü Müdürü

ETİK BEYAN

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

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

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

Fırat GÖKKİS

03/06/2022

ZARARLI YAZILIMLARIN MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE TESPİT EDİLMESİ

(Yüksek Lisans Tezi)

Fırat GÖKKİS

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

Haziran 2022

ÖZET

Günümüzün dijital dünyasında zararlı yazılımlar ve zararlı yazılımları tespit etmek için geliştirilen güvenlik sistemleri üstünlük kurma yarışı içindedirler. Kurumsal ve bireysel bilgi sistemlerinin zararlı yazılım saldırılarından korunması için klasik tespit yöntemlerinin kabiliyeti sorgulanır hale gelmiştir. Çünkü klasik yöntemler gün geçtikçe sayısı ve karmaşıklığı artan zararlı yazılımları tespit etmekte yetersiz kalmaktadır. Zararlı yazılımlar artık imza tabanlı ve sezgisel analiz temeline dayanan güvenlik sistemlerini atlatabilecek özelliklere sahiptir. Bu durum zararlı yazılım dosyalarının tespitinde yeni yaklaşımlara başvurulmasını zorunlu hale getirmiştir. Bu yaklaşımlardan günümüzde en çok rağbet gören ve geliştirilmeye çalışılan yaklaşım makine öğrenmesidir. Makine öğrenmesinde amaçlanan, zararlı yazılımları temsil edebilecek modeller oluşturmak ve bir dosyanın zararlı olup olmadığını mümkün olan en küçük hata ile tespit edebilmektir. Oluşturulan model yüksek başarı oranına sahip ise zararlı yazılımın bir Gelişmiş Sürekli Tehdit (Advanced Persistent Threat(APT)) ya da sıfırıncı gün (zeroday) saldırısı olması fark etmeyecektir. Çalışmada, bu alanda yapılan diğer çalışmalardan farklı olarak, biri açık kaynaktan olmak üzere statik analiz ile; diğeri bu çalışmaya özgü olarak dinamik analiz ile oluşturulan iki farklı veri seti üzerinde literatür taraması neticesinde belirlenen öznitelik seçme yöntemleri ve sınıflandırma algoritmaları denenmiş, karşılaştırma yöntemi ile öznitelik seçme yöntemleri ve sınıflandırma algoritmalarının performansları analiz edilmiştir. Analiz sürecinde, Ubuntu ve Windows işletim sistemi yüklü bir fiziksel makinede iki farklı kum sandığı (Cuckoo ve ZeroWine), WEKA açık kaynak veri madenciliği programı, Windows 7 sanal makineleri ile Python programlama dili kullanılmıştır.

Bilim Kodu : 92432
Anahtar Kelimeler : Zararlı yazılım, makine öğrenmesi, sınıflandırıcı, öznitelik seçimi
Sayfa Adedi : 61
Tez Yöneticisi : Doç. Dr. Tahsin ÇETİNYOKUŞ

DETECTION OF MALWARES BY MACHINE LEARNING ALGORITHMS

(M.Sc. Thesis)

Fırat GÖKKİS

GAZİ UNIVERSITY

INFORMATICS INSTITUTE

Haziran 2022

ABSTRACT

In today's dijital world, malwares and security systems developed to detect malwares are in a challenging race to overcome each other. The capability of legacy detection systems has been questioned in protection of organizational and individual information systems since these systems are inadequate in detecting malwares of which the numbers and the complexity are increasing day by day. Malwares have characteristics to evade security systems that rely on signiture-based and heuristic analysis. This situation has made it obligatory to aly new aroaches to detect malwares. Of all the most aealing and popular one that is being improved is the machine learning. The ultimate aim of the machine learning is to produce models which are able to represent the very features of malwares and to detect if a new file is malware with the minimum error. If the model has a high accuracy rate then it will not be a deal for the model to detect if the malware is an Advanced Persistent Threat (APT) or zeroday attack. In this study, unlike the other studies in the field, feature selection methods and classification algorithms, determined as a result of literature review, are tried on two different datasets, one of which has been created through static analysis and obtained from open source and the other, specific to this study, has been created through dynamic analysis, have been analyzed via comparison method. Throughout the analysis, two different sandboxes (Cuckoo and Zerowine), an open source data mining program WEKA, Windows 7 virtual machines and Python programming language on a physical host containing Ubuntu and Windows operating systems have been employed.

Science Code : 92432
Key Words : Malware, machine learning, classifier, feature selection
Page Number : 61
Superviser : Assist. Prof. Tahsin ÇETİNYOKUŞ

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla ben yönlendiren danışman hocam Doç.Dr. Tahsin ÇETİNYOKUŞ'a, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli eşim Gülsün YILMAZ GÖKKİS'e ve zamanından çaldığım bir tanecik kızım Zeynep Alya GÖKKİS'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

1. GİRİŞ

Günümüzde siber saldırılar ve bu saldırılara karşı geliştirilen güvenlik sistemleri birbirlerine karşı üstünlük kurma yarışı içindedirler. Her gün sayısı milyonları bulan zararlı yazılımlar, çeşitli yöntemlerle kullanıcı sistemlere girerek saldırganların meşru olmayan amaçlarına hizmet etmek için kullanılmaktadır [1]. Artık her bilginin dijital olarak saklandığı ve işlendiği göz önüne alındığında, bu saldırıların geri dönüşü olmayan sonuçlar doğurabileceği yadsınamaz bir gerçek haline gelmiştir. Bununla birlikte zararlı yazılımlar, gerek maddi çıkarlar sağlamak; gerekse hedef birey, kurum ve/veya bazı durumlarda devletlerin itibarını düşürmek amacıyla gün geçtikçe daha karmaşık ve tespiti zor bir şekilde oluşturulmaktadır. İmza tabanlı (signature-based) ve sezgisel (bahviour/knowledge-based) tespit sistemleri gibi klasik güvenlik sistemleri bu gelişim karşısında yeteri kadar koruma sağlayamamaktadır [3]. Bugün polimorfik ve metamorfik özellikli bir zararlı yazılım aktif olduğu her durumda, kendi imzasını ve/veya davranışını değiştirebilmekte ve ana fonksiyonunu aynı şekilde yerine getirebilmektedir. Bu durum, zararlı yazılımları birebir şekilde tespit edecek yöntemler yerine zararlı yazılımların ortak özellik ve davranışlarını tespit edebilecek güvenlik sistemlerinin ortaya çıkmasını zorunlu kılmıştır. Bu noktada zararlı yazılım tespiti, veri madenciliği ve makine öğrenmesi süreçleri kesişmektedir [7].

Veri madenciliği ile veriler arasındaki ilişkiler ortaya koyularak, yeni girdilere göre en az hata oranı ile en doğru çıktılar elde edebilmek hedeflenmektedir. Veri madenciliği bir kurumda üretilen tüm verilerin belirli yöntemler kullanılarak var olan ya da gelecekte ortaya çıkabilecek gizli bilgiyi ortaya çıkarma sürecidir. Veri madenciliği ile makine öğrenmesi birbirinden ayrılmaz kavramlardır. Makine öğrenmesi yöntemleri ile verinin anlamlı hale getirilmesi ve işlenmiş olan veri modeli ile sisteme giren yeni verinin doğru sınıflandırılabileceği bir yapıya sahip olunması amaçlanmaktadır [2].

Çalıştırılan her program/yazılım zararlı olsun veya olmasın bulunduğu işletim sisteminde çok sayıda değişikliğe yol açar. Bu değişikliklere örnek olarak kayıt defteri kaydı, var olan bir dosyada değişiklik yapılması, yeni bir dosya oluşturulması, ağ trafiği oluşturması, ağ ayarlarında değişiklik yapılması verilebilir [8]. Bu değişiklikler, yazılımların özelliklerini içeren veriler olarak değerlendirilebilir. Elde edilen bu fazla sayıda ve farklı yapıdaki veri, doğru bir analiz yöntemi ile yazılımların zararlı veya zararsız olarak sınıflandırılmasında

etkin rol oynayabilmektedir. Bu sayede imza tabanlı tespit yöntemi ile çalışan bir anti virüs sistemince tespit edilemeyen bir zararlı yazılım, bahse konu yaklaşım ile zararlı yazılım olarak sınıflandırılabilecektir [9].

Yapılan çalışmanın amacı çok sayıda ve farklı biçimde (sayısal, metin) özellikleri bulunan zararlı yazılımları, makine öğrenmesi yaklaşımlarından olan sınıflandırma algoritmaları ile tespit edebilmektir.

Çalışmada öncelikle elde edilen ve normalize edilmiş iki farklı veri setinde bulunan özniteliklerden hangilerinin sınıf değerine daha fazla etkili olduğu Bilgi Kazanımı, Korelasyon ve Temel Bileşen analizi öznitelik seçme yöntemleri kullanılarak tespit edilmiştir. Bu sayede sınıf değerine etkisi/kazanımı daha yüksek olan özelliklerin sınıflandırıcıların performanslarını olumlu yönde etkileyip etkilemediği ortaya koyulmuştur. Sonraki adımda literatür taraması sonucunda öne çıkan Naif Bayes, Bayes Ağı, C4.5, Lojistik Regresyon, Destek Vektör Makineları ve Rastgele Orman algoritmalarının performansları öncelikle boyutu indirgenmemiş ham veri setleri üzerinde; daha sonra belirlenen öznitelik seçme yöntemleri ile boyutu indirgenmiş veri setleri üzerinde kıyaslanmıştır.

Bahse konu analizde ilk olarak statik zararlı yazılım analizi ile elde edilmiş olan ve açık kaynakta bulunan bir veri seti üzerinde, daha sonra “kelime torbası (Bag of words)” yöntemi ile sıfırdan oluşturulan diğer bir veri seti üzerinde yukarıda belirtilen yöntemler ve algoritmalar uygulanarak sonuçlar gözlemlenmiştir.

Çalışma Windows işletim sistemine uyumlu yürütülebilir dosyalar üzerinde yapılan analizleri kapsamaktadır. Farklı işletim sistemlerinde görülen zararlı yazılımlar çalışma kapsamında değildir.

2. ZARARLI YAZILIMLAR VE KULLANIM AMAÇLARI

Kötü amaçlı yazılım, tek tek bilgisayarlara veya bir ağda bulunan tüm sistemlere bulaşmak hedefiyle kötü amaçlı yazılım uygulayan kişiler tarafından oluşturulur. Zararlı yazılımlar genellikle hedef sistemde bulunan açıklıklardan faydalanılarak veya sistem yöneticilerinin (admin) yetkileri kullanılarak aktif hale getirilebilirler [6]. Zararlı yazılım, bir kurumun ağ trafiğine, süregelen hizmetine veya programlanabilir kritik bir cihazına, zarar vermek veya bunları istismar etmek üzere tasarlanmış yazılımları temsil eden terimdir. Zararlı yazılımları siber suçlular genellikle maddi kazanç odaklı kullanırlar. Bu kapsamda istismar edilmek istenen veriler finansal, tıbbi ve sistem kullanıcı bilgileri olabilmektedir [16]. Kötü amaçlı yazılım, en bilinen şekli ile virüsleri, fidye yazılımlarını ve casus yazılımları kapsar. Bunların dışında farklı amaçlarla (reklam vb.) kötü amaçlı yazılım çeşitleri de mevcuttur. Kötü amaçlı yazılımlar, siber saldırganlarca oluşturulan ve hedefi; veri bütünlüğüne, sistemin işlerliğine zarar vermek veya sisteme yetkisiz erişim sağlamak olan kodlardır [7].

Kötü amaçlı yazılımların, genellikle kullanıcıların e-postasına gönderilen bir bağlantı veya ek olarak sunulan bir dosya biçiminde sistemde yürütülmesi hedeflenmektedir. Kötü amaçlı yazılımların sistemlerde aktif hale gelebilmesi için kullanıcının zararlı dosyayı çalıştırması, linke tıklaması veya saldırganın kullanıcıya ait oturum bilgileri vb. bilgileri elde etmesi gereklidir. Kötü amaçlı yazılımlar, 1970'li yıllardan bu zamana, kurumlar ve bireyler için bir tehdit unsuru olmaktadır. Bugüne kadar, milyonlarca farklı kötü amaçlı yazılım geliştirilmiş ve çeşitli siber saldırılarda kullanılmıştır. Özellikle 2018 yılı ilk çeyreğinde, 40 milyondan fazla zararlı yazılımın ortaya çıktığı göz önüne alındığında, bu alanda çok büyük gayret sarf edildiği anlaşılabilir. Bu durum, zararlı yazılım saldırılarına karşı alınması gereken tedbirler için de aynı oranda gayretin sarf edilmesini gerekli kılmaktadır [7].

Gün geçtikçe zararlı yazılımlar daha karmaşık hale getirilmektedir. Zararlı yazılım oluşturucuları, imza tabanlı anti virüs sistemleri gibi klasik (legacy) güvenlik çözümlerini atlatmayı amaçlamaktadır. Bu durum sistemlerin ve sistemler içinde tutulan verilerin güvenliğini sağlamaya yönelik olarak farklı ve yeni çözümler bulma konusunda çalışmalar yapılmasını zorunlu hale getirmiştir [50].

Zararlı yazılımlar siber suçlular tarafından başta aşağıda sunulanlar olmak üzere başka birçok sebeple de kullanılabilmektedir:

- Kişisel verileri çalmak ve kimlik hırsızlığı
- Kredi kartı bilgilerinin veya diğer finansal bilgilerin elde edilmesi
- Ağ üzerinde herkese verilen bir hizmeti kesintiye uğratmak; bu maksatla birden fazla bilgisayarı uzaktan kontrol etmek
- Kullanıcıların bilgisayarlarını dijital para kazanmak amacıyla kullanmak [16].

2.1. Zararlı Yazılım Türleri

Zararlı yazılımların sayısı günümüzde milyonlarla ifade edilmekle birlikte, genel olarak her bir zararlı yazılım aşağıda açıklanan kategorilerden birisine üye olduğu söylenebilir.

2.1.1. Virüsler

Bir bilgisayar virüsü, başka dosyalar içinde gizlenerek sistemin çalışma şeklini değiştirmeyi amaçlayan bir programdır. Bilgisayar virüsleri kendi başlarına diğer programlara ve/veya belgelere yayılırlar. Bir bilgisayar virüsünün hedefi, açıklığı bulunan sistemlere nüfuz etmek, yönetici rolünü elde etmek ve kullanıcıların hassas bilgilerini çalmaktır [41].

2.1.2. Fidye yazılımı (Ransomware)

Fidye yazılımı, siber saldırganların kullanıcıların dosyalarını veya tüm sabit disklerini karmaşık bir algoritma ile şifreledikten sonra kullanıcılardan para (genellikle dijital para Bitcoin, Eterium vb.) talep etmek için kullandıkları bir zararlı yazılım çeşididir. Diğer zararlı yazılımlardan farklı olarak kendini gizlemek yerine bir an önce kullanıcının farkına varması ve istenilen fidye miktarını saldırganlara ulaştırması hedeflenmektedir. Kullanıcı ödeme yapana kadar veriler şifreli kalır [42].

2.1.3. Solucanlar (Worms)

Virüsler ile solucanlar arasındaki önemli bir fark, virüslerin aktif olabilmesi için mevcut bir program gerekirken, solucanlar kullanıcı eylemi olmadan çoğalabilen zararlı programlardır. Bir solucan kendisinin birden fazla kopyasını oluşturur ve ağda açıklıkları bulunan ve gerektiği gibi korunmayan tüm bilgisayarları ve sunuculara bulaşabilir [43].

2.1.4. Casus yazılım (Spyware)

Casus yazılım, kullanıcıların bilgisi dışında sistemlere yüklenen ve kişisel bilgileri veya internette ziyaret edilen sitelerden kullanıcının davranış analizini çıkararak saldırgana ileten bir zararlı yazılım çeşididir. Bu yazılımlar ile kullanıcıların cihaz üzerindeki her tür etkileşimi izlemeleri mümkün hale gelir. Saldırgan casus yazılımlar aracılığıyla sisteme yüklenmeye müteakip kullanıcıların eylemlerini kendi bilgisayarından izleyebilir [16].

2.1.5. Truva atı (Trojan Horse)

Truva atları, kullanıcıları kandırmak için zararsız uygulamalar şeklinde davranırlar. Amaç yayılmaktan ziyade girdikleri sistemde fark edilmeden ana fonksiyonu (bilgi çalmak vb.) yürütebilmektir. Çoğu zaman meşru programların içine bilinçli olarak yerleştirilirler. Bu sebeple lisanssız programlar kullanırken bu husus göz önünde bulundurulmalıdır. Genellikle saldırgan için sistemde bir arka kapı (backdoor) açmak için kullanılırlar [44].

2.1.6. Reklam yazılımı (Adware)

Çevrimiçi reklamlar oluşturmak amacıyla kullanılan kötü amaçlı yazılımlardır. Bu tür programlar, kullanıcılara çevrimiçi iken istenmeyen reklamlar veya açılır pencereler gösterir. Reklam yazılımı programları genellikle, başka bir program yüklerken sisteme yüklenir [17].

2.2. Saldırı Vektörleri ve Klasik Zararlı Yazılım Analiz Yöntemleri

Yukarıda açıklanan zararlı yazılımların kullanıcı sistemlerinde aktif hale getirilmesi için saldırganlar tarafından kullanılan yaygın siber saldırı vektörleri şunlardır:

- Güvenliği İhlal Edilmiş Kimlik Bilgileri
- Zayıf ve Çalıntı Kimlik Bilgileri
- Fidye yazılımı
- E-dolandırıcılık
- Sıfır Gün Güvenlik Açıkları
- Eksik veya Yetersiz Şifreleme
- Yanlış yapılandırma
- Güven İlişkileri

- Sıfır gün güvenlik açıkları
- Kaba kuvvet saldırısı
- Dağıtılmış Hizmet Reddi (DDoS)

Kötü amaçlı yazılımların çeşitliliği, karmaşıklığı ve kullanılabilirliği, ağları ve bilgisayar sistemlerini saldırılardan güvence altına almak için muazzam zorluklar ortaya çıkarır. Kötü amaçlı yazılımlar güvenlik analistlerini zorlayacak şekilde sürekli gelişmektedirler. Araştırmacıların siber savunma yöntemlerini geliştirerek bu duruma ayak uydurmaları bir zorunluluk haline gelmiştir. Polimorfik ve metamorfik yöntemler ile zararlı yazılımın güvenlik sistemlerine yakalanmasının önlenebilmesi ve gerçek amacının saklanabilmesi ile zararlı yazılımların sayısı ve ortaya çıkardığı güvenlik tehdidi oldukça artmaktadır. Polimorfik kötü amaçlı yazılım, kodun yapısını değiştirecek bir kod barındırır. Bu sayede hash değeri değişse de yazılım orijinal fonksiyonunu korur. Ayrıca statik analiz yapılmasını neredeyse imkânsız hale getirir. Paketleme ve şifreleme yaygın olarak kullanılan yöntemlerdendir [7].

Metamorfik kötü amaçlı yazılım, aynı fonksiyonu sürdürecek bir şekilde kodu yeniden oluşturur. Kötü amaçlı yazılım yazarları birden fazla dönüşüm kullanabilir. Buna örnek olarak; kayıt defteri (registry) yeniden adlandırma, kod genişletme (extension), kod küçültme ve çöp (garbage) kodu ekleme buna örnek olarak verilebilir [7].

Literatür taramalarına bakıldığında zararlı yazılım analizlerinin genel itibari ile 3 temel metot kullanılarak analiz edildiği görülmektedir:

Bunlar statik analiz, dinamik analiz ve hibrid analiz metotlarıdır. Bununla birlikte zararlı yazılım tespit yöntemleri imza-tabanlı ve sezgisel (davranış tabanlı) olarak karşımıza çıkmaktadır. Statik analiz kötü amaçlı yazılımı çalıştırmadan incelemeyi içerir. Diğer yandan dinamik analiz yönteminde yazılımın çalıştırılmasını ön koşuldur. Hibrid analiz her iki yöntemin harmanlanmış halidir [27].

Kötü amaçlı yazılımın nasıl çalıştığını anlamak, işlevselliğini, kaynağını ve potansiyel etkisini belirlemek için izlenen inceleme süreci kötü amaçlı yazılım analizi olarak adlandırılır.

Güvenlik analistlerinin karşılaştığı kötü amaçlı yazılım sayısı, önceden tespit edilen programların farklı versiyonları ile sayıları milyonları bulacak şekilde sürekli artmaktadır. Bu sebeple kötü amaçlı yazılım analizi tüm işletmeler için kritiktir.

Dinamik analiz sayesinde, zararlı yazılımın statik analiz ile tespit edilebilmesini zorlaştıran faktörler ortadan kalkar ancak; dinamik analiz sürecinde göz önüne alınması gereken bazı hususlar bulunmaktadır:

1. Dinamik analiz için zararlı yazılım gerçek zamanlı olarak çalıştırılmalıdır.

2. Çalıştırılan sistem güvenli bir ortam olmalıdır. Bu sayede zararlı yazılımın ortaya çıkarabileceği geri dönülmez zararlar önlenebilir.

3. Bahse konu güvenli ortam gözden çıkarılabilir ve internet bağlantısı olmayan bir fiziksel sistem olabilir. Ancak bu durumda zararlı yazılım işlevine göre internete çıkış yapamayacağı için hangi domain, IP ile bağlantı kurduğu tespit edilemez.

4. Bu durumda sanal makinelar kullanılabilir. Fakat zararlı yazılım yazarları, zararlı yazılımın sanal veya kum havuzu vb. ortamlarda bulunduklarını tespit etmeye ve bu durumda farklı davranmaya yönelik kod parçacıklarını zararlı yazılıma ekleyebilecekleri unutulmamalıdır [18].

3. LİTERATÜR TARAMASI

Çalışma öncesinde bu alanda yapılmış olan çalışmaları incelemek adına YÖK Tez Merkezi veritabanında "Zararlı Yazılım Tespiti" ve "Malware Detection" anahtar kelimeleri ile arama yapılmıştır. Arama sonucunda 15 adet tez bulunmuştur. Bu tezlerden iki tanesinin doktora, diğerlerinin yüksek lisans tezi olduğu ve Türkiye'de bu alanda ilk kez 2012 yılında tez çalışması yapıldığı görülmüştür. Çalışmaların bazıları bu tezde incelenen Windows işletim sistemlerine uyumlu yazılımlardan ziyade android işletim sistemlerine karşı üretilen zararlı yazılımlara yöneliktir. Bu tezde incelenen problem sahası ile benzerlik gösteren çalışma konusuna sahip tezler hakkındaki bilgiler aşağıdaki tabloda sunulmuştur:

Çizelge 3.1. Literatür taramasında incelenen tez bilgileri

Yazar/Yazarlar	Yıl	Problem	Çözüm Yöntemi
Tahıllıoğlu Erşan	2020	Sıfırıncı gün zararlı yazılımlarının tespiti	Zararlı yazılımların bellek üzerinde bıraktığı izler resim formatına dönüştürülerek özellik çıkarımı yapılmıştır.
Alsumaidaee Yaseen Ahmed	2019	Kötü amaçlı yazılımların analizi ve tespiti	Güç Spektral Yoğunluğu (PSD) ile özellik çıkarımı yapılmış, elde edilen veri seti Destek Vektör Makinesi (SVM), Radyal Temel Ağ (RBF) ve çok katmanlı perceptron (MLP) yöntemleri ile analiz edilmiştir.
Mohamed, Sefu	2019	Sıfırıncı gün zararlı yazılım adı verilen "yakalanmayan" zararlı yazılımların tespiti	Çalıştırılabilir dosyaların nümerik ve metinsel özellikleri kullanılarak oluşturulan veri seti üzerinde Rastgele Orman, KNN, Karar Ağacı ve Light GBM sınıflandırıcıları çalıştırılarak analiz edilmiştir.

Çizelge 3.1. (devam) Literatür Taramasında İncelenen Tez Bilgileri

Atasever, Kübra Nur	2019	Metamorfik virüslerin tespiti	Çalıştırılabilir dosyalardan elde edilen, assembly dosyalarından çıkarılan yerel ve harici fonksiyonları ile oluşturulan grafların benzerlik oranı ve bu fonksiyonlara ait opcode dizilerinin benzerliği en uzun ortak küme (longest common subsequence) algoritması kullanılarak metamorfik virüs varyantlarının benzerliği analiz edilmiştir.
Bulut, İrfan	2017	Analiz sürecini atlatmaya çalışan modern zararlı yazılımların tespit edilmesi	Derin öğrenme yaklaşımı ile bu tür zararlı yazılımların tespitine yönelik açık kaynak bir veri seti üzerinde çalışma yapılmıştır.
Spafford ,E. H.	1989	Bilgisayar Solucanlarının Yayılması ve Sistemsel Davranışları	Kodları üzerinden genel bir analiz yapılmıştır.

Ayrıca Cornell Üniversitesi “arxiv.org”, ScienceDirect veri tabanlarında “malware detection via machine learning” ve ULAKBİM/Harman veri tabanında “makine öğrenmesi ile zararlı yazılım tespiti” anahtar kelimeleri yapılan arama sonucunda toplam 168 adet makale bulunmuştur. Çalışmaların bazıları bu tezde incelenen Windows işletim sistemlerine uyumlu yazılımlardan ziyade android işletim sistemlerine karşı üretilen veya ağ trafiği analizinde tespit edilmeye çalışılan zararlı yazılımlara yöneliktir. Bu tezde incelenen problem sahası ile benzerlik gösteren çalışma konusuna sahip makaleler hakkındaki bilgiler aşağıdaki tabloda sunulmuştur:

Çizelge 3.2. Literatür Taramasında İncelenen Makale Bilgileri

Yazar/Yazarlar	Yıl	Problem	Çözüm Yöntemi
Galinkin, Erick	2020	Zararlı yazılım trafiğinin analizi ve zararlı yazılım tespiti	Sinir ağları kullanılarak analiz çalışması yapılmıştır.
Sharif, Mahmood Shintre, Saurabh Bauer, Lujo Reiter, Michael	2019	Bilinmeyen zararlı yazılımların klasik olmayan yöntemler ile tespiti	Zararlı yazılım yapısı değiştirilerek/optimize edilerek Derin Sinir ağları ile tespit çalışması yapılmıştır.
Joshi, Levi Lloyd, Paul Westin, Srini	2019	Zararlı URL tespiti	Zararlı URL lerin statik özellikleri (strings) çıkarılarak elde edilen veri seti ile analiz yapılmıştır.
Kolter	2006	Bilinen zararlı yazılımların tespiti	N-gram ayırma sonucu elde edilen veri seti Naif Bayes, DVM ve karar ağaçları ile analiz edilmiştir.
Wang, Xin Yiu, Siu Ming	2016	Zararlı yazılım analizi için daha az ama daha kapsamlı veri seti oluşturma	Recurrent Neural network (RNN) ve API çağrım dizisi kullanılarak elde edilen veri seti üzerinde analiz yapılmıştır.
Perdisci	2008	Paketlenmiş zararlı programların tespiti	Paketlerin açılması ve imza tabanlı tespit yöntemi kullanılmıştır.
Shafiq	2009	Bilinmeyen zararlı yazılımların tespiti	PE Miner ile statik öznitelikler çıkarılmış, öznitelik eleme sonrası veri madenciliği yöntemlerinden Naif Bayes, DVM, Regresyon kullanılmıştır.

Çizelge 3.2. (devam) Literatür Taramasında İncelenen Makale Bilgileri

Shabtai	2009	Bilinmeyen zararlı yazılımların tespiti	Öznitelik seçme teknikleri (Bilgi kazanımı, Fisher skoru, doküman sıklığı (TF)) ve sınıflandırma algoritmaları (K-NN, Bayes Net, Naif Bayes) kullanılmıştır.
Bazrafshan	2013	Kötü amaçlı yazılımları tespit etmek	(1) imza tabanlı yöntemler, (2) sezgisel tabanlı yöntemler ve davranış tabanlı yöntemleri kullanarak statik özellikleri çıkarılan zararlı yazılımların tespiti için çalışma yapılmıştır.
Dube T. Raınes R. Peterson, G. Bauer K.	2012	Kötü amaçlı yazılımları tespit etmek	Zararlı yazılımların sezgisel olarak ortaya koyulmuş statik özellikleri karar ağacı (ID3, C4.5) algoritmaları ile analiz edilmiştir.
Mohaisen A. Alrawi Omar Mohaisen M.	2015	Klasik yöntemler ile modern zararlı yazılımların tespit edilememesi	Yazılımlar sanal ortamlarda çalıştırılarak elde edilen davranışsal özellikleri ile denetimsiz öğrenme yaklaşımı kapsamında kümelenmişlerdir.

4. VERİ MADENCİLİĞİ, ZARARLI YAZILIMLAR VE MAKİNE ÖĞRENMESİ

Veri madenciliği temelde, çok sayıda yapısal ve yapısal olmayan büyük veriden çeşitli matematiksel yöntem ve yaklaşımlar ile anlamlı çıkarımlar elde edilmesidir [13]. Bu tanıma verilebilecek en güncel örnek online alışveriş platformlarıdır. Bahse konu platformlarda müşteri özellikleri (yaş, cinsiyet, ülke vb.), müşteri hareketleri (alınan ürünlerin sırası, ilişkisi vb.) gibi özellikler sonucu çok sayıda müşteri ve satılan, göz atılan ürün bilgisi elde edilmektedir. Bu özelliklerin oluşturduğu büyük veri, analiz edilerek müşterilere satın alabilecekleri ürünler, satın alınan ürünlere göre önerilecek ürünler ortaya çıkarılabilmektedir. Büyük verinin üç temel özelliği bulunmaktadır. Bunlar verinin miktarı, hızı ve değişkenliğidir [8].

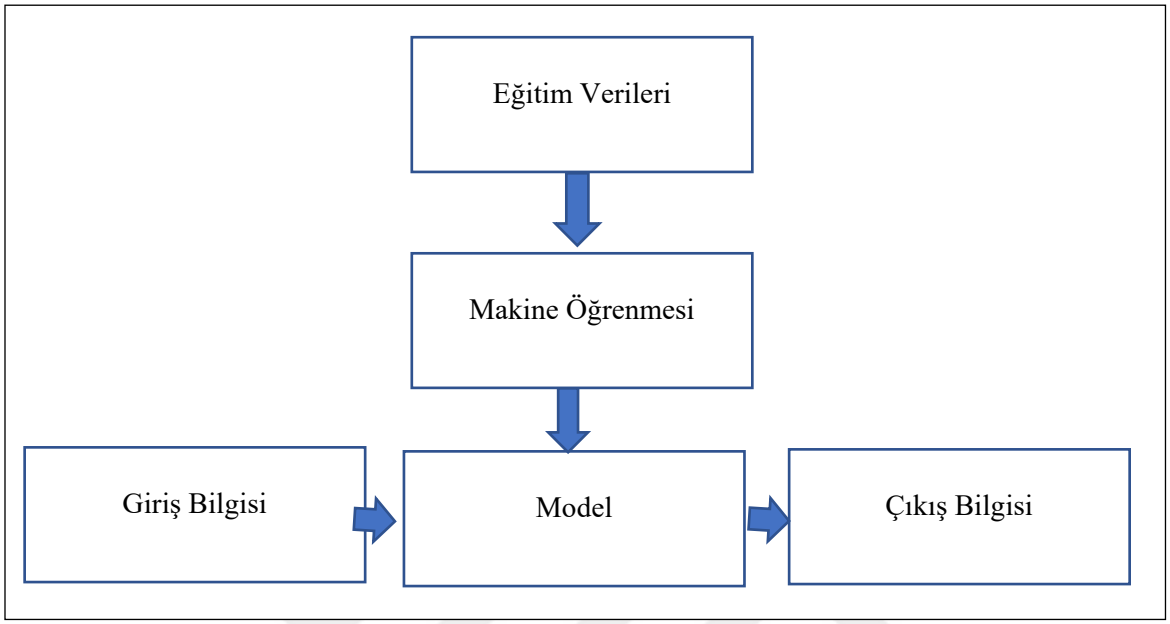
Benzer şekilde zararlı yazılım analizlerinin yapıldığı platformlara (çevrim içi bulut yapıları, lokal bilgisayarlar vb.) bağlı olarak miktar ve hız değişkenlik gösterebilecektir. Bulut yapılarında çok sayıda veri çok kısa sürede oluşabilirken; lokal çalışma alanlarında (kişisel bilgisayar) bu sayı nispeten sabit olabilmektedir. Ancak değişkenlik özelliği zararlı yazımların içerdiği yapısal, yarı yapısal ve yapılsa olmayan veriler göz önüne alındığında her iki taraf için de aynı olabilmektedir [8].

Zararlı yazılımların statik ve dinamik analizleri sonucunda elde edilen özellikleri de yukarıda belirtildiğine benzer şekilde sayısal, metin vb. değerlere sahiptir. Bu özellikler ve değerlerin hepsi zararlı yazılımların analizleri için kullanılan büyük veri halindedir. Bahse konu değerler normalize edilmeye müteakip veri madenciliği yöntemleri ile birbirleri ve çıktı değerleri ile ilişkileri ortaya koyulur. Nihai olarak bir modelde kullanılabilecek hale getirilir. Elde edilen model yeni verileri en yüksek doğruluk oranında kategorize edebilmek adına bir karar destek aracı olarak kullanılır [13].

4.1. Makine Öğrenmesi, Denetimli-Denetimsiz Öğrenme Modelleri

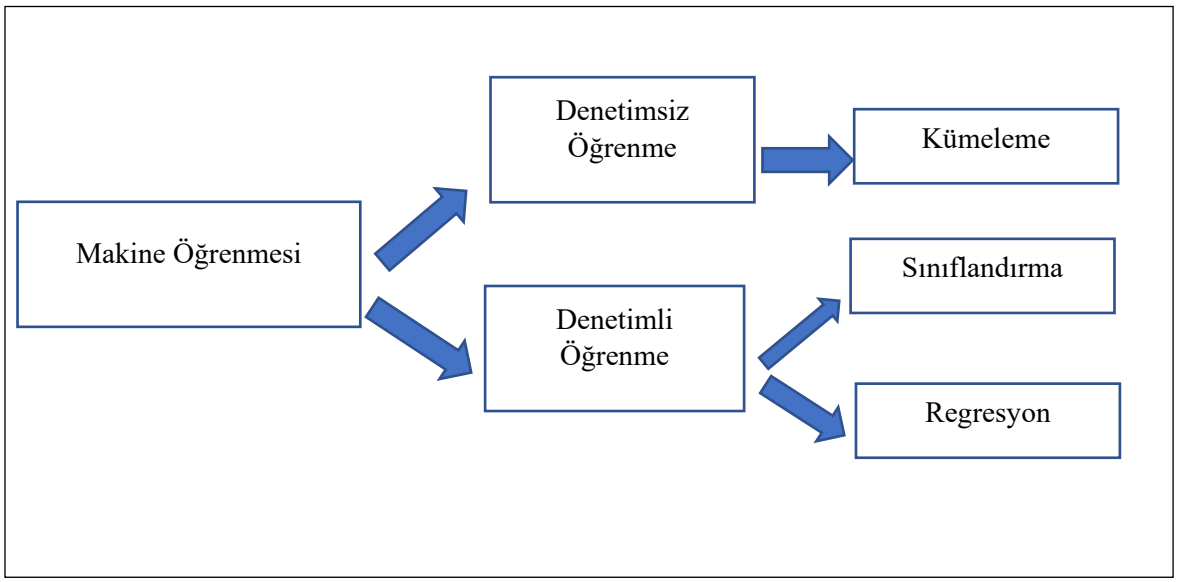
Makine Öğrenimi, öğrenme yeteneğine sahip olan, kendisine verilen verileri kullanarak yeni veriler hakkında tahminde bulunan algoritmaları kapsayan bir sistemdir. Bu algoritmalar kendisine girdi olarak verilen verilerden bir model oluştururlar. Bu modelin çıktı kalitesi

(tahmin/karar) girdilerin niteliği ile doğru orantılıdır. Bu sebeple algoritmalara verilecek olan verilerin seçilmesi, elenmesi süreçleri önem arz etmektedir [9].



Şekil 4.1. Makine öğrenmesi temel akış şeması [9]

Makine öğrenimi algoritmaları, verilerden öğrenen, tecrübi olarak sürekli geliştirebilen ve nihai olarak insan müdahalesi olmadan çalışabilmesi hedeflenen programlardır. Denetimli ve denetimsiz öğrenme modelleri makine öğrenmesi algoritmalarını kapsayan iki temel kategori olarak karşımıza çıkmaktadır.

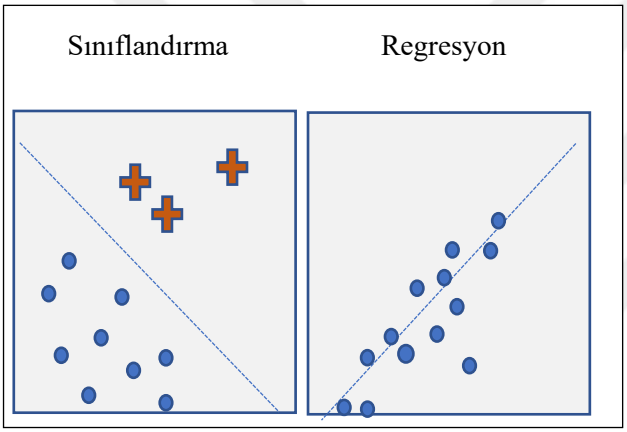


Şekil 4.2. Makine öğrenmesi kategorileri [9]

4.1.1. Denetimli öğrenme

Denetimli öğrenmede model etiketlenmiş veriler kullanılarak eğitilir. Modelin eğitiminde kullanılacak bir eğitim veri seti mevcuttur. Benzer durum olarak, bir öğretmenin desteğiyle gerçekleşen öğrenme süreci verilebilir. Denetimli öğrenme sürecinde model, etiketli eğitim verileri ile öğrenme işlemini gerçekleştirir, müteakiben yeni gelen verilerden en doğru sonuca ulaşılmasında yardımcı olur. [10].

Denetimli öğrenme, önceki deneyimden veri toplamamıza veya bir veri çıktısı oluşturmamıza olanak tanır. Deneyimi kullanarak performans kriterlerini optimize etmemize ve çeşitli gerçek dünya hesaplama problemlerini çözmemize yardımcı olur [10].



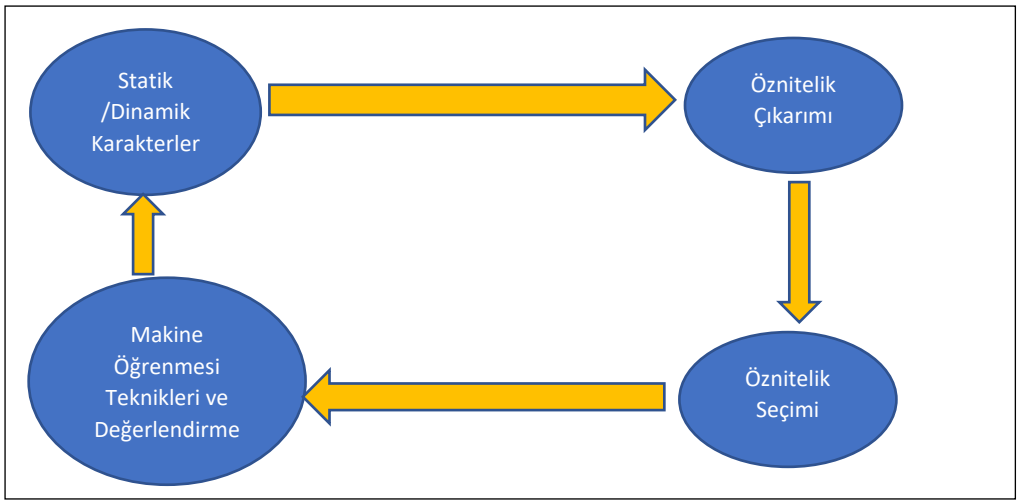
Şekil 4.3. Denetimli öğrenme yöntemleri [10]

4.1.2. Denetimsiz öğrenme

Denetimsiz öğrenmede bir sınıf değeri/kategori çıktısı bulunmamaktadır. Denetimsiz öğrenmede amaç, verilerin özelliklerine göre kümeler altında toplayabilmektir. Kümeleme ve birliktelik kurallarına göre verileri gruplara ayırmak, denetimsiz öğrenme yönteminin temel metotlarıdır [11].

Çizelge 4.1. Denetimli-denetimsiz öğrenme karşılaştırması [10]

Perametreler	**Denetimli Makine Öğrenmesi Tekniği**	**Denetimsiz Makine Öğrenmesi Tekniği**
İşlem	Girdi ve çıktı değerleri verilir	Yalnızca girdi verisi verilir.
Girdi Verisi	Algoritmalar etiketli veri ile eğitilir.	Algoritmalar etiketli veri kullanmaz.
Kullanılan Algoritmalar	SVM, NN, Regresyon, Rastgele Ağaç, Snıflandırma Ağaçları	Kümeleme algoritmaları, En yakın komşu algoritması, Hiyerarşik kümeleme vb.
Hesaplama Karmaşıklığı	Denetimsiz öğrenmeye göre daha basittir.	Hesaplama karmaşıklığı nispeten yüksektir.
Veri Kullanımı	Eğitim seti ile modeli öğrenir ve girdi çıktı arasında bağlantı kurmaya çalışır.	Çıktı verisi kullanmaz. Girdiyi uygun kümeye ilintiler.
Doğruluk Oranı	Yüksek seviyede güvenilirlik	Daha az doğruluk oranı
Gerçek Zamanlı Öğrenme	Öğrenme eğitim seti ile çevrimdışı gerçekleşir	Öğrenme kümeleme sırasında gerçek zamanlı olur.
Sınıf sayısı	Sınıf sayısı bilinir.	Sınıf yoktur. Optimize küme sayısını belirlemek ayrıca hesaplama gerektirir.
Temel Sorun	Değişik yapıdaki çok büyük verinin analizi sırasında problemler ortaya çıkabilir.	Kesin ve tam olarak doğru sonuca her zaman ulaşılamaz.



Şekil 4.4. Makine öğrenmesi tabanlı sınıflandırma sürecinde izlenen döngü [12]

4.2. Makine Öğrenmesi ile Zararlı Yazılım Tespit ve Sınıflandırmasında Karşılaşılan Güçlükler [18]

Bu kapsamda sınıf dengesizliği, konsept kayması ile karşı saldırı öğrenimi ve açık kaynak riski kavramları karşımıza çıkmaktadır.

4.2.1. Sınıf dengesizliği [19]

Makine öğrenmesi temelli oluşturulan bir model kendisine verilen eğitim setinin kalitesi ile doğru orantılı olarak sonuç üretebilecektir. Veri seti içinde farklı sınıflara ait dengesiz veri dağılımı, modelin çoğunlukta olan sınıfa göre sonuç üretmesine dolayısı ile doğruluk oranının düşmesine sebebiyet verebilecektir.

Literatürde doğruluk çelişkisi olarak yer alan bu duruma bağlı olarak doğruluk kendi başına yeterli bir gösterge olmamaktadır. Bu kapsamda karmaşıklık matrisi (confusion matrix) içinden elde edilen Kesinlik, Duyarlılık ve F score değerleri de bir sınıflandırıcının performansını ölçmek için kullanılmaktadır.

Kesinlik (Precision) değeri; doğru pozitif (TP) sayısının doğru pozitif ve yanlış pozitif sayılarının toplamına oranıdır. Kesinlik değeri sınıflandırıcının pozitif olarak sınıflandırdığı değerlerden kaçının gerçekten pozitif olduğunu ortaya koyar. Yanlış doğru sayısı azaldıkça doğruluk oranının arttığı matematiksel olarak görülebilmektedir.

Kesinlik değeri formülü Eş. 4.1'de görüldüğü gibidir.

Kesinlik= TP/(TP+FP) (4.1)

Duyarlılık (Recall) değeri; doğru pozitif (TP) sayısının doğru pozitif ve yanlış negatif (FN) sayılarının toplamına oranıdır. Sınıflandırıcının pozitif olarak sınıflandırması gereken değerlerden kaçını pozitif olarak sınıflandırdığını ortaya koyar. Yanlış negatif sayısı azaldıkça doğruluk oranının arttığı matematiksel olarak görülebilmektedir. Zararlı bir yazılımın zararsız olarak kabul edilmesinin riski göz önüne alınarak, çalışmada doğruluk değerleri ile birlikte duyarlılık değerleri de dikkatle incelenmiştir. Eş. 4.2'de görüldüğü gibidir.

Duyarlılık=TP/(TP+FN) (4.2)

F score değeri; Kesinlik ve Duyarlılık değerlerinin çarpımının toplamına oranının 2 ile çarpılması ile edilmektedir. Uç değerlerin de hesaba katılması gereken eşit dağılımın olmadığı veri setleri analizinde F-score değeri önem taşımaktadır. Eş. 4.3'te görüldüğü gibidir.

F-Score=2*((Kesinlik*Duyarlılık)/(Kesinlik+Duyarlılık)) (4.3)

4.2.2. Konsept kayması

Çalışılan zararlı yazılım tespit yöntemlerinin ileriki dönemde ortaya çıkabilecek tüm zararlıları tespit edebileceği yanılgısı bulunmaktadır. Hâlbuki zararlı yazılımlar gün geçtikçe daha karmaşık hale gelmekte ve içerdiği öznitelikler farklılaşabilmektedir. Örneğin; zararlı yazılım kodları üzerinde çalıştıkları işletim sistemini analiz ederek buna göre eğer sanal makine üzerinde ise asıl fonksiyonlarını gerçekleştirmeyecek şekilde oluşturuldukları göz önüne alındığında konsept kaymasının dikkate alınması gerektiği daha iyi anlaşılabilmektedir. [19].

4.2.3. Karşı saldırı öğrenimi ve açık kaynak riski

Makine öğrenmesi her ne kadar yeni nesil bir metot ve zararlı yazılım tespit ve sınıflandırmasında etkin ve modern bir yaklaşım olarak kullanılsa da; saldırganlar da modelin yapısını atlatmaya yönelik öznitelikler oluşturabileceği akıldan çıkarılmamalıdır. Modelin tespiti hangi esasalar göre yaptığını öğrenen bir saldırgan modeli aldatmaya yönelik şekilde yazılımı şekillendirebilir. Bununla birlikte makine öğrenmesi ile zararlı yazlım

analizine yönelik çalışmaların büyük çoğunluğu açık kaynaktan ulaşılabilecek veri setleri üzerinde yapıldığından, saldırganların bu veri setlerine ulaşabileceği ve belirgin öznitelikleri saklayacak şekilde yazılımı oluşturabileceği bir gerçektir.

Bu geleneksel saldırı vektörleri yanında saldırganlar yapay zekâyı da bir saldırı vektörü olarak kullanmaya başlamışlardır. Günümüzde yapay zekâ kabiliyetleri ile bir insanın sesini telefonda taklit edip kişi veya organizasyondan para/bilgi çalmak yaygınlaşmaktadır [19].

5. ZARARLI YAZILIM TESPİTİNDE KULLANILAN MAKİNE ÖĞRENMESİ METOTLARI

Zararlı yazılım tespitinde kullanılan makine öğrenmesi metotları istatistiksel, kural tabanlı ve uzaklık tabanlı metotlar olmak üzere üç başlık altında incelenmiştir.

5.1. İstatistiksel Metotlar

Sonuçları olasılık hesaplamaları ile üreten Naif bayes, Bayes ağları ile regresyon ve lojistik regresyon metotları aşağıda açıklanmıştır.

5.1.1. Naif bayes (Naive bayes)

Naif Bayes olasılığa dayanan bir sınıflandırıcıdır. Bayes Kuralı koşullu bağımsızlıkları esas alan bir formül ile açıklanabilir [12]. Eş. 5.1'de görüldüğü gibidir.

$$P(A|B) = P(A)\frac{P(B|A)}{P(B)} \quad (5.1)$$

Bir Naive Bayes sınıflandırıcı, özniteliklerin birbirinden koşulsal bağımsız ve erişilmek istenen çıktının (sınıf değeri) tüm bu özniteliklere koşulsal bağlı olarak değerlendirildiği bir sınıflandırıcıdır [38].

P(A|B) ; B olayı gerçekleştiği durumda A olayının meydana gelme olasılığıdır.

P(B|A) ; A olayı gerçekleştiği durumda B olayının meydana gelme olasılığıdır.

P(A) ve P(B) ; A ve B olaylarının sırasıyla olasılıksal oranlarıdır.

Bu makine öğrenimi algoritması ile bir eğitim veri kümesi üzerinden koşullu ve koşulsuz olasılıklar elde edilmektedir. Elde edilen çıktı, sorgulanan bir olasılıktır. Bu nedenle, sınıflandırma işlemi sonucunda verilen karar, karşılık gelen sınıf kümesinden bir maksimum değere dayanılarak verilecektir [20]. Naif Bayes algoritması tüm özniteliklerin birbirinden bağımsız, sınıf değerine koşullu bağımlı olduğu ilkesine dayanır. Özniteliklerin birbirleri ile olan bağımlılıklarını göz ardı ettiği için doğruluk değeri nispeten yüksek olmamakla birlikte; bellek ve hesaplama karmaşıklığı oldukça düşüktür [38].

5.1.2. Bayes ağları

Bayes Ağları, yönlendirilmiş döngüsel olmayan grafik kullanarak koşullu bağımlılıkları gösteren, olasılık odaklı döngüsel olmayan bir grafik modeldir (Literatürde Bayesian Belief Networks olarak da adlandırılır).

Bir Bayes ağı, yönlü dönüşsüz bir çizge modelidir. Naif Bayes'ten farklı olarak birbirleriyle koşulsal bağımlılığa sahip bir değişkenler kümesini temsil eder [1]. Bayes ağları; bir olayı olası sebepleri ile bağdaştırarak açıklamak için kullanılan bir modelleme türüdür. Veri kümesine ait tüm öznitelikler koşulsal bağımlı olarak kabul edilerek karar verilmeye çalışılır. Bu kapsamda, çalışmada olduğu gibi yazılım öznitelikleri kullanılarak bir programın zararlı olup olmadığı, semptomlara bağlı olarak hastalık türü gibi sonuçlara ulaşılabilir. Bir olayın bileşenleri ile olay arasında istatistiksel bir neden-sonuç ilişkisi ortaya koyabilmektedir. Eş. 5.2'de görüldüğü gibidir.

$$\frac{P(y)P(x|y)P(z|y)}{P(y)P(x|y)} \tag{5.2}$$

Bayes Ağları sınıflandırma için kullanılabilir, bu nedenle uygulanabilir kötü amaçlı yazılım tespiti ve sınıflandırması için de uygun bir metottur. Bayes Ağını sınıflandırmaya uygun hale getirmek için, üst düğümlere sahip olmayan sınıflar en üst ebeveyn (parent) düğümler olarak belirlenmelidir [20].

Bayes ağında temel olarak izlenen yöntemin, belli bir sonucu (bu çalışmada programın zararlı ya da zararsız olması) etkileyen parametrelerin (niteliklerin) birbirlerine bağımlılık durumlarına göre sonuca yönelik olasılıksal olarak tahminde bulunmak olduğu söylenebilir [20].

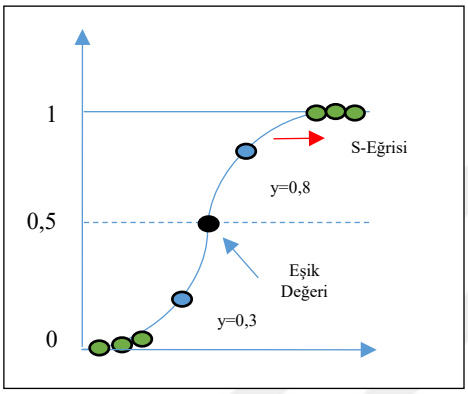
5.1.3. Regresyon ve lojistik regresyon

Verilen bir değerden sınıfı tahmin etme işlemi olarak bilinir. Regresyonda yönteminin üç adımı mevcuttur:

- R^2 : Veriler arasındaki korelasyon;
- p-değeri : R^2 değerinin önemi;

- Tahmin edilen sonuç.

Lojistik regresyonun lineer regresyondan farkı, lojistik regresyon herhangi bir durum hakkında sonuca yönelik bir olasılık ortaya koyar. (%100 doğru-yanlış; %50 sınıf-1 sınıf-2 gibi). Lineer regresyondan farklı olarak bir doğru yerine 'S' şeklinde bir eğriye uyumlu değerler üretilir. Bu kapsamda sonuca yönelik bir olasılık ortaya çıkarır. Sınıflandırmada olasılık yaklaşımından ziyade en sağlıklı tahmini yapmak esas olduğundan, elde edilmek istenen sonuca uygun olarak modele deneysel (empirical) olarak girilecek bir eşik değeri (threshold) ne göre lojistik regresyon bir sınıf değeri ortaya koyar.



Şekil 5.1. Lojistik regresyon eğrisi

Bu uygulamada sezgisel bir eşik değerinin üzerindeki değerler bir sınıfa altındaki değerler diğer sınıfa atanabilmektedir. Burada ulaşılmak istenen sonuca yönelik hangi girdilerin baz alınacağı belirlenmelidir. Bu kapsamda olasılığı artırmak için kazanımı en yüksek olan niteliklerin belirlenmesinde fayda vardır. Lineer regresyon gibi sürekli (ağırlık, yaş) ve kesik veriler (tip) ile çalışabilir. Wald Test hangi niteliğin tahminde etkinliği ne kadar olduğunu belirlemede kullanılır. Lojistik regresyonda R^2 kullanılmaz. Bunun yerine maksimum benzerlik yaklaşımı kullanır [22].

5.2. Kural Tabanlı Metotlar

Kural tabanlı algoritmalar, analiz sonucu kesin veya belirsiz kurallar ortaya koyabilmek için kullanılır [13]. Kötü amaçlı yazılım sınıflandırmasına dâhil olan mantık kurallarına sahip olmanın temel avantajı, eşit, daha büyük, küçük ya da eşit gibi mantıksal kuralların donanım düzeyinde çalıştırılabilir olmalarıdır. Bu durum karar verme hızını önemli ölçüde artırır.

5.2.1. C4.5

Karar ağaçları inşa etmek için özel olarak önerilmiştir [14]. Bahse konu ağaçlar kötü amaçlı yazılım tespiti gibi sınıflandırma için kullanılabilir. Ağaç eğitimi süreci, önceden sınıflandırılmış veri setinin işlenmesini içerir. Her adımda, en iyi bilgi kazancı sağlayan öznitelikler bulunarak veri ağaç yapısına dönüştürülür [50].

C4.5'in diğer karar ağacı algoritmalarına kıyasla birçok faydası vardır:

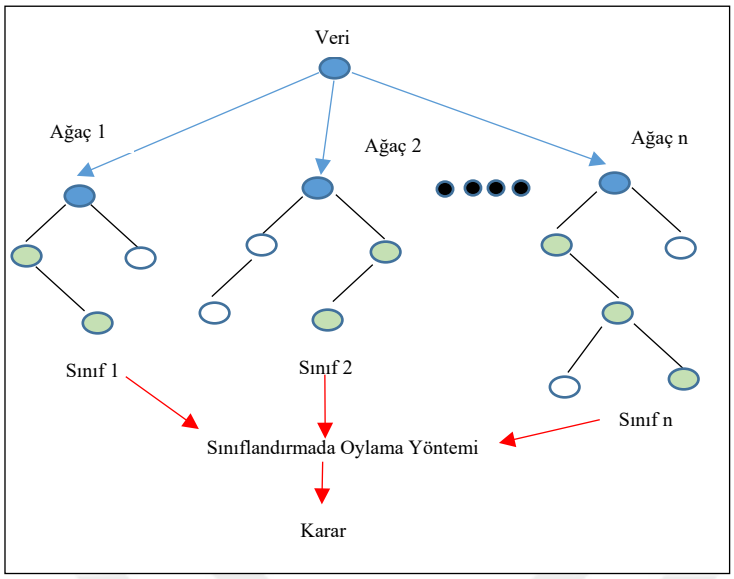
• Yalnızca ayrık değil, sürekli niteliklerle de çalışır.

• Eksik nitelik değerlerini hesaba katar.

• Farklı maliyetlere sahip özniteliklerle çalışır. Ağaçta geriye doğru süreç işletilerek düşük bilgi kazancı sağlayan dallar kaldırılır (Ağaç Budama). Bu işlem otomatik olarak fazlalık veriyi modelden kaldırır. Fazlalık (redundant) verinin sınıflandırıcının (özellikle Naif Bayes) performansına etkisi deneysel çalışmada açıklanmıştır. Çalışmada öznitelik seçme yöntemleri ile boyutu indirgenen veri setleri analizi ile ham veri setlerinin C4.5 sınıflandırıcı ile analizinde performansta çok büyük fark oluşmayacağı öngörülmektedir.

5.2.2. Rastgele orman

Adından da anlaşılacağı gibi, bir topluluk olarak çalışan çok sayıda bireysel karar ağacından oluşur. Rastgele ormanda bulunan her bir ağaç bir sınıf tahmini ortaya çıkarır, en çok seçilen sınıf, modelin kararı olarak kabul edilir [50]. Arkasındaki temel kavram, basit ama güçlü bir yaklaşım olan - kalabalıkların bilgeliğidir. Bir komite olarak çalışan çok sayıda, görece ilişkisiz modeller (ağaçlar), her bir kurucu modelden daha iyi performans gösterecektir [29].



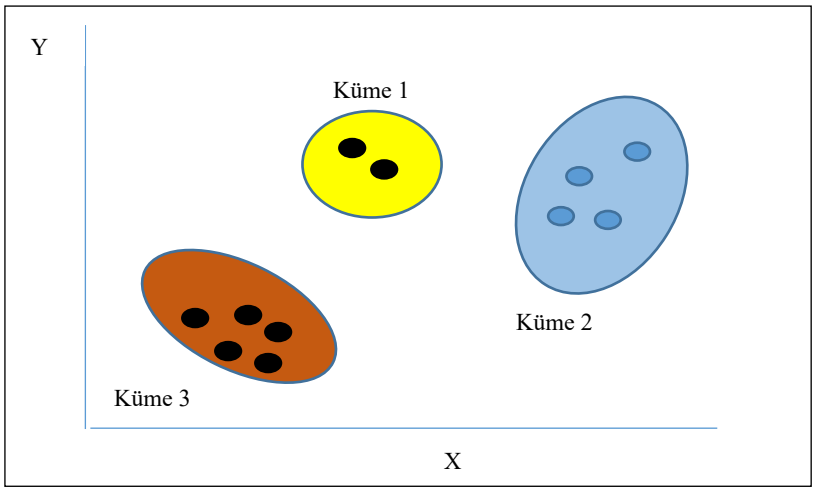
Şekil 5.2. Rastgele orman algoritması şekilsel gösterim

5.3. Uzaklık Tabanlı Metotlar

Bu yöntem seti, önceden tanımlanan mesafe ölçüsünü baz alarak sınıflandırma yapar. Uzaklık temelli yöntemler için veriler dikkatlice hazırlanmalıdır. Çünkü hesaplama karmaşıklığı, eğitim örnekleri sayısına bağlı olarak önemli ölçüde artmaktadır. Bu nedenle uygun özelliklerin seçimine ve bazen veri normalleştirmesine ihtiyaç vardır [12].

5.3.1. K-en yakın komşular veya k-nn

Bir sınıflandırma ve regresyon yöntemidir. K-NN algoritmasında gerçek "eğitime" ihtiyaç duyulmaz. Veri setinin etiketli olmasına gerek duymaz, veriyi etiketlemekten ziyade kümeleme yaparak ayırır. Algoritma bir sınıflandırılması gereken ve eğitimden numunelere olan mesafeleri hesaplayan numune veri kümesi, daha sonra k en yakın komşuyu (en kısa mesafelerle) seçer ve bu k en yakın komşunun sınıfına dayalı karar verir. K-NN, kötü amaçlı yazılım sınıflandırması ve tespiti için kullanıldığında, aykırı değerler ve çok karışık verilerle başa çıkmak için bir metodolojinin yanı sıra özellik seçimine dikkat etmek gereklidir [12]. Denetimsiz bir öğrenme metodu olması sebebi ile K-NN çalışmada kullanılmamıştır.



Şekil 5.3. Üç kümeli bir k-means algoritması şekilsel gösterimi

5.3.2. Destek vektör makinesi veya SVM

Denetimli bir öğrenme yöntemi olan destek vektör makinesinde veri setinin bölünmesi için hiper düzlemler kullanılır. Alt düzlem, en yakın veri noktalarına olan mesafeyi en üst düzeye çıkarmak için yapılmıştır. Hiper düzlemleri basitleştirmek için çekirdek dönüşümler kullanılabilir. Bir hiper düzlem inşa etmek genellikle "bire bir, bire karşı çok" olmak üzere iki sınıflı bir probleme dönüşür [12].

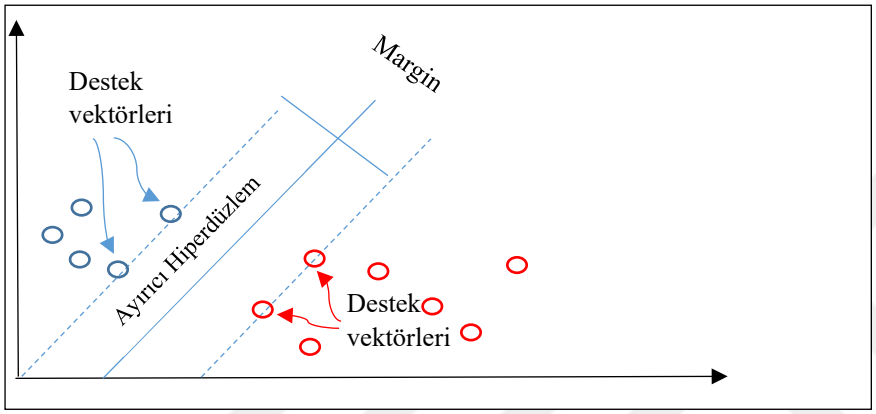
Bir Destek Vektör Makinesi (DVM), ayırıcı olarak hiper düzlemi kullanan bir sınıflandırıcıdır. Diğer bir deyişle, etiketli eğitim verilerine göre DVM sınıflandırıcı algoritması yeni verileri denetimli öğrenmede uygulandığı hali ile kategorize edebilecek optimal bir hiper düzlem çıkarmayı amaçlar.

Bu düzlemin gelen yeni girdileri doğru kategorize etmesi için optimize bir şekilde belirlenmesi gerekmektedir. Rastgele çizilen bir çizgi özellikle belirgin bir şekilde bir sınıfa ait olmayan ve her iki sınıfın da özelliklerini içerebilen yeni girdilerin yanlış sınıflandırılmalarına sebep olabilecektir. Bunu önlemek adına SVM eğitim setinde bulunan en uç (extreme) örnekleri baz alır. Her iki sınıfın en uç diğer bir deyişle birbirlerine en yakın örneklerinden çizgi çekerek düzlemi oluşturur. Bu noktaları vektör kabul eder ve diğer örnekleri kullanmaz.

Her iki sınıfın en uç noktaları arasındaki en kısa mesafeye margin adı verilir. Bu mesafenin yarısı eşik (threshold) olarak alınır. Bu sayede margin her iki noktaya da eşit mesafede uzaklığa sahip olarak olabilecek en uzun mesafeye sahip olacaktır.

Maksimal marjin sınıflandırıcılar aykırı verilere karşı çok hassastır. Bunu önlemek için yanlış sınıflandırmalara izin verilmelidir. Önyargı/Varyans çelişkisi ortaya çıkmaktadır. Sonuç olarak eşik çizgisinin yeri eğitim verisinden en az etkilenecek şekilde belirlenmelidir. Eğer herhangi bir sınıfta aykırı veriler var ise bu örnekler veri setinden kaldırılmalı ya da görmezden gelinerek az sayıda yanlış sınıflandırma göze alınmalıdır.

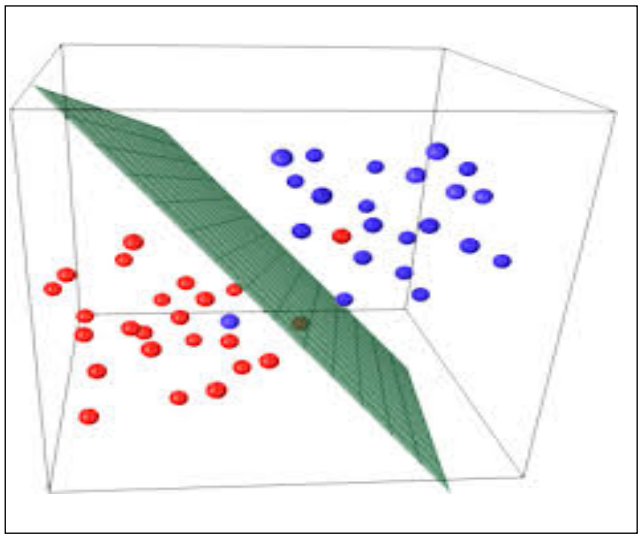
Destek Vektör Sınıflandırıcıda toplamda iki öznitelik mevcut ise, ayırıcı (margin) bir doğru olarak ortaya çıkacaktır.



Şekil 5.4. İki sınıflı bir problemde maksimum marjin hiper düzlem

Maksimal Kenar Sınıflandırıcı: Sınıf elemanları ayrık veriler dışında doğru olarak ayrılmış durumdadır. Ayırıcı bir doğru olarak karşımıza çıkmaktadır.

Destek Vektör Sınıflandırıcı (SVC): Öznitelik sayısı üç olduğunda ayırıcı *bir düzlem*; dört ve daha fazla olduğunda SVC *bir hiper uzay* olarak karşımıza çıkar.



Şekil 5.5. Üç boyutlu destek vektör sınıflandırıcı

Destek Vektör Makineleri (SVM): İki sınıf olan ancak lineer bir şekilde ayrılamayan veri setlerinde kullanılır. Tek boyutlu veri seti verilerin kareleri alınarak iki boyutlu x/y düzlemlerinde görülebilecek hale getirilir. SVM leri Kernel fonksiyonlarını bu tür durumlarda veriyi dönüştürmek ve Destek Vektör Sınıflandırıcı ortaya koyabilmek için kullanır.

Polinominal kernelde D polinom derecesi. D=1 her bir veri çiftinin ilişkisinden SVC bulur. D nin uygun değeri cross validation ile bulunur. Eş. 5.3'te görüldüğü gibidir.

$$\text{Polynomial Kernel} = (axb+r)^d \tag{5.3}$$

a ve b her iki sınıfa ait gözlem noktalarını; r polinom katsayısını ve d polinom derecesini temsil etmektedir.

Örtüşen verilerin olduğu veri setini analiz etmenin bir yolu da DVM i Radial Kernel diğer bir deyişle Radial Temel Fonksiyonu (RBF) ile kullanmaktır. Eş. 5.4'te görüldüğü gibidir.

$$\text{Radial Kernel} = e^{y(a-b)2} \tag{5.4}$$

Radial Kernel *sonsuz boyutta* verinin Destek Vektör Sınıflandırıcısını bulmaktadır. Radial Kernel in temel çalışma prensibi Ağırlıklı En Yakın Komşu yaklaşımı ile yeni gelen verinin kendisine en yakın verilere dayalı olarak sınıflandırılmasıdır. SVM, veri setinin dağılımına ve nitelik sayısına göre en uygun SVC bulmak için kernel fonksiyonlarını; kernel fonksiyonlarındaki parametrelerin (d polinom derecesi) en optimize değeri için de çapraz doğrulama (cross validation) yöntemini kullanan algoritmalardır.

Çapraz doğrulama, veri setinin belli bir oranının eğitim seti kalan oranının test seti olarak kullanılmasına dayanır. Bu uygulamada veri seti belirlenecek bir tamsayı değerinde eşit parçaya bölünür. Bu parçalardan büyük kısım eğitim seti kalan tek kısım ise test seti olarak değerlendirilir. Her bir parça bir kere test setinde "tamsayı-1" kadar eğitim setinde yer almalıdır. Diğer bir önemli nokta ise her bir parçada eşit oranda sınıf elemanlarının bulunmasıdır. Bu şekilde elde edilen doğruluk oranlarının ortalaması alınarak modelin performansı ortaya koyulabilmektedir.

e Euler sayısı y gözlemlenen noktaların birbirlerine olan etki katsayısı a ve b her iki sınıfa ait gözlem noktalarını temsil etmektedir. Bu denklem Taylor Dizi Genişletmesi ne dayanmaktadır. Bu denklemin açılımı tez çalışmasında incelenmeyecektir. Sonuç olarak Radial Kernel denklemi ile iki nokta arasında sonsuz boyuttaki ilişkinin sonucu elde edilir [22].

Çalışmada polynomial Kernel metodu kullanılmıştır. Tanımlardan da anlaşılabileceği üzere DVM, çok boyutlu (öznitelik sayısı çok olan) veri setleri analizinde, yüksek kaynak kullanımına bağlı olarak bellek ve zamansal karmaşıklığı arttırmaktadır.

6. EĞİTİM SETİNİN OLUŞTURULMASI VE ÖNEMİ

Etiketli verilerin ayrılmasında kullanılan lineer doğru veya eğrilerin çok daha sağlıklı bir şekilde yeni verileri sınıflandırabilmesi, yeterli eğitim verisi ile modelin eğitilmesine bağlıdır. Bu konuda dikkat edilmesi gereken husus, çok fazla eğitim verisinin aşırı eğitilmeye (overfitting); az sayıda eğitim verisinin ise yeterli eğitilmemeye (underfitting) yol açmasıdır. Aşırı eğitilme (overfitting) ile tabir edilen durumda, model eğitilmesinden ziyade veriyi ezberlemiş hale gelir. Böylece yeni veriyi ancak eğitim verisi ile birebir örtüşüyorsa doğru sınıflandırabilir. Diğer durumda (underfitting) ise model yeterince eğitilemediğinden yeni veriyi düşük doğruluk seviyesinde sınıflandırabilecektir [33].

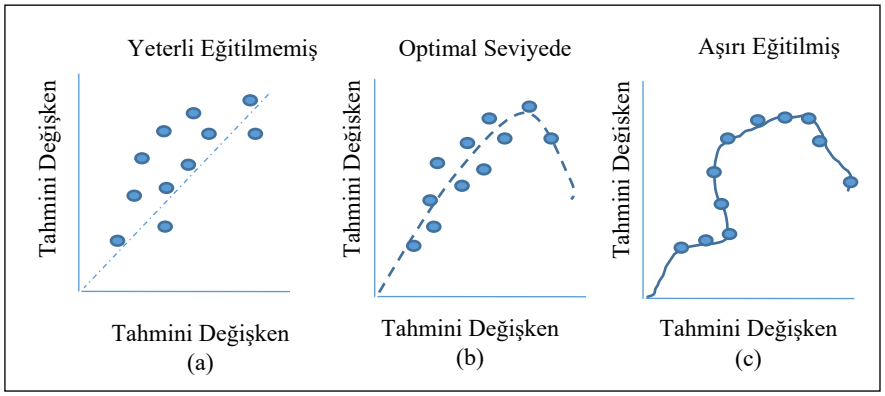
Konu ile ilgili olarak iki kritik kavram bulunmaktadır. Bunlardan ilki önyargı (Bias) diğeri varyanstır. Önyargı (bias), ortalama tahmin ile doğru değer arasındaki farktır. Önyargı Hatası veya Önyargı nedeniyle Hata olarak da bilinir. Varyans, farklı eğitim veri setleri kullanıldığında tahminin değişeceği miktardır. Farklı eğitim veri setleri nedeniyle doğru değerden tahmin edilen değerlerin ne kadar dağınık (tutarsız) olduğunu ölçer. Varyans Hatası veya Varyanstan Kaynaklanan Hata olarak da bilinir.

Aşırı eğitilmiş bir model düşük önyargılı ve yüksek varyanslı bir modeldir. Yeterli eğitilmemiş bir model ise yüksek önyargılı ve düşük varyanslı bir modeldir.

Yüksek Önyargı - Düşük Varyans (Yetersiz Uydurma-Underfit): Tahminler tutarlıdır, ancak ortalama olarak yanlıştır. Bu, model çok az parametre kullandığında gerçekleşebilir.

Düşük Sapma - Yüksek Varyans (Fazla Uydurma-Overfit): Tahminler ortalamada tutarsız ve doğrudur. Bu, model çok sayıda parametre kullandığında olabilir [33].

Algoritmik olarak tüm verilerin birbirinden ayrılmasının mümkün olmadığı veri setlerinde hangi veri önemli ise onun doğru sınıflandırılmasına yönelik bir yöntem üzerinde durulmalıdır [21].



Şekil 6.1 Makine öğrenmesi modelleri öğrenim seviyeleri

Denetimli öğrenmede kullanılan veri setlerinde;

1. Tüm veri setinde bulunan verilerin bazıları parametre tahmini yani modeli eğitmek için kullanılır. (TRAIN)

2. Deneysel (empirical) olarak belirlenecek bir kısmı doğrulama (validation) seti olarak kullanılır. (VALIDATE)

3. Makine öğrenmesi metodu ne kadar iyi iş çıkarıyor sorusunun cevaplanması gerekmektedir. (TEST)

Denetimli öğrenmede kullanılan veri setlerinde;

Genellikle iki yöntem ile analiz gerçekleştirilir. Bunlardan ilkinde tüm verinin eğitim seti olarak kullanılmasından ziyade veri setinin belli bir oranını (genel olarak bu oran %75) eğitim kalan %25 lik kısmını test için kullanmaktır [21]. Diğer yöntemde ise çapraz doğrulama (k-fold cross validation) metoduyla veri seti belirlenecek sayıda (k) bloklara bölünür her blok kendi içinde k sayısı kadar bölünerek k-1 parça eğitim seti kalan parça test seti olarak kullanılır. Her bloktan elde edilen analiz değerlerinin ortalaması modelin son çıktısı olarak kabul edilir [33].

6.1. Eğitim Setinde Öznitelikler

Özellik önemi, hangi değişkenlerin ilgilenilen bir hedef değişkeni etkileme olasılığı olduğuna karar vermek için kullanılabilir, Her değişken, verilere dayalı istatistiksel anlamlılık testleri kullanılarak hedef değişkenle karşılaştırılır [33].

Bazı temel algoritmaların performansının, iki veya daha fazla değişkenin sıkı ilişkilerine bağlı olarak bozulması durumu çoklu doğrusal bağlantı (multicollinearite) olarak adlandırılır [26].

Modeli oluştururken ulaşılmak istenen sınıfı en fazla etkileyen nitelikleri analiz etmek daha doğru bir sonuç ortaya koyacaktır. Bununla birlikte öznitelik seçimi veri boyutunun çok yüksek olduğu veri setlerinin analiz sürecindeki hesaplama karmaşıklığında gözle görülür bir azalamaya sebep olacaktır [32].

Özellik seçimi için süzme (filter) ve sıralama (rank) işlemleri yapılmaktadır. Süzme işlemi sonucunda hedeflediğimiz etiketi/sınıfı en çok etkileyen özellikler bulunurken, sıralama algoritmaları ile bu özellikler en önemliden önemsize doğru sıralanabilmektedir [31]. Çalışmada süzme yaklaşımı ile sınıflandırıcı performanslarını optimum düzeyde etkileyen ve deneysel olarak belirlenmiş sayıda öznitelik seçilerek analizler gerçekleştirilmiştir.

6.2. Öznitelik Değerlendiricileri

Veri kümenizdeki her özniteliğin (sütun veya özellik olarak da adlandırılır) çıktı değişkeni bağlamında (ör. Sınıf) değerlendirildiği tekniktir [32]. Veri kümesindeki farklı öznitelik birleşimlerinin denenmesi için kullanılan teknik arama yöntemi olarak bilinir [22].

6.2.1. Bilgi kazanımı (İnfo gain)

Bu öznitelik değerlendirme yönteminde; çıktı değişkeninin her bir özniteliği için bilgi kazancı (entropi de denir). Giriş değerleri 0 (bilgi yok) ile 1 (maksimum bilgi) arasında değişmektedir. Özniteliklerden, yüksek bilgi kazanım değerine sahip olanlar seçilebilir; bunun yanında bilgi kazanımı düşük olan öznitelikler de kaldırılabilir [33].

6.2.2. Korelasyon

Bu yöntem istatistik analizlerinde Pearson korelasyon katsayısı olarak bilinir. Her bir özellik ile çıktı değişkeni arasındaki korelasyon hesaplanmasına müteakip, yüksek korelasyonu (-1 veya 1'e yakın) olan yani çıktıyı belirlediği değerlendirilen öznitelikler seçilebilir [22].

Pearson korelasyon katsayısı (Karl Pearson olarak adlandırılır), iki veri örneği arasındaki doğrusal ilişkinin büyüklüğünü görebilmek için kullanılabilir.

Pearson'ın korelasyon katsayısı, formülsel olarak iki değişkenin kovaryansının her bir veri örneğinin standart sapmasının ürününe bölünmesiyle hesaplanır. Yorumlanabilir bir puan elde edebilmek için iki değişken arasındaki kovaryansın normalleştirilmesi olarak da açıklanabilir [24].

Veri kümesindeki değişkenler arasında karmaşık ilişkiler olabilir. Veri kümesindeki değişkenlerin birbirlerine bağımlılık seviyelerinin ölçülmesi önemlidir. Bu sayede, makine öğrenimi algoritmalarının performanslarını yükseltmek için modelin daha iyi veri ile daha nitelikli hale getirilmesi başarılabilecektir [32].

Bir veri kümesindeki değişkenler birçok nedenden dolayı ilişkilendirilebilir.

Örneğin:

Bir değişken, başka bir değişkenin değerlerinin sebebi olabilir veya bunlara bağlı olabilir.

Bir değişken, başka bir değişkenle düşük seviyede ilişkilendirilebilir.

İki değişken, üçüncü bir bilinmeyen değişkene bağlı olabilir.

Değişkenler arasındaki ilişkileri daha iyi anlamak için veri analizi de faydalı olabilir. İki değişken arasındaki istatistiksel ilişki, korelasyonları olarak adlandırılır [23].

Bir korelasyon pozitif ise her iki değişken de aynı yönde hareket eder. Negatif ise bir değişkenin değeri arttığında diğer değişkenlerin değerleri azalır. Korelasyon nötr veya sıfır ise değişkenlerin birbirleri ile tamamen bağımsız olduğu sonucuna varılabilir.

Olumlu Korelasyon: Her iki değişkenin aynı yönde değiştiği durumda ortaya çıkar.

Nötr Korelasyon: Değişkenlerin değişiminde birbirlerine etkileri yoktur.

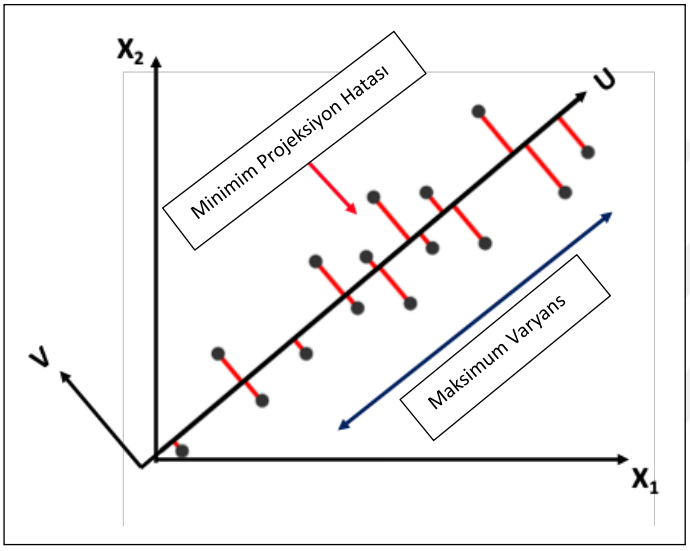
Negatif Korelasyon: Değişkenler zıt yönlerde değişiklik gösterir.

6.2.3. Temel bileşenler analizi (PCA)

Bu kapsamda kullanılan yöntemlerden bir tanesi olan Temel Bileşenler Analizi (PCA)'in amacı, öznitelik sayısı çok fazla olan veri setlerinde, yüksek varyans ile veri setini temsil edebilecek şekilde yeni öznitelikler oluşturmak ve verilerin sıkıştırılması ile boyut indirgemeyi sağlamaktır. Bu yöntemde bazı öznitelikler kaybolacaktır ancak bu öznitelikler

sınıf değerine önemli ölçüde katkı sağlamayan veya diğer bir deyişle sınıf değeri hakkında çok az bilgi içeren öznitelikler olmaları itibari ile göz ardı edilebileceklerdir. Ortaya çıkan yeni öznitelikler yüksek korelasyonlu değişkenleri içeren "temel bileşenler" kümesini oluştururlar [26].

PCA veri seti içindeki önem olan bilgileri ayırt edebilen etkili bir yöntemdir. Bu yaklaşımda temel mantık, verideki temel özellikleri barındıran daha az sayıda değişkenle çok boyutlu bir veriyi göstermektir [26].



Şekil 6.2. Temel bileşen analizi (PCA)

PCA yaklaşımında biz dizi matematik işlemi (Eigenvalue, varyasyon vb.) sonucunda hangi niteliklerin sınıflandırmaya veya etiketli veriye daha fazla etkisi olduğu ortaya koyulmaya çalışılır. Bu sayede etkisiz nitelikler veri setinden ayrılarak daha önemli nitelikler üzerinde çalışılabilecektir. Daha önemli nitelikleri ayırt edebilmek için, noktalar kümesinde ortalama uzaklığı bütün noktalara en düşük değere sahip "en uygun doğru" seçilir [31].

7. UYGULAMA

Açıklanan kavramlar ve metotlar ışığında farklı iki veri seti üzerinde sınıflandırıcıların ve öznitelik seçme yöntemlerinin etkileri incelenmiş ve çıkarımlar yapılmıştır.

7.1. Çalışmanın İçeriği

Çalışmada iki adet veri seti kullanılmıştır. Analizin tekdüze olmasını önlemek adına ilk veri setinde statik analiz metodu ile elde edilen öznitelikleri; ikinci veri setinde dinamik analiz metodu ile elde edilen öznitelikleri kullanılmıştır. Her iki veri setinin ilk olarak normalize edilmiş ham halleri, daha sonra Bilgi Kazanımı, Korelasyon ve Temel Bileşen Analizi öznitelik seçme yöntemleri uygulanmasına müteakip indirgenmiş boyutlu halleri üzerinde, denetimli (supervised) öğrenme yaklaşımından literatür taramasında en sık karşılaşılan sınıflandırıcı algoritmalarının (Naif Bayes, Bayes Ağı, Lojistik Regresyon, Destek Vektör Makineleri, C4.5 ve Rastgele Ağaç) performansları gözlemlenmiştir.

Deneylerde öznitelik değerlerinin birbirinden çok farklı değerlere sahip olmalarından ötürü analiz sonuçlarını olumsuz etkileyebileceği göz önüne alınarak değerler normalize edilmiştir. Veri setleri 10 lu çapraz doğrulama (cross validation) yöntemi ile kendi içinde eğitim ve test veri setlerine ayrılarak performans analizi yapılmıştır. Bu yaklaşım veri setinin yüzdesel olarak tek seferlik bölünmesi ile ortaya koyulacak olan performansına kıyasla daha güvenilir bir yöntem olduğu değerlendirilmektedir.

Birinci veri seti, açık kaynaktan elde edilmiştir [27]. Veri setinde bulunan öznitelikler statik zararlı yazılım analizi sonucunda elde edilmiştir.

İkinci veri seti açık kaynakta bulunan 50 farklı zararlı yazılım ve 17 Windows yazılımının sistem davranışlarının bir python kodu ile toplandığı kayıt dosyalarında bulunan kelimeler (strings) inden “kelime torbası (bag of words)” yöntemi ile oluşturulmuştur. En sık tekrarlanan ilk 200 kelime öznitelik olarak veri setinde sütunları; tekrarlanma sayıları da her bir zararlı/iyi yazılım kaydı için ilgili öznitelik değerini oluşturmaktadır.

Uygulamada, ilk veri seti ile bir yazılımın statik özelliklerinin; ikinci veri seti ile sistemde meydana gelen değişiklik kayıtlarının, yazılımın zararlı olup olmadığı konusunda ne derece kullanılabilir olduğu görülmeye çalışılmıştır. Makine öğrenmesi teknikleri öznitelik seçme

yöntemleri ile birlikte kullanılarak tahmin sonucunun mümkün olan en yüksek doğruluk ile elde edilmesine yönelik bir dizi işlem gerçekleştirilmiştir.

Çalışmada daha önceki çalışmalardan farklı olarak yeni bir veri seti oluşturulmasının yanında belirtilen üç öznitelik seçme yönteminin de karşılaştırılması yapılmıştır. Her iki veri seti de öncelikle normalize edilerek öznitelik elemesi yapılmadan sınıflandırıcılarda analiz edilmiştir. Daha sonra öznitelik seçme yöntemleri ile boyutu indirgenmiş veri setleri üzerinde sınıflandırıcılar tekrar analiz edilerek belirtilen öznitelik seçme yöntemlerinin sınıflandırıcıların performansına olan etkileri gözlemlenmiş ve sınıflandırıcıların performansları kıyaslanmıştır.

7.2. Çalışmanın Amacı

Çalışmanın ilk amacı veri setlerinde bulunan özniteliklerden hedef sınıfı belirleyenleri ayıklamak diğer bir deyişle hedef sınıfımıza etkisi en az olanları eleyebilmektir. Bu sayede sınıflandırıcılar gürültülü veriden ziyade daha sağlıklı veri ile çıktı üretebileceklerdir. Bu sürecin performans artırımını olumlu yönde etkileyeceği değerlendirilmektedir.

İkinci safhada sınıflandırıcıların veri setleri üzerindeki performanslarının karmaşıklık matrisi ile kıyaslanması hedeflenmektedir.

Çalışmada odaklandığımız noktalar;

- Öznitelik seçme metotlarının sınıflandırma sürecine etkisi
- Sınıflandırıcıların performans kıyaslaması olarak özetlenebilir.

Bununla birlikte makine öğrenimini, zararlı yazılımları tespit etmek ve sınıflandırmak için kullanmak, dinamik olarak ayıklanan davranışsal ve bellek özelliklerini kullanarak hedeflenen kötü amaçlı yazılım analizi ile ikili dosyaları iyi huylu ve kötü amaçlı olacak şekilde sınıflandırmaya odaklanmaktadır.

Özetle bu tez literatüre aşağıdaki katkıları yapar:

* Hedeflenen kötü amaçlı yazılımları klasik yöntemlerden ziyade yeni yaklaşımlar ile tespit etmek ve sınıflandırmak için bir yöntem sunar.

* Analiz sisteminin, yöntem adımlarının, çıkarılan özelliklerin ve makine öğrenimi algoritmalarının ayrıntılı bir açıklamasını sağlar.

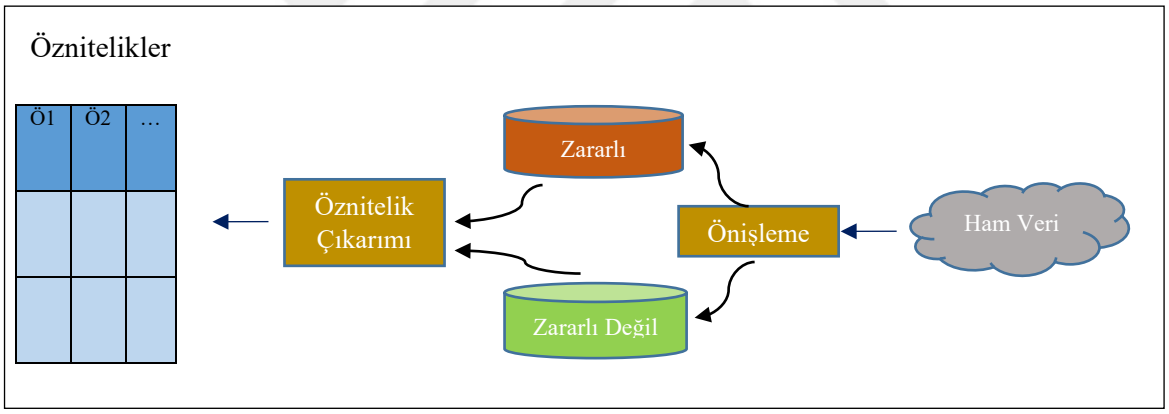
* Önerilen yöntemin kapsamlı bir değerlendirmesini sunar.

7.3. Birinci Veri Seti Analizi

Birinci veri setinin analizi için Windows 10 İşletim Sistemi üzerinde açık kaynak veri madenciliği yazılımı olan ve makine öğrenmesi kullanımına imkan veren Weka (versiyonunu da yazalım) kullanılmıştır. Sistem 16 GB RAM'e sahiptir.

İlk veri seti açık kaynaktan edindiğimiz bir veri setidir [27] Bu veri setinde 2501 adet iyi (benign) 2683 adet zararlı (malware) toplam 5184 örnek mevcuttur. Her bir örneğe ait 55 adet öznitelik ve 1 adet sınıf (iyi-zararlı) bulunmaktadır.

Analizde kullanılan veri setinin elde edilme süreci Şekil-7.1'de gösterilmiştir.



Şekil 7.1. Öznitelik elde etme adımları [27]

Öznitelik dağılımı aşağıda sunulmuştur:

Çizelge 7.1. Birinci veri seti öznitelikleri

IMAGE_DOS_HEADER (19 adet öznitelik)	FILE_HEADER (7 adet öznitelik)	OPTIONAL_HEADER (29 adet öznitelik)
e_magic e_cblp e_cp e_crlc e_cparhdr e_minallo e_maxallo c e_ss e_sp e_csum e_ip e_cs e_lfarlc e_ovno e_res e_oemid e_oeminfo e_res2 e_lfanew	Machine NumberOfSections CreationYear PointerToSymbolTable NumberOfSymbols SizeOfOptionalHeader Characteristics	Magic MajorLinkerVersion MinorLinkerVersion SizeOfCode SizeOfInitializedData SizeOfUninitializedData AddressOfEntryPoint" BaseOfCode BaseOfData ImageBase SectionAlignment FileAlignment MajorOperatingSystemVersion MinorOperatingSystemVersion MajorImageVersion MinorImageVersion MajorSubsystemVersion MinorSubsystemVersion SizeOfImage SizeOfHeaders CheckSum Subsystem DllCharacteristics SizeOfStackReserve SizeOfStackCommit SizeOfHeapReserve SizeOfHeapCommit LoaderFlags NumberOfRvaAndSizes

Veri seti incelendiğinde statik analiz sonucu programlar yürütülmeden elde edilmiş olan öznitelikler olduğu görülmektedir.

İstatistiklerde, bir aykırı değer, diğer gözlemlerden önemli ölçüde farklı bir veri noktasıdır. Bir aykırı değer, ölçümdeki değişkenliğe bağlı olabilir veya deneysel hatayı gösterebilir; ikincisi bazen veri kümesinden çıkarılır. Bir aykırı değer istatistiksel analizlerde ciddi sorunlara neden olabilir [28]. Bu kapsamda veri setimizde bulunan ayrık (outlier) veriler diğer çalışmalardan farklı olarak veri setlerinden çıkarılmamış; öznitelik seçme yöntemlerinin bu ayrık verileri eleyip elemediği analiz sonucunda ortaya koyulmuştur.

İlk olarak veri setinde herhangi bir öznitelik elemeden (55 özniteliğin hepsi ile) sınıflandırıcılar ile analiz edilmiştir. Yapılan analiz sonucunda elde edilen değerler:

Çizelge 7.2. Birinci veri setinin ilk halinde sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%94,46**	**%62,61**	**%95,85**	**%93,42**	**%96,83**	**%98,43**
Duyarlılık	%92,7	%64,3	%95,3	%90,8	%96	%97,9
Kesinlik	%95,7	%56,6	%96,1	%95,4	%97,4	%98,8
F-Score	%94,2	%59,4	%95,7	%86,9	%96,7	%98,4

Veri setinin ham hali ile yapılan analiz sonucunda “Random Forest algoritmasının” diğer sınıflandırıcılara göre doğruluk, duyarlılık ve kesinlik değerlerinin belirgin şekilde yüksek olduğu tespit edilmiştir.

Nitelikli öznitelik seçimi ile sınıflandırıcıların performanslarında değişikliği tespit edebilmek adına ilk olarak **infogain (bilgi kazanımı)** öznitelik değerlendirici ile en iyi 25 öznitelik seçilmiştir. Bu öznitelikler ve bilgi kazanım oranları:

Çizelge 7.3. Bilgi kazanımına göre seçilen en iyi 25 öznitelik

Bilgi Kazanım Oranı	Seçilen Öznitelik
0,4695	Characteristics
0,3883	DllCharacteristics
0,2942	CheckSum
0,2762	ImageBase
0,2565	MinorLinkerVersion
0,2466	SizeOfStackReserve
0,2049	AddressOfEntryPoint
0,2017	MajorImageVersion
0,1949	CreationYear
0,1942	MajorOperatingSystemVersion
0,1865	MajorLinkerVersion
0,1762	MajorSubsystemVersion
0,1606	SizeOfStackCommit
0,1581	Subsystem
0,1571	SizeOfImage
0,152	MinorImageVersion
0,1465	MinorOperatingSystemVersion
0,1448	MinorSubsystemVersion
0,1371	SizeOfInitializedData
0,1218	NumberOfSections
0,1144	SizeOfCode
0,095	SizeOfHeapReserve
0,093	e_lfanew
0,0812	BaseOfData

Elde edilen bilgi kazanım oranları incelendiğinde veri setindeki özniteliklerin nispeten düşük bilgi kazanımına (yüksek entropi) sahip oldukları görülmektedir. Bu şekilde sınıflandırıcılar ile yapılan analiz sonucunda elde edilen değerler:

Çizelge 7.4. Bilgi Kazanımına Göre Sınıflandırıcıların Elde Ettiği Değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%94,65**	**%68,83**	**%95,71**	**%93,44**	**%96,93**	**%98,3**
Duyarlılık	%92,1	%67,6	%94,8	%90,8	%96	%97,6
Kesinlik	%96,6	%67,3	%96,3	%95,4	%97,7	%98,7
F-Score	%94,3	%67,4	%95,5	%93	%96,8	%98,2

Değerler incelendiğinde doğruluk oranının tek başına yeterli olmamasına rağmen, doğruluk oranı düşük olduğunda kesinlik ve/veya duyarlılık değerinin de düşük olduğu görülmektedir. Bu çizelgede Naif Bayes sınıflandırıcısının zararlı sınıfı için yanlış negatif (FN) sayısını çok yüksek hesapladığı ve buna bağlı olarak doğruluk başta olmaz üzere diğer istatistiksel değerlerinin de düşük olduğu tespit edilmiştir. Yani diğer bir deyişle zararlı olduğu bilinen örneklerden birçoğunun iyi olarak sınıflandırıldığı söylenebilir.

Infogain öznitelik değerlendiricinin seçtiği öznitelikler ile yapılan analiz veri setinin ilk hali ile kıyaslandığında çok belirgin bir iyileşme olmadığı görülmüştür. Bilgi kazanım değerleri incelendiğinde en yüksek bilgi kazanımının bile 0,5 ten düşük olduğu, dolayısı ile sınıf değeri hakkında belirgin seviyede bilgi sahibi olan bir özniteliğin bulunmadığı söylenebilir. Buna rağmen özellikle Naif Bayes algoritması için %6 oranında bir iyileşme sağladığı görülmüştür. Bunun nedeni olarak bilgi kazanımı ile ayrık verilerin bir ölçüde veri setinden çıkarılması gösterilebilir.

Bununla birlikte diğer sınıflandırıcıların başarı oranlarında çok belirgin bir düşüş/yükseliş tespit edilmemiştir.

Özniteliklerin bağımsızlığı prensibine göre sınıflandırma yapan bir algoritma olduğu için bilgi kazanımı ile öznitelik seçiminin, deneysel bir eşik değeri ile sınırlandırılmadığı sürece

performansı pozitif yönde etkilemediği çıkarımı yapılabilir. Bu önermenin ispatı niteliğinde yapılan çalışmada bilgi kazanımı 0,2 üstü olan öznitelikler seçilmiştir.

Çizelge 7.5. Bilgi kazanımında eşik değerine (0.2) göre seçilen en iyi öznitelikler

Bilgi Kazanım Oranı	Seçilen Öznitelik
0,4695	Characteristics
0,3883	DllCharacteristics
0,2942	CheckSum
0,2762	ImageBase
0,2565	MinorLinkerVersion
0,2466	SizeOfStackReserve
0,2049	AddressOfEntryPoint
0,201	MajorImageVersion

Bu öznitelikler ile yapılan analizde Naif Bayes sınıflandırıcısında doğruluk, kesinlik (precision) ve Duyarlılık (recall) değerlerinin %85 seviyesine ulaştığı görülmüştür. Öznitelik seçme algoritmaları, farklı matematiksel yaklaşımlar ile özniteliklerin sınıf değerlerine etkisine göre sıralanabilmesi maksadıyla kullanılır. Eşik değeri olarak performansa etki edebilecek öznitelik sayısı veya belirlenecek bir oranın üzerindeki (bilgi kazanımı, korelasyon, skor vb.) özniteliklerin seçilmesi ile sınıflandırıcıların daha yüksek doğruluk ve duyarlılık ile çıktı üretmeleri sağlanabilecektir [37].

Nitelikli öznitelik seçimi ile sınıflandırıcıların performanslarında değişikliği tespit edebilmek adına ikinci olarak **korelasyon** öznitelik değerlendirici ile en iyi 25 öznitelik seçilmiştir. Bu öznitelikler ve bilgi kazanım oranları:

Çizelge 7.6. Korelasyona göre seçilen en iyi 25 öznitelik

Korelasyon Değerleri	Seçilen Öznitelik
0,3801	ImageBase
0,3376	MinorSubsystemVersion
0,3191	DllCharacteristics
0,3004	MajorSubsystemVersion
0,2805	MinorOperatingSystemVersion
0,2662	Subsystem
0,2572	Characteristics
0,2439	SizeOfStackReserve
0,2357	SizeOfHeapCommit
0,2352	SizeOfHeapReserve
0,2117	MajorOperatingSystemVersion
0,1964	SectionAlignment
0,1921	CreationYear
0,188	FileAlignment
0,17	e_ovno
0,1296	e_lfanew
0,1248	NumberOfSections
0,0993	MinorLinkerVersion
0,0928	MajorLinkerVersion
0,0769	e_cparhdr
0,0698	e_lfarlc
0,0695	e_maxalloc
0,0649	SizeOfImage
0,0589	BaseOfData
0,0569	SizeOfInitializedData

Elde edilen korelasyon oranları incelendiğinde veri setindeki özniteliklerin sınıf değeri (iyi/zararlı) ile düşük korelasyona sahip oldukları görülmektedir. Bu şekilde sınıflandırıcılar ile yapılan analiz sonucunda elde edilen değerler:

Çizelge 7.7. Korelasyona göre sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%93,96**	**%75,93**	**%95,48**	**%93,5**	**%96**	**%97,8**
Duyarlılık	%92,4	%84	%94,3	%90,9	%94,8	%97,2
Kesinlik	%94,9	%78,2	%96,3	%95,4	%96,8	%98,3
F-Score	%93,7	%80,8	%95,3	%93,1	%95,8	%97,7

Korelasyon öznitelik değerlendiricinin seçtiği öznitelikler ile yapılan analiz sonucunda göze çarpan ilk husus Naif Bayes sınıflandırıcısının performansında "Bilgi Kazanımı" metoduna kıyasla gerçekleşen iyileşmedir. Naif Bayes her bir özniteliğin birbirinden bağımsız olduğu prensibine dayanarak çalışan bir algoritmadır [12]. Korelasyon katsayılarına bakıldığında en yüksek değerin (0,38) de nispeten düşük bir değer olmasına rağmen, sınıf değerini doğru orantılı olarak belirleyen özniteliklerin bulunduğu ve veri setinin bu doğrultuda iyileştirildiği söylenebilir. Elde edilen bu sonuç ile, matematiksel değer olarak bilgi kazanımına göre daha düşük değerler ortaya korelasyon yönteminin Naif Bayes performansına daha olumlu katkı sağladığı sonucuna da ulaşılmıştır.

Bununla birlikte diğer sınıflandırıcıların başarı oranlarında çok belirgin bir düşüş/yükseliş tespit edilmemiştir.

Nitelikli öznitelik seçimi ile sınıflandırıcıların performanslarında değişikliği tespit edebilmek adına son olarak Temel Bileşen Analizi (PCA) öznitelik değerlendirici, sezgisel olarak belirlenmiş eşik değeri ("1") ile çalıştırılmış böylece ortaya çıkan yapay değişkenlerden en iyi 25 öznitelik seçilmiştir. Bu öznitelikler ve bilgi kazanım oranları:

Çizelge 7.8. PCA'e göre oluşturulan öznitelikler ve sapma değerleri

Standart Sapma	**PCA ile Oluşturulan Yeni Yapay/Yeni Değişkenler**
2,513	Öznitelik 1
2,279	Öznitelik 2
2,108	Öznitelik 3
1,918	Öznitelik 4
1,876	Öznitelik 5
1,822	Öznitelik 6
1,749	Öznitelik 7
1,688	Öznitelik 8
1,571	Öznitelik 9
1,551	Öznitelik 10
1,533	Öznitelik 11
1,47	Öznitelik 12
1,415	Öznitelik 13
1,4	Öznitelik 14
1,383	Öznitelik 15
1,361	Öznitelik 16
1,327	Öznitelik 17
1,298	Öznitelik 18
1,279	Öznitelik 19
1,273	Öznitelik 20
1,242	Öznitelik 21
1,234	Öznitelik 22
1,2	Öznitelik 23
1,19	Öznitelik 24
1,187	Öznitelik 25

Elde edilen öznitelikler ile amaçlanan sınıf değerini etkileyen ve mevcut özniteliklerden daha etkili yeni öznitelikler oluşturarak veri setini daha sağlıklı hale getirebilmektir. Bu şekilde sınıflandırıcılar ile yapılan analiz sonucunda elde edilen değerler:

Çizelge 7.9. PCA'e göre sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%92,4**	**%83,1**	**%93,7**	**%92**	**%95**	**%96,91**
Duyarlılık	%91,6	%83,7	%91,8	%87,8	%94,6	%96,7
Kesinlik	%92,6	%81,7	%95	%95,4	%94,9	%96,9
F-Score	%92,1	%82,7	%93,4	%91,4	%94,8	%93,8

Temel Bileşen Analizi öznitelik değerlendiricinin seçtiği öznitelikler ile yapılan analiz sonucunda göze çarpan ilk husus Naif Bayes sınıflandırıcısının veri setinin ilk haline ve infogain metoduna kıyasla performansında gerçekleşen iyileşmedir. Veri seti özniteliklerinin sınıf değerine göre bilgi kazanımı oranlarının yüksek olmadığını belirtmiştik. Özniteliklerin sınıf değerine etkilerinin bireysel değerlendirildiği Naif Bayes gibi analiz metotlarında bu sebeple düşük performans elde edildiği görülmektedir. PCA metodunda ise veri seti, tüm özellikleri korunurken daha az veri ve daha geniş varyans ile temsil edilmesi amaçlanmaktadır. Bu sayede özellikle Naif Bayes gibi bağımsız öznitelik prensibine dayanan algoritmalarda her bir öznitelik daha önemli hale gelebilmektedir. Karmaşıklık matrisinde görülen belirgin artışın sebebi bu şekilde açıklanabilir.

Diğer sınıflandırıcılar arasında Rastgele Orman algoritmasının en başarılı sınıflandırıcı olduğu görülmektedir. Farklı karar ağaçları ile her bir girdinin sonucu değerlendirilerek en çok seçilen çıktı algoritmanın çıktısı olarak belirlenmektedir. Bununla birlikte analiz sırasında seçilen 10 lu çapraz doğrulama diğer bir kontrol mekanizması olarak kullanılmaktadır. Algoritma yeteneği ile bahse konu doğrulama mekanizması birlikte çalıştığında bu sonucun elde edilmesinin beklenen durum olduğu söylenebilir.

Bununla birlikte diğer sınıflandırıcıların başarı oranlarında çok belirgin bir düşüş/yükseliş tespit edilmemiştir.

Diğer sınıflandırıcıların önemsiz öznitelikleri analiz sürecinde elediği göz önüne alındığında, öznitelik seçme yöntemleri ile çok belirgin bir iyileşme sağlanmadığı sonucuna varılmıştır.

Ayrıca aynı veri seti üzerinde yapılan diğer bir çalışma incelenmiştir [48]. Bu çalışmada da Rastgele Orman algoritmasının diğer etiketli sınıflandırma tekniklerine göre %98 doğruluk oranı ile daha iyi bir performans gösterdiği görülmüştür. Bu kıyaslama, çalışmada elde edilen sonuçların, uygulanan yaklaşımın ve öznitelik seçme yöntemlerinin tutarlı olduğu hipotezini güçlendirmiştir.

7.4. İkinci Veri Seti Analizi [45]

İkinci veri setimizde metodolojiye sıfırdan başlayarak ilerlemek esas alınmıştır. Bu maksatla kendi veri setimizi oluşturmak adına Ubuntu 18.04.4 işletim sistemi üzerinde VirtualBox, Cuckoo Sandbox 2.0.7 ve Zero Wine kurulmuştur. Bahse konu sanal ortamlar içinde dinamik analiz gereği Windows işletim sisteminde çalışabilecek şekilde oluşturulmuş 67 adet PE dosyası (17 iyi [40], 50 zararlı [39]) kum havuzunda çalıştırılarak işletim sisteminde meydana gelen değişiklikler (sistem çağrıları (API calls), kütüphaneler (.dll), kayıt defteri değişiklikleri vb.) 67 farklı kayıt dosyasında saklanmıştır. Müteakiben bahse konu kayıt dosyaları üzerinde Python 3.6 programlama dilinde yazılan betik çalıştırılarak 67 örnekli, 200 öznitelikli ve 2 sınıflı bir veri seti elde edilmiştir. Zararlı yazılımların sanal ortamı algılayabilecek yetenekleri göz önüne alınarak sanal makinelerde 2 CPU ve 2048 MB RAM olacak şekilde konfigüre edilmiştir. Aynı zamanda programların çalışma zamanında olası insan hareketleri (Mouse hareketi vb.) otomatik olarak uygulanmaktadır. Sınıflandırıcı scriptleri Python 3.6 programlama dilinde yazılmış ve karmaşıklık matrisi oluşturulmuştur. Bununla birlikte Weka, sınıflandırma analizi için ayrıca kullanılmıştır.

Kullandığımız phyton kod parçası ile kayıt dosyalarında geçen tüm kelimeler tekil (distinct) diğer bir deyişle tekrarlamalı olmayacak şekilde çıkarılarak (extract) bir liste oluşturulmuştur. Bunların arasından en sık tekrarlanan 200 kelimeyi öznitelik olarak çıkarmaktadır. Bu literatürde kelime torbası (bag of words) metodu olarak bilinmektedir [30] Bahse konu yaklaşımda dokümanlarda bulunan kelimelerin dizilişi veya içerdiği anlam (semantic) yerine sadece kelimelerin birbirinden bağımsız olarak tekrarlanma sayıları dikkate alınır. Diziliş ve anlamsal farklılıklar doğal dil işleme problemlerinde dikkate alınmaktadır [38] Daha sonra bir vektör/öznitelik olarak kullanılacak bu kelimelerin

tekrarlanma sayılarının (Term Frequency/TF IDF) öznitelik değeri olduğu bir csv dosyası oluşturmaktadır. İlk veri setinde olduğu gibi bu veri setimizde de etiketli veri yani sınıflarımız (iyi "-1"; zararlı "1") olarak tanımlanmıştır. İşlemi yapan python kodu EK-1'de sunulmuştur.

Oluşan veri setimizde değerler normalize edilmeye müteakip aynı sınıflandırıcılar ile analiz yapılmıştır:

Veri setinin geçerliliğini kontrol etmek maksadıyla tüm veri seti kendi içinde eğitim ve test veri kısımlarını içerecek şekilde 10'a bölünerek (10-fold cross validation) Naif Bayes, Bayes Ağı, Lojistik Regresyon, Destek Vektör Makinesi ve Rastgele Orman sınıflandırıcıları performansları karşılaştırılmıştır.

Çizelge 7.10. İkinci veri setinin ilk halinde sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%77,6**	**%64**	**%85,5**	**%91**	**%92,5**	**%94**
Duyarlılık	%72	%66	%84,2	%94	%94	%94,2
Kesinlik	%97,3	%68,5	%85,3	%94	%96	%93,8
F-Score	%82,8	%67,4	%85,1	%94	%95	%94

Elde edilen sonuçlar incelendiğinde Rastgele Orman sınıflandırıcısının en yüksek doğruluk, kesinlik, duyarlılık ve F-skor değerlerine sahip olduğu görülmektedir. Bununla birlikte oluşturduğumuz veri setinin sınıflandırma analizlerine uygun olduğunu ve elde edilen karmaşıklık matrisi değerlerinin tatmin edici seviyede olduğu söylenebilir.

Bir sonraki aşamada sırası ile bilgi kazanımı (infogian), korelasyon ve temel bileşen analizi ile veri setimizde bulunan 200 öznitelik arasından en iyi 15 (eşik değeri) öznitelik seçilmiştir.

Çizelge 7.11. Bilgi kazanımına göre sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%88**	**%94**	**%91**	**%92,5**	**%92,5**	**%95,5**
Duyarlılık	%86	%94	%92	%96	%94	%98
Kesinlik	%97,7	%98	%96	%94,1	%96	%96,1
F-Score	%91,5	%96	%94	%95	%95	%97

Bilgi kazanımına göre seçilen öznitelikler ile yapılan analizde Rastgele Orman sınıflandırıcısı en iyi değerlere sahip olmakla birlikte, diğer tüm sınıflandırıcılarda performansın yükseldiği görülmektedir. Yöntem ile veri setinde bulunan ayrık verilerin elendiği sonucuna varılabilir. Bilgi kazanım oranlarının 0,6-0,4 aralığında olduğu görülmüştür.

Çizelge 7.12. Korelasyona göre sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%89,5**	**%91**	**%91**	**%94**	**%92,5**	**%94,3**
Duyarlılık	%88	%90	%92	%96	%94	%96
Kesinlik	%97,8	%97,8	%96	%96	%96	%96
F-Score	%92,6	%93,8	%94	%96	%95	%95,4

Korelasyona göre seçilen öznitelikler ile yapılan analizde Rastgele Orman ile Destek Vektör Makinesi sınıflandırıcıları en iyi değerlere sahip olmakla birlikte, diğer tüm sınıflandırıcılarda performansın ilk duruma göre yükseldiği görülmektedir. Korelasyın oranlarının 0,8-0,6 aralığında olduğu görülmüştür. Korelasyon değerlerine göre seçilen özniteliklerin sınıf değerini önemli ölçüde etkilediğini söylenebilir.

Çizelge 7.13. PCA'e göre sınıflandırıcıların elde ettiği değerler

Sınıflandırıcı / Değer	Bayes Ağı	Naif Bayes	Lojistik Regresyon	SVM	C4.5	Random Forest
Doğruluk	**%80,6**	**%88**	**%86,6**	**%92,5**	**%92,7**	**%94,6**
Duyarlılık	%84	%96	%88	%96	%92	%92
Kesinlik	%89,4	%88,9	%93,6	%94,1	%92	%93,6
F-Score	%86,6	%92,3	%90,7	%95	%91,6	%92,3

Temel Bileşen Analizi'ne göre seçilen öznitelikler ile yapılan analizde Rastgele Orman sınıflandırıcısının en iyi performansa sahip olduğu görülmektedir. Bununla birlikte diğer sınıflandırıcıların performanslarında ilk duruma göre bir miktar iyileşme olduğu tespit edilmiştir.

8. SONUÇ VE ÖNERİLER

Günümüzde zararlı yazılımların gün geçtikçe daha karmaşık ve klasik tespit sistemlerini yanıltıcı hale gelmesinden ötürü makine öğrenmesi ile zararlı yazılım tespiti bir seçenekten ziyade zorunluluk haline gelmiştir. Bu kapsamda statik ve dinamik analiz yöntemleri ile zararlı yazılımların sahip olduğu ve/veya çalıştırılmasına müteakip sistemde sebep olduğu değişiklikler elde edilerek büyük veri analizi/veri madenciliği yaklaşımları ile analiz edilmektedir. Bu yaklaşımlarda zararlı yazılımları en fazla temsil edebilecek özellikleri (öznitelikleri) bulacak bir model geliştirmek ve yeni bir yazılımın zararlı olup olmadığını mümkün olan en yüksek doğruluk oranı ile tespit edebilmek hedeflenmektedir.

Bu kapsamda statik veriler ile oluşturulan bir veri seti, dinamik analiz sonucu oluşturulan diğer veri seti ile aynı yöntemlerle iki ayrı analiz gerçekleştirilmiştir. Her iki yöntemde seçilen öznitelik seçme ve sınıflandırma algoritmalarının sonuçları karşılaştırılarak çıkarımlar yapılmıştır. Her iki çalışmada da Rastgele Orman sınıflandırıcısının diğerlerine göre daha iyi sonuçlar ürettiği sonucuna varılmıştır. Bununla birlikte öznitelik seçme (feature selection) yöntemleri birbirlerine bariz bir üstünlük kurmadıkları ancak veri setinin ilk hali ile kıyaslandığında sınıflandırıcıların performansında bir iyileştirme sağladıkları tespit edilmiştir.

Öznitelik seçme sürecinde eşik değeri (öznitelik sayısı ve/veya bilgi kazanım/korelasyon vb. oran) belirlemenin önemi ortaya koyulmuştur. Naif Bayes algoritmasının özniteliklerin bağımsız olarak sınıf değerini belirlemesi yaklaşımına bağlı olarak diğerlerine göre nispeten daha düşük sınıflandırma kabiliyeti olduğu ancak tecrübi olarak seçilecek özniteliklere ve özniteliklerin sınıf değerine etkisine göre Naif Bayes algoritmasının da yüksek doğruluk oranına sahip olabileceği görülmüştür [45]. Aynı zamanda ayrık verilerin öznitelik seçme yöntemleri ile elenmesinin de Naif Bayes performansına olumlu katkı sağladığı görülmüştür. Bu iyileşmenin diğer sınıflandırıcı performanslarında nispeten daha düşük seviyede olduğu; bu durumun, diğer sınıflandırıcıların analiz sürecinde etkisi düşük öznitelikleri hesaplamalarının dışında tutması (budaması) sonucu gerçekleştiği sonucuna varılmıştır.

Çalışmada kullanılan veri setinin oluşturulması için kullanılan String/Kelime Torbası mevcut olan birçok yöntemden bir tanesidir. Elde edilen öznitelikler ve değerleri ile yapılan analizlerin sonuçları tatmin edici seviyede olmakla birlikte, zararlı yazılım konusunda

sistemlerin daha yüksek tespit oranına sahip algoritmalar ile korunması gerekmektedir. Bu kapsamda farklı yöntemler ve öznitelikler ile elde edilecek yeni veri setleri ve analiz için kullanılacak etiketsiz sınıflandırma algoritmaları gibi farklı algoritmalar ile %100 e yakın doğrulukla sonuçlar elde edilmesi bu alanda yapılacak müteakip çalışmalarda hedeflenmelidir. Bu çerçevede analizler sonucunda yalnızca doğruluk oranı değil yanlış negatif (FN) oranını da kapsayan duyarlılık (Recall) değeri de göz önüne alınmıştır. Öznitelik seçme algoritmaları ile optimize edilen veri setlerinin bu kapsamda da tatmin edici (%90 üzeri) seviyede duyarlı sonuç ürettiği söylenebilir.

Çalışmada kullanılan etiketli sınıflandırma teknikleri farklı alanlarda (hastalık teşhisi [46], balık yaş gruplandırma [47] vb.) da kullanılmıştır. Bahse konu çalışmalarda elde edilen sonuçlar incelenerek bu çalışmada varılan sonuçların tutarlılığı ayrıca incelenmiştir. Bahse konu çalışmalardan hastalık teşhisi analizinde [46] genetik algoritma öznitelik seçme yöntemi temelli rasgele orman algoritmasının ve balıkların 3 özniteliğe bağlı yaş gruplarına ayrılması analizinde [47] yine rastgele orman algoritmasının nispeten daha yüksek doğruluk oranına sahip olduğu tespit edilmiştir.

Bununla birlikte zararlı yazılım tespitinde statik analiz yöntemi ile elde edilen özniteliklerin bulunduğu ayrı bir veri setinde yapılan analizde [49] RF algoritmasının Sci-kit kütüphanesi kullanıldığında %99 civarında doğruluk ve duyarlılık oranı ile diğer etiketli sınıflandırma algoritmalarından daha iyi bir sonuç verdiği tespit edilmiştir.

Yapılan karşılaştırmalar, çalışmada kullanılan veri setinin ve elde edilen sınıflandırıcı performanslarının doğruluk/duyarlılık oranlarını pekiştirir niteliktedir.

Müteakip çalışmalarda farklı işletim sistemlerine uyumlu zararlı yazılımlar üzerinde derin öğrenme ve yapay sinir ağları ile yapılacak analizlerin literatüre katkı sağlayabileceği; doğruluk, duyarlılık ve kesinlik değerlerini artırabileceği öngörülmektedir.

KAYNAKLAR

1. Spafford, E. H. (1989). The Internet Worm Program: An Analysis, SIGCOMM Comput. Commun. Rev., 19 (1), 17-57.

2. Schultz, M. G., Eskin, E., Zadok, E., and Stolfo, S. J. (2001). Data Mining Methods for Detection of New Malicious Executables. *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 23-26.

3. Kolter, J. Z., and Maloof, M. A. (2004). Learning to Detect Malicious Executables in the Wild. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 8-11.

4. Perdisci, R., Lanzi, A., and Lee, W. (2008). McBoost: Boosting Scalability in Malware Collection and Analysis Using Statistical Classification of Executables, *Computer Security Alications Conference*, 7-9.

5. Shafiq, M., Tabish, S., Mirza, F., and Farooq, M. (2009). Lecture Notes On computer Science. *PE-Miner: Mining Structural Information to Detect Malicious Executables*, 121-141

6. İnternet: Forcepoint, Advanced Malware Detection, URL: https:// www. forcepoint. com / product/advanced-malware-detection Son Erişim Tarihi: 19.12.2020

7. Çatak,F.Ö. (2019). Classification of Methamorphic Malware with Deep Learning Conference Paper, 2-4

8. İnternet: EDUCBA Corp., "What is Big Data Analysis", https://www.educba.com/what-is-big-data-analytics/ Son Erişim Tarihi 19.12.2020

9. Zoi, T., Ceccarelli A. (2001). "Prepare for trouble and make it double! Supervised – Unsupervised stacking for anomaly-based intrusion detection", Journal of Network and Computer Alications, 189 (1), 13-16.

10. İnternet: " Supervised vs Unsupervised Learning: Key Differences", https://www. guru99.com/supervised-vs-unsupervi-learning.html Son Erişim Tarihi: 13.11.2020

11. İnternet: "Denetimsiz öğrenme (Unsupervised learning)", https://yavuz.github.io/denetimsiz-ogrenme/ Son Erişim Tarihi:16.10.2020

12. Dehghantanha A., Dargahi T., Conti M., (2018). Cyber Threat Intelligence. 70(1), 8-45.

13. Kononenko, I, and Kukar, M. (2007). Machine learning and data mining: introduction toprinciples and algorithms. Horwood Publishing, 12-15.

14. RC Quinlan. (1993) 4.5: Programs for machine learning morgan kaufmann publishers inc. *San Francisco, United States of America*, 5-7.

15. Gibert, D., Mateu, C., Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Alications*, 153.

16. R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, J. M. Pedersen and A. Czech, (2015). "Analysis of Malware behavior: Type classification using machine learning," *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 1-7.

17. R. S. Pirscoveanu, S. S. Hansen, T. M. T. Larsen, M. Stevanovic, J. M. Pedersen and A. Czech, (2015). "Analysis of Malware behavior: Type classification using machine learning," *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 1-7.

18. A.Afianian, S.Niksefat, S. (2020, November). Malware Dynamic Analysis Evasion Techniques: A Survey", ACM Computing Surveys 52(6), 1–28

19. İnternet: T.Emmanuel, "Why A Malware Solution Must Include A Machine Learning Component", https://hub.packtpub.com/emmanuel-tsukerman-on-why-a-malware-solution-must-include-a-machine-learning-component/
Son Erişim Tarihi: 30.12.2020

20. Hall, M.A. (1999, April). Correlation-based Feature Selection for Machine Learning", The University of Waikato, 5(2), 132-133

21. İnternet: Deep Learning: A Crash Course https://www.youtube.com/watch?v=r0Ogt-q956I Son Erişim Tarihi:15.01.2021

22. Platt, J. (1999). Probabilistic outputs for suort vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*. United States of America:MIT Press, 121-133.

23. İnternet: Bayes Server Websitesi, "Feature importance", https: //www .bayesserver. com /docs /learning/feature-selection Son Erişim Tarihi:25.11.2020

24. İnternet: B.Jason, "How to Perform Feature Selection With Machine Learning Data in Weka" https://machinelearningmastery.com/perform-feature-selection-machine-learning-data-weka/ Son Erişim Tarihi: 12.12.2020

25. Hall, M.A. (1999, April). Correlation-based Feature Selection for Machine Learning", The University of Waikato, 61-67

26. Smith, S., Perrin, S., Dyregrov, A. (2003). Principal components analysis of the impact of event scale with children in war. *Personality and Individual Differences,* 34(2), 315-322

27. İnternet: Urwithajit; "Classification of Malware with PE headers", https://github.com/ Son Erişim Tarihi:10.12.2020

28. Yu, D., Sheikholeslami, G., Zhang, A. (2002). FindOut: Finding Outliers in Very Large Datasets. *Knowl Inform Sys 4*, 387–412.

29. İnternet: Y.Tony, “How the Algorithm Works and Why it Is So Effective” https://towardsdatascience.com/understanding-random-forest-58381e0602d2 Son Erişim Tarihi: 15.01.2021

30. Tsai, C., Bag-of-Words Representation in Image Annotation: A Review”, International Scholarly Research Network ISRN Artificial Intelligence, 2012(1), 13-15.

31. Paul,G., Kim, E., Svante, W. (1987, August). Chemometrics and Intelligent Laboratory Systems (2nd Ed.), Netherlands: Elsevier Science Publishers, 37-52

32. Danny, R., Grigoris, K., Chawla, V. (2006, November). Information Gain, Correlation and SupportVector Machines (3rd Ed.), Berlin: Springer Heidelberg, 463-470

33. S.Sharma, C. Rama, S. K. Sahay, K. (2019, March). Detection of Advanced Malware by Machine Learning Techniques. 5-8.

34. Shabtai, A., Moskovitch, R., Elovici, Y., Glezer, C., 2009. Detection of Malicious Code by Alying Machine Learning Classifiers on Static Features: A State-Of-The-Art Survey. Information Security Technical Report 14 (1). , . 16–29 malware http:// www.sciencedirect.com/science/article/pii/S1363412709000041.

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

36. Yanfang Ye, Lifei Chen, Dingding Wang, Tao Li, Qingshan Jiang, Min Zhao, (2009). *An interpretable string based malware detection system using SVM ensemble with bagging*, 6-11.

37. Mrs. Radha R, Mr. Muralidhara S. (2016, July). Removal Of Redundant And Irrelevant Data From Traınıng Datasets Usıng Speedy Feature Selectıon Method, IJCSMC, 5(7), 359 – 364

38. Prof. Calix Purdue University Northwest:” Machine Learning for Cyber Security: Naive Bayes https://www.youtube.com/watch?v=goovixQBb2k Son Erişim Tarihi: 30.01.2021

39. İnternet: VX malware archive: https://vx-underground.org/apts.html Son Erişim Tarihi: 22.01.2021

40. İnternet: Microsoft Utilities, https://Docs.Microsoft.Com/En-Us/Sysinternals/ Downloads, Son Erişim Tarihi: 25.01.2021

41. Ijaz, M., Durad, H., and Ismail, M. (2019). "Static and Dynamic Malware Analysis Using Machine Learning," *2019 16th International Bhurban Conference on Alied Sciences and Technology (IBCAST)*, 687-691.

42. Clouston, M., Haslhofer, B. (2019) “Ransomware payments in the Bitcoin ecosystem”, Journal of Cybersecurity, 1–11.

43. Chale M., Stewart J., Gibson D., Certified Information Systems Security Professional, United States of America:Sybex, (8th ed. 633-641)

44. Chale M., Stewart J., Gibson D., Certified Information Systems Security Professional, United States of America:Sybex, (8th ed. 645)

45. Gökkis F. (Ağustos 2021). "Öznitelik Seçme Yöntemlerinin Sınıflandırıcıların Performansına Etkisi"; *ICAR-Uluslararası Akademik Araştırmalar Kongresi*; 72 (*Bildiri bu tezden üretilmiştir*).

46. Azar Ahmad, Elshazly Ismail, Hassanien Aboul, Elkorany Abeer, 'A random forest classifier for lymph diseases', *Science Direct 2014*. 465-473

47. Benzer R., Benzer S., (2022). Investigation of Some Machiene Learning Algorithms in Fish age Classification. *Fisheries Research 2022*, 245 (1), 106151.

48. Kumar Ajit, Kuusamy K.S., 'A learning model to detect maliciousness of portable executable using integrated feature set', Journal of King Saud University 31(1), 252

49. Gülburun S., Dener M. (2021). Büyük Veri Ortamlarında Zararlı Yazılım Tespiti Kapsamında Makine Öğrenmesi Algoritmalarının Performansının İncelenmesi. El-Cezeri (8), 1536-1549

50. Gökkis F., Çetinyokuş T. (Nisan 2022). J48 ve Rastgele Orman Sınıflandırıcılarının Bilgi Kazanımı Ve Temel Bileşen Analizi Yöntemleri İle Oluşturulan Farklı Boyutlu Veri Setleri Üzerinde Performanslarının Kıyaslanması. *7'nci Uluslararası Mühendislik ve Teknoloji Kongresi*, s.411-418 (*Bildiri bu tezden üretilmiştir*).

EKLER

EK-1. Kayıt dosyalarından veri seti oluşturan python kod parçacığı

```
import os
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer # kelime torbası için#
Dizin='kayıt_dosyaları'
etiket = []
metin = []
#Dizin = ('kayıt_dosyaları')
for dosya_a in os.listdir(Dizin):
        if "m" in dosya_a:
            labels.aend("-1")
        else:
            labels.aend("1")
        dosya_a = os.path.join(Dizin, dosya_a)
        print(dosya_a)
        with open(dosya_a) as f:
            içerik= f.read()
        icerik.replace(",", " ") # "," karakterini silmek ve aralara boşluk koymak için#
        icerik.replace("", " ") # " ' " karakterini silmek ve aralara boşluk koymak için#
        metin.aend(icerik)  # dosya içeriğini cümleye çevirmek için#
vectorizer = CountVectorizer(stop_words='english', max_features=200)
 dtm = vectorizer.fit_transform(text)
 df = pd.DataFrame(dtm.toarray(), index=etiket, \
                    columns=vectorizer.get_feature_names()) #veri seti burada oluşuyor#
df.index.name = "labels"
df.to_csv(r'BilesikMatris.csv')
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : GÖKKİS, Fırat

Uyruğu : T.C.

Eğitim

Derece	**Eğitim Birimi**	**Mezuniyet tarihi**
Lisans	Kara Harp Okulu/Sistem Mühendisliği	2008
Lise	Işıklar Askeri Lisesi	2004

İş Deneyimi

Yıl	**Yer**	**Görev**
2008-2022	K.K.K.lığı	Bilgi Sistem Personeli

Yabancı Dil

İngilizce

Tezden Üretilen Yayınlar

Gökkis, F. (2021). Öznitelik Seçme Yöntemlerinin Sınıflandırıcıların Performansına Etkisi. *ICAR-Uluslararası Akademik Araştırmalar Kongresi* (Ağustos 2021), 72-78

Gökkis, F., Çetinyokuş T. (2022). J48 ve Rastgele Orman Sınıflandırıcılarının Bilgi Kazanımı ve Temel Bileşen Analizi Yöntemleri İle Oluşturulan Farklı Boyutlu Veri Setleri Üzerinde Performanslarının Kıyaslanması. *7'nci Uluslararası Mühendislik ve Teknoloji Kongresi* (Nisan 2022), 411-418

Hobiler

Futbol, bilgisayar teknolojileri, müzik

GAZİLİ OLMAK AYRICALIKTIR…