T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE İÇERİK TABANLI SİBER TEHDİT TESPİTİ

Emre KOÇYİĞİT

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman

Prof. Dr. Banu DİRİ

Haziran, 2021

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE İÇERİK TABANLI SİBER TEHDİT TESPİTİ

Emre KOÇYİĞİT tarafından hazırlanan tez çalışması 02.06.2021 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Banu DİRİ
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Banu DİRİ, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. Özgür Koray ŞAHİNGÖZ
Biruni Üniversitesi _____

Dr. Öğr. Üyesi Göksel BİRİCİK
Yıldız Teknik Üniversitesi _____

Danışmanım Prof. Dr. Banu DİRİ sorumluluğunda tarafımca hazırlanan Derin Öğrenme ile İçerik Tabanlı Siber Tehdit Tespiti başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Emre KOÇYİĞİT

İmza

Biricik eşime…

TEŞEKKÜR

Tez çalışmam boyunca benden bilgi ve desteklerini esirgemeyen yol gösterici yaklaşımları ile araştırmacı kimliğimde derin tesirleri bulunan kıymetli hocalarım Prof. Dr. Banu Diri ve Prof. Dr. Özgür Koray Şahingöz'e, iş hayatı ile üniversite hayatının eş zamanlı olması nedeniyle yaşadığım yoğunlukta her daim desteğini sunan sevgili eşim Rana Koçyiğit'e teşekkürü borç bilirim.

Emre KOÇYİĞİT

İÇİNDEKİLER

KISALTMA LİSTESİ

ALG	Algoritma
COM	Commercial
CSS	Cascading Style Sheets
DESA	Derin Evrişimsel Sinir Ağları
DOĞ	Doğruluk
DÖ	Derin Öğrenme
DVM	Destek Vektör Makinesi
ESA	Evrişimsel Sinir Ağları
F1S	F1Skoru
GAN	Generative Adversarial Network
GB	Gigabyte
GNB	Gaussian Naive Bayes
GNO	Gerçek Negatif Oranı
GOV	Goverment
GPO	Gerçek Pozitif Oranı
GYB	Geçitli Yinelenen Birim
HAS	Hassasiyet
HTML	Hypertext Markup Language
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
IP	Internet Protocol
KA	Karar Ağaçları
KES	Kesinlik
KYK	K-En Yakın Komşu
LR	Lojistik Regresyon
LSTM	Long Short-Term Memory
MÖ	Makine Öğrenmesi
NB	Naive Bayes
ÖO	Öğrenme Oranı
ORG	Organizasyon
PHP	Hypertext Preprocessor
PSO	Parçacık Sürü Optimizasyonu
RELU	Rectified Linear Unit

RO	Rastgele Orman
SGA	Stokastik Gradyen Azaltma
SMS	Short Message Service
SSL	Secure Sockets Layer
TGO	Tümleşik Geliştirme Ortamı
UL	Unordered List
URL	Uniform Resource Locator
WWW	World Wide Web
YNO	Yanlış Negatif Oranı
YPO	Yanlış Pozitif Oranı
YSA	Yinelenen Sinir Ağları

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Derin Öğrenme ile İçerik Tabanlı Siber Tehdit Tespiti

Emre KOÇYİĞİT

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Banu DİRİ

Bilgisayarların insan hayatına girmesiyle karşılaşılan güvenlik problemlerinin başında siber tehditler yer almaktadır ve bu çalışmada özellikle en yaygın siber tehditlerden biri olan oltalama saldırıları ele alınmıştır. E-posta ya da SMS gibi çeşitli iletişim kanalları ile başlayan oltalama saldırıları çoğunlukla kullanıcıların bilgilerini çalmak amacıyla tasarlanmış oltalama internet sitelerinde devam etmektedir. Kullanıcı eğitimi, kara liste kullanımı gibi önleyici çözümler saldırıları engellemeye yeterli olmamaktadır ve saldırıların yol açtığı finansal kayıplar günden güne artmaktadır. Bu doğrultuda oltalama internet sitelerini gerçek zamanlı tespit edebilen yazılım sistemleri geliştirilmektedir. Bu projede sırasıyla Makine Öğrenmesi ve Derin Öğrenme teknikleri kullanılarak oltalama internet sitelerinin tespit edilmesi hedeflenmiştir. Oltalama saldırılarının başarılı bir şekilde sınıflandırılabilmesi için URL, metin veya görsel bazlı yaklaşımlar kullanılabilmektedir. Bu çalışmada içerik-tabanlı yaklaşım benimsenmiştir. İlk aşamada içerik-tabanlı olanlar önceliklendirilerek literatürdeki oltalama tespit çalışmaları araştırılmış ve içlerinden yedi adet içerik-tabanlı çalışma ele alınmıştır. Bu çalışmalardaki 168 eşsiz özelliğin kullanım sıklığı ölçülmüş ve "Gizlenmiş Etiket, Pop-up" sayısı gibi içerik-tabanlı 48 özellik seçilmiştir. Ayrıca veri setindeki örnekler analiz edilerek daha önce literatürde görülmemiş olan 9 içerik-tabanlı özellik daha eklenmiştir. Toplamda 57 özellik hem analiz yöntemleri hem de Scikit-

learn kütüphanesindeki fonksiyonlar yardımıyla modele olan etkilerine göre sıralanmıştır. Python, TensorFlow ve BeautifulSoup gibi araçlar kullanılarak veri setinde yer alan oltalama ve meşru internet sitesi içeriklerinin özellikleri çıkarılmıştır. Yedi farklı Makine Öğrenmesi sınıflandırma algoritmasıyla oluşturulan modeller için karışıklık matrisleri elde edilmiştir. En başarılı Makine Öğrenmesi algoritması %97'nin üzerinde doğruluk ve %3'ün altında Yanlış Pozitif Oranı ile Rastgele Orman algoritması olmuştur. Devamında Yinelenen Sinir Ağları, Çekişmeli Üretken Ağ Modelleri gibi Derin Öğrenme teknikleri ile çeşitli sınıflandırma modelleri denenmiştir. Farklı aktivasyon fonksiyonları, katman tipleri ve parametreler kullanılarak yapılan deneysel çalışmalar sonucunda Makine Öğrenmesi algoritmalarından daha başarılı oltalama internet sitesin tespit edebilen Derin Öğrenme modelleri elde edilmiştir. Gelecekteki çalışmalarda içerik tabanlı özelliklerin artırılması, evrimsel algoritmalarla hiper-parametre optimizasyonu ve hibrit yaklaşımların kullanılması ile sınıflandırma modellerinin başarısı artırılabilir.

Anahtar Kelimeler: içerik-tabanlı sınıflandırma, oltalama tespiti, makine öğrenmesi, derin öğrenme

ABSTRACT

Content-based Cyber Threat Detection with Deep Learning

Emre KOÇYİĞİT

Department of Computer Engineering

Master of Science Thesis

Advisor: Prof. Dr. Banu DİRİ

Cyber threats are at the top of the security problems encountered with the introduction of computers into human life, and in this study, phishing attacks, one of the most common cyber threats, are addressed. The phishing attack that started with various communication channels such as e-mail or SMS continues mostly on phishing websites designed to steal users' information. Preclusive solutions such as user education and blacklist usage are not enough to prevent attacks. Therefore, software systems that can detect phishing websites in real time should be developed. In this project, it is aimed to classify phishing and legitimate websites using both Machine Learning and Deep Learning techniques, respectively. Content-based approach is adopted in this study. In the first stage, content-based ones were prioritized, phishing detection studies in the literature were researched and seven content-based studies were addressed. Content-based features such as "Hidden Tags, Number of Pop-ups" were selected. After analyzing phishing content, 9 content-based features that were not previously seen in the literature were added. Totally 57 features are listed according to their effects on the model with analysis methods and functions in the Scikit-learn library. Thanks to the functions created using Python, Scikit-learn, BeautifulSoup, 57 content-based features were extracted.

The most successful Machine Learning algorithm has been the Random Forest algorithm with an Accuracy of over 97% and a False Positive Rate below 3%. Subsequently, various classification models were established with Deep Learning techniques such as Recurring Neural Networks, Generative Adversarial Network Models. As a result of experimental studies using different activation functions, layer types and parameters, Deep Learning models that can detect phishing website more successful than Machine Learning algorithms have been obtained. In future studies, the success of classification models can be incremented by increasing content-based features, hyper-parameter optimization with evolutionary algorithms, and using hybrid approaches.

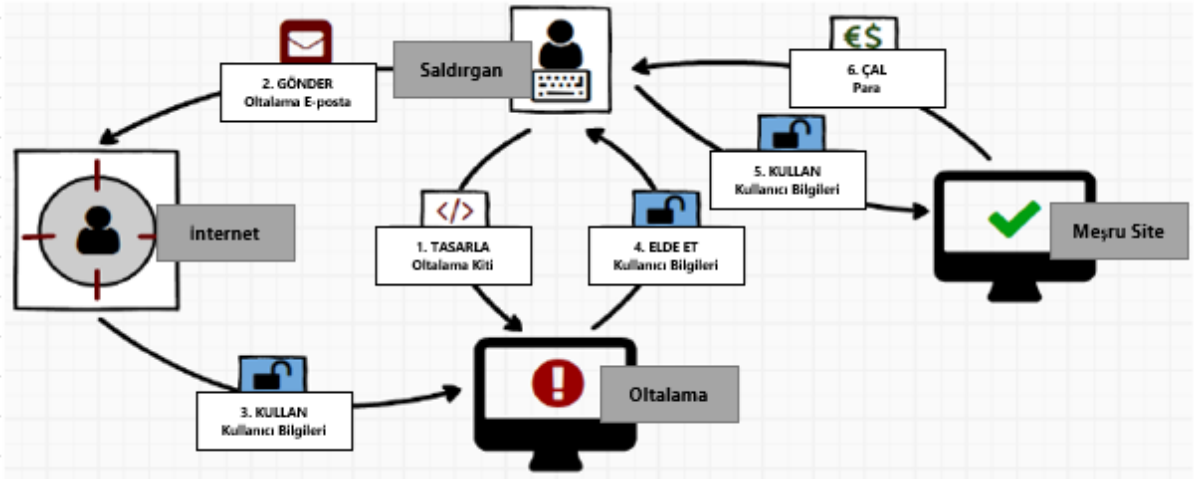
Keywords: content-based classification, phishing detection, machine learning, deep learning

1
GİRİŞ

Bilgisayarların insan hayatına girmesiyle eğitimden sağlığa, küçük çaplı özel kurumlardan devlet kurumlarına kadar birçok alanda ve/veya organizasyon yapısında önemli dönüşümler yaşanmaktadır. Bu dijital dönüşüm; izlenebilirlik, büyük ölçekli veri akışını mümkün kılma, süreçleri hızlandırma, iletişim kanallarını genişletme gibi pek çok kazanım sağlarken daha önce dünya gündeminde olmayan güvenlik problemlerini de beraberinde getirmiştir. Bu güvenlik problemlerinin başında siber tehditler yer almaktadır ve siber tehditleri incelediğimizde hem nitelik hem de nicelik yönüyle dramatik bir artış görülmektedir.

İletişim kanallarına ve hedeflerine göre çeşitlilik arz eden siber tehditlerin en eski ve yaygın olanlarından biri de "Oltalama Saldırılarıdır". Oltalama; isminden de anlaşılacağı üzere saldırganların kullanıcıları tuzaklarına çekebilmek için attıkları masum görünümlü yemler ile başlar. Amaç tuzağa düşen kullanıcıların kullanıcı adı, şifre ve/veya kredi kartı bilgileri gibi kişisel verilerini ele geçirebilmektir. Saldırganlar genellikle meşru ve resmi bir internet sitesinin ödeme sayfası gibi kişisel veri girişi yapılan bir sayfasının benzerini tasarlarlar. Hazırladıkları sahte internet sitesine doğrudan bağlantı içeren e-posta ya da SMS gibi bir araç ile gönderilen mesajı kurbanlarına ulaştırırlar. Bu bağlantıyı kullanan kurbanlar kişisel veri girişi yaptıkları anda bu bilgiler saldırganlar tarafından ele geçirilmiş olur. Bu bilgiler ile saldırganlar gizli bilgilere erişebilir ya da finansal işlem yapabilirler. Örnek bir işleyiş Şekil 1.1'de verilmiştir.

Oltalama saldırıları hedeflerine göre dört ayrı şekilde kategorize edilebilir: *Bulk, Spear, Clone ve Whaling*. "*Bulk*", belli bir hedef olmaksızın geniş ölçekli gerçekleştirilen saldırılardır. "*Spear",* öncesinde bir keşif süreci barındıran ve doğrudan belirlenmiş bir hedefe yönelik olarak gerçekleştirilen saldırılardır. "*Clone*", meşru ve resmi bir e-postanın kopyalanıp ya da diğer bir ifade ile klonlanıp içerisine eklenti veya bağlantı yerleştirilmesi sonucu gerçekleştirilir. "*Whaling*" saldırıları ise, "*Spear"* saldırısı ile benzerlik göstermekte olup, bu saldırı tipinde hedef birden fazladır ve finansal yönden getirisi çok daha yüksek olabilecek hedefler seçilir [1].

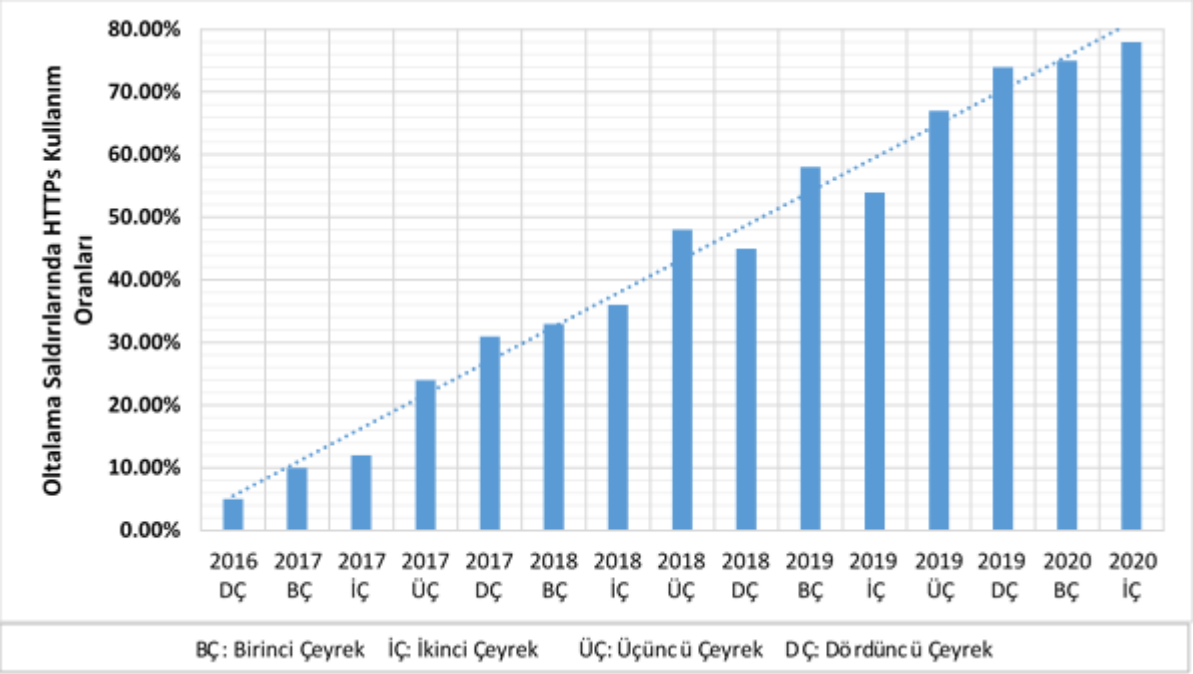


Şekil 1.1 Oltalama saldırısı adımları

1995'ten bu yana icra edilen Oltalama Saldırıları internet kullanıcıları için en büyük tehditlerden biridir. İnternet kullanıcıları ve şirketler bu saldırılar sebebiyle büyük miktarlarda finansal kayıp yaşamaktadır. 2019 yılında Deloitte firması tarafından yapılan açıklamada şirketlerin e-posta dolandırıcılığı kaynaklı güvenlik ihlali maliyetinin yaklaşık on iki milyar dolar büyüklüğünde olduğu belirtilmiştir [2]. İnternet kullanıcı sayısının ve oltalama saldırılarının artması ile orantılı olarak yaşanan finansal kayıplar da giderek artmaktadır. Saldırganlar HTTP'nin yanı sıra HTTPs üzerinden de oltalama saldırıları gerçekleştirmekte ve bu miktar her geçen yıl daha da artmaktadır. Şekil 1.2'de APWG'nin açıkladığı verilere göre HTTPs protokolü kullanılarak gerçekleştirilen oltalama saldırılarının yıllara bağlı oranları gösterilmektedir [3]. Bu doğrultuda oltalama saldırılarını önlemek veya başarısız kılmak hem bireysel kullanıcıları hem de şirketleri önemli kayıplardan kurtaracaktır.

Oltalama Saldırıları; uygulanmasının kolay olması, yüksek teknoloji ürünlerine ihtiyaç duymaması ve insani zayıflıklardan faydalanması gibi özellikleri nedeniyle en yaygın siber tehditlerden birisi olarak karşımıza çıkmaktadır. Bu saldırıların önlenebilmesi için çeşitli çalışmalar yapılmaktadır ve bu çalışmalarda en temelde iki yaklaşım benimsenmektedir. Bunlardan birincisi "Kullanıcı Eğitimi", ikincisi de "Siber Tehdit Tespiti Yapabilen Yazılım Uygulamalarının Geliştirilmesi"dir. Kullanıcı eğitiminin siber saldırılara karşı başarılı olduğu görülse de bu başarı oldukça sınırlıdır. Çeşitli sosyal mühendislik teknikleriyle kullanıcıları hedef alan bu saldırılara karşı eğitimli ve dikkatli bir kullanıcı tek başına yeterli olmamaktadır. İş,

eğitim ve sosyal hayatın pek çok anında internet kullanan kişilerin sürekli tetikte olmasını beklemek ya da yeni geliştirilen saldırı tekniklerine karşı kendilerini güncel tutmalarını beklemek gerçekçi bir senaryo değildir. Bu yüzden oltalama saldırıları ve diğer birçok siber tehditler için en başarılı ve güvenilir önlemler kapsamlı, gelişmiş ve kendini sürekli güncelleyen yazılım uygulamaları ile alınabilmektedir.



Şekil 1.2 Oltalama saldırılarındaki HTTPs kullanımının yıllara bağlı değişimi

Siber tehditlere karşı geliştirilmiş yazılım sistemlerinde temel olarak iki yaklaşım kullanılmaktadır. Birinci yaklaşım tehditlerin saldırı öncesi belirlenmesi üzerine odaklanırken, ikinci yaklaşım saldırı ile gerçek zamanlı olarak tehdidin tespit edilmesi ve önlenmesine odaklanır. Birinci yaklaşım; önceki saldırılardan veya internet kullanıcılarının raporladığı şüpheli internet sitelerinden elde edilen bilgiler doğrultusunda oluşturulan kara veya beyaz listelerden faydalanır. Kullanıcının olası bir oltalama tuzağına düşmemesi için kara listede yer alan bir internet sitesine erişimi doğrudan engeller ya da kullanıcıyı internet sitesi ile ilgili uyarır. İkinci yaklaşım ise herhangi bir kara listede olmayan ve ilk defa tasarlanan bir oltalama saldırısını dahi tespit edebilmeyi hedefler. Bunun için "Görsel Benzerlik, Metin Benzerliği" gibi tekniklerin yanı sıra "URL-bazlı, Kural-bazlı veya İçerik-bazlı" yaklaşımlardan da faydalanır.

Özellikle son dönemlerde birçok alanda olduğu gibi siber tehditler için geliştirilen önlemlerde de Makine Öğrenmesi ve Derin Öğrenme algoritmalarından faydalanılmaya başlanmıştır. İstenmeyen veya güvenlik ihlali barındıran e-postaların tespiti bu çalışmaların başında gelmektedir. Oltalama saldırılarında e-posta, SMS gibi araçlarla başlayan süreçlerin büyük bir kısmı sahte internet sitesinde tamamlanır. Makine Öğrenmesi ve Derin Öğrenme algoritmaları kullanılarak gerçekleştirilen çalışmaların da önemli bir kısmı güvenlik ihlali barındıran, sahte ve dolandırıcılık amaçlı tasarlanan internet sitelerini tespit edebilmeyi hedeflemektedir.

1.1 Literatür Özeti

İnternet dünyasında dolandırıcılık ve bilgi hırsızlığı maksadıyla tasarlanan saldırıların tespit edilebilmesi için ilk aşamada içerisinde oltalama sitelerine ait bağlantı içeren e-postaların tespit edilebilmesi üzerine odaklanılmıştır. Bu kapsamda e-postaların sınıflandırılması problemi en yaygın çalışma alanlarından biri olmuştur. G. Mujtaba ve arkadaşları çalışmalarında 2006 ile 2016 yılları arasındaki 10 yıllık süreçte yapılmış 96 farklı e-posta sınıflandırma çalışmasını derlemişlerdir. Bu çalışmaların yarısından çoğu "*spam*" yani istenmeyen e-postaların sınıflandırılması ile ilgili olup, bunların da yaklaşık %15'i oltalama e-postalarının sınıflandırılması ile alakalıdır. En yaygın kullanılan tekniklerin "eğiticili, yarı eğiticili ve eğiticisiz makine öğrenmesi, içerik-bazlı öğrenme ve istatistiksel öğrenme" yöntemleri olduğunu tespit etmişlerdir [4].

Oltalama maksadıyla gönderilen e-postalar aslında "*spam*" e-postaların özel bir şekli olup, sosyal mühendislik teknikleri ile meşru/resmi bir kurumdan gelen e-posta görünümünde olmaktadır. A. Almomani ve arkadaşlarının da oltalama e-postalarının filtrelenmesi üzerine yaptıkları inceleme çalışmasında çoğu sınıflandırıcıların eğiticili, eğiticisiz veya hibrit öğrenme yöntemlerini kullandıklarını belirtmişlerdir. Eğiticili öğrenme teknikleri daha hızlı olmasına rağmen eğiticisiz öğrenme ile elde edilen doğruluk oranı eğiticili öğrenmede alınan doğruluk oranının altında kalmaktadır. Bu kapsamda birçok algoritmadan faydalanılsa da oltalama maksadıyla gönderilen e-postaları engelleyebilmek ya da filtreleyebilmek için standart bir teknik henüz geliştirilememiştir [5].

E-posta ve SMS'in yanı sıra sosyal medya platformları da oltalama saldırılarında kullanılmaya başlanmıştır. Özellikle son yıllarda artan sosyal medya platformu ve kullanıcı sayısı, saldırganlar için bu iletişim kanalını da öne çıkarmıştır. Oltalama saldırıları sosyal medya siber tehditleri içerisinde en yaygın olanlarındandır [6]. Bu doğrultuda çeşitli sosyal medya uygulamalarında yer alan oltalama maksadıyla yerleştirilmiş bağlantıları tespit edebilmek için çalışmalar yapılmıştır. Örneğin [7] çalışmasında popüler sosyal medya uygulaması olan Twitter'daki oltalama saldırılarını otomatik olarak tespit edebilen bir sistem geliştirmiştir.

İletişim kanalları çeşitlilik gösterse de oltalama saldırılarının nihai noktası dolandırıcılık maksadıyla hazırlanmış internet siteleridir. "*Spam*" filtreleme, kara listedeki hesaplardan gelen e-postaları engelleme gibi özellikler ile oltalama saldırıları engellenmeye çalışılsa da saldırıları tamamen önleyebilmek şu ana kadar mümkün olmamıştır. Bu yüzden erişim sağlanan internet sitelerinin güvenlik kontrolü için çeşitli çalışmalar yapılmıştır. Başta tarayıcı uygulamaları kendi kontrol sistemleri ile internet sitelerini analiz edip, tehlikeli olanlar için internet kullanıcılarını uyarmaktadır. Tarayıcılar ile ilgili yapılan çalışmalarda bu önlemlerin çok da başarılı olmadığı gözlemlenmiştir [8]. Bir diğer çalışmada Google Chrome'un oltalama sitelerine karşı yeterliliği incelenmiş ve mevcut çözümlerin yeterli olmadığı belirtilmiştir [9].

Bağlantı manipülasyonları ya da mesajlara güvenilir bir internet sitesinden geliyormuş izlenimi verilmesi gibi sosyal mühendislik teknikleri kullanılarak tasarlanan oltalama internet sitelerine karşı birçok çözüm önerisinde bulunulmuştur. Bazı çalışmalarda bu tür öneriler temel olarak "İçerik-bazlı" ve "İçerik-bazlı Olmayan" şeklinde iki açıdan ele alınmıştır. "İmaj Analizi, Görsel Benzerlik Yaklaşımı, URL Analizi, HTML Kaynak Kodu Analizi, Sezgisel Yaklaşımlar ve Makine Öğrenmesi Yaklaşımı" içerik-bazlı, "Liste-bazlı Yaklaşım, Arama Motoru Bazlı ve Kitle Kaynak Kullanımı" ise içerik-bazlı olmayan çözümler altında kategorize edilmiştir. [10]. Z. Dou ve arkadaşları da yazılım bazlı oltalama internet sitelerinin tespiti hakkında yaptıkları çalışmada en çok kullanılan yöntemleri şu şekilde sıralamıştır: görsel benzerlik bazlı, içerik bazlı, URL-bazlı, kara liste bazlı ve hibrit yöntemlerdir. Ayrıca, çalışmalarında özellikle Destek Vektör Makineleri ve Lojistik Regresyon yönteminin en sık kullanılan sınıflandırıcılar olduğunu

belirtmişlerdir [11]. Bir diğer kapsamlı anket çalışmasında ise yazılım tabanlı oltalama tespit yaklaşımları "kara liste, sezgisel, görsel benzerlik ve makine öğrenmesi" başlıkları altında ele alınmıştır [12].

Makine Öğrenmesi teknikleri oltalama internet sitelerinin tespiti için son zamanlarda kullanılan en yaygın tekniklerdendir. Son yıllarda birçok problem sahasında başarılı sonuçlar üreten Makine Öğrenmesi algoritmaları, oltalama saldırılarına karşı çözüm arayan araştırmacılar tarafından da sıklıkla kullanılmaktadır. Bazı çalışmalarda yapay zekâ araçlarının hız ve verimlilik itibariyle daha başarılı uygulamaların geliştirilmesine olanak sağlayacağı ifade edilmiştir [13]. Bir başka çalışmada ise kara listede yer alan internet sitelerinin tespiti için Makine Öğrenmesinden faydalanılmıştır [14]. Makine Öğrenmesi tekniklerinin kullanıldığı bir başka çalışmada ise üç farklı yaklaşımın kullanıldığı hibrit bir çözüm benimsenmiştir. Hem kara liste – beyaz liste hem de sezgisel ve görsel benzerlik yaklaşımları ile oltalama internet siteleri tespit edilmeye çalışılmıştır [15]. M. N. Alam ve arkadaşları ise Makine Öğrenmesi algoritmalarından Karar Ağaçları ve Rasgele Orman ile oltalama internet sitelerini sınıflandıran ve %97 oranında doğruluk başarısı elde eden bir çalışma önermişlerdir. Veri setlerinin tanımlama ve sınıflandırma süreçlerinde analiz yöntemi olarak Temel Bileşen Analizini tercih etmişlerdir [16]. Makine Öğrenmesi temelli çalışmalarda genelde yüksek sınıflandırma başarısı elde edilmiştir. Örneğin bir başka çalışmada çeşitli makine öğrenmesi algoritmaları kullanılmış ve Destek Vektör Makinesi ile %95'in üzerinde başarı elde edilmiştir [17]. URL-bazlı ve içerik-bazlı 36 farklı özelliğin kullanıldığı bir başka çalışmada ise özelliklerin kombinasyonları denendikten sonra 29 farklı özellik ile %98'i geçen başarı oranı elde edilmiştir [18]. Ağırlıklı olarak URL-bazlı özelliklerin kullanıldığı bir başka çalışmada ise aralarında Destek Vektör Makinesi, Lojistik Regresyon, Karar Ağaçları ve Naive Bayes'in olduğu dört farklı Makine Öğrenmesi modeli geliştirilmiş ve en iyi skoru Lojistik Regresyon ile alınmıştır [19]. Literatürde yer alan K- En Yakın Komşu, Naive Bayes, Karar Ağaçları, Destek Vektör Makinesi, Yapay Sinir Ağları ve Rasgele Orman gibi Makine Öğrenmesi tekniklerini kullanan çalışmalara yer veren yayınlardan, M. V. Kunju ve arkadaşları Makine Öğrenmesi temelli oltalama tekniklerinin evrimini incelemişlerdir [20]. Bu doğrultuda Makine Öğrenmesi temelli birçok çalışmanın

sayısının arttığını ve odaklandıkları konu kapsamının giderek genişlediğini söyleyebiliriz.

Oltalama internet sitelerinin tespiti için başvurulan bir diğer trend yaklaşım Derin Öğrenme tekniklerinin kullanılmasıdır. Derin Öğrenme teknikleri, özellikle performans konusunda ortaya koyduğu başarılı sonuçlar ile birçok araştırmacının odak noktası haline gelmiştir. S. Singh ve arkadaşları URL tabanlı bir yaklaşım ile Evrişimsel Sinir Ağlarını uygulayarak başarılı sonuç elde ettiklerini ve özellikle zaman yönüyle verimli bir sistem önerdiklerini ifade etmişler [21]. Bir başka çalışmada ise Derin Evrişimsel Sinir Ağları kullanılarak %99 oranında bir doğruluk başarısı elde edildiği ve Makine Öğrenmesi algoritmalarına göre çok daha başarılı sonuçlar alındığı belirtilmiştir [22]. Derin Öğrenme tekniklerinin kullanıldığı bir diğer model olan Web2Vec ile oltalama internet sayfasının tespitinde başarılı sonuçlar elde edilmiştir [23]. I. Saha ve arkadaşları da çalışmalarında Derin Öğrenme temelli bir yaklaşım benimseyerek Çok Katmanlı Algılayıcı tekniğini kullanmış ve URL veri seti üzerinden oltalama tespiti yapmaya çalışmışlardır [24]. Oldukça büyük veri seti ile çalışan bir başka Derin Öğrenme temelli çalışmada ise URL'in sıralı karakter, internet sitesi içeriği ve metni kullanılarak elde edilen çok boyutlu özellik yapısı sayesinde %98'i geçen doğruluk oranı ve %1'in altında YPO oranı elde etmiştir [25]. Oltalama internet sitesi tespitinde %99 doğruluk oranını elde eden bir diğer çalışmada ise Uzun Kısa Süreli Bellek Yinelenen Sinir Ağı kullanılmıştır [26]. Ayrıca, Derin Öğrenme temelli birçok hibrit çalışma da yürütülmektedir. Bu çalışmalardan biri *Otokod* ve Evrişimsel Sinir Ağları kullanarak verimli bir Hibrit Sinir Ağı sistemi geliştirmiş olan çalışmadır [27].

Makine Öğrenmesi ve Derin Öğrenme temelli çalışmaların geneli incelendiğinde özellikle URL-bazlı özelliklerin yoğun olarak kullanıldığı görülmektedir. "IP adresi, URL uzunluğu" gibi özellikler oltalama ve meşru internet sitelerinin ayırt edilmesinde önemli bilgi vermektedir. Örneğin; H. Chapla ve arkadaşları sınıflandırıcı olarak bir Bulanık Mantık modeli kullandıkları Makine Öğrenme temelli çalışmalarında URL-bazlı bir sistem geliştirmişlerdir. MatLab ve 1000 adet URL kullandıkları çalışmada yaklaşık olarak %91'lik doğruluk oranı elde etmişlerdir [28]. Veri setinin büyüklüğü, çeşitliliği ve dengeli olması bu başarıda elbette etkilidir. Bazı çalışmalarda, URL-bazlı özelliklerin yanı sıra Google arama motoru

sonuçları gibi üçüncü parti uygulamalardan da faydalanılarak doğruluk oranı artırılmaya çalışılmıştır [29]. Bir başka çalışmada ise, sınırlı sayıda URL ile ilgili özellikler kullanılarak Rasgele Orman algoritması ile oltalama tespiti yapılırken [30], bir diğer çalışmada Bayes optimizasyonu yapılarak Destek Vektör Makinesi kullanılmıştır [31]. Y. Huang ve arkadaşları yaptıkları kapsamlı çalışmada URL özelliklerini çıkarmak için karakter seviyesinde Evrişimsel Sinir Ağları ve dikkat tabanlı hiyerarşik Yinelenen Sinir Ağlarını kullanmıştır. 4,8 milyonluk URL veri seti kullandıkları çalışmalarında %99'ı geçen doğruluk oranı ve %1'in altında YPO oranı elde etmişlerdir [32]. Literatürdeki çalışmaların ağırlıklı kısmı sınıflandırıcılar üzerine yapılmıştır ancak kümeleme algoritmaları kullanılarak gerçekleştirilen çalışmalar da mevcuttur. Bazı çalışmalar da bu iki teknik birbirleri ile karşılaştırılmıştır [33]. Ayrıca çeşitli optimizasyon algoritmalarından yararlanarak oltalama tespit sistemlerinin verimlilik ve başarısı artırılmaya çalışılmıştır. Örneğin [34] çalışmasında Yapay Sinir Ağlarının eğitim safhasında Parçacık Sürü Optimizasyonu kullanılmıştır.

İçerik-bazlı yaklaşımın kullanıldığı oltalama tespiti çalışmaları URL-bazlı çalışmalara nazaran daha az sayıdadır. [35] çalışmasında toplamda 58 adet içerik-bazlı özellik kullanarak 8 farklı Makine Öğrenmesi tekniği ile oltalama internet sitelerinin tespit edilmesi üzerine çalışmalar yapmış ve %97'e varan başarı oranı elde edilmiştir. Ancak içerik-bazlı çalışmaların ağırlıklı kısmı e-posta içeriği ile yapılan çalışmalardan oluşmaktadır. Örneğin H. Che ve arkadaşları yaptıkları çalışmada oltalama e-postalarını tespit etmeye çalışmışlardır. Durum çalışması ile algoritmalarını detaylı olarak aktardıkları yayınlarında geleneksel filtreleme yöntemlerine göre daha başarılı sonuçlar elde ettiklerini ifade etmişlerdir [36]. Bir başka içerik-bazlı çalışmada da "spam" e-postaların Bayesian sınıflandırıcısı ile tespiti üzerine çalışılmış ve %96'nın üzerinde tespit başarısı elde edilmiştir [37]. Karşılaştırmalı analiz yapılan bir diğer çalışmada ise e-posta içerisinde yer alan zararlı URL bağlantıları yine içerik-bazlı yaklaşım ile tespit edilmeye çalışılmıştır [38]. Oltalama tespitinin yanı sıra özellikle "spam" e-postaların tespit edilebilmesi için içerik-bazlı yaklaşımlara sıklıkla başvurulmaktadır [39]. E-postaların yanı sıra sosyal medya uygulamaları ile yapılan içerik-bazlı çalışmalar da mevcuttur. Örneğin [40] çalışmasında Twitter uygulaması içerisinde yer alan radikal hesapların otomatik olarak tespit edilebilmesi için içerik-bazlı yaklaşımdan yararlanmıştır.

Oltalama sürecinin son noktası olan oltalama internet sitelerinin tespit edilebilmesi için yapılan içerik-bazlı çalışmalarda farklı tekniklerden yararlanılmıştır. Bazı çalışmalarda metin ve görsel içerik kullanılarak oltalama tespiti yapılmıştır [41]. Bir başka çalışmada ise internet site içeriğinde yer alan resimler kullanılarak oltalama analizi yapılmaya çalışılmıştır [42]. Oltalama internet sitelerinin ömrü oldukça kısadır bu yüzden oltalama internet sitesi içeriği elde etmek, URL elde etmekten daha zor olmaktadır. Bu zorluğa bağlı olarak içerik-bazlı çalışmalarda kullanılan veri setlerinde yer alan örnek sayısı URL-bazlı çalışmalara göre çok daha az olmaktadır. Örneğin, araştırmacılar URL-bazlı bir çalışmada 4,8 milyon örnek kullanırken [32], içerik bazlı bir başka çalışmada ise yaklaşık 18 bin örnek kullanılmıştır [43]. Oltalama tespiti çalışmalarının başarısını gösterene en kritik değerlerden biri YPO değeridir. Çünkü sistem için oltalama olduğu halde güvenli olarak sınıflandırılan internet siteleri en büyük güvenlik tehdidini oluşturmaktadır. Bu kapsamda YPO değerini düşürebilmek için çeşitli çalışmalar yürütülmüştür [44].

Literatür özeti bölümünde incelenen çalışmalarla ilgili temel bilgiler karşılaştırmalı olarak Tablo 1.1 ve Tablo 1.2'de gösterilmiş olup, çalışmaların kullandığı yaklaşım ve yöntem oltalama internet sayfalarının tespitinde makine öğrenmesi ve derin öğrenme tekniklerine sıklıkla başvurulduğu, URL-tabanlı yaklaşımın ise en çok tercih edilen yaklaşım olduğu görülmüştür. İçerik-tabanlı yaklaşım kullanılan çalışmalarda ise URL-tabanlı özelliklerin de ayrıca kullanıldığı gözlemlenmiş, bu tez çalışmasında URL ve alan adı bazlı özelliklerin etkisini engellemek için sadece içerik-tabanlı özellikler kullanılmıştır. Aynı zamanda makine öğrenmesi ve derin öğrenme teknikleri kullanılarak yapılan oltalama tespiti çalışmalarının diğer yöntemlere göre çok daha başarılı olduğu görülmüş ve bu tez çalışmasında da bu tekniklerden faydalanılmıştır.

Tablo 1.1 Oltalama tespit çalışmaları (2015-2019)

Yıl	Yayın	Yöntem	Hedef	Yaklaşım
2015	[37]	Makine Öğrenmesi	E-posta	İçerik
	[38]			
	[42]	Arama Motoru	İnternet Sayası	Görsel
2016	[29]	Kara Liste	İnternet Sayfası	URL
	[39]	Kural-tabanlı	E-posta	İçerik
	[40]	Makine Öğrenmesi	Sosyal Medya	
2017	[34]	Derin Öğrenme	İnternet Sayfası	URL
	[36]	Bulanık Mantık	E-posta	İçerik
2018	[15]	Makine Öğrenmesi	İnternet Sayfası	Beyaz Liste, URL, Alan adı, Trafik bazlı, Hibrit
	[27]	Derin Öğrenme		URL, Alan adı, İçerik
	[30]	Makine Öğrenmesi, Sezgisel		URL
2019	[13]	Makine Öğrenmesi	İnternet Sayfası	URL
	[20]			İnceleme
	[28]			URL

Tablo 1.2 Oltalama tespit çalışmaları (2019-2020)

Yıl	Yayın	Yöntem	Hedef	Yaklaşım
2019	[22]	Makine Öğrenmesi, Derin Öğrenme		URL, İçerik
	[33]	Makine Öğrenmesi, Hibrit	İnternet Sayfası	URL
	[25]	Derin Öğrenme		
	[32]			
2020	[14]	Makine Öğrenmesi	İnternet Sayfası	URL
	[31]			
	[16]			URL, İçerik
	[17]			
	[18]			
	[19]			URL, Görsel
	[35]			İçerik
	[23]	Derin Öğrenme		Hibrit
	[21]			URL
	[24]			
	[26]			

1.2 Tezin Amacı

Yapılan araştırmalar sonucunda siber tehditlerin hem bireysel internet kullanıcıları hem de şirketler için önemli kayıplara yol açtığı, oltalama saldırılarının en yaygın ve en eski siber tehditlerden biri olduğu tespit edilmiştir. Farklı yaklaşımlar ve

teknikler kullanılmasına rağmen oltalama saldırılarının tamamen engellenemediği, alternatif çözüm arayışlarının devam ettiği, özellikle Makine Öğrenmesi ve Derin Öğrenme teknikleri ile yapılan çalışmalardaki başarı oranlarının yüksek olduğu, ancak bu başarı oranına rağmen saldırıların ve yol açtıkları hasarın nicelik ve nitelik yönü itibariyle günden güne artış gösterdiği görülmektedir. Oltalama saldırılarının genellikle e-posta, SMS ve özellikle son yıllarda sosyal medya platformları gibi farklı iletişim kanalları aracılığıyla başlatıldığı, nihai nokta olarak internet kullanıcılarının oltalama maksadıyla tasarlanan sahte internet sitelerine yönlendirilmek istendiği saptanmıştır. Tarayıcılar ve e-posta uygulamaları kendi filtreleme ve güvenlik önlemlerini yürütmeye çalışsa da saldırıları tamamıyla önleyememektedirler. Bu doğrultuda oltalama internet sitelerini gerçek zamanlı olarak tespit edilebilen ve internet kullanıcılarını uyaran bir sistemin geliştirilmesi gerekmektedir.

Oltalama saldırıları gibi sosyal mühendislik tekniklerinden faydalanılan siber tehditlere karşın internet kullanıcılarının eğitilmesi veya dikkatli olmaları yeterli olmamaktadır. Geliştirilecek yazılım sistemleri yardımıyla bu tehditler saptanmalı ve kullanıcılar bilgilendirilmelidir. URL ve görsel benzerlik ile insanlar aldatılabilir ancak internet sitesinin arka planında yer alan içeriği görebilen bilgisayar sistemleri sosyal mühendislik teknikleri ile aldatılamaz. Bu kapsamda içerik-bazlı yaklaşım kullanılarak oltalama internet sitelerini tespit edebilen bir sistemin geliştirileceği tez çalışması amaçlanmıştır.

Bu tez kapsamında incelenecek olan içerik unsurları kullanıcının görebildiği metinsel, görsel veya işitsel içerikler yerine arka planda internet sitesinin yapısal olarak iskeletini oluşturan HTML, CSS gibi internet sayfalarını inşa ederken kullanılan temel teknolojilerdir.

Oltalama internet sitelerinin tespiti çalışmalarının ilk aşamasında URL özellikleri üzerinden bir sınıflandırma çalışması yapılmaktadır. URL içerisinde kullanılan '&$~*!' gibi özel karakter sayısı, URL uzunluğu gibi özellikler kullanılarak tasarlanan Makine Öğrenmesi ve Derin Öğrenme modelleri bir internet sitesinin oltalama olarak yani siber tehdit olarak doğru bir şekilde sınıflandıramadığı durumlarda o internet sitesinin içerik-bazlı özellikleri incelenmeli ve daha hassas bir sınıflandırma gerçekleştirilebilmelidir. Bu tez çalışması kapsamında en ayırt edici

içerik-bazlı özelliklerin tespiti ve bu özelliklerin kullanıldığı yüksek hassasiyetle tespit yapabilen modellerin geliştirilmesi hedeflenmiştir.

Siber tehdit tespit sistemlerinde özellikle *Yanlış Pozitif Oranı* değeri oldukça önemlidir. Oltalama internet sitesinin güvenli bir internet sitesi olarak sınıflandırılması, internet kullanıcısı ve sistem güvenliği açısından en istenmeyen durumdur. Tehdit olduğu halde tehdit olmayan internet sitesi olarak sınıflandırılan örneklerin oranını veren YPO değeri minimize edilmelidir. YNO değeri YPO kadar kritik olmasa da kullanıcı güvenini etkileyebileceği için minimize edilmesi gereken bir diğer değerdir. Bu çalışma kapsamında inşa edilen modeller ile sadece doğruluk başarısının artırılması değil, aynı zamanda YPO ve YNO değerlerinin sıfıra yaklaştırılması amaçlanmıştır.

İnternet sitelerinin siber tehdit durumlarına göre kategorize edilmesi ya da siber tehdit olup olmadıklarının belirlenmesi bir tür sınıflandırma problemidir. Sınıflandırma tipleri temel olarak "İkili" ve "Çoklu" sınıflandırma olmak üzere ikiye ayrılır. "İkili Sınıflandırma" için kullanılabilecek algoritmalardan bazıları Lojistik Regresyon, K-En Yakın Komşu Algoritması, Karar Ağaçları, Destek Vektör Makineleri ve Naive Bayes olarak verilebilir. "Çoklu Sınıflandırma" için ise K-En Yakını Komşu Algoritması, Karar Ağaçları ve Naive Bayes'in yanı sıra Rastgele Karar Ormanları, Gradyan Artırma ve Yapay Sinir Ağları gibi teknikler de kullanılmaktadır. Herhangi bir internet sitesinin "Oltalama" ya da "Meşru (Oltalama Değil)" şeklinde sınıflandırması problemi ise ikili sınıflandırma problemidir ve bu çalışmada ikili sınıflandırma yapılarak oltalama saldırılarının doğru bir şekilde sınıflandırılması amaçlanmıştır.

1.2.1 İnternet Sayfası Bileşenleri

Bir internet sitesi sayfa veya sayfalarından oluşmaktadır. İnternet sayfaları ise metin, görsel ve işitsel içerikler barındıran ve ziyaretçisine bu kapsamda bilgi aktaran dokümanlardır. İnternet kullanıcıları bu sayfalara HTTP veya HTTPS protokollerini ve bu sayfaların adresleri aracılığıyla erişebilmektedir. İnternet sitesi adresi "www" yani "World Wide Web'in" kısaltılmış hali, alan adı ve "com, org, gov vb." üst seviye alan adı unsurlarından oluşmaktadır. Her bir unsur arasında "nokta" ile ayrım yapılmaktadır.

İnternet sitelerinde bulunan sayfalar temel olarak iki bileşenden oluşmaktadır. Birincisi, internet sayfasına erişimi sağlayan protokol ve adres bilgilerinin girildiği "URL", ikincisi ise yazılı, görsel ve işitsel tüm unsurların yer aldığı içerik kısmıdır.

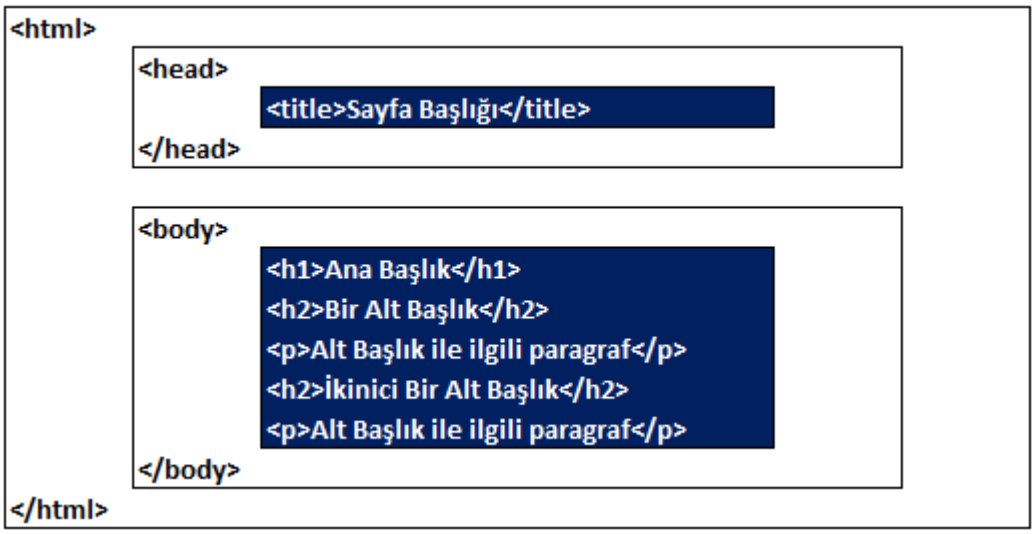
Şekil 1.3 URL bileşenleri

1.2.2 URL Nedir?

İnternet üzerinde bulunan tüm cihazlar eşsiz IP adrese sahiptir. İnternet kullanıcılarının kullanımını kolaylaştırmak, internet sayfasının adresini daha hatırlanabilir bir hale getirmek ve dünya genelinde standardizasyon sağlamak amacıyla internet iletişim protokolleri ve alan adları oluşturulmuştur. Bu protokolleri, alan adlarını ve diğer detay bilgilerini bünyesinde barındıran ve her internet sayfası için eşsiz şekilde var olan talimatlar kümesine "*Uniform Resource Locater*" yani URL denilmektedir. Örnek bir URL ve bileşenleri Şekil 1.3'te detaylı olarak gösterilmiştir.

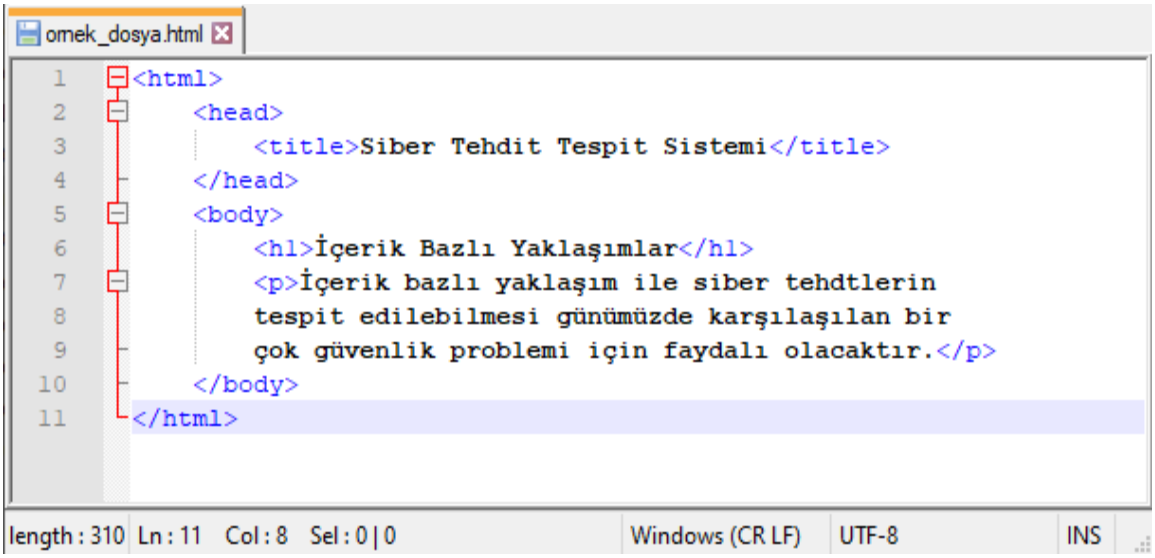
1.2.3 İnternet Sayfası İçeriği Hangi Bileşenlerden Oluşur?

İnternet sayfası içeriğinin en temel unsuru HTML bileşenleridir. HTML yani "*Hyper Text Markup Language*" internet sayfasının iskeletini tasvir eden standart bir biçimleme dilidir. Çeşitli etiketler ve elementler ile internet sayfası içeriklerinin nasıl gösterileceğini ve kategorize edileceğini belirler. Bir dokümanın HTML olduğu "<!doctype html>" etiketi ile anlaşılır. İnternet sayfası içerisinde yazı, resim, video gibi çeşitli içerik bileşenlerinin tarayıcı tarafından nasıl işleneceği, hangi işlemlerin ne ölçüde uygulanacağı HTML etiketleri aracılığıyla belirlenir. Bir HTML etiketi "<" ile başlar ve "/>" ile sonlanır. Aynı doğrultuda HTML etiketleri ve diğer içerikler de "<html>" ve "</html>" etiketleri içerisinde yer alır. Tarayıcılar bu HTML etiketlerini internet kullanıcılarına göstermez. Temel olarak "*head*" ve "*body*" kısımlarından oluşan bir HTML sayfası Şekil 1.4'te gösterilmiştir.



Şekil 1.4 HTML <head>, <body> blokları

"*Head*" etiketi üst kısım bilgilerini, "*body*" kısmı ana gövdeyi yani içerik kısmındaki bilgileri tarif eder. Şekil 1.4'te mavi arka plan ile renklendirilmiş kısım kullanıcı tarafından görülebilir ancak diğer kısımlar görülemez. Üst kısımdaki "<title>" etiketi ile işaretlenmiş alan sayfa başlığını tarif ederken aynı zamanda tarayıcının başlık barında ya da sayfa tab kısmında gösterilmektedir. Gövde kısmında yer alan "<h1>" ve "<h2>" etiketleri sayfa içerisindeki başlıkları tarif ederken, "<p>" etiketi paragrafları yani başlık altında yer alan ilgili metinleri tarif etmektedir. Şekil 1.5'te örnek bir HTML dosyası ve Şekil 1.6'da ise bu dosyanın tarayıcılardaki görünümü gösterilmiştir.



Şekil 1.5 Örnek HTML dosyası

Şekil 1.6 Örnek HTML dosyasının tarayıcıda gösterimi

1.3 Hipotez

- Siber tehdit oluşturan internet sayfalarının tespitinde sayfa içeriğinden elde edilen özellikler belirleyici olabilir.
- Literatürde yer almayan yeni içerik-tabanlı özellikler ile oltalama internet sayfalarının sınıflandırma başarısı artırılabilir.
- Makine öğrenmesi temelli sınıflandırma algoritmaları ile oltalama ve meşru internet sayfaları başarılı bir şekilde sınıflandırılabilir.
- Sınıflandırma çalışmalarında yanlış sınıflandırılan örneklerin analizi ile elde edilecek içerik-tabanlı özellikler ayırt edici olabilir.
- Derin öğrenme teknikleri ile makine öğrenmesi algoritmalarından daha başarılı siber tehdit tespiti yapılabilir.

2
VERİ SETİ

Literatürdeki çalışmalarda oltalama internet siteleri için çeşitli veri setleri bulunmaktadır. Bu çalışma kapsamında kullanılan veri setine ait detaylar, veri seti üzerinde uygulanan işlemler ve oltalama saldırıları için kullanılan diğer popüler veri setleri hakkındaki bilgilendirme Bölüm 3.1'de, ham veri halinde bulunan veri setinin işlenmesi de Bölüm 3.2'te anlatılmıştır.

2.1 Veri Seti Detayları

İçerik-tabanlı oltalama tespiti çalışmalarındaki en büyük zorluklardan biri veri setini oluşturmaktır. Oltalama internet sitelerini barındıran çeşitli kaynaklar bulunmaktadır [52]. Ancak bu kaynakların büyük bir kısmında oltalama internet sitelerinin sadece URL bilgileri yer almaktadır. Bunun yanı sıra veri setleri içerisinde yer alan oltalama internet sitelerinin bir kısmı tespit edildikten sonra kapatıldığı için aktif değildir. Oltalama internet sitelerinin oldukça kısa bir yaşam süresine sahip olduğu göz önünde bulundurulduğunda bu URL'ler aracılığıyla elde edilebilecek internet sitesi içeriği sayı olarak toplam veri setine kıyasla düşük sayıda kalmaktadır. Bu tez çalışmasında "*Benchmarking*" çalışmaları için oluşturulmuş ve düzenlenmiş bir veri seti kullanılmıştır [53]. Bu veri setinde 15 bin oltalama, 15 bin meşru olmak üzere toplam 30 bin adet internet sitesi örneği bulunmaktadır. Örneklerin büyük çoğunluğu için beş adet klasör mevcuttur. Bunlardan "SCREEN-SHOT" klasöründe internet sitesine ait ekran görüntüsü, "URL" klasöründe internet sitesinin URL bilgisi, "WEBPAGE" klasöründe internet sitesine ait logo, icon vb. görseller, "WHOIS" klasöründe ise internet sitesinin alan adı bilgileri ve "RAW-HTML" klasöründe de ".html, .php vb." uzantılı içerik dosyaları yer almaktadır. Bu tez kapsamında ise sadece "RAW-HTML" klasörü içerisindeki veriler kullanılmıştır. "RAW-HTML" klasöründe yer alan örnek bir içerik dosyasından alınan kesit Şekil 2.1'de gösterilmiştir.

```
<!doctype html>
<html>

<head>

    <meta charset="utf-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>


<title>Service-Free-France.com</title>

<meta name="keywords" content="cash advance debt consolidation insurance service-free-france.com" />

<meta name="description" content="Find Cash Advance, Debt Consolidation and more at Service-Free-France.com. Get the best of Ins

<script src='http://code.jquery.com/jquery-latest.min.js' type='text/javascript'></script>
<script language='JavaScript' src='/js/standard.js?rte=1&tm=2&dn=service-free-france.com&tid=1020'></script>

<meta name='google' value='notranslate' />
<script type='text/javascript' language='JavaScript' src='/js/google_caf.js?rte=1&tm=2&dn=service-free-france.com&tid=1020'></sc
<script type='text/javascript' language='JavaScript' src='http://www.google.com/adsense/domains/caf.js'></script>

<script type='text/javascript'>
var pageOptions =
{
    'domainRegistrant' : 'as-drid-2421601518898051',
    'relatedSearchBaseUrl': 'http://mobile.free.fr.service-free-france.com/?ac=2&slt=8&slr=1&lpt=1',
    'resultsPageBaseUrl': 'http://mobile.free.fr.service-free-france.com/?ac=2&slt=8&slr=1&lpt=1',
    'pageLoadedCallback': google_callback,
    'pubId': 'dp-demandmedia31_3ph',
    'channel': '100001',
    'terms': '',
    'optimizeTerms': true,
```

Şekil 2.1 Veri setinde yer alan örnek bir içerik dosyası

Veri setinde yer alan örneklerin bazılarında tüm klasörler mevcut olmayıp bazı içerik dosyaları da okunabilir formda değildir. Bundan dolayı veri setinde yer alan örnekler kontrol edilmiş ve yapılan veri temizleme sürecinden sonra 13422 meşru ve 12896 oltalama olmak üzere toplamda 26318 internet sitesi örneği ile veri seti oluşturulmuştur. Sonuç olarak meşru ve oltalama örnekleri sırasıyla yaklaşık %51 ve %49 oranlarında dengeli bir veri seti elde edilmiştir.

2.2 Verilerin İşlenmesi

İşlenmemiş halde bulunan tüm içerik dosyaları, Python programlama dili ve BeautifulSoup modülü ile ayrıştırılıp, yazılan fonksiyonlar ile her bir örnek için çalıştırılmıştır. Bu fonksiyonlar yardımıyla Bölüm 2.1'de belirtilen 57 özellik için sayısal değerler üretilmiştir. Tüm örnekler için üretilen değerler; satırlar örnekleri ve sütunlar özellikleri temsil edecek şekilde bir ".csv" dosyasına aktarılmıştır. Oluşturulan ".csv" dosyasından alınan bir kesit Şekil 2.2'de gösterilmiştir.

Ayrıca veri setinde bulunan internet sitelerinden oltalama olanlar "1", meşru olanlar da "0" olarak etiketlenmiştir. Etiket değerleri oluşturulan ".csv" dosyasına "*Label*" başlığı altında elli sekizinci sütun olarak eklenmiştir.

Span_count	Is_favicon	Title_length	Table_Count	TH_count	TR_count	div_count	li_count	ul_count	p_count	h1_count	h2_count	button_count	meta_count	style_count	label_count	select_count
0	0	12	1	0	1	8	0	0	5	1	0	0	1	0	0	0
9	1	14	0	0	0	72	7	1	0	0	0	3	17	0	6	0
8	1	50	1	0	1	268	15	2	2	0	0	0	15	0	0	0
18	1	11	0	0	0	42	18	2	0	0	0	0	2	0	0	0
175	1	72	0	0	0	397	96	13	28	1	7	9	29	0	48	13
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
14	0	107	0	0	0	32	16	4	0	0	0	0	9	0	0	0
0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	30	0	0	0	51	10	3	1	0	0	0	2	5	0	0
1	1	50	0	0	0	66	35	7	15	1	10	2	18	1	3	1
36	0	43	0	0	0	77	15	3	1	1	10	0	2	1	0	0
46	1	1	0	0	0	141	26	2	56	1	0	0	5	2	0	1
27	1	14	1	0	2	161	184	58	40	1	5	1	4	1	1	0
15	1	5	0	0	0	132	42	9	10	1	5	1	6	4	0	2
33	0	47	0	0	0	108	25	4	0	19	1	0	8	2	0	0
617	0	64	12	0	14	433	10	1	1	2	6	4	9	1	0	2
49	0	73	0	0	0	62	52	7	17	1	0	2	11	0	0	0
33	1	10	0	0	0	138	32	10	24	1	9	1	7	1	0	0
127	0	69	4	0	18	318	181	35	52	1	7	0	13	4	3	0
75	1	21	0	0	0	89	69	12	41	0	0	10	4	0	2	5

Şekil 2.2 Veri seti işlendikten sonra elde edilen .csv dosyası

3
YÖNTEMLER

Makine Öğrenmesi ve Derin Öğrenme yöntemleri ile geliştirilen içerik-tabanlı siber tehdit tespiti çalışmalarında çeşitli özellikler kullanılmakta olup bunların bir kısmı URL-bazlı, bir kısmı da Domain-bazlı özelliklerdir. Bu tez çalışmasında sadece içerik-tabanlı özellikler kullanılmış olup, diğer yaklaşımlarla elde edilen özelliklerin kullanımından kaçınılmıştır. Özelliklerin belirlenmesi ile ilgili detaylar Bölüm 4.1'de verilmiştir. Son yıllarda elde edilen başarılı sonuçlar neticesinde oltalama tespit çalışmalarında Makine Öğrenmesi ve Derin Öğrenme temelli sınıflandırma algoritmaları kullanılmaya başlanmış olup, algoritmaların başarı ve performansı için özellik seçimi önem arz etmektedir. Bu kapsamda yapılan özellik seçimi çalışmaları Bölüm 4.2'de anlatılmıştır. Son olarak bu çalışmada kullanılan Makine Öğrenmesi temelli sınıflandırma algoritmaları Bölüm 4.3'te, Derin Öğrenme temelli tekniklere de Bölüm 4.4'te yer verilmiştir.

3.1 Özelliklerin Belirlenmesi

Oltalama tespit sürecinde ilk aşama olan e-posta ve/veya SMS gibi iletişim kanallarının kara liste, filtreleme ve güvenlik sistemlerinin yakalayamadığı saldırılarda kullanıcılar oltalama maksadıyla tasarlanmış internet sitelerine erişim sağlamaktadır. Bu noktada URL üzerinden yapılan değerlendirme süreci sonucunda internet sitesinin güvenli olup olmadığı saptanmaya çalışılmaktadır. Oltalama tespit çalışmalarında ağırlıklı olarak URL-bazlı yaklaşım benimsendiği için bu noktada kullanılan özelliklerin de büyük çoğunluğu "URL uzunluğu" gibi URL ile ilgili özellikler olmaktadır. URL-bazlı özellikler kullanılarak tespit edilemeyen oltalama internet sitesinin saptanması için de son olarak içerik-bazlı özellikler üzerinden bir değerlendirme yapılmaktadır. Bu doğrultuda seçilecek içerik-bazlı özellikler hem sistem başarısını hem de performansını etkileyeceği için dikkatli seçilmelidir.

İçerik-bazlı özelliklerin belirlenebilmesi için ilk olarak literatürde yer alan içerik-bazlı oltalama tespiti çalışmaları araştırılmış ve onlardan içerik-bazlı özellikleri paylaşanlar tespit edilmiştir. İncelenen çalışmaların güncel olması için 2015 yılı ve sonrasında yayınlanmış çalışmalara yer verilmiştir. Bu kapsamda içerik-bazlı yedi çalışma incelenmiştir. Seçilen çalışmalar sırasıyla şunlardır: [45], [46], [47], [48],

[49], [50] ve [51]. Bu çalışmalarda kullanılan özellik sayıları sırasıyla 58, 76, 48, 31, 32, 30 ve 16 olmak üzere toplamda 198 adet özellik tespit edilmiştir. Bu özellikler detaylı olarak incelendiğinde bazılarının isimlendirmede farklılık içerse de aynı fonksiyona sahip olduğu görülmüştür. Bu şekilde tespit edilen tekrar eden özellikler ayıklandıktan sonra 168 adet eşsiz özellik belirlenmiştir. Bu özelliklerden en sık kullanılanları tespit edebilmek için Tablo A.1'de verilen "Özellik Frekans Tablosu" oluşturulmuştur. İncelenen yedi yayın sırasıyla yedi farklı kolonla eşleştirilmiş ve özelliğin kullanılması durumunda ilgili çalışmanın bulunduğu sütundaki hücreye "1", kullanılmaması durumunda ise "0" değeri verilmiştir.

Literatür araştırması sonucu listelenen 168 özelliğin tamamını kullanmak yerine model başarısına olumlu etki edecek yeterli sayıda özelliğin kullanımı amaçlanmıştır. Oluşturulacak sınıflandırma modellerinin performansı da göz önünde bulundurularak çalışmalarda en sık kullanılan özellikler önceliklendirilmiştir. Bu doğrultuda ilk olarak Tablo 3.1 ve Tablo 3.2'de listelenen 168 özellikten en az iki yayında kullanılan özellikler yani tablodaki ilk 50 özellik seçilmiştir. Bu özellikler içerisinde "*Tiny URL, Abnormal URL*" gibi URL-bazlı özellikler, "*Age of Domain*" gibi alan adı ile ilgili özellikler elenmiş ve HTML ile ilişkili özellikler önceliklendirilerek toplamda 27 adet özellik elde edilmiştir. Literatürde yapılan diğer çalışmalardaki özellik sayısı ve incelenen çalışmalardaki ortalama özellik sayısının yaklaşık olarak 42 olduğu değerlendirildiğinde, 27 adet özelliğin yetersiz olacağı görülmüştür.

Tablo 3.1 Literatür araştırması sonucu listelenen 168 özellik - 1

Özellik	Özellik	Özellik	Özellik
Link	Suspicious action upon submitted information	Button Count	TLD evaluation in the part of the URL
HTTPS	Websites forwarding	Title Length	Country- code and TLD comparison
Submit Element	Count of hidden tags	Longest Word Length	Path Level
E-mail Input	Count of external links	Shortest Word Length	Num Dash
Iframe	Shortining_Service	Content Length	Num Dash In Hostname
Favicon	SSLfinal_State	Using free hosting domains	Tilde Symbol
Redirect	Domain_registeration_length	Count of digit	Number of UnderScore
IP Address	on_mouseover	Registration Date of Domain	Num Percent
@' symbol	POST Method	Port No in the URL	Num Query Components
Sub-domain	Input Element	Number of triplets in the path pf URL	Num Ampersand
Length of URL	Button	Number of triplets in domain name	Num Hash
Disabling Right Click	Non UTF-8 Char	Number of Phishing Keywords in URL	Num Numeric Chars
Pop-up Window	Checkbox	Website Owner	Random String
Age of Domain	BlackListed Link	Abnormal DNS record	Domain In Paths
URL of Anchor	Title has special char	Abnormal anchors	Hostname Length
Prefix or Suffix	Date Time	Abnormal server form handler	Path Length
Request URL	Name or Surname	Abnormal certificate in SSL	Query Length
Abnormal URL	Phone Number	Number of web pages	Double Slash In Path
DNS Record	Meta Tag	Average number of inbound links	Num Sensitive Words
Web Traffic	Downloadable Content	Average number of internal links	Embedded Brand Name
Google Index	Cookie	Average number of input boxes	Pct Ext Resource Urls
Number of Links Pointing to Web	Cache	Average number of password boxes	Insecure Forms
Statistics Report Based Features	Copyright	Proportion of form links	Relative Form Action
Page Rank	Readable HTML	Dynamic web page proportion	Ext Form Action
Title	Blacklist Word Usage	Count of unsymmetric tags	Abnormal Form Action
Tiny URL	Option Element	Count of JavaScript segments	Pct Null Self Redirect Hyperlinks
'//' usage - redirect	Select Element	Count of plugins	Frequent Domain Name Mismatch
Number of dots in URL	T H Element	Count of Active X controls	Images Only In Form

Tablo 3.2 Literatür araştırması sonucu listelenen 168 özellik - 2

Özellik	Özellik	Özellik	Özellik
Links in tags	T R Element	Count of long string	Subdomain Level RT
Status Bar Customization	Table Element	Count of Unicode char	Pct Ext Resource URLs RT
Server Form Handler	LI Element	Count of Hex and Octal coding	Abnormal Ext Form Action R
Form	UL Element	Count of replace () function call	Ext Meta Script Link RT
Image	Div Element	Count of eval () and exec () function	Pct Ext Null Self Redirect Hyperlinks RT
Password	Span Element	Count of string functions	Result
Hidden Element	Article Element	Count of obfuscation function	Website in Search Engine Results
Input Element Count	P Element	Evaluation of meta description	Frequency of domain in anchor links
Href Element	Content Word	Evaluation of meta keywords	Frequency of domain in CSS, image, Script links
Content Spec Char	Blacklist Word Count	Evaluation of script	Common page detection ratio in Website
Image Count	HTTP Link	Grayscale histogram	Common Page detection ratio in Footer
Long URL that hides suspicious part	Meta Tag Count	Color histogram	Null Links ratio in website
Using Non-Standard Port	HTML Element Count	Spatial Relationship btw subgraphs of image	NULL Links ratio in Footer
Special Char	Checkbox Count	TLD evaluation in the domain name	Broken links ratio

Bunun üzerine listedeki özelliklerden içerik-bazlı ve spesifik olarak HTML tabanlı olan 21 özellik, bir önceki aşamada elde edilen 27 özelliğe eklenmiş ve içerik-tabanlı toplam özellik sayısı 48'e çıkarılmıştır. Bu 48 özellik Tablo 3.3'te gösterilmiştir. Geliştirilecek olan Makine Öğrenmesi ve Derin Öğrenme temelli modellerde kullanımı kolaylaştırmak için tüm özellikler sayısal değere sahip olacak şekilde kurgulanmıştır. Özellik ismi İngilizce "Is" ya da "Has" ifadeleri ile başlıyorsa, özellik "Boolean" değere sahip olabilir ve bu değer o özelliğin sayfa içerisinde var olup olmadığını gösterir. "True" değeri için 1, "False" değeri için de 0 kabul edilmiştir. Özellik isminde İngilizce "Count" ifadesi bulunan özellikler ise sayfa içerisinde o özelliğin kaç kez geçtiğini gösterir.

Tablo 3.3 İçerik-tabanlı olmayan çalışmaların elenmesi sonucu elde edilen 48 özellik

Özellik	Özellik	Özellik	Özellik
Link Count	Has Hidden Tag	Popup Count	Is Onmouseover
Href Count	Hidden Tag Count	Title Count	Table Count
Has Submit Input	Has Iframe	Form Count	TH Count
Submit Input Count	Iframe Count	Has Server Form Handler	TR Count
Has Email Input	Footer Count	Image Count	Span Count
Email Input Count	Has Popup	Has Password Input	Div_Count
Paragraph_Count	Is Favicon	List_Count	Header1_Count
Button_Count	Title Length	UnorderedList_Count	Header2_Count
Meta_Count	Style_Count	Label_Count	Select_Count
Base Count	Video_Count	Audio_Count	Script_Count
Address_Count	Canvas_Count	Nav_Count	Figure_Count
Section_Count	Option_Count	Stylesheet_Count	Ins_count

Tablo 3.3'te belirtilen özelliklere ait tanımlar aşağıda verilmiştir:

Link_Count. İnternet sayfa içeriğinde yer alan toplam bağlantı sayısıdır.

Href_Count. İnternet sayfasında yer alan tüm "<a>" etiketlerini bulur ve bunların içerisinde yer alan toplam "*Href*" elementlerinin sayısını temsil eder. "*Href*" elementi HTML dosya içeriğindeki "*anchor*" etiketinin bir özelliğidir ve "*Hypertext REFerence*" unsurlarını ifade eder.

Has_Submit_Input / Submit_Input_Count. Saldırganlar genellikle kullanıcıların hassas bilgilerini ele geçirebilmek için kullanıcılardan bazı bilgileri "*submit*" etmesini isterler. Bu özellik internet sayfasında "*submit*" elementinin olup olmadığını kontrol etmekte ve varsa toplam sayısını temsil etmektedir.

Has_Email_Input / Email_Input_Count. İnternet sayfası içerisinde yer alan tüm "*input*" elementleri bulur. İçlerinde "*email input*" elementi olup olmadığını kontrol eder, varsa toplam sayısını temsil eder.

Has_Hidden_Tag / Hidden_Tag_Count. Oltalama saldırısını planlayanların en temel hedeflerinden biri de tasarladıkları sahte internet sayfasının içeriğini kullanıcıdan

gizlemektir. Bu özellik ile internet sayfası içerisinde "*Hidden*" yani gizlenmiş etiketin olup olmadığı kontrol edilir, varsa toplam sayısını temsil eder.

Has_Iframe / Iframe_Count. "*Iframe*" yani "*Inline Frame*" internet sayfa içeriğindeki dikdörtgen kısımları tanımlar ve HTML içerisine başka dosyaları gömmek için kullanılır. Bu özellik ile internet sayfa içeriğinde "*iframe*" etiketinin olup olmadığı kontrol edilir, varsa toplam sayısını temsil eder.

Has_Popup / Popup_Count. İnternet sayfa içeriğinde özellikle reklam amacıyla ya da kullanıcının dikkatini çekmek için kullanılan "*Popup*" elementlerinin olup olmadığın kontrol eder, varsa toplam sayısını temsil eder.

Has_Password_Input. Oltalama saldırılarını planlayanların elde etmek istediği en kritik bilgilerin başında "*password*" yani şifre bilgisini ele geçirmek gelir. Bu özellik internet sayfa içeriğinde "*password*" elementinin olup olmadığını kontrol eder.

Has_Server_Form_Handler. İnternet sayfa içeriğinde "*Server Form Handler*" olup olmadığını kontrol eder.

Is_Onmouseover. İnternet sayfa içeriğinde "*Onmouseover*" özelliğinin kullanıp kullanmadığını kontrol eder.

Title_Length. Bu özellik HTML içeriğindeki "<*title*>" etiketlerini bulur ve içlerinden en uzun elementi temsil eder.

Tablo 3.3'te yer alan diğer özellikler ise internet sayfa içeriğindeki HTML elementinden kaç adet bulunduğunu temsil etmektedir.

Literatürde yapılan çalışmalarda kullanılan içerik-tabanlı özelliklerin inceledikten sonra HTML-bazlı özelliklerin tercih edildiği görülmüştür. Bu noktada yeni özellik oluşturabilmek ya da keşfedebilmek amacıyla HTML "*tutorial*" ve dosyalar detaylı olarak analiz edilmiştir. Literatürde gözlemlenemeyen ancak yapılan analizler sonucu oltalama tespit amacıyla kurulan Makine Öğrenmesi ve Derin Öğrenme modellerinde ayırt edici olabileceği öngörülen 9 yeni özellik belirlenmiştir. Bu özellikler ve açıklamaları Tablo 3.4'te detaylandırılarak verilmiştir. Bu özelliklerin de eklenmesi ile tamamı içerik-tabanlı olan toplam 57 özellik elde edilmiştir.

Tablo 3.4 Analiz sonrası oluşturulan özellikler

Özellik	Açıklama
Meta Content Count	HTML "*meta*" etiketi içerisinde yer alan "*content*" öznitelik sayısını temsil eder.
Script Language Count	HTML "*script*" etiketi içerisinde yer alan "*lang*" öznitelik sayısını temsil eder.
Div Class Count	HTML "*div*" etiketi içerisinde yer alan "*class*" öznitelik sayısını temsil eder.
BR Count	HTML "*br*" etiketi sayısını temsil eder.
Text Size	Tüm HTML etiketleri içerisinde yer alan metin uzunluğunu temsil eder.
Line Size	HTML dosyasında yer alan toplam satır sayısın temsil eder.
Comment Line Count	HTML dosyasında yer alan toplam yorum satırı sayısını temsil eder.
Is Div in Lowercase	HTML "div" etiketlerini kontrol eder ve hem büyük hem de küçük harf ile yazılı etiket olup olmadığını kontrol eder.
Is Script in Lowercase	HTML "script" etiketlerini kontrol eder ve hem büyük hem de küçük harf ile yazılı etiket olup olmadığını kontrol eder.

3.2 Özellik Seçimi

Yapılan literatür araştırmasında rastlanmamış olan ve bu tez kapsamında yapılan analizler sonucunda belirlenmiş olan özelliklerin başarısını ve performansını görebilmek adına iki aşamalı bir özellik seçimi kurgulanmıştır:

- Birinci Aşama: Python dili için yazılmış ve açık kaynak bir kütüphane olan Scikit-learn ile özellik sıralaması çalışması yapılmıştır. Scikit-learn'de özellik seçimi için geliştirilmiş farklı modüller kullanılarak karşılaştırmalı sonuçlar listelenmiştir.
- İkinci Aşama: Tüm özellikler oltalama ve meşru internet siteleri için ayrı ayrı analiz edilerek, özelliklerin hangi tür internet sitelerinde ne oranda kullanıldığı hesaplanmıştır. Tüm özellikler sayısal değere sahip olduğu için ayrıca her özelliğin oltalama ve meşru internet siteleri için ortalama değerleri bulunmuştur. Böylece kullanım oranlarına ve ortalama değerlerine göre ayırt edici veya sistem için önemli özellikler listelenmiştir.

3.2.1 Birinci Aşama

Birinci aşamada Scikit-learn kütüphanesinden özellik seçimi için hazırlanmış modül olan "*sklearn.feature_selection*" bünyesindeki sınıflandırma yöntemleri incelenmiş ve özellik sıralaması için üç farklı yöntem seçilmiştir. Bunlar sırasıyla "*SelectKBest*", "*SelectPercentile*" ve "*GenericUnivariateSelect*" sınıflarıdır. Bu sınıflar ile yapılan çalışmada parametre olarak

"*chi2*" kullanılmıştır. Üç farklı yöntem ile sistemin başarısı için etkili olacak özellikler sıralanmıştır. Bu sıralamalar "*SelectKBest*" için Tablo 3.5'te, "*SelectPercentile*" için Tablo 3.6'da ve "*GenericUnivariateSelect*" için ise Tablo 3.7'de verilmiştir.

Tablo 3.5 SelectKBest ile elde edilen özellik sıralaması

No	Özellik	No	Özellik	No	Özellik	No	Özellik
1	Line Size	16	Meta Count	31	Footer Count	46	Submit Input
2	Text Size	17	Meta Content	32	Stylesheet	47	Base Count
3	Href Count	18	Popup Count	33	Button Count	48	Style Count
4	Div Count	19	Link Count	34	Has Password	49	Audio Count
5	Div Class	20	Option Count	35	Form Count	50	Email Input
6	List Count	21	Is Onmouseover	36	Has Popup	51	Has Email
7	Span Count	22	Section Count	37	Has Iframe	52	Has Hidden
8	Image Count	23	Figure Count	38	Script Count	53	Canvas Count
9	Comment	24	H1 Count	39	Select Count	54	Has Submit
10	P Count	25	TH Count	40	Server Form	55	Address
11	UL Count	26	Table Count	41	Hidden Tag	56	Is Div Case
12	Title Length	27	Nav Count	42	Has Favicon	57	Is Script Case
13	BR Count	28	Insert Count	43	Video Count		
14	TR Count	29	Label Count	44	Title Count		
15	H2 Count	30	Iframe Count	45	Script		

Tablo 3.6 SelectPercentile ile elde edilen özellik sıralaması

No	Özellik	No	Özellik	No	Özellik	No	Özellik
1	Line Size	16	Meta Count	31	Has Password	46	Video Count
2	Text Size	17	Meta Content	32	Stylesheet	47	Base Count
3	Href Count	18	Popup Count	33	Button Count	48	Style Count
4	Div Count	19	Link Count	34	Footer Count	49	Audio Count
5	Div Class	20	Option Count	35	Form Count	50	Email Input
6	List Count	21	Is Onmouseover	36	Has Popup	51	Canvas Count
7	Span Count	22	Section Count	37	Has Iframe	52	Address Count
8	Image Count	23	Figure Count	38	Script Count	53	Has Hidden
9	Comment	24	H1 Count	39	Title Count	54	Has Email
10	P Count	25	TH Count	40	Server Form	55	Has Submit
11	UL Count	26	Table Count	41	Hidden Tag	56	Is Div Case
12	Title Length	27	Nav Count	42	Has Favicon	57	Is Script Case
13	BR Count	28	Insert Count	43	Submit Input		
14	TR Count	29	Label Count	44	Select Count		
15	H2 Count	30	Iframe Count	45	Script Language		

Üç farklı fonksiyon ile yapılan özellik sıralamalarının birbirleri ile hemen hemen aynı olduğu görülmüştür. Aralarındaki farklar sıralamaların son kısımlarında olmakta ve farklılıkların boyutu da oldukça küçüktür. Örneğin "SelectKBest" ile yapılan sıralamada 43'üncü sırada olan "Video Count" özelliği, "SelectPercentile" fonksiyonu ile yapılan sıralamada 46'ncı sırada yer almaktadır. Benzer bir durum "SelectPercentile" ve "GenericUnivariateSelect" fonksiyonları ile elde edilen sıralamalar arasında da tespit edilmiştir. Örneğin "SelectPercentile" ile elde edilen sıralamada 51'inci sırada olan "Canvas Count" özelliği, "GenericUnivariateSelect" fonksiyonu ile elde edilen sıralamada 55'inci sırada yer almaktadır. Aynı özellik "SelectKBest" için 53'üncü sırada yer almaktadır. Üç fonksiyon için de benzer sonuçlar elde edildiğinden dolayı çalışmanın ilerleyen kısımlarında kullanılmak üzere "SelectKBest" ile elde edilen sıralama referans olarak alınacaktır.

Tablo 3.7 GenericUnivariteSelect ile elde edilen özellik sıralaması

No	Özellik	No	Özellik	No	Özellik	No	Özellik
1	Line Size	16	Meta Count	31	Footer Count	46	Submit Input
2	Text Size	17	Meta Content	32	Form Count	47	Base Count
3	Href Count	18	Popup Count	33	Button Count	48	Style Count
4	Div Count	19	Link Count	34	Has Password	49	Audio Count
5	Div Class	20	Option Count	35	Stylesheet	50	Video Count
6	List Count	21	Is Onmouseover	36	Has Popup	51	Has Hidden
7	Span Count	22	Section Count	37	Has Iframe	52	Has Email
8	Image Count	23	Figure Count	38	Script Count	53	Has Submit
9	Comment	24	H1 Count	39	Select Count	54	Address Count
10	P Count	25	TH Count	40	Server Form	55	Canvas Count
11	UL Count	26	Table Count	41	Hidden Tag	56	Is Div Case
12	Title Length	27	Nav Count	42	Has Favicon	57	Is Script Case
13	BR Count	28	Insert Count	43	Email Input		
14	TR Count	29	Label Count	44	Title Count		
15	H2 Count	30	Iframe Count	45	Script Language		

3.2.2 İkinci Aşama

İkinci aşamada ise Scikit-learn kütüphanesinden yararlanılarak gerçekleştirilen özellik sıralaması ile veri seti üzerinde yapılan analizlerin karşılaştırılması yapılmıştır. Öncelikle tüm özelliklerin oltalama ve meşru örnekler için ayrı ayrı kullanım oranları hesaplanmış, oltalama ve meşru örnekler için ayrı ayrı elde edilen oranların ortalaması alınarak genel bir kullanım oranı ortalaması elde edilmiştir. Sonrasında tüm özellikler oltalama örneklerindeki kullanım oranı ortalamalarına göre büyükten küçüğe sıralandılar (Tablo 3.8 ve Tablo 3.9).

Tablo 3.8 Özelliklerin nümerik değerleri arasındaki fark ve oranı - 1

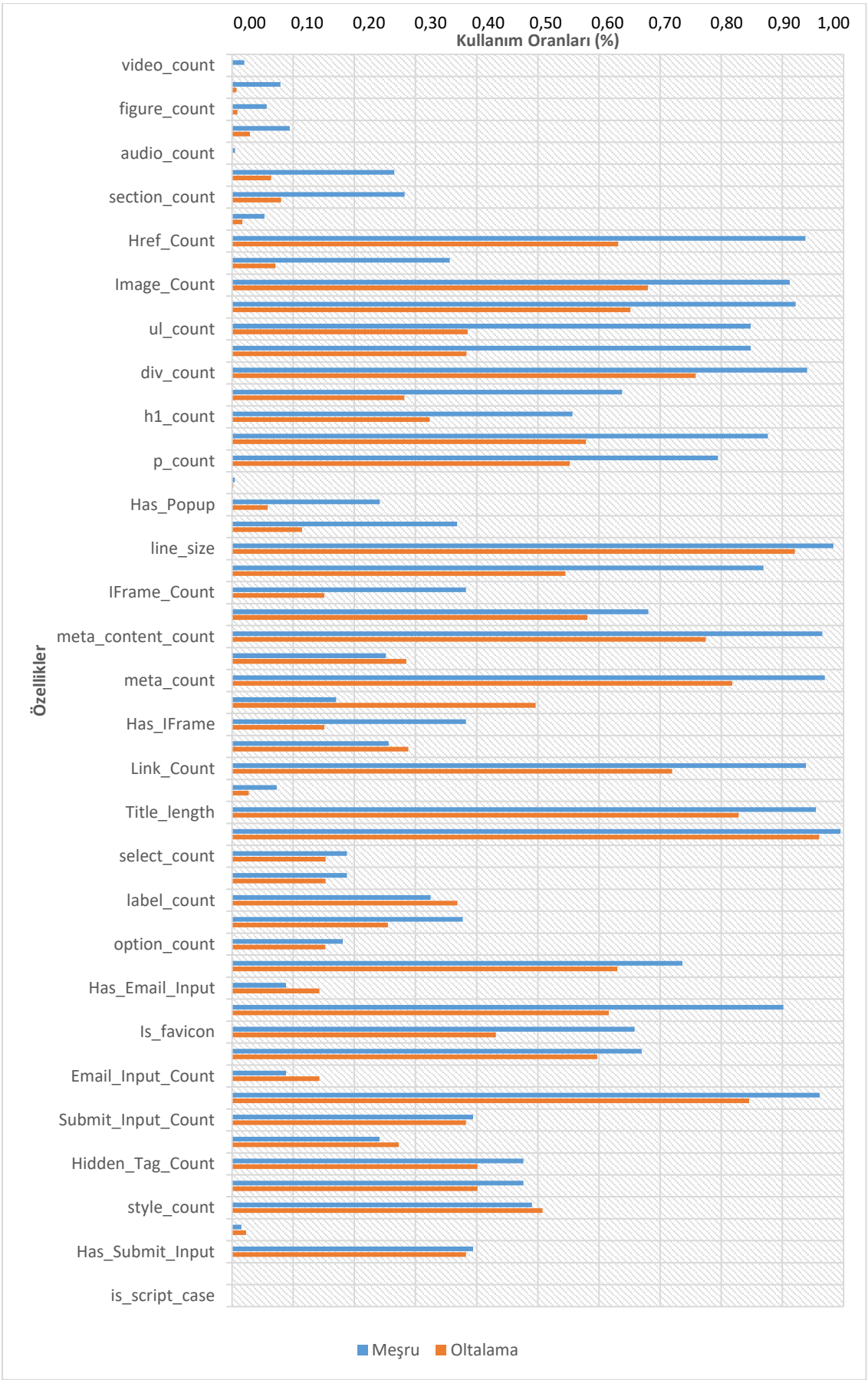
Özellik	Meşru	Oltalama	Mutlak Fark	Farkın Oranı (%)
video_count	0,0382	0,0003	0,0379	13.590,84
ins_count	0,4279	0,0094	0,4185	4.474,22
figure_count	1,6622	0,0803	1,5820	1.970,70
Is_Onmouseover	2,0222	0,0981	1,9241	1.960,51
audio_count	0,0155	0,0008	0,0147	1.751,63
Popup_Count	4,4697	0,3923	4,0774	1.039,37
section_count	2,2781	0,2075	2,0706	998,11
TH_count	0,9649	0,1018	0,8631	848,10
Href_Count	214,8447	27,8954	186,9493	670,18
nav_count	0,8184	0,1122	0,7062	629,15
Image_Count	48,8817	6,8926	41,9891	609,19
div_class_count	175,5251	27,3739	148,1513	541,21
ul_count	18,5291	3,2095	15,3195	477,31
li_count	109,8922	19,2076	90,6846	472,13
div_count	197,7597	36,4185	161,3411	443,02
h2_count	8,0132	1,4812	6,5319	440,98
h1_count	2,0077	0,4197	1,5880	378,41
Span_count	86,2278	18,5922	67,6356	363,78
p_count	27,2887	6,0799	21,2089	348,84
canvas_count	0,0168	0,0038	0,0130	344,72
Has_Popup	0,2414	0,0581	0,1833	315,70
Footer_Count	0,6849	0,1744	0,5106	292,81
line_size	23.252,4534	6.152,4412	17.100,0122	277,94
comment_count	34,2120	9,1525	25,0595	273,80
IFrame_Count	0,7957	0,2143	0,5814	271,31
br_count	17,3235	4,7642	12,5593	263,62
meta_content_count	10,2212	3,0768	7,1444	232,20
TR_count	17,8100	5,7396	12,0703	210,30
meta_count	10,9367	3,5595	7,3772	207,25
Has_Password_Input	0,2540	0,6772	0,4233	166,67

Tablo 3.9 Özelliklerin nümerik değerleri arasındaki fark ve oranı - 2

Özellik	Meşru	Oltalama	Mutlak Fark	Farkın Oranı (%)
Has_IFrame	0,3824	0,1511	0,2314	153,17
Table_Count	2,7078	1,1241	1,5837	140,89
Link_Count	9,6406	4,1019	5,5387	135,03
base_count	0,0733	0,0317	0,0416	131,36
Title_length	53,0078	26,4137	26,5942	100,68
text_size	26.666,1285	14.477,9169	12.187,2115	84,18
select_count	0,5701	0,3110	0,2591	83,29
script_count	0,5701	0,3110	0,2591	83,29
label_count	2,6191	1,4308	1,1883	83,05
button_count	1,6958	0,9779	0,7178	73,40
option_count	19,9325	11,6797	8,2527	70,66
Form_Count	1,6345	1,0006	0,6339	63,36
Has_Email_Input	0,0882	0,1430	0,0548	62,08
stylesheet_count	3,9998	2,5894	1,4104	54,47
Is_favicon	0,6583	0,4317	0,2266	52,51
Server_Form_Handler	1,3209	0,9144	0,4064	44,45
Email_Input_Count	0,1397	0,1853	0,0455	32,57
Title_Count	1,1637	0,8885	0,2753	30,98
Submit_Input_Count	0,7105	0,5442	0,1663	30,56
script_language_count	0,9673	1,2221	0,2549	26,35
Hidden_Tag_Count	3,3827	2,7367	0,6460	23,61
Has_Hidden_Tag	0,4766	0,4015	0,0751	18,70
style_count	1,4531	1,6644	0,2113	14,54
address_count	0,0251	0,0228	0,0023	10,15
Has_Submit_Input	0,3940	0,3827	0,0114	2,97
is_div_case	0,0000	0,0000	0,0000	0,00
is_script_case	0,0000	0,0000	0,0000	0,00

Tez çalışması kapsamında çıkarılan tüm özelliklerin sayısal karşılıkları bulunmaktadır. Özelliklerin değerlerini elde etmek için oluşturulan fonksiyonlar veri setinde yer alan örneklerin tamamı için çalıştırılmış olup, oltalama ve meşru

örnekler için ayrı ayrı ortalama değerler hesaplanmıştır. Değerler farklı spektrumda olduğu için öncelikle her bir özelliğin ortalama örneklerdeki değeri ile meşru örneklerdeki değeri arasındaki fark alınmıştır. Aradaki fark değerlerini birbirleri ile karşılaştırabilmek için bu farkın düşük olan değere oranı hesaplanmıştır. Bu sayede meşru ve oltalama örnekleri arasında özellik bazlı nasıl farklılıklar olduğu ölçülmüştür. Özelliklerin sayısal değerlerinin farkını ortaya koyan grafik, oranlar büyükten küçüğe olacak şekilde Tablo 3.8'de detaylı olarak gösterilmiştir. Tablo 3.8 ve Tablo 3.9'da de görüldüğü üzere kalın harflerle vurgulanan ilk 35 özelliğin değerleri incelendiğinde, oltalama ve meşru örnekler için elde edilen ortalama değerlerin arasında en az iki kat fark bulunmaktadır. Bu fark, özelliklerin oltalama ve meşru örnekler için ayırt edici özellikler olduğunu göstermiştir. Son iki özellik ise veri setindeki örneklerde hiç kullanılmadığı için sıfır değerini almıştır. Şekil 3.1'de oltalama örnekler için turuncu, meşru örnekler için mavi renk ile özelliklerin kullanım oranları gösterilmiştir. Tablo 3.8 ve Şekil 3.1 birlikte değerlendirildiğinde özelliklerin Makine Öğrenmesi ve Derin Öğrenme modellerinde etkili olabilmesi için hem kullanım oranı hem de ayırt ediciliğinin yüksek olması gerektiği görülmüştür. Ayırt edici özellik tablosunda "*Video Count*" yani içerikte bulunan "*video*" etiketli elementlerin sayısını temsil eden özellik, oldukça ayırt edici olmasına rağmen kullanım oranı çok düşük olduğu için Scikit-learn kütüphanesinde yer alan modüller için önem sırası düşük bir özellik olarak belirlenmiştir.



Şekil 3.1 Özelliklerin kullanım oranları

3.3 Makine Öğrenmesi Temelli Sınıflandırma Algoritmaları

Makine Öğrenmesi algoritmaları temel olarak eğitim verisi ile öğrenme sürecini tamamlayan ve yeni veriyi doğru şekilde sınıflandırabilen veya tahmin edebilen algoritmalardır. Bu algoritmaların eğitimi için yapısal ya da yapısal olmayan veriler kullanılabilir. Sınıflandırma algoritmalarında temel amaç, öğrenme sürecini tamamlayan algoritmanın daha önce karşılaşmadığı veriyi doğru sınıfa atayabilmesidir. Sınıflandırma algoritmaları iki ana eksende toplanmaktadır. İlki "İkili Sınıflandırma" ve ikincisi "Çoklu Sınıflandırmadır". İkili sınıflandırma olası iki sonuç söz konusu iken çoklu sınıflandırmada ikiden daha fazla olası sonuç söz konusudur. Eğiticili sınıflandırma algoritmaları "Mantıksal Öğrenme Algoritmaları", "Destek Vektör Makineleri", "İstatistik Tabanlı Algoritmalar" ve "Tembel Öğrenme Algoritmaları" olarak dört ana başlıkta toplanabilir. P. C. Sen ve arkadaşları yaptıkları çalışmada sınıflandırma hızı ve netliği konusunda karar ağaçları gibi mantıksal öğrenme algoritmalarının daha başarılı olduğunu göstermiştir [54].

Makine Öğrenmesi algoritmalarının başarısı problemin yapısına ve veri setine göre değişiklik göstermektedir. Bu tezde sınıflandırma problemlerinde oldukça başarılı olan Karar Ağaçları (KA), Rastgele Ormanı (RO) kullanılmıştır. Aynı zamanda Naive Bayes (NB), Destek Vektör Makinesi (DVM), K-En Yakın Komşu (KYK), Stokastik Gradyen Azaltma (SGA), Adaboost algoritmaları kullanılmış ve elde edilen sınıflandırma başarıları karşılaştırılmıştır. İlerleyen bölümlerde bu tez çalışmasında kullanılan Makine Öğrenmesi algoritmaları ve detayları verilmektedir.

3.3.1 Karar Ağaçları

Karar Ağaçları algoritması bilgisayar bilimindeki en yaygın veri yapılarından olan ağaç yapısını kullanmaktadır. Özellikle sınıflandırma problemlerinde sıklıkla kullanılan karar ağaçları kök, düğüm ve yapraklardan oluşmaktadır. Eğiticili sınıflandırma algoritmalarından biridir. Entropiyi en aza indirebilmenin amaçlandığı karar ağaçlarındaki kök düğüm ve alt düğümlerin seçiminde Gini, En Küçük Kareler" gibi yöntemler kullanılır. Aşırı öğrenmeden kaçınmak için "Minimum Description Length (MDL)" prensibinden faydalanılır [55]. Nihai olarak entropiyi en aza indirecek yöntemler ve prensipler ile tahmin başarısı maksimize edilir.

3.3.2 Rastgele Ormanı

Rastgele ağaçlardan oluşan bir yapıya sahiptir ve ağaç sayısı ile sınıflandırma başarısı arasında doğrudan bir ilişki bulunmaktadır. Karar Ağaçları ile Rastgele Ormanın farklılaştığı nokta aynı zamanda Rastgele Ormanın karakteristik özelliğini tanımlamaktadır. Aralarındaki en temel fark; Rastgele Ormanda kök düğüm ve alt düğümlerin seçiminde Karar Ağaçlarından farklı olarak rastgele seçim yapılır. Bu rastgelelilik yeterli sayıda ağaç olması durumunda aşırı öğrenme yani ezberlemeden kaçınmayı sağlamaktadır. Karar Ağaçlarındaki temel sorunlardan biri öğrenmenin sayısal olarak zor olmasıdır ve bu zorluk Rastgele Orman gibi algoritmalarda, sezgisel yaklaşımların dahil edilmesi ile aşılmaya çalışılır [56].

3.3.3 Naive Bayes

Makine Öğrenmesi problemlerinde oldukça yaygın olarak kullanılan ve özellikle öğrenme sürecini basitleştiren bir algoritmadır. 1812 yılında Thomas Bayes tarafından ortaya konulan "Bayes Teoremi" baz alınarak geliştirilen Naive Bayes sınıflandırıcısı temel olarak tembel bir öğrenme algoritmasıdır. Algoritma her bir durumun olasılığını hesaplar ve en yüksek olasılık değerine sahip olma durumuna göre sınıflandırma işlemini geçekleştirir. Burada kritik olan hususlardan biri özelliklerin birbirinden bağımsız olması ve sonuca olan etkilerinin aynı oranda olduğunun kabul edilmesidir.

3.3.4 Destek Vektör Makinesi

Eğiticili Makine Öğrenmesi sınıflandırıcılarından DVM'nin en temel özelliği sınıfları marjinler aracılığıyla birbirinden ayırması ve yeni gelen örneğin marjinin ne tarafında olduğuna göre konumlandırılmasını sağlamasıdır. Marjin içerisinde istisnai örnekler olmasına bağlı olarak "Hard" veya "Soft "olarak isimlendirilmektedir. Uygulamanın kolay olması ve ufak düzeltmelerle birden fazla problem için kullanılabilir olması kullanım yaygınlığını artırmaktadır. DVM'nin bir diğer ayırt edici özelliği kullanılacak olan veri seti hakkında bir ön bilgi söz konusu değildir. Aşırı öğrenme sorunu ile karşılaşmadan hem doğrusal hem de doğrusal olmayan veriler için kullanılabilmektedir. En büyük dezavantajı ise özellikle geniş veri setlerinde ve çoklu sınıflandırma problemlerinde çalışma süresinin çok uzun olmasıdır. Literatürde bu problemle ilgili çalışmalar yapılmaya devam etmektedir.

Örneğin [57] çalışmasında DVM ile ağırlıklandırılmış Öklid mesafesi kullanarak öğrenme süresini kısaltmıştır.

3.3.5 K-En Yakın Komşuluk

Hem regresyon hem de sınıflandırma problemlerinde kullanılabilen K-En Yakın Komşuluk (KYK), uygulaması kolay bir algoritmadır. Temel olarak örnekler k sayıdaki komşuluklarına olan uzaklıklarına göre sınıflara ayrılmaktadır. Öklid, Manhattan gibi farklı uzaklık ölçümleri kullanılabilmektedir. Özellikle gürültülü verilerde oldukça başarılı sonuçlar veren bu algoritma, komşuluk hesabı mekanizmasında tuttuğu ve işleme aldığı veri sebebiyle veri miktarı arttıkça ihtiyaç duyduğu bellek miktarı da daha fazla artmaktadır. Mevcut veri için en iyi K değerini görebilmek için farklı K değerleri ile sınıflandırma yapılır. KYK algoritması literatürde sıklıkla kullanılmaktadır ve hesaplama yükü dezavantajını giderebilmek için çeşitli çalışmalar yapılmaktadır [58].

3.3.6 Stokastik Gradyen Azaltma

Geniş ölçekli Makine Öğrenmesi problemlerinde oldukça başarılı sonuçlar veren Stokastik Gradyen Azaltma (SGA), özellikle doğrusal sınıflandırıcılar için oldukça etkili bir yaklaşıma sahiptir. Her ne kadar uygulanması kolay ve verimli bir algoritma olsa da özellik ölçeklendirme bakımından hassas bir algoritmadır. Aynı zamanda "iterasyon" sayısı gibi hiper parametrelere ihtiyaç duymaktadır. Çoklu sınıflandırma problemleri için de uygulanabilmektedir. SGA yapay ağların öğrenme sürecinde de başarılı bir şekilde kullanılabilmektedir. R. G. J. Wijnhoven ve arkadaşları nesne algılamada öğrenme sürecini hızlandırabilmek için yaptıkları çalışmada SGA'yı kullanmış ve SGA'nın sınıflandırma performansına ciddi bir katkıda bulunduğunu belirtmişlerdir [59].

3.3.7 AdaBoost

İngilizce "Adaptive" yani uyarlanabilir ve "Boosting" yani arttırma kelimelerinin birleşiminden oluşan AdaBoost algoritması, arttırma algoritmalarının başında gelmektedir. AdaBoost; Yoav Freund ve Robert Schapire tarafından geliştirilmiş ve 2003 Gödel Ödülü kazanmıştır. Derinliği "1" olan karar ağaçlarından oluşan AdaBoost algoritmasında; önceki zayıf öğrenicilerin hatalı sınıflandırmalarından elde edilen çıktı sonraki adımın girdisi olup, algoritma bu girdi ve çıktı bilgilerine göre güncellenir. Mevcut sınıflandırıcıların performans artırımı için de

kullanılabilmektedir [60]. Uygulanması kolay ve hızlı olan bu algoritmayla küçük ve dengesiz veri seti kullanılarak başarılı çalışmalar yürütülmüştür. Örneğin bir çalışmada indüksiyon motorlarındaki hata tespiti için AdaBoost algoritmasından yararlanılmış ve yüksek hassasiyette sonuçlar elde edilmiştir [61].

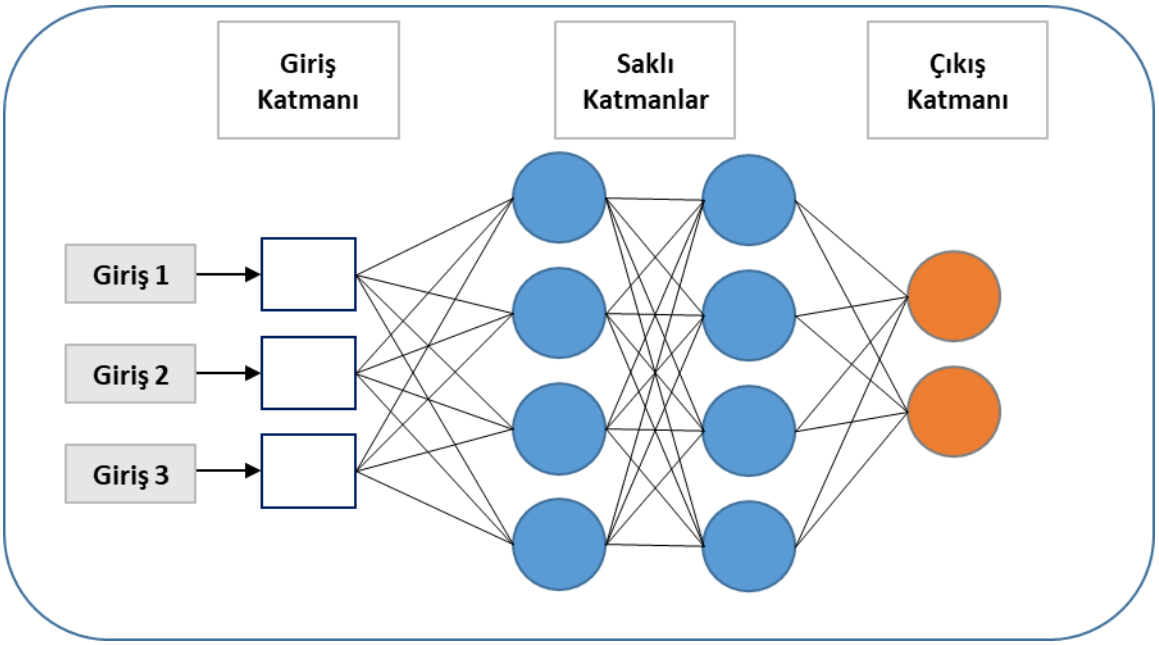
3.4 Derin Öğrenme Temelli Teknikler

Derin Öğrenme Makine Öğrenmesinin bir alt kümesi olup Yapay Sinir Ağları yapısından esinlenerek oluşturulmuştur. Örnek bir Yapay Sinir Ağı yapısı Şekil 3.2'de gösterilmiştir. Gerçek yapay sinir ağından esinlenilen bu yapıdaki önemli kavramlar Girdiler, Ağırlıklar, Birleştirme Fonksiyonu, Aktivasyon Fonksiyonu ve Çıktılardır. Girdiler öncelikle geldikleri bağlantı yolunun ağırlığı ile çarpılmaktadır. Sonrasında ağırlıklandırılmış tüm girdiler Birleştirme Fonksiyonu ile o hücrenin nihai girdisi olarak hesaplanır. Bu noktada Aktivasyon Fonksiyonu toplam girdiyi kullanarak yeni bir Çıkış değeri üretmektedir. Bu değer de bir sonraki katman için girdi değerini oluşturmaktadır. Derin Öğrenmede eğiticili, yarı-eğiticili veya eğiticisiz öğrenme de kullanılabilmektedir. Temel olarak Giriş Katmanı, Gizli/Saklı Katman ve Çıkış Katmanından oluşmaktadır. Çok daha hızlı ve başarılı sonuçlar elde edebilmek için geliştirilen Derin Öğrenme teknikleri büyük veriye ihtiyaç duymaktadır. Birçok öğrenme algoritması için veri miktarının artışı bir noktadan sonra performans ve başarı bakımından olumlu etki etmezken, derin öğrenme kavramında veri miktarındaki artış performansı olumlu etkilemeye devam etmektedir. Aslında bu öğrenme biçimine "derin" denilmesinin ve yapay sinir ağlarından farklılaşmasının temel sebebi de Giriş ve Çıkış katmanları arasında yer alan katmanların ve kullanılan verinin artmasıdır. Diğer bir deyişle Derin Öğrenme kavramı öğrenme sürecinde "Geri Yayılım" kullanan büyük yapay ağlardır denilebilir. "Çok Katmanlı Algılayıcı Ağları", "Evrişimsel Sinir Ağları" ve "Uzun Kısa-Süreli Bellek Yinelenen Sinir Ağları" en yaygın Derin Öğrenme tekniklerindendir.

3.4.1 Çok Katmanlı Algılayıcı Ağları

Çok Katmanlı Algılayıcı ağ yapısı temel olarak Tek Katmanlı Algılayıcı'nın birden fazla katmandan oluşan halidir. Tek Katmanlı Algılayıcı yapısı giriş, çıkış ve saklı katmandan oluşmaktadır. Giriş katmanı, veri setindeki her bir örnek için özellikleri temsil ederken, sınıflandırma problemleri için çıkış katmanı her bir sınıfı temsil etmektedir. Tek Katmanlı Algılayıcıların yapısı karmaşık ve doğrusal bir çizgi ile

ayırt edilemeyen sınıflandırma problemleri için yetersiz kalmaktadır. Basit bir XOR kapısı örneği için bile Tek Katmanlı Algılayıcı ağ yapısı yeterli olmamaktadır. Çok Katmanlı Algılayıcı'da ise Giriş ve Çıkış Katmanları arasında birden çok Saklı Katman yer almaktadır. Örnek bir Çok Katmanlı Algılayıcı yapısı Şekil 3.2'de gösterilmiştir. Bu yüzden Çok Katmanlı Algılayıcı yapısı daha çok tercih edilmektedir. Çok Katmanlı Algılayıcı modelinin her katmanında ayrı bir aktivasyon fonksiyonu kullanılabilir. En popüler aktivasyon fonksiyonları "*Rectified Linear Unit (RELU)*", "*Sigmoid*" ve "*Tanh*"dır.



Şekil 3.2 Çok katmanlı algılayıcı yapısı

3.4.2 Evrişimsel Sinir Ağları

Evrişimsel Sinir Ağları'nın (ESA) yapısı Çok Katmanlı Yapay Sinir Ağları ile aynı karakteristik özelliklere sahip olmakla birlikte farklılaştığı temel nokta sahip olduğu katman yapısıdır. Özellikle görüntü işleme alanında sıklıkla kullanılmakta ve oldukça başarılı sonuçlar üretmektedir. 2012 yılında Bilgisayarla Görü yarışmasında elde ettiği başarıdan sonra giderek yaygınlaşan ESA; özellik çıkarımını kendi kendine yapması, sınıflandırma başarısını maksimize etmesi yönleriyle bu alandaki çalışanlar için tercih edilme oranını her geçen gün artırmaktadır. ESA'nın katmanları dört ana eksende toplanabilir. Bunlardan birincisi ESA'nın en belirleyici ve kritik katmanı olan Evrişimsel Katman'dır. Bu katman ilk sırada yer almakta ve bu katmanda filtreleme işlemi uygulanmaktadır (örneğin ayırt edici özellikleri

belirlenmeye çalışılır). Matris formatında piksellerden oluşan her bir örnek, yine bir başka matris olan filtre ile işleme sokularak yeni bir matris elde edilir. Birden fazla özellik tespit edebilmek için katman sayısı da birden fazla olmaktadır. İlk filtre uygulandıktan sonra küçülen giriş verisinin boyutunu korumak için boş kalan hücrelere sıfır eklenerek "*padding*" işlemi yapılır. İkinci Katman ise evrişimsel katmanlardan sonra gelen "*Pooling*" Katmanı'dır. Bu katmanda verinin karakteristik özellikleri korunurken boyutu azaltılır. Bu işlemler ağdaki hesaplama ve parametre sayısını azaltmak için yapılır. Performansı artıran bu işlem aynı zamanda aşırı öğrenmeden kaçınmak için de faydalıdır. Üçüncü katman ise tüm negatif değerlerin sıfıra eşitlendiği Rektifiye Doğrusal Birim veya literatürde bilinen adıyla "*Rectified Linear Unit* (*RELU*)" Düzeltme Katmanıdır. "*Sigmoid, Tanh*" gibi aktivasyon fonksiyonlarına göre "*RELU*"nun çok daha başarılı olduğu görüldükten sonra bu katmanda genellikle "*RELU*" kullanılmaktadır. Evrişimsel Sinir Ağlarının son ve dördüncü katmanı ise Tam Bağlantılı Katman'dır. Bu katman her daim son kısımda yer alır ve örnekteki özellikler ile sınıf arasındaki ilişkiyi belirler.

3.4.3 Uzun Kısa-Süreli Bellek Yinelenen Sinir Ağları

Uzun Kısa Süreli Bellek ile Yinelenen Sinir Ağlarını yani literatürde kullanılan yaygın kısaltmaları ile ele alırsak sırasıyla LSTM ve YSA'yı öncelikle ayrı ayrı ele almalıyız. İlk olarak YSA kavramını açmak gerekirse, en temel anlamıyla geleneksel sinir ağları modeli bir önceki adımı hatırlama özelliğine sahip değildir ve YSA bu noktadaki döngüsel yapısıyla hatırlama özelliğine sahiptir. Aynı zamanda döngüsel yapısıyla ilintili olarak YSA'deki girişler de birbirleriyle ilişkilidir. Hatırlama özelliği girişler arasındaki ilişki nedeniyle sağlanır. Hatırlama özelliği sayesinde birçok alanda başarılı bir şekilde kullanılır. Sınıflandırma ve regresyon problemleri başta olmak üzere metin ve ses ile ilgili çalışmalarda oldukça etkili sonuçlar elde edilmesine olanak sağlar. En büyük dezavantajı ise giriş verilerinin birbirleriyle ilişkili yapısı nedeniyle çok derin yapılar kurulması ve bunların işlenmesi zordur. Buna bağlı olarak YSA uzun bağlılık gerektiren durumlarda sorun yaşamaktadır. Örneğin bir metin verisini ele alırsak "Birçok maden topraktadır" cümlesindeki son kelimeyi tahmin edebilmek için bu cümlenin öncesindeki içeriğe ihtiyaç yoktur. Ancak bazı durumlarda çok daha uzun bağlama ihtiyaç duyulabilmektedir. İşte bu noktada LSTM ortaya koyduğu yaklaşım ile YSA'da yaşanılan problemi gidermektedir. YSA'nın özel bir türü olan LSTM ağları uzun dönemli bağlılık gerektiren pek çok

problem sahasında başarılı sonuçlar vermektedir. Bunun en önemli sebebi LSTM'de bir sonraki katmana aktarım tek bir kanal üzerinden iletilmez. Örneğin [62] çalışmasında metin sınıflandırması için ESA ve LSTM kullanılmıştır. Yaptıkları çalışmada LSTM ESA'ya göre az bir farkla daha başarılı iken birlikte kullanımlarında bu başarı daha da artmıştır.

4
DENEYSEL SONUÇLAR

Sınıflandırma problemlerinde eğiticili Makine Öğrenmesi algoritmaları ile oldukça başarılı sonuçlar elde edilmektedir. Bu çalışmada ilk olarak farklı Makine Öğrenmesi algoritmaları ile sınıflandırma modelleri oluşturulmuş ve bu modellerin "Karışıklık Matrisleri" elde edilmiştir. Bu matristeki değerler aracılığıyla algoritmaların başarıları ölçülmüş ve başarılar karşılaştırılmıştır. Makine Öğrenmesi algoritmaları ile yapılan sınıflandırma çalışmalarına ait detaylar Bölüm 5.1'de detaylandırılmıştır. Bölüm 5.2'de ise aynı veri seti ile Derin Öğrenme teknikleri kullanılarak oltalama tespit çalışmaları yapılmış ve burada kullanılan tekniklere ait başarı ölçütleri raporlanmıştır. Son olarak Bölüm 5.3'te Makine Öğrenmesi ve Derin Öğrenme ile yapılan çalışmaların karşılaştırılması yapılmıştır. Deneysel sonuçlar Tablo 4.1'deki donanım üzerinde gerçekleştirilmiştir.

Tablo 4.1 Test makine özellikleri

Özellik	Değer
Bilgisayar	Lenovo – HuronRiver Platform
İşletim Sistemi	Microsoft Windows 10 Pro
Merkezi İşlemci Birimi	Intel® Core™ i5-CPU @ 2.30GHz, 2301 MHz, 2 Cores, 4 Logical
Rasgele Erişimli Bellek	4,00 GB
Tümleşik Geliştirme Ortamı	PyCharm 2020.1(CE)

4.1 Makine Öğrenmesi ile Sınıflandırma Deneyleri

Çalışmanın bu aşamasında kullanılmış olan tüm algoritmaların başarılarının ölçülebilmesi için "Karışıklık/Hata Matrisleri" elde edilmiştir. Örnek bir Karışıklık Matrisi Tablo 4.2'de verilmiştir. Gerçek Pozitif değeri bu çalışma için oltalama olarak etiketlenmiş ve algoritma tarafından da oltalama olarak sınıflandırılmış örnekleri temsil etmektedir. Yanlış Pozitif değeri ise bu çalışma için meşru internet sitesi olarak etiketlenmiş ve algoritma tarafından da meşru olarak sınıflandırılmış örnekleri temsil etmektedir. Yani GP ve YP değerleri oltalama olarak etiketlenmiş örnekleri temsil etmektedir. "YN" değeri ise gerçekte meşru olarak etiketlenmiş

ancak algoritma tarafından yanlış bir tahminle oltalama olarak sınıflandırılan örnekleri temsil etmekte iken, "GN" değeri ise gerçekte meşru olarak etiketlenmiş ve algoritma tarafından da meşru olarak sınıflandırılmış örnekleri temsil etmektedir. Diğer bir deyişle "YN" ve "GN" değerleri meşru örnekleri temsil etmektedir.

Tablo 4.2 Karışıklık/hata matrisi

		Gerçek	
		Pozitif	**Negatif**
Tahmin	**Pozitif**	Gerçek Pozitif	Yanlış Pozitif
	Negatif	Yanlış Negatif	Gerçek Negatif

Makine Öğrenmesi algoritmalarının başarılarını sağlıklı bir şekilde ölçebilmek için bazı değerlere birlikte bakmak önem arz etmektedir. Örneğin 90 adet pozitif ve 10 adet negatif örnekten oluşan test verisinin tamamını pozitif olarak sınıflandıran bir algoritmanın sadece GP değerine bakarsak %90 oranını elde ederiz. Ancak tek başına bu değer algoritmanın genel başarısı için bize sağlıklı bir bakış açısı sunmaz çünkü algoritma hatalı bir şekilde çalışıp tüm örnekleri pozitif olarak sınıflandırıyor olabilir. Bu durumda YN ve GN değerlerine bakıldığında algoritmanın hiç negatif sınıflandırma yapmadığı görülebilir.

Oltalama örneklerinin tespit edilebilmesi için kullandığımız sınıflandırma algoritmalarının "Doğruluk (*Accuracy*)", "Kesinlik (*Precision*)", "Duyarlılık/Hassasiyet (*Recall/Sensitivity*)", F-1 Skoru, Gerçek Pozitif Oranı (GPO), Gerçek Negatif Oranı (GNO), Yanlış Negatif Oranı (YNO), Yanlış Pozitif Oranı (YPO) değerleri elde edilmiştir. Bu değerleri elde etmek için kullanılan denklemler **Hata! B aşvuru kaynağı bulunamadı.**'de gösterilmiştir. Doğruluk değeri, başarılı olarak sınıflandırılan örneklerin tüm örneklere oranı ile elde edilir. Kesinlik değeri oltalama olarak sınıflandırılan oltalama örneklerinin oltalama olarak sınıflandırılan tüm örneklere oranı ile, Duyarlılık değeri de oltalama olarak sınıflandırılan örneklerin gerçekten tüm oltalama örneklerine oranı ile elde edilmektedir. F-1 Skoru ise Kesinlik ve Duyarlılık değerlerinin harmonik ortalaması alınarak elde edilir. GPO ise Duyarlılık değerinin bir diğer ifadesidir. GNO değeri ise meşru olarak

sınıflandırılan örneklerin tüm meşru örneklere oranı ile elde edilir. Diğer bir ifade ile "Seçicilik (*Selectivity*)" olarak adlandırılabilir. YNO değeri ise meşru olarak sınıflandırılan ancak oltalama saldırısı olan örneklerin tüm oltalama örneklere oranı ile elde edilir. Kaçırma oranı olarak da adlandırılabilen bu değer özellikle bu tarz güvenlik problemleri için oldukça kritiktir çünkü bu değer saldırı olmasına rağmen doğru sınıflandırılamayan yani sistemin gözden kaçırdığı tehditlerin oranını göstermektedir. Son olarak YPO değeri ise oltalama olarak sınıflandırılan ancak gerçekte meşru olan örneklerin tüm meşru örneklere olan oranı ile elde edilir. Bu değer de önem arz etmekle birlikte oltalama olmayan internet sitesinin tehdit olarak algılanması güvenlik bakımından doğrudan zafiyet yaratmaz ancak kullanıcının sisteme olan güvenini zedeler.

$\text{Doğruluk} = \frac{\text{Gerçek Pozitif + Gerçek Negatif}}{\text{Pozitif + Negatif}}$	$\text{Hassasiyet} = \frac{\text{Gerçek Pozitif}}{\text{Gerçek Pozitif + Yanlış Negatif}}$
$\text{Kesinlik} = \frac{\text{Gerçek Pozitif}}{\text{Gerçek Pozitif + Yanlış Pozitif}}$	$\text{F1 Skoru} = \frac{\text{2 x Gerçek Pozitif}}{\text{2 x Gerçek Pozitif + Yanlış Pozitif + Yanlış Negatif}}$
$\text{GPO} = \frac{\text{Gerçek Pozitif}}{\text{Gerçek Pozitif + Yanlış Negatif}}$	$\text{YPO} = \frac{\text{Yanlış Pozitif}}{\text{Yanlış Pozitif + Gerçek Negatif}}$
$\text{GNO} = \frac{\text{Gerçek Negatif}}{\text{Gerçek Negatif + Yanlış Pozitif}}$	$\text{YNO} = \frac{\text{Yanlış Negatif}}{\text{Yanlış Negatif + Gerçek Pozitif}}$

Şekil 4.1 Model başarısı ölçüt değerleri

İkili sınıflandırma yapan algoritmanın başarısı hesaplanırken hangi ölçüt değerlerin baz alınacağı veri setinin yapısına bağlı olarak değişiklik göstermektedir. Dengesiz bir sınıf yapısı olması durumunda "F1 Skoru" baz alınırken, veri setinde dengeli iki sınıf bulunması durumunda ise "Doğruluk" değeri baz alınmalıdır. Bu tez çalışmasında kullanılan veri setinde iki sınıf bulunmakta ve bu sınıflar yaklaşık %50 oranında dengeli bir dağılıma sahiptir. Bu doğrultuda modellerin başarısı karşılaştırılırken birincil olarak "Doğruluk" değeri referans alınmıştır.

4.1.1 Literatürden Seçilen 48 Özellik İçin Makine Öğrenmesi Algoritmalarının Performansı

İlk olarak Bölüm 4.2'de detayları verilmiş olan ve Tablo 3.3'de sıralanan literatürden elde edilmiş 48 özellik için Makine Öğrenmesi algoritmaları çalıştırılmıştır. Tablo

4.3'te RO, KA, DVM, GNB, SGA, KYK ve AdaBoost algoritmaları için elde edilen Doğruluk, GPO, GNO, YNO, YPO, Kesinlik, Duyarlılık ve F-1 Skoru değerleri sırasıyla listelenmiştir. Yapılan sınıflandırma çalışmalarında en başarılı Makine Öğrenmesi algoritması RO ve KA olarak tespit edilmiştir. Özellikle Doğruluk ve F-1 Skoru yönüyle değerlendirildiğinde en iyi algoritma RO olarak gözlemlenmiştir. Diğer algoritmalardan GNB hariç tüm algoritmaların %80'in üzerinde doğruluğa sahip olduğu ve Duyarlılık değeri göz önünde bulundurulduğunda da en düşük değere sahip algoritmanın %88,99 ile SGA olduğu tespit edilmiştir.

Tablo 4.3 Makine Öğrenmesi algoritma başarı değerleri (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	94,53	95,86	91,65	4,13	8,34	96,16	95,86	96,01
KA	93,42	96,11	87,97	3,88	12,02	94,18	96,11	95,14
DVM	83,73	90,62	71,38	9,37	28,61	84,96	90,62	87,70
GNB	60,89	94,84	44,47	5,15	55,52	45,24	94,84	61,26
SGA	81,90	88,99	69,05	11,00	30,94	83,89	88,99	86,37
KYK	89,38	92,59	82,65	7,4	17,34	91,82	92,59	92,20
AdaBoost	87,90	89,75	86,46	10,24	13,53	83,84	89,75	86,70

4.1.2 Analiz Sonrası Eklenen Yeni Özellikler ile 57 Özellik İçin Makine Öğrenmesi Algoritmalarının Performansı

İkinci aşamada ise oltalama site içeriklerinin analizi sonrası eklenen yeni özellikler ile oluşturulan 57 özelliğin tamamı aynı Makine Öğrenmesi algoritmaları için kullanılmıştır. Tablo 4.4'te sırasıyla RO, KA, DVM, GNB, SGA, KYK ve AdaBoost algoritmaları için elde edilen Doğruluk, GPO, GNO, YNO, YPO, Kesinlik, Duyarlılık ve F-1 Skoru değerleri listelenmiştir. Bu deneyde de en başarılı algoritmalar RO ve KA olurken aynı zamanda eklenen özellikler ile doğru orantılı olarak başarı ölçütlerinde de artış gözlemlenmiştir. Ancak özellikle DVM ve SGA algoritmalarında özelliklerin artması negatif bir etkide bulunmuştur. Bu iki algoritma için tüm değerlerde ciddi bir düşüş gözlemlenmiştir. GNB için ise Doğruluk değerinde bir miktar artış olmasına rağmen diğer değerlerde de önemli bir negatif etki gözlemlenmiştir. Eklenen özellikler KYK algoritmasında da negatif bir etki oluşturmuş ancak DVM ve

SGA'daki kadar dramatik bir etki görülmemiştir. AdaBoost algoritmasında ise az miktarda olumlu değişim gözlemlenmiştir.

Tablo 4.4 Makine Öğrenmesi algoritmaları başarı değerleri (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,67	97,51	97,82	2,49	2,18	97,54	97,51	97,53
KA	96,25	96,90	95,69	3,10	4,31	95,06	96,90	95,97
DVM	67,26	72,07	64,95	27,92	35,04	49,53	72,08	58,71
GNB	68,03	84,19	63,47	15,80	36,52	39,38	84,19	53,66
SGA	60,17	72,07	64,95	27,92	35,04	49,53	72,08	58,71
KYK	86,41	88,64	84,70	11,36	15,30	81,52	88,64	84,93
AdaBoost	88,23	90,31	86,61	9,69	13,38	83,97	90,31	87,03

4.1.3 Özellik Sayısındaki Değişimin Makine Öğrenmesi Algoritmalarının Performanslarına Etkisi

İlk olarak Bölüm 5.1.1'de literatürde elde edilen 48 özellik için deneysel sonuçlar alınmış ve devamında analiz sonrası eklenen 57 özellik için de deneyler tekrarlanmıştır. Algoritmaların bazılarında olumlu bir değişim gözlemlenirken bazılarında ise özellik sayısındaki artışın olumsuz etkisi görülmüştür. Bu kapsamda özellik sayısındaki artışın algoritma başarısına her zaman olumlu etki etmediği görülürken, bazı özelliklerin çıkarılmasının algoritma başarısına olumlu etkide bulunabileceği öngörülmüştür. Eklenen özellikler RO, KA ve AdaBoost için olumlu etkide bulunurken diğer algoritmalarda olumsuz etkide bulunmuştur. Bu durumda algoritmalar için ideal özellik sayısının değişiklik gösterebileceği görülmüştür. Aynı zamanda farklı özellik sayıları için en başarılı algoritmada da değişiklik gösterebilir. Bu durumda farklı özellik sayıları için en başarılı Makine Öğrenmesi algoritmasının ve en başarılı algoritma için ideal özellik sayısının bulunabilmesi amacıyla sırasıyla 5, 10, 15, 20, 25, 30 ,35, 40, 45, 50, 57 özellik için tüm algoritmalar tekrar çalıştırılmıştır. Bu aşamada özellikler, "Scikit-learn" kütüphanesindeki "SelecktKBest" fonksiyonu ile analiz edilmiş ve tüm özellikler en etkili özellikten en az etkili olan özelliğe doğru sıralanmıştır. Diğer bir ifade ile özellik sayısının 5 olduğu durumda "SelectKBest" fonksiyonunun sonuçlarına göre 57 özellik içerisindeki en iyi 5 özellik kullanılmıştır. Her özellik grubu için başarı ölçütlerini

gösteren on bir tablo oluşturulmuş ve en başarılı algoritma koyu font ile işaretlenmiştir.

Tablo 4.5 5 özellik için algoritma değerleri (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	96,71	97,03	96,43	2,96	3,56	95,94	97,03	96,48
KA	95,31	96,44	94,37	3,55	5,62	93,48	96,44	94,94
DVM	67,18	71,97	64,90	28,02	35,09	49,44	71,97	58,61
GNB	67,85	83,72	63,37	16,27	36,62	39,24	83,72	53,44
SGA	56,68	53,11	61,83	46,88	38,16	66,81	53,11	59,18
KYK	86,32	88,05	84,97	11,94	15,02	82,03	88,05	84,93
AdaBoost	83,76	81,93	85,47	18,06	14,52	83,98	81,93	82,94

Tablo 4.6 10 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	96,98	96,83	97,12	3,16	2,87	96,75	96,83	96,79
KA	95,66	96,98	94,56	3,01	5,43	93,68	96,98	95,30
DVM	67,15	71,80	64,91	28,19	35,08	49,60	71,80	58,67
GNB	69,41	84,50	64,70	15,50	35,29	42,78	84,49	56,80
SGA	57,81	53,96	63,78	46,03	36,21	69,77	53,96	60,86
KYK	86,03	88,27	84,33	11,72	15,66	81,05	88,27	84,51
AdaBoost	84,60	84,07	85,06	15,92	14,93	82,97	84,07	83,51

Tablo 4.7 15 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RA	97,05	97,32	96,81	2,67	3,18	96,38	97,32	96,85
KA	95,80	97,02	94,78	2,98	5,22	93,95	97,02	95,46
DVM	67,10	71,81	64,85	28,18	35,14	49,41	71,81	58,54
GNB	68,99	84,41	64,32	15,58	35,67	41,75	84,41	55,87
SGA	54,03	50,66	66,24	49,33	33,75	84,45	50,67	63,30
KYK	86,33	88,83	84,46	11,16	15,53	81,13	88,83	84,81
AdaBoost	85,38	85,68	85,12	14,31	14,87	82,71	85,68	84,17

Tablo 4.8 20 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,36	97,47	97,26	2,52	2,73	96,89	97,47	97,18
KA	95,96	96,93	95,15	3,07	4,85	94,40	96,92	95,65
DVM	67,31	72,12	65,00	27,87	34,99	49,64	72,12	58,81
GNB	69,12	84,30	64,46	15,69	35,53	42,17	84,30	56,22
SGA	60,60	55,76	70,02	44,24	29,97	78,34	55,76	65,15
KYK	86,08	87,93	84,65	12,07	15,35	81,59	87,93	84,64
AdaBoost	86,32	87,50	85,38	12,50	14,62	82,71	87,50	85,04

Tablo 4.9 25 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,29	97,26	97,33	2,73	2,67	96,97	97,27	97,12
KA	95,68	96,84	94,71	3,15	5,28	93,87	96,84	95,33
DVM	67,14	71,85	64,88	28,15	35,12	49,48	71,85	58,60
GNB	68,45	84,26	63,85	15,73	36,15	40,45	84,27	54,66
SGA	58,75	54,91	64,21	45,09	35,78	68,56	54,90	60,98
KYK	86,18	87,97	84,80	12,03	15,20	81,80	87,97	84,77
AdaBoost	86,06	86,91	85,37	13,09	14,62	82,82	86,91	84,82

Tablo 4.10 30 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,69	97,83	97,58	2,17	2,42	97,25	97,83	97,54
KA	95,94	97,06	95,00	2,94	5,00	94,22	97,06	95,62
DVM	67,19	71,97	64,91	28,02	35,09	49,46	71,98	58,63
GNB	67,95	84,07	63,42	15,93	36,58	39,27	84,07	53,53
SGA	57,10	52,68	71,71	47,32	28,29	86,00	52,68	65,34
KYK	86,07	88,28	84,40	11,72	15,60	81,15	88,28	84,56
AdaBoost	87,34	89,43	85,72	10,57	14,27	82,86	89,43	86,02

Tablo 4.11 35 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,47	97,37	97,57	2,63	2,43	97,25	97,36	97,31
KA	96,56	97,57	95,72	2,43	4,28	95,07	97,57	96,30
DVM	67,48	70,61	65,79	29,38	34,20	52,81	70,61	60,43
GNB	68,42	84,59	63,79	15,41	36,21	40,15	84,58	54,45
SGA	60,15	55,10	72,00	44,89	28,00	82,20	55,10	65,98
KYK	86,30	88,55	84,60	11,45	15,40	81,38	88,55	84,81
AdaBoost	88,00	90,13	86,35	9,86	13,64	83,62	90,14	86,76

Tablo 4.12 40 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,73	97,96	97,53	2,04	2,47	97,19	97,96	97,57
KA	96,44	97,13	95,83	2,86	4,17	95,22	97,14	96,17
DVM	67,24	71,97	64,97	28,03	35,03	49,64	71,97	58,76
GNB	68,14	84,30	63,57	15,70	36,43	39,62	84,30	53,91
SGA	56,71	52,42	70,78	47,58	29,22	85,49	52,42	64,99
KYK	86,47	88,33	85,03	11,67	14,97	82,06	88,33	85,08
AdaBoost	87,91	90,08	86,22	9,91	13,78	83,45	90,09	86,64

Tablo 4.13 45 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,67	97,55	97,78	2,45	2,22	97,49	97,55	97,52
KA	96,46	97,50	95,58	2,49	4,41	94,90	97,51	96,18
DVM	67,30	72,02	65,03	27,98	34,97	49,77	72,02	58,87
GNB	67,97	84,17	63,42	15,83	36,58	39,22	84,17	53,52
SGA	55,70	51,71	71,01	48,29	28,99	87,23	51,71	64,93
KYK	86,38	88,29	84,90	11,71	15,10	81,87	88,29	84,96
AdaBoost	88,04	90,24	86,35	9,75	13,65	83,61	90,24	86,80

Tablo 4.14 50 özellik için algoritma başarıları (%)

ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,52	97,53	97,51	2,46	2,49	97,18	97,53	97,35
KA	96,41	97,13	95,79	2,86	4,20	95,17	97,14	96,14
DVM	67,24	71,95	64,97	28,05	35,02	49,66	71,94	58,76
GNB	67,99	84,01	63,46	15,99	36,54	39,41	84,01	53,65
SGA	58,94	54,31	69,18	45,68	30,82	79,60	54,31	64,57
KYK	86,30	88,04	84,95	11,96	15,05	82,00	88,04	84,91
AdaBoost	87,98	89,79	86,55	10,20	13,44	83,97	89,80	86,79

Tablo 4.15 57 özellik için algoritma başarıları (%)

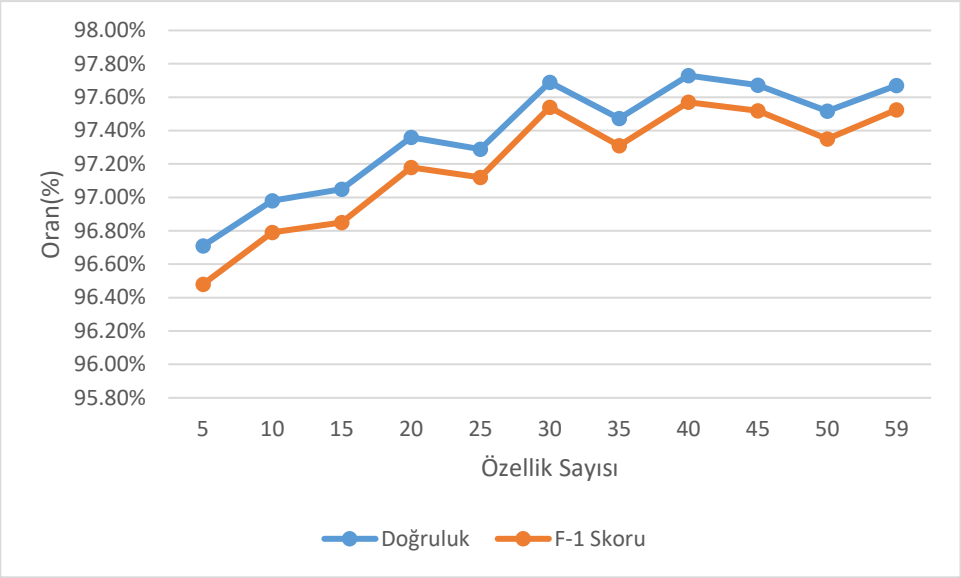
ALG	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
RO	97,67	97,51	97,82	2,49	2,18	97,54	97,51	97,53
KA	96,25	96,90	95,69	3,10	4,31	95,06	96,90	95,97
DVM	67,26	72,07	64,95	27,92	35,04	49,53	72,08	58,71
GNB	68,03	84,19	63,47	15,80	36,52	39,38	84,19	53,66
SGA	60,17	56,22	65,54	43,77	34,45	68,96	56,23	61,95
KYK	86,41	88,64	84,70	11,36	15,30	81,52	88,64	84,93
AdaBoost	88,23	90,31	86,61	9,69	13,38	83,97	90,31	87,03

Tablo 4.5'ten Tablo 4.15'e kadar görüldüğü üzere RO algoritması tüm özellik gruplarında en başarılı algoritma olarak gözlemlenmiştir. İkinci sırada her durum için KA yer alırken üç ve dördüncü en başarılı algoritmalar KYK ve AdaBoost algoritmaları olmuştur. DVM, GNB ve SGA algoritmaları tüm özellik grupları için %70 Doğruluk oranının üzerine çıkamamıştır.

4.1.4 En Başarılı Makine Öğrenmesi Algoritması İçin İdeal Özellik Sayısının Tespiti

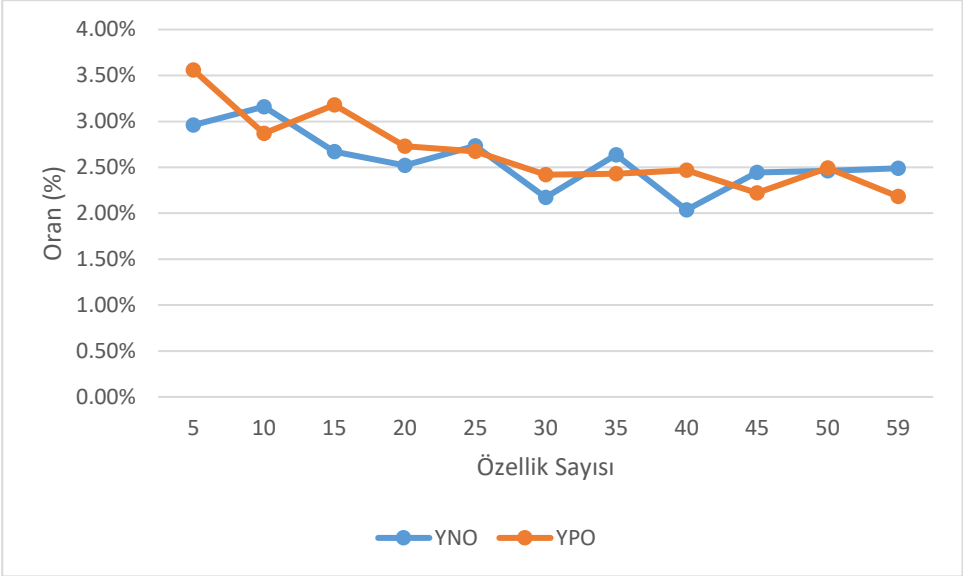
Bu bölümde en başarılı algoritma olarak gözlemlenen RO için özellik sayılarındaki değişimin algoritma başarısına etkisi analiz edilmiştir. İlk olarak algoritmanın oltalama örnekleri hangi doğrulukla sınıflandırdığını gösteren Doğruluk değeri, Kesinlik ve Duyarlılık değerlerinin harmonik ortalaması ile elde edilen F-1 Skoru birlikte incelenmiştir. Şekil 4.2'de gösterildiği gibi özellik sayısı arttıkça bu iki değer

önce artmış ancak sonrasında varyansı düşük olmakla birlikte inişli çıkışlı bir hale bürünmüştür. Şekil 4.2'de görüldüğü üzere bu iki değer için de en ideal özellik sayısı 40 olarak tespit edilmiştir.



Şekil 4.2 Doğruluk ve F-1 Skoru'nun özellik sayısına bağlı değişimi

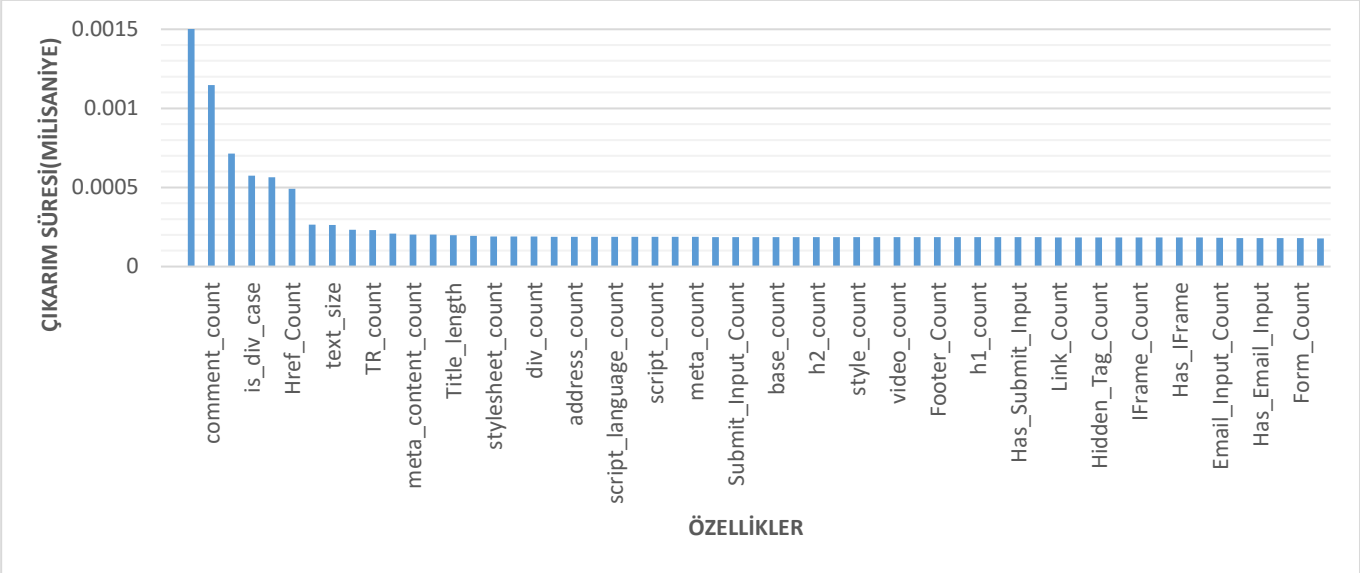
Siber tehdit tespit sistemlerinde sistemin güvenilirliği bakımından en önemli performans ölçütlerinden bir diğeri de YPO ve YNO değerleridir. Özellikle sistemin saldırıları gözden kaçırma oranını veren YNO değeri oldukça hayati öneme sahiptir. Şekil 4.3'te sırasıyla tüm özellikler için RO'ya ait YPO ve YNO değerleri verilmiştir. YPO için ideal özellik sayısı 45 ve 57 olurken, YNO için bu değer 40 olarak tespit edilmiştir. Bu durumda 40 adet özellik kullanılan modelde Doğruluk, F-1 Skoru ve YNO değeri optimal noktasına erişirken, YPO değeri için eklenen özellikler olumlu etkide bulunmaktadır. Aynı zamanda 45 ve 57 özellik olması durumunda Doğruluk ve F-1 Skoru değerlerinin YPO gibi olumlu anlamda değiştiği görülmektedir.



Şekil 4.3 YNO ve YPO değerlerinin özellik sayısına bağlı değişimi

Elde edilen bu değerler neticesinde ideal özellik sayısının 40 veya 57 olduğu ifade edilebilir. Bu noktada özellik sayısının artışına bağlı olarak algoritmanın çalışma süresinin de artabileceği göz önünde bulundurulmalıdır. Bunun için de özelliklerin ortalama çıkarım süreleri hesaplanmıştır. Tüm veri seti için özelliklerin çıkarım süreleri toplanmış ve toplam örnek sayısına bölünerek ortalaması alınmıştır. Tüm özelliklerin ortalama çalışma süresi $0{,}2375 \times 10^{-3}$ saniye olarak hesaplanmıştır. Şekil 4.4'te tüm özelliklerin çalışma süreleri büyükten küçüğe sıralı olarak verilmiştir. Soldan ilk beş özellik haricinde tüm özellikler ortalama sürenin altında veya çok yakınında bir değere sahiptir. Ortalamanın çok üzerinde çalışma süresine sahip özellikler için de ayrı bir değerlendirme yapılmalıdır çünkü bazı özellikler çalışma süresi bakımından dezavantaj oluştursa da algoritmanın başarısına olumlu etki edebilir. Bu da bir tür "*Trade-off*" durumudur. Yani bir değiş tokuş veya bir avantajı elde edebilmek için başka bir avantajdan vaz geçme durumu olarak adlandırabiliriz. Bu doğrultuda ortalamanın çok üzerinde olan beş özelliğin modele etkisi incelendiğinde "*Comment_Count*" ve "*Href_Count*" özelliklerinin ilk on özellik arasında yer aldığı, "*Has_Favicon*" özelliğinin kırk ikinci sırada yer aldığı ve "*Is_Div_Case*" ve "*Is_Script_Case*" özelliklerinin ise son iki sırada yer aldığı görülmektedir. Bu doğrultuda "*Trade-off*" kapsamında "*Is_Div_Case*" ile "*Is_Script_Case*" özelliklerinin çıkarılması algoritma performansı ve başarısı için

olumlu etkide bulunmuştur. Modelde kullanılacak tüm özelliklerin toplam çıkarım süresi bir internet sayfası için 0,1353 milisaniye olarak hesaplanmıştır. Bu nedenle



Şekil 4.4 Özelliklerin ortalama çıkarım süreleri

modelde kullanılan tüm özelliklerin model performansına etkisi kabul edilebilir seviyede olduğu görülmüş olup, deneysel çalışmalarda daha önceki aşamalarda elde edilen 57 özelliğin tamamı kullanılmıştır.

4.1.5 Farklı Dağılımlara Sahip Veri Setlerinin Model Performansına Etkisi

Bu tez çalışmasında kullanılan veri setinde toplamda 26.318 adet örnek bulunmakta olup, yaklaşık olarak %50-%50 oltalama ve meşru oranına sahip olan veri seti dengeli bir yapıya sahiptir. Gerçek dünyada internet kullanıcılarının karşılaştığı örnekler dengeli olmamakta ve oltalama örnekleri ile karşılaşma oranı çok daha düşüktür. Ayrıca farklı dağılımlara sahip veri setleri ile eğitilen modellerin sınıflandırma başarısı da farklılık göstermektedir. Bu kapsamda tez çalışmasında beş farklı dağılıma sahip veri seti için deneysel çalışmalar yapılmıştır. Sırasıyla %20, %40, %50, %60 ve %80 oranlarında oltalama örneklerinden oluşan veri setleri için beş farklı senaryo oluşturulmuştur. Bu beş farklı senaryo için en başarılı Makine Öğrenme algoritması olan RO'nun sınıflandırma başarısı hesaplanmıştır. Deneysel çalışmalarda elde edilen sonuçlar Tablo 4.16'da detaylı olarak verilmiştir.

Tablo 4.16 Farklı dağılımlardaki veri setleri için model performansı (%)

Senaryo	DOĞ	GPO	GNO	YNO	YPO	KES	HAS	F1S
%20 Oltalama	96,45	96,95	94,28	3,04	5,71	98,66	96,95	97,80
%40 Oltalama	94,91	94,98	94,80	5,01	5,19	96,16	94,98	95,79
%50 Oltalama	94,28	94,05	94,51	5,94	5,48	94,55	94,05	94,30
%60 Oltalama	95,01	93,80	95,81	6,19	4,18	93,71	93,80	93,76
%80 Oltalama	96,35	93,58	97,00	6,41	2,99	88,05	93,58	90,73

4.2 Derin Öğrenme ile Sınıflandırma Deneyleri

Çalışmanın bu aşamasında ise Makine Öğrenmesi algoritmaları ile elde edilen sınıflandırma modelleri Derin Öğrenme teknikleri ile gerçekleştirilmiştir. Özellikle görüntü işleme, doğal dil işleme gibi alanlarda Makine Öğrenmesi algoritmalarına göre oldukça başarılı sonuçlar üreten Derin Öğrenmenin içerik tabanlı oltalama sitelerinin sınıflandırılmasındaki başarısını ölçebilmek için farklı tekniklerden yararlanılmıştır. Derin Öğrenmenin katmanlı yapısında katmanlar bir önceki katmanın çıktısını giriş verisi olarak kullanmakta ve bu yapı birçok problemin çözümünde avantaj sağlamaktadır.

İlk aşamada "*Sequential*" olarak adlandırılan "Sıralı" formatta 5 katmanlı bir model kurulmuştur. Ara katmanlarda aktivasyon fonksiyonu olarak "*RELU*" kullanılırken son katmanda "sigmoid" fonksiyonu tercih edilmiştir. Şekil 4.5'te detayları gösterilen model için Doğruluk, Kesinlik, Duyarlılık ve F-1 Skor değerleri elde edilmiştir. Elde edilen değerler Tablo 4.17'da verilmiştir. Elde edilen doğruluk oranı Makine Öğrenmesi algoritmalarından RO ve KA ile elde edilen değerlerden daha düşüktür. Dolayısıyla bir sonraki bölümde farklı Derin Öğrenme teknikleri kullanılarak bu doğruluk oranının değişimi gösterilecektir.

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 32)                1472
_________________________________________________________________
dense_2 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_3 (Dense)              (None, 16)                528
_________________________________________________________________
dense_4 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 9
=================================================================
Total params: 3,201
Trainable params: 3,201
Non-trainable params: 0
```

Şekil 4.5 Sıralı model özeti

Tablo 4.17 Sıralı model değerleri (%)

Ölçüt	Değer
Doğruluk	91,85
Kesinlik	90,53
Duyarlılık	93,13
F-1 Skoru	91,18

4.2.1 Farklı Derin Öğrenme Teknikleri ile Sınıflandırma Deneyleri

Derin Öğrenme tekniklerinden ilk olarak Basit YSA'dan faydalanılmıştır. Şekil 4.6'te detayları verilen model ile elde edilen başarı Basit YSA'nın kullanılmadığı modele göre daha düşük olmuştur. Parametre sayısında büyük bir artış gözlemlenirken başarı skorlarında düşüş olduğu tespit edilmiştir, Model değerleri Tablo 4.18'de detaylandırılmıştır.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
simple_rnn_1 (SimpleRNN)     (None, 128)               20608
_________________________________________________________________
dense_1 (Dense)              (None, 32)                4128
_________________________________________________________________
dense_2 (Dense)              (None, 16)                528
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 9
=================================================================
Total params: 425,409
Trainable params: 425,409
Non-trainable params: 0
```

Şekil 4.6 Basit YSA model özeti

İkinci aşamada GYB kullanılmış ve model başarısında Basit YSA'ya göre artış gözlemlenmiştir. Ancak RO ve KA ile kıyaslandığında elde edilen başarı yeterli

değildir. Şekil 4.7'da GYB kullanılan model özeti detaylandırılmış, model ile elde edilen değerler de Tablo 4.18'de belirtilmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
gru_1 (GRU)                  (None, 128)               61824
_________________________________________________________________
dense_1 (Dense)              (None, 32)                4128
_________________________________________________________________
dense_2 (Dense)              (None, 16)                528
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 9
=================================================================
Total params: 466,625
Trainable params: 466,625
Non-trainable params: 0
```

Şekil 4.7 GYB model özeti

Kullanılan Derin Öğrenme tekniklerinden bir diğeri de detayları Bölüm 4.4.3'te anlatılan LSTM'dir. Özellikle metin bazlı çalışmalarda oldukça başarılı sonuçlar veren bu teknik ile model oluşturulmuş ve sınıflandırma başarısı ölçülmüştür, Model özeti Şekil 4.8'de belirtilmiştir, LSTM ile elde edilen değerler ise Tablo 4.18'de gösterilmiştir,

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
lstm_1 (LSTM)                (None, 128)               82432
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 482,561
Trainable params: 482,561
Non-trainable params: 0
```

Şekil 4.8 LSTM model özeti

Bir sonraki aşamada ise İki yönlü LSTM tekniğinden faydalanılarak modelde değişiklikler yapılmış ve model başarısı tekrar ölçülmüştür, Model özeti Şekil 4.9'de belirtilmiş olup, İki yönlü LSTM ile elde edilen değerler Tablo 4.18'de gösterilmiştir,

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               164864
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 257
=================================================================
Total params: 565,121
Trainable params: 565,121
Non-trainable params: 0
```

Şekil 4.9 İki yönlü LSTM model özeti

Son aşamada GAN tekniğinden faydalanılmış ve model başarısı ölçülmüştür, Model özeti Şekil 4.10'da detaylandırılmıştır, Modelin başarı değerleri ise Tablo 4.18'de gösterilmiştir,

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 784)               545552
_________________________________________________________________
sequential_2 (Sequential)    (None, 1)                 533505
=================================================================
Total params: 1,079,057
Trainable params: 545,552
Non-trainable params: 533,505
```

Şekil 4.10 GAN model özeti

İlk aşamada elde edilen Doğruluk değerleri %90 civarında olsa da RO ile elde edilen başarı değerlerinin altında kalmıştır. Bu kapsamda Basit YSA, GYB, LSTM, İki yönlü LSTM ve GAN modelleri için katman sayısı ve buna bağlı olarak "*epoch*" parametre değeri artırılmıştır. Daha fazla parametreye sahip yeni modellerin başarı ölçümleri ve elde edilen değerlerin detayları Bölüm 5.2.2 ve Bölüm 5.2.3'te anlatılmıştır.

Tablo 4.18 Farklı Derin Öğrenme modelleri başarı değerleri (%)

Model	Doğruluk	Kesinlik	Duyarlılık	F-1 Skoru
Basit YSA	75,64	76,54	72,77	74,61
GYB	89,35	88,60	90,07	89,29
LSTM	89,11	86,42	92,46	89,34
İki yönlü LSTM	89,59	86,84	93,44	90,02
GAN	92,12	89,77	95,16	92,39

4.2.2 Derin Öğrenme Modellerindeki Katman Sayısının Artırılması: Birinci Kademe

Birinci kademede ilk olarak Basit YSA için katman sayısı ve "*epoch*" sayısı artırılıp, modelin başarısındaki değişim gözlemlenmiştir. Yeni elde edilen modele ait bilgiler Şekil 4.11'da verilmiştir. Modelin başarısına ait detaylar ise Tablo 4.19'de gösterilmiştir. Katman sayısındaki artışın Basit YSA için olumsuz etkide bulunduğu gözlemlenmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
simple_rnn_1 (SimpleRNN)     (None, 128)               20608
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_3 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_4 (Dense)              (None, 16)                528
_________________________________________________________________
dense_5 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 9
=================================================================
Total params: 432,673
Trainable params: 432,673
Non-trainable params: 0
```

Şekil 4.11 BasitYSA model özeti

İkinci aşamada ise GYB ile oluşturulan modelin katman sayısı ve "*epoch*" sayısı artırılmış ve modelin özeti Şekil 4.12'de detaylı olarak verilmiştir. Model başarısına ait detaylar ise Tablo 4.19'de gösterilmiştir. Katman sayısı artırıldıktan sonra GYB kullanılan modeldeki tüm değerlerde olumlu bir değişim gözlemlenmiştir. Örneğin, Doğruluk değeri %89,35'ten %93,30'a yükselmiştir.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
gru_1 (GRU)                  (None, 128)               61824
_________________________________________________________________
dense_1 (Dense)              (None, 32)                4128
_________________________________________________________________
dense_2 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_3 (Dense)              (None, 16)                528
_________________________________________________________________
dense_4 (Dense)              (None, 16)                272
_________________________________________________________________
dense_5 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 9
=================================================================
Total params: 467,953
Trainable params: 467,953
Non-trainable params: 0
```

Şekil 4.12 GYB model özeti

LSTM kullanılan modelin de katman sayısı ve "*epoch*" değeri artırılmıştır. Şekil 4.13'de detayları belirtilen model için elde edilen yeni başarı değerleri Tablo 4.19'de gösterilmiştir. Aynı şekilde LSTM kullanılan modelde de dört değer için artış gözlemlenmiştir. Doğruluk değeri %89,11'den %91,54'e yükselmiştir.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
lstm_1 (LSTM)                (None, 128)               82432
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 33
=================================================================
Total params: 492,801
Trainable params: 492,801
Non-trainable params: 0
```

Şekil 4.13 LSTM model özeti

Katman sayısı artırılan bir diğer model de İki yönlü LSTM kullanılan modeldir. Artırılmış katman sayısına bağlı olarak parametre sayısında da artış olmuş ve yeni modele ait detaylar Şekil 4.14'te gösterilmiştir. Elde edilen başarı değerleri ise Tablo 4.19'de verilmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               164864
_________________________________________________________________
dense_1 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_2 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 33
=================================================================
Total params: 583,425
Trainable params: 583,425
Non-trainable params: 0
```

Şekil 4.14 İki yönlü LSTM model özeti

Son olarak GAN modelinde de katman sayısı artırılmış ve yeni modelin başarısındaki değişim gözlemlenmiştir. Katman sayısı artırılmış model detayları Şekil 4.15'te gösterilmiş olup yeni modele ait başarı değerleri Tablo 4.19'de verilmiştir. GAN tekniğinden yararlanılarak oluşturulan modeldeki parametre artışı da başarı değerlerine olumlu etki yapmıştır. %92,12 olan Doğruluk değeri yeni modelde %93,61'e yükselmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 784)               808208
_________________________________________________________________
sequential_2 (Sequential)    (None, 1)                 599297
=================================================================
Total params: 1,407,505
Trainable params: 808,208
Non-trainable params: 599,297
```

Şekil 4.15 GAN model özeti

Farklı teknikler kullanılarak oluşturulan Derin Öğrenme modellerindeki katman sayısının artırılması modellerin başarılarına olumlu etki etmiştir. Sadece Basit YSA'da bu artış olumsuz yönde olmuştur. Modellerde kullanılan aktivasyon fonksiyonları, öğrenme katsayısı ya da "*Optimizer*" parametrelerinde herhangi bir değişiklik yapılmamıştır. Yalnızca katman sayısı ve ona bağlı olarak "*epoch*" sayısı artırılmıştır. Bu durumda katman sayısındaki artışın genel olarak model başarılarına olumlu etki ettiği görülmekle birlikte, Makine Öğrenmesi algoritmalarının başarı değerlerine erişilebilmesi için katman yapısının daha da

derinleştirilmesine ihtiyaç duyduğu görülmüştür. En başarılı Derin Öğrenme modelleri GYB ve GAN tekniklerinden yararlanan modeller olmuştur. Sırasıyla Doğruluk değerleri %93,30 ve %93,61 olarak elde edilmiştir. Çalışmanın bir sonraki aşamasında modellerdeki katman sayısının artırılmasına devam edilecektir. Tüm modeller için ikinci kademe katman sayısı artırımının etkileri ölçülecektir.

Tablo 4.19 Birinci kademe model başarı değerleri (%)

Model	Doğruluk	Kesinlik	Duyarlılık	F-1 Skoru
Basit YSA	49,21	49,21	100	65,96
GYB	93,30	91,62	95,73	93,63
LSTM	91,54	88,64	95,80	92,08
İki yönlü LSTM	91,93	88,65	96,25	92,30
GAN	93,61	93,23	94,17	93,70

4.2.3 Derin Öğrenme Modellerindeki Katman Sayısının Artırılması: İkinci Kademe

Derin Öğrenme modellerindeki katman sayısının artırılmasının ikinci kademesinde sırasıyla GYB, LSTM, İki yönlü LSTM ve GAN modellerinin katman sayıları artırılmış ve tüm modeller için yapılan değişikliğin etkisi ölçülmüştür. Güncellenen GYB modeline ait detaylar Şekil 4.16'da gösterilmiştir. 8 katmanlı model yapısına iki katman daha eklenerek 10 katmanlı bir yapıya dönüştürülmüştür. Modelin başarısına ait detaylar ise Tablo 4.20'de verilmiştir. İkinci kademe katman artırımı da modelin başarısına olumlu etkide bulunmuş ve doğruluk değerinin %93,30'ten %94,16'ya yükselmesini sağlamıştır. İkinci kademe katman sayısı artırılan bir diğer model LSTM ile oluşturulmuş modeldir. Güncellenen modele ait detaylar Şekil 4.17'de gösterilmiştir. 5 katmanlı LSTM model yapısına 4 katman daha eklenerek 9 katmanlı bir yapıya dönüştürülmüştür. Modele 5856 parametre eklenerek toplamda 498.657 parametreli bir model oluşturulmuştur. Modelin başarısına ait detaylar ise Tablo 4.20'de verilmiştir. LSTM modelinde de katman artırımının olumlu etkisi gözlemlenmiştir. Doğruluk değeri %91,54'ten %92,87'ye yükselmiştir.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 32)	400000
gru_1 (GRU)	(None, 128)	61824
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 64)	4160
dense_3 (Dense)	(None, 32)	2080
dense_4 (Dense)	(None, 32)	1056
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 16)	272
dense_7 (Dense)	(None, 8)	136
dense_8 (Dense)	(None, 1)	9

Total params: 478,321
Trainable params: 478,321
Non-trainable params: 0

Şekil 4.16 GYB model özeti

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, None, 32)	400000
lstm_1 (LSTM)	(None, 128)	82432
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 64)	4160
dense_3 (Dense)	(None, 32)	2080
dense_4 (Dense)	(None, 32)	1056
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 8)	136
dense_7 (Dense)	(None, 1)	9

Total params: 498,657
Trainable params: 498,657
Non-trainable params: 0

Şekil 4.17 LSTM model özeti

İkinci kademe katman artırımı yapılan bir diğer model ise İki yönlü LSTM olmuştur. Güncellenen model özeti Şekil 4.18'de gösterilmiş olup modele ait başarı değerleri Tablo 4.20'de verilmiştir. GYB ve LSTM modellerindeki gibi İki yönlü LSTM modelinde de katman sayısındaki artış olumlu etki yapmıştır ve doğruluk oranı %91,93'ten %93,46'ya yükselmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, None, 32)          400000
_________________________________________________________________
bidirectional_1 (Bidirection (None, 256)               164864
_________________________________________________________________
dense_1 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_2 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_3 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_4 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_5 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_6 (Dense)              (None, 32)                1056
_________________________________________________________________
dense_7 (Dense)              (None, 16)                528
_________________________________________________________________
dense_8 (Dense)              (None, 16)                272
_________________________________________________________________
dense_9 (Dense)              (None, 8)                 136
_________________________________________________________________
dense_10 (Dense)             (None, 1)                 9
=================================================================
Total params: 594,769
Trainable params: 594,769
Non-trainable params: 0
```

Şekil 4.18 İki yönlü LSTM model özeti

İkinci kademe katman sayısı artırılan son model GAN modelidir. Güncellenen GAN model özeti Şekil 4.19'da belirtilmiş olup modelin başarısına ait değerler Tablo 4.20'de sunulmuştur. Diğer modellerde olduğu gibi GAN modelinde de katman sayısındaki artış parametre artışına neden olmuş ve sonuç olarak da modelin başarısının yükselmesini sağlamıştır. %93,61 olan doğruluk değeri %95,53'e yükselmiş ve GAN modeli ikinci kademe sonucundaki en başarılı model olmuştur.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
sequential_1 (Sequential)    (None, 784)               874000
_________________________________________________________________
sequential_2 (Sequential)    (None, 1)                 861953
=================================================================
Total params: 1,735,953
Trainable params: 874,000
Non-trainable params: 861,953
```

Şekil 4.19 GAN model özeti

Tablo 4.20 İkinci kademe model başarı değerleri (%)

Model	Doğruluk	Kesinlik	Duyarlılık	F-1 Skoru
GYB	94,16	92,01	96,91	94,40
LSTM	92,87	91,48	94,01	92,73
İki yönlü LSTM	93,46	91,12	96,11	93,55
GAN	95,53	95,22	95,68	95,45

4.2.4 Derin Öğrenme Parametrelerindeki Değişimin Model Başarılarına Etkisi

Derin Öğrenme modellerinde katman sayısında yapmış olduğumuz artışın modellerin başarısına olumlu etki yaptığını göstermiştik. Ancak Derin Öğrenmede en başarılı model olan GAN'ın doğruluk oranı en başarılı Makine Öğrenmesi algoritması olan RO'nun ulaştığı doğruluk değerinin altında kalmıştır. RO %97,73 doğruluk oranına sahip iken ikinci kademe katman sayısını artırdıktan sonra GAN modelinin doğruluk oranı %95,53'e ulaşmıştı. Bu bölümde ise modellere uygulanan farklı değişikliklerin modellerin başarılarına olan etkileri gözlemlenecektir, Aynı zamanda bu değişiklikler ile model başarısı artırılmaya ve RO'nun verdiği doğruluk değerinden daha iyi bir sonuca ulaşmak hedeflenmiştir. Bu doğrultuda parametrelerdeki değişimin etkisini görebilmek ve Makine Öğrenmesi algoritmalarından daha iyi bir sonuç elde edebilmek için şu ana kadar yapılan deneylerde en başarılı sonuçları üreten Derin Öğrenme modeli olan GAN seçilmiştir.

İlk olarak "*epoch*" ve "*batch_size*" değerleri sabit tutularak öğrenme oranı için beş farklı değer seçilmiştir. Kullanılan öğrenme oranı değerleri ve bu değerlere bağlı olarak elde edilen başarı skorları Tablo 4.21'de detaylı şekilde verilmiştir. "*Optimizer*" olarak "*Adam*" seçilmiştir. Tablo 4.21'nin en alt satırında da her bir öğrenme oranı için model çalışma süresi değerleri eklenmiştir. Tüm öğrenme oranı

değerlerinde "*epoch"* sayısı 4 olarak sabit tutulmuş ve aktivasyon fonksiyonu olarak "*RELU*" kullanılmıştır. Tablo 4.21'deki değerler incelendiğinde bu çalışma için en iyi öğrenme oranı değerinin 0,0001 olduğu tespit edilmiştir. 0,0002 değerinden büyük değerlerde ise başarı oranlarının ciddi şekilde düştüğü gözlemlenmiştir.

Tablo 4.21 Öğrenme oranına göre başarı değerleri ve çalışma süreleri (%)

Öğrenme Oranı	**0,0001**	**0,0002**	**0,0005**	**0,001**
Doğruluk	88,05	87,47	50,15	50,97
Kesinlik	87,49	90,85	50,15	50,97
Hassasiyet	88,12	83,88	99,21	100,00
F-1 Skoru	87,80	87,23	66,63	67,53
Çalışma Süresi (sn)	145,504	156,220	167,550	166,451

Derin Öğrenme yapısındaki diğer bir önemli parametre katmanlarda kullanılan aktivasyon fonksiyonlarıdır. "*RELU*", "*Tanh*", "*Sigmoid*", "*Softmax*" gibi çeşitli aktivasyon fonksiyonları bulunmaktadır. Oltalama internet sitelerinin tespiti bir tür ikili sınıflandırma problemidir. Bu kapsamda en doğru aktivasyon fonksiyonunu seçebilmek için farklı aktivasyon fonksiyonları kullanımının model başarılarına etkisi incelenmiştir. Bu deney için GAN modeli tercih edilmiştir, Her bir aktivasyon fonksiyonu için elde edilen değerler Tablo 4.22'de detaylandırılmıştır. Tüm başarı değerleri göz önünde bulundurulduğunda "*RELU*" ve "*Tanh*" aktivasyon fonksiyonlarının en başarılı fonksiyonlar olduğu gözlemlenmiştir.

Tablo 4.22 Aktivasyon fonksiyonlarına göre başarı değerleri (%)

Aktivasyon Fonksiyonu	**RELU**	**Tanh**	**Sigmoid**	**Softsign**	**Softplus**
Doğruluk	88,05	89,85	82,88	86,92	85,82
Kesinlik	87,49	86,98	83,85	83,54	87,27
Hassasiyet	88,12	93,62	81,83	92,53	83,85
F-1 Skoru	87,80	90,18	82,82	87,81	85,53

Derin Öğrenme algoritmalarında performans ve başarıyı artırabilmek için kullanılabilen bir diğer teknik de "*Dropout*" kullanımıdır. "*Dropout*" kullanımı ile katmanlar arasındaki bağlantı sayısı belirlenen oranda seyreltilir ve özellikle aşırı

öğrenme (*overfitting*) önlenmeye çalışılır. Bu kapsamda en başarılı GAN modeli içerisinde "*Dropout*" katmanları kullanılarak model başarısına olan etkisi gözlemlenmiştir. GAN modelindeki katman sayısı artırılmış ve "*Generator*" ve "*Discriminator*" kısımlarına ait katman yapısı ayrı ayrı belirtilmiştir. "*Dropout*" kullanılmayan "*Generator*" katman yapısı Şekil 4.20'de "Discriminator" katman yapısı Şekil 4.21'de gösterilmiştir. "*Epoch*" değeri 200, "*batch_size*" değeri ise 20 olarak ayarlanmıştır. "*Dropout*" katmanı kullanılmayan bu modele ait başarı değerleri Tablo **4.23**'te verilmiştir. GAN modelindeki katman sayısının artırılması, ara katmanlarda aktivasyon fonksiyonu olarak en başarılı sonuçları veren "*Tanh*" fonksiyonun kullanılması ve öğrenme oranı olarak da 0,0001'in seçilmesi ile elde edilen doğruluk oranı, RO ile elde edilen doğruluk oranını geçmiştir.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	11776
dense_2 (Dense)	(None, 256)	65792
dense_3 (Dense)	(None, 512)	131584
dense_4 (Dense)	(None, 512)	262656
dense_5 (Dense)	(None, 784)	402192
dense_6 (Dense)	(None, 784)	615440
dense_7 (Dense)	(None, 1012)	794420

Total params: 2,283,860
Trainable params: 2,283,860
Non-trainable params: 0

Şekil 4.20 GAN – "Generator" model özeti

Bir sonraki adımda aynı model için parametrelerde bir değişiklik yapmaksızın "Dropout" katmanları eklenmiştir. 0,2 ve 0,3 değerlerinde eksiltme yapılarak güncellenen modele ait özet bilgiler Şekil 4.22 ve Şekil 4.23'te belirtilmiştir. "Dropout" kullanılan modelin başarı değerleri ise Tablo 4.23'te detaylandırılmıştır. "Dropout" katmanlarının kullanımı GAN modelinin başarısını artırmıştır. Doğruluk oranı %97,54'ten %97,92'ye yükselmiştir. Ayrıca, diğer tüm değerlerde artış gözlemlenmiştir.

Layer (type)	Output Shape	Param #
dense_8 (Dense)	(None, 784)	794192
dense_9 (Dense)	(None, 784)	615440
dense_10 (Dense)	(None, 512)	401920
dense_11 (Dense)	(None, 512)	262656
dense_12 (Dense)	(None, 256)	131328
dense_13 (Dense)	(None, 256)	65792
dense_14 (Dense)	(None, 1)	257

Total params: 2,271,585
Trainable params: 2,271,585
Non-trainable params: 0

Şekil 4.21 GAN – "*Discriminator"* model özeti

Derin Öğrenme modeli son haliyle Makine Öğrenmesi algoritması RO'dan daha iyi bir skor elde etmiştir. Sonuç olarak başlangıçta daha düşük olan Derin Öğrenme modelinin başarısı katman sayısının artırılması ve bazı parametrelerin değiştirilmesi ile artırılmıştır.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	11776
dense_2 (Dense)	(None, 256)	65792
dropout_1 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 512)	131584
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dense_5 (Dense)	(None, 784)	402192
dropout_3 (Dropout)	(None, 784)	0
dense_6 (Dense)	(None, 784)	615440
dense_7 (Dense)	(None, 1012)	794420

Total params: 2,283,860
Trainable params: 2,283,860
Non-trainable params: 0

Şekil 4.22 GAN – "*Generator*" model özeti

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_8 (Dense)              (None, 784)               794192
_________________________________________________________________
dense_9 (Dense)              (None, 784)               615440
_________________________________________________________________
dense_10 (Dense)             (None, 512)               401920
_________________________________________________________________
dense_11 (Dense)             (None, 512)               262656
_________________________________________________________________
dropout_6 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_12 (Dense)             (None, 256)               131328
_________________________________________________________________
dropout_7 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_13 (Dense)             (None, 256)               65792
_________________________________________________________________
dense_14 (Dense)             (None, 1)                 257
=================================================================
Total params: 2,271,585
Trainable params: 2,271,585
Non-trainable params: 0
```

Şekil 4.23 GAN – "*Discriminator*" model özeti

Tablo 4.23 Dropout kullanımına göre gan model değerleri (%)

Ölçüt	Dropout Kullanılan Model	Dropout Kullanılmayan Model
Doğruluk	97,54	97,92
Kesinlik	95,83	96,78
Duyarlılık	98,12	98,64
F-1 Skoru	96,87	97,43

5
SONUÇ VE ÖNERİLER

Dijital devrim ile eğitimden endüstriye pek çok alanda önemli kazanımlar elde edilmiştir. Her geçen gün daha fazla verinin dolaşıma girmesiyle birlikte bazı güvenlik problemleri açığa çıkmıştır. Siber tehditler de bu güvenlik problemleri içerisinde internet kullanıcılarından şirketlere, sivil toplum kuruluşlarından devletlere kadar birçok kişi ve kurumu etkileyen problemlerin başında gelmektedir. Siber tehditler içerisinde ise en yaygın ve giderek hem nitelik hem de nicelik yönüyle etkisini artıran problemlerden biri olan oltalama saldırılarıdır. E-posta, SMS, sosyal medya uygulamaları gibi çeşitli iletişim kanalları aracılığıyla insanları dolandırmak ve kullanıcıların bilgilerini çalmak amacıyla gerçekleştirilen bu saldırılarda nihai aşama kurbanların sahte internet sitelerine yönlendirilmesidir. İnternet kullanıcıları için 90'lı yıllardan bu yana önemli finansal sorunlara yol açan bu saldırılara karşı iki temel çözüm önerisi geliştirilmiştir. Bunlardan ilki kullanıcı eğitimidir ancak bu çözüm sınırlı kapasitede etkili olmakta, birçok saldırıya karşı yetersiz kalmaktadır. Diğer temel çözüm önerisi ise gelişmiş yazılımlar ile bu saldırıların tespiti ve önlenmesidir. Bu tez kapsamında içerik tabanlı bir yaklaşım ve yapay zekâ temelli algoritmalar kullanılarak oltalama internet sitelerinin tespiti sağlanmıştır.

İlk aşamada geniş bir literatür taraması ile içerik tabanlı sınıflandırma çalışmalarına odaklanılmış, bu sayede kullanılmış olan özellikler belirlenmiştir. Sonrasında oltalama internet sitelerinin içerikleri manuel olarak incelenmiş ve analizler sonucunda yeni özellikler çıkarılarak toplamda 57 içerik tabanlı özellik oluşturulmuştur. Denemeler için 15 bin adet oltalama ve 15 bin adet meşru internet sitesi içeriğinden oluşan bir veri seti seçilmiştir. Python programlama dili kullanılarak her bir özellik için sayısal değer üreten fonksiyonlar yazılmış ve veri setinde yer alan tüm örnekler bu fonksiyonlar aracılığıyla yapay zekâ modelleri tarafından kullanılabilir hale getirilmiştir. Bu noktada çeşitli veri bilimi tekniklerinden faydalanılarak özellik seçimi çalışmaları yapılmıştır. Çalışmanın devamında ise sınıflandırma problemlerinde kullanılan Makine Öğrenmesi algoritmaları ile modeller oluşturulmuş ve bu modellerin sınıflandırma başarıları

ölçülmüştür. En başarılı algoritmalar Rastgele Orman ve Karar Ağaçları olurken, Rastgele Orman algoritmasının doğruluk oranı %97,73 olarak elde edilmiştir. Çalışmanın son aşamasında ise Derin Öğrenme tekniklerinden faydalanılarak çeşitli modeller tasarlanmış ve Derin Öğrenme modelleri üzerinde yapılan çeşitli değişiklikler ile Makine Öğrenmesi algoritmaları ile elde edilen başarı sonuçlarından daha iyi sonuçlar elde edilmeye çalışılmıştır. YSA, GYB, LSTM ve GAN gibi Derin Öğrenme teknikleri ile geliştirilen modellerin katman sayısı derinleştirilerek modellerin başarısında iyileştirme yapılmıştır. Sonuç olarak GAN modeli ile Rastgele Orman algoritmasının başarısı geçilmiştir. Hem Mekine Öğrenmesi hem de Derin Öğrenme algoritmaları ile oldukça başarılı sınıflandırma modelleri inşa edilmiştir. Sadece içerik tabanlı özellikler kullanılarak elde edilen bu başarı ile oltalama internet sitelerinin içerikleri ile bu saldırıların büyük ölçüde önlenebileceği gösterilmiştir.

Gelecek çalışmalarda içerik-tabanlı özelliklerin artırılması, URL ve "*Domain*" bazlı yaklaşımlarla entegre, hibrit bir model tasarlanarak elde edilen doğruluk oranları ve model performansları iyileştirilebilir. Ayrıca, hiper-parametrelerin optimizasyonu için genetik algoritma gibi evrimsel algoritmalar kullanılabilir. Sürekli kendini güncelleyen internet dünyasında oltalama saldırıları ve bu doğrultuda tasarlanan oltalama internet sitesi içerikleri de güncellenmektedir. Bu kapsamda daha güncel örneklerle oltalama ve meşru internet site içeriklerinden oluşan veri setleri genişletilebilir ve güncellenebilir. Bu sayede özellikle Yanlış Pozitif Oranı minimize edilerek sistemin tespit edemediği oltalama saldırı sayısı sıfırlanmaya çalışılabilir.

KAYNAKÇA

[1] S. Furnell, M. Papadaki, and K. Millet, "Fifteen years of Phishing: can technology save us?" Computer Fraud & Security, vol. 7, pp. 11-16, 2017.

[2] Understanding Phishing techniques, Deloitte, December 2019 https://www2,deloitte,com/content/dam/Deloitte/sg/Documents/risk/sea-risk-cyber-101-part10.pdf.

[3] APWG Phishing Activity Trends Report 3rd Quarter 2020, Phishing Attacks Hosted in HTTPs, https://docs.apwg.org/reports/apwg_trends_report_q3_2020.pdf, pp. 8, 24 November, 2020.

[4] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, "Email Classification Research Trends: Review and Open Issues," IEEE Access, vol. 5, pp. 9044-9064, 2017.

[5] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A Survey of Phishing Email Filtering Techniques" IEEE Communications Surveys & Tutorials, vol. 15, pp. 2070-2090, 2013.

[6] P. Patel, K. Kannorpatti, B. Shanmugam, S. Azam, and K. C. Yeo, "A theoretical review of social media usage by cyber-criminals" International Conference on Computer Communication and Informatics (ICCCI), 5-7 Jan, 2017, Coimbatore, India.

[7] A. Aggarwal, A. Rajadesingan, and P. Kumaraguru, "PhishAri: Automatic realtime phishing detection on twitter" 2012 eCrime Researchers Summit, 23-24 Oct, 2012, Las Croabas, PR, USA.

[8] N. Mazher, I. Ashraf, and A. Altaf, "Which web browser work best for detecting Phishing" 2013 5th International Conference on Information and Communication Technologies, 14-15 Dec, 2013, Karachi, Pakistan.

[9] G. Varshney, S. Bagade, and S. Sinha, "Malicious browser extensions: A growing threat: A case study on Google Chrome: Ongoing work in progress" 2018 International Conference on Information Networking (ICOIN), 10-12 Jan, 2018, Chiang Mai, Thailand.

[10] A. A. Athulya, and K. Praveen, "Towards the Detection of Phishing Attacks" 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), 15-17 July 2020, Tirunelveli, India.

[11] Z. Dou, I. Khalil, A. Khreishah, A. Al-Fuqaha, and M. Guizani, "Systematization of Knowledge (SoK): A Systematic Review of Software-Based Web Phishing Detection" IEEE Communications Surveys & Tutorials, vol. 19, pp. 2797-2819, 2017.

[12] M. Khonji, Y. Iraqi, and A. Jones, "Phishing Detection: A Literature Survey" IEEE Communications Surveys & Tutorials, vol. 15, pp. 2091-2021, 2013.

[13] I. O. Garces, M. F. Cazares, and R. O. Andrade, “Detection of Phishing Attacks with Machine Learning Techniques in Cognitive Security Architecture” 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 5-7 Dec, 2019, Las Vegas, NV, USA.

[14] M. H. Alkawa, S. J. Steven, and A. I. Hajamydeen, “Detecting Phishing Website Using Machine Learning” 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 28-29 Feb, 2020, Langkawi, Malaysia.

[15] V. Patil, P. Thakkar, C. Shah, T. Bhat, and S. P. Godse, “Detection and Prevention of Phishing Websites Using Machine Learning Approach” 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 16-18 Aug, 2018, Pune, India.

[16] M. N. Alam, D. Sarma, F. F. Lima, I. Saha, R. E. Ulfath, and S. Hossain, “Phishing Attacks Detection using Machine Learning Approach” 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 20-22 Aug, 2020, Tirunelveli, India.

[17] J. Rashid, T. Mahmood, M. W. Nisar, and T. Nazir, “Phishing Detection Using Machine Learning Technique” 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 3-5 Nov, 2020, Riyadh, Saudi Arabia.

[18] A. Alswailem, B. Alabdullah, N. Alrumayh, and A. Alsedrani, “Detecting Phishing Websites Using Machine Learning” 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), 1-3 May 2019, Riyadh, Saudi Arabia.

[19] W. Bai, “Phishing Website Detection Based on Machine Learning Algorithm”, 2020 International Conference on Computing and Data Science (CDS), 1-2 Aug, 2020, Stanford, CA, USA.

[20] M. V. Kunju, E. Daniel, H. C. Anthony, and S. Bhelwa, “Evaluation of Phishing Techniques Based on Machine Learning” 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 15-17 May 2019, Madurai, India.

[21] S. Singh, M. P. Singh, and R. Pandey, “Phishing Detection from URLs Using Deep Learning Approach” 2020 5th International Conference on Computing, Communication and Security (ICCCS), 14-16 Oct, 2020, Patna, India.

[22] K. M. Z. Hasan, M. Z. Hasan, and N. Zahan, “Automated Prediction of Phishing Websites Using Deep Convolutional Neural Network” 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 11-12 July 2019, Rajshahi, Bangladesh.

[23] J. Feng, L. Zou, O. Ye, and J. Han, “Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning” IEEE Access, vol. 8, pp. 221214-221224, 2020.

[24] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana, and S. Hossain, "Phishing Attacks Detection using Deep Learning Approach" 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 20-22 Aug, 2020, Tirunelveli, India.

[25] P. Yang, G. Zhao, and P. Zeng, "Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning" IEEE Access, vol. 7, pp. 15196-15209, 2019.

[26] Y. Su, "Research on Website Phishing Detection Based on LSTM RNN" 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 12-14 June 2020, Chongqing, China.

[27] X. Zhang, D. Shi, H. Zhang, W. Liu, and R. Li, "Efficient Detection of Phishing Attacks with Hybrid Neural Networks" 2018 IEEE 18th International Conference on Communication Technology (ICCT), 8-11 Oct, 2018, Chongqing, China.

[28] H. Chapla, R. Kotak, and M. Joiser, "A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier" 2019 International Conference on Communication and Electronics Systems (ICCES), 17-19 July 2019, Coimbatore, India.

[29] V. R. Hawanna, V. Y. Kulkarni, and R. A. Rane, "A novel algorithm to detect phishing URLs" 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 9-10 Sept, 2016, Pune, India.

[30] S. Parekh, D. Parikh, S. Kotak, and S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection" 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 20-21 April 2018, Coimbatore, India.

[31] S. Shukla, and P. Sharma, "Detection of Phishing URL using Bayesian Optimized SVM Classifier" 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 5-7 Nov, 2020, Coimbatore, India.

[32] Y. Huang, Q. Yang, J. Qin, and W. Wen, "Phishing URL Detection via ESA and Attention-Based Hierarchical RNN" 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 5-8 Aug, 2019, Rotorua, New Zealand.

[33] S. Mondal, D. Maheshwari, N. Pai, and A. Biwalkar, "A Review on Detecting Phishing URLs using Clustering Algorithms" 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), 20-21 Dec, 2019, Mumbai, India.

[34] S. Gupta, and A. Singhal, "Phishing URL detection by using artificial neural network with PSO" 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), 10-11 Aug, 2017, Noida, India.

[35] U. Ozker, and O. K. Sahingoz, "Content Based Phishing Detection with Machine Learning" 2020 International Conference on Electrical Engineering (ICEE), 25-27 Sept, 2020, Istanbul, Turkey.

[36] H. Che, Q. Liu, L. Zou, H. Yang, D. Zhou, and F. Yu, "A Content-Based Phishing Email Detection Method" 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 25-29 July 2017, Prague, Czech Republic.

[37] S. B. Rathod, and T. M. Pattewar, "Content based spam detection in email using Bayesian classifier" 2015 International Conference on Communications and Signal Processing (ICCSP), 2-4 April 2015, Melmaravathur, India.

[38] S. B. Rathod, and T. M. Pattewar, "A comparative performance evaluation of content based spam and malicious URL detection in E-mail" 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), 2-3 Nov, 2015, Bhubaneswar, India.

[39] P. Liu, and T. Moh, "Content Based Spam E-mail Filtering" 2016 International Conference on Collaboration Technologies and Systems (CTS), 31 Oct, – 4 Nov, 2016, Orlando, FL, USA.

[40] U. Xie, J. Xu, and T. Lu, "Automated classification of extremist Twitter accounts using content-based and network-based features" 2016 IEEE International Conference on Big Data (Big Data), 5-8 Dec, 2016, Washington, DC, USA.

[41] H. Zhang, G. Liu, T. W. S. Chow, and W. Liu, "Textual and Visual Content-Based Anti-Phishing: A Bayesian Approach" IEEE Transactions on Neural Networks, vol. 22, pp. 1532-1546, 2011.

[42] M. Dunlop, S. Groat, and D. Shelly, "GoldPhish: Using Images for Content-Based Phishing Analysis" 2010 Fifth International Conference on Internet Monitoring and Protection, 9-15 May, 2015, Barcelona, Spain.

[43] B. Wardman, T. Stallings, G. Warner, and A. Skjellum, "High-performance content-based phishing attack detection" 2011 eCrime Researchers Summit, 7-9 Nov, 2011, San Diego, CA, USA.

[44] S. Nakayama, I. Echizen, and H. Yoshiura, "Preventing False Positives in Content-Based Phishing Detection" 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 12-14 Sept, 2009, Kyoto, Japan.

[45] U. Ozker, "Content-based Phishing detection with machine learning" https://ugurozker,medium,com/content-based-phishing-detection-with-machine-learning-9837f809b884, 2020.

[46] S. Chanti, and T. Chithralekha, "Classification of Anti-phishing Solutions" SN Computer Science, vol. 1, article number. 11, 2020.

[47] C. L. Tan, "Phishing Dataset for Machine Learning: Feature Evaluation" Mendeley Data, V1, doi: 10,17632/h3cgnj8hft,1, 2018.

[48] R. M. A. Mohammad, L. McCluskey, and F. Thabtah, "An assessment of features related to phishing websites using an automated technique" 2012 International Conference for Internet Technology and Secured Transactions, 10-12 Dec, 2012, London, UK.

[49] M. Kaytan, and D. Hanbay, "Effective Classification of Phishing Web Pages Based on New Rules by Using Extreme Learning Machines" Anatolian Journal of Computer Sciences, vol. 2, no. 1, pp. 15-36, 2017.

[50] R. M. A. Mohammad, F. Thabtah, and L. McCluskey, "Phishing Website Features" http://eprints,hud,ac,uk/id/eprint/24330/6/MohammadPhishing14July2015,pdf, 2015.

[51] R. S. Rao, and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework" Neural Computing and Applications, vol. 31, pp. 3851-3873, 2019.

[52] Dataset for Phishing websites, http://data,phishtank,com/data/online-valid,csv, 2021.

[53] K. L. Chiew, E. H. Chang, C. L. Tan, J. Abdullah, and K. S. C. Yong, "Building Standard Offline Anti-Phishing Dataset for Benchmarking", International Journal of Engineering and Technology, 7 (4,31) pp. 7-14, 2018.

[54] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised Classification Algorithms in Machine Learning: A Survey and Review" Emerging Technology in Modelling and Graphics, pp. 99-111, 2019.

[55] S. Shalew-Shwartz, and S. Ben-David, "Understanding Machine Learning: From Theory to Algorithms – Decision Tree" Cambridge University Press, pp. 250, 2014.

[56] S. Shalew-Shwartz, and S. Ben-David, "Understanding Machine Learning: From Theory to Algorithms – Random Forest" Cambridge University Press, pp. 256, 2014.

[57] H. Dai, "Research on SVM improved algorithm for large data classification" 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), 9-12 March 2018, Shanghai China.

[58] Y. Tan, "An Improved KNN Text Classification Algorithm Based on K-Medoids and Rough Set" 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 25-26 Aug, 2018, Hangzhou, China.

[59] R. G. J. Wijnhoven, and P. H. N. De With, "Fast Training of Object Detection Using Stochastic Gradient Descent" 2010 20th International Conference on Pattern Recognition, 23-26 Aug, 2010, Istanbul, Turkey.

[60] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, “A comparative study on machine learning algorithms for indoor positioning” 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), 2-4 Sept, 2015, Madrid, Spain.

[61] I. Martin-Diaz, D. Morinigo-Sotelo, O. Duque-Perez, and R. De J. Romero-Troncoso, “Early Fault Detection in Induction Motors Using AdaBoost With Imbalanced Small Data and Optimized Sampling” IEEE Transactions on Industry Applications, vol. 53, pp. 3066-3075, 2016.

[62] Y. Luan, and S. Lin, “Research on Text Classification Based on ESA and LSTM” 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 29-31 March 2019, Dalian, China.

A
ÖZELLİKLER

Tablo A.1 Özelliklerin Kullanım Frekansı

No	Yayın No / Özellik	1	2	3	4	5	6	7
1	Link	1	1	1	1	1	1	1
2	HTTPS	1	1	1	1	1	1	1
3	Submit Element	1	1	1	1	1	1	0
4	E-mail Input	1	1	1	1	1	1	0
5	Iframe	1	1	1	1	1	1	0
6	Favicon	1	1	1	1	1	1	0
7	Redirect	1	1	1	1	1	1	0
8	IP Address	1	1	1	1	1	1	0
9	@' symbol	0	1	1	0	1	1	1
10	Sub-domain	0	1	1	1	1	1	0
11	Length of URL	0	1	1	1	1	0	1
12	Disabling Right Click	0	1	1	1	1	1	0
13	Pop-up Window	0	1	1	1	1	1	0
14	Age of Domain	0	1	1	1	1	0	1
15	URL of Anchor	0	1	0	1	1	1	1
16	Prefix or Suffix	0	1	0	1	1	1	0
17	Request URL	0	1	0	1	1	1	0
18	Abnormal URL	0	1	0	1	1	1	0
19	DNS Record	0	1	0	1	1	1	0
20	Web Traffic	0	1	0	1	1	1	0
21	Google Index	0	1	0	1	1	1	0
22	Number of Links Pointing to Web Page	0	1	0	1	1	1	0

Tablo A.1 Özelliklerin Kullanım Frekansı (devamı)

	Yayın No	**1**	**2**	**3**	**4**	**5**	**6**	**7**
No	**Özellik**							
23	Statistics Report Based Features	0	1	0	1	1	1	0
24	Page Rank	0	0	0	1	1	1	1
25	Title	1	1	1	0	0	0	0
26	Tiny URL	0	1	0	0	1	1	0
27	'//' usage - redirect	0	1	0	0	1	1	0
28	Number of dots in URL	0	1	1	0	0	0	1
29	Links in tags	0	1	0	1	0	1	0
30	Status Bar Customization	0	1	1	0	0	1	0
31	Server Form Handler	0	0	0	1	1	1	0
32	Form	1	2	0	0	0	0	0
33	Image	1	2	0	0	0	0	0
34	Password	1	2	0	0	0	0	0
35	Hidden Element	1	2	0	0	0	0	0
36	Input Element Count	1	2	0	0	0	0	0
37	Href Element	1	2	0	0	0	0	0
38	Content Spec Char	1	2	0	0	0	0	0
39	Image Count	1	2	0	0	0	0	0
40	Long URL that hides suspicious part	0	1	0	0	0	1	0
41	Using Non-Standard Port	0	1	0	0	0	1	0
42	Special Char	1	1	0	0	0	0	0
43	Suspicious action upon submitted	0	1	1	0	0	0	0
44	Websites forwarding	0	1	0	0	0	1	0
45	Count of hidden tags	1	1	0	0	0	0	0
46	Count of external links	0	1	1	0	0	0	0
47	Shortining_Service	0	0	0	1	0	1	0

Tablo A.1 Özelliklerin Kullanım Frekansı (devamı)

	Yayın No	**1**	**2**	**3**	**4**	**5**	**6**	**7**
No	**Özellik**							
48	SSLfinal_State	0	0	0	1	1	0	0
49	Domain_registeration_length	0	0	0	1	0	1	0
50	on_mouseover	0	0	0	1	1	0	0
51	POST Method	1	0	0	0	0	0	0
52	Input Element	1	0	0	0	0	0	0
53	Button	1	0	0	0	0	0	0
54	Non UTF-8 Char	1	0	0	0	0	0	0
55	Checkbox	1	0	0	0	0	0	0
56	BlackListed Link	1	0	0	0	0	0	0
57	Title has special char	1	0	0	0	0	0	0
58	Date Time	1	0	0	0	0	0	0
59	Name or Surname	1	0	0	0	0	0	0
60	Phone Number	1	0	0	0	0	0	0
61	Meta Tag	1	0	0	0	0	0	0
62	Downloadable Content	1	0	0	0	0	0	0
63	Cookie	1	0	0	0	0	0	0
64	Cache	1	0	0	0	0	0	0
65	Copyright	1	0	0	0	0	0	0
66	Readable HTML	1	0	0	0	0	0	0
67	Black List Word Usage	1	0	0	0	0	0	0
68	Option Element	1	0	0	0	0	0	0
69	Select Element	1	0	0	0	0	0	0
70	T H Element	1	0	0	0	0	0	0
71	T R Element	1	0	0	0	0	0	0
72	Table Element	1	0	0	0	0	0	0
73	LI Element	1	0	0	0	0	0	0

Tablo A.1 Özelliklerin Kullanım Frekansı (devamı)

	Yayın No	**1**	**2**	**3**	**4**	**5**	**6**	**7**
No	**Özellik**							
74	UL Element	1	0	0	0	0	0	0
75	Div Element	1	0	0	0	0	0	0
76	Span Element	1	0	0	0	0	0	0
77	Article Element	1	0	0	0	0	0	0
78	P Element	1	0	0	0	0	0	0
79	Content Word	1	0	0	0	0	0	0
80	Blacklist Word Count	1	0	0	0	0	0	0
81	HTTP Link	1	0	0	0	0	0	0
82	Meta Tag Count	1	0	0	0	0	0	0
83	HTML Element Count	1	0	0	0	0	0	0
84	Checkbox Count	1	0	0	0	0	0	0
85	Button Count	1	0	0	0	0	0	0
86	Title Length	1	0	0	0	0	0	0
87	Longest Word Length	1	0	0	0	0	0	0
88	Shortest Word Length	1	0	0	0	0	0	0
89	Content Length	1	0	0	0	0	0	0
90	Using free hosting domains	0	1	0	0	0	0	0
91	Count of digit	0	1	0	0	0	0	0
92	Registration Date of Domain	0	1	0	0	0	0	0
93	Port No in the URL	0	1	0	0	0	0	0
94	Number of triplets in the path pf URL	0	1	0	0	0	0	0
95	Number of triplets in domain name	0	1	0	0	0	0	0
96	Number of Phishing Keywords in URL	0	1	0	0	0	0	0
97	Website Owner	1	0	0	0	0	0	0
98	Abnormal DNS record	0	1	0	0	0	0	0

Tablo A.1 Özelliklerin Kullanım Frekansı (devamı)

	Yayın No	1	2	3	4	5	6	7
No	**Özellik**							
99	Abnormal anchors	0	1	0	0	0	0	0
100	Abnormal server form handler	0	1	0	0	0	0	0
101	Abnormal certificate in SSL	0	1	0	0	0	0	0
102	Number of web pages	0	1	0	0	0	0	0
103	Average number of inbound links	0	1	0	0	0	0	0
104	Average number of internal links	0	1	0	0	0	0	0
105	Average number of input boxes	0	1	0	0	0	0	0
106	Average number of password boxes	0	1	0	0	0	0	0
107	Proportion of form links	0	1	0	0	0	0	0
108	Dynamic web page proportion	0	1	0	0	0	0	0
109	Count of unsymmetric tags	0	1	0	0	0	0	0
110	Count of JavaScript segments	0	1	0	0	0	0	0
111	Count of plugins	0	1	0	0	0	0	0
112	Count of Active X xontrols	0	1	0	0	0	0	0
113	Count of longstring	0	1	0	0	0	0	0
114	Count of Unicode char	0	1	0	0	0	0	0
115	Count of Hex and Octalcoding	0	1	0	0	0	0	0
116	Count of replace() function call	0	1	0	0	0	0	0
117	Count of eval() and exec() function	0	1	0	0	0	0	0
118	Count of string functions	0	1	0	0	0	0	0
119	Count of obfuscation function	0	1	0	0	0	0	0
120	Evaluation of meta description	0	1	0	0	0	0	0
121	Evaluation of meta keywords	0	1	0	0	0	0	0
122	Evaluation of script	0	1	0	0	0	0	0
123	Grayscale histogram	0	1	0	0	0	0	0
124	Color histogram	0	1	0	0	0	0	0

Tablo A.1 Özelliklerin Kullanım Frekansı (devamı)

	Yayın No	**1**	**2**	**3**	**4**	**5**	**6**	**7**
No	**Özellik**							
125	Spatial Relationship between subgraphs of image	0	1	0	0	0	0	0
126	TLD evaluation in the domain name	0	1	0	0	0	0	0
127	TLD evaluation in the part of the URL	0	1	0	0	0	0	0
128	Country- code and TLD comparison	0	1	0	0	0	0	0
129	Path Level	0	0	1	0	0	0	0
130	Num Dash	0	0	1	0	0	0	0
131	Num Dash In Hostname	0	0	1	0	0	0	0
132	Tilde Symbol	0	0	1	0	0	0	0
133	Number of UnderScore	0	0	1	0	0	0	0
134	Num Percent	0	0	1	0	0	0	0
135	Num Query Components	0	0	1	0	0	0	0
136	Num Ampersand	0	0	1	0	0	0	0
137	Num Hash	0	0	1	0	0	0	0
138	Num Numeric Chars	0	0	1	0	0	0	0
139	Random String	0	0	1	0	0	0	0
140	Domain In Paths	0	0	1	0	0	0	0
141	Hostname Length	0	0	1	0	0	0	0
142	Path Length	0	0	1	0	0	0	0
143	Query Length	0	0	1	0	0	0	0
144	Double Slash In Path	0	0	1	0	0	0	0
145	Num Sensitive Words	0	0	1	0	0	0	0
146	Embedded Brand Name	0	0	1	0	0	0	0
147	Pct Ext Resource Urls	0	0	1	0	0	0	0
148	Insecure Forms	0	0	1	0	0	0	0
149	Relative Form Action	0	0	1	0	0	0	0
150	Ext Form Action	0	0	1	0	0	0	0

Tablo A.1 Özelliklerin Kullanım Frekansı (devamı)

No	Yayın No / Özellik	1	2	3	4	5	6	7
151	Abnormal Form Action	0	0	1	0	0	0	0
152	Pct Null Self Redirect Hyperlinks	0	0	1	0	0	0	0
153	Frequent Domain Name Mismatch	0	0	1	0	0	0	0
154	Images Only In Form	0	0	1	0	0	0	0
155	Subdomain Level RT	0	0	1	0	0	0	0
156	Pct Ext Resource Urls RT	0	0	1	0	0	0	0
157	Abnormal Ext Form Action R	0	0	1	0	0	0	0
158	Ext Meta Script Link RT	0	0	1	0	0	0	0
159	Pct Ext Null Self Redirect Hyperlinks RT	0	0	1	0	0	0	0
160	Result	0	0	0	1	0	0	0
161	Website in Search Engine Results	0	0	0	0	0	0	1
162	Frequency of domain in anchor links	0	0	0	0	0	0	1
163	Frequency of domain in CSS links, image links, and Script links	0	0	0	0	0	0	1
164	Common page detection ratio in Website	0	0	0	0	0	0	1
165	Common Page detection ratio in Footer	0	0	0	0	0	0	1
166	Null Links ratio in website	0	0	0	0	0	0	1
167	NULL Links ratio in Footer	0	0	0	0	0	0	1
168	Broken links ratio	0	0	0	0	0	0	1

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. E. Kocyigit, M. Korkmaz, O. K. Sahingoz, and B. Diri, "Real-Time Content-based Cyber Threat Detection with Machine Learning" 20th International Conference on Intelligent Systems Design and Applications (ISDA 2020), 12-15 December 2020.

2. M. Korkmaz, E. Kocyigit, O. K. Sahingoz, and B. Diri, "Deep Neural Network based Phishing Classification on a High-Risk URL Dataset" Proceedings of the 12th International Conference on Soft Computing and Pattern Recognition (SOCPAR 2020), vol. 1383, pp. 648-657, 16 April 2021.