T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ÖZNİTELİK TABANLI OLTALAMA TESPİT SİSTEMLERİNİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE DEĞERLENDİRMESİ

Selahattin ALİYAZICIOĞLU

DANIŞMAN
Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN

II. DANIŞMAN
Dr. Öğr. Üyesi Ebu Yusuf GÜVEN

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Ocak, 2024

TEZ KABUL VE ONAYI

Selahattin ALİYAZICIOĞLU tarafından, **Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN** danışmanlığında hazırlanan **"ÖZNİTELİK TABANLI OLTALAMA TESPİT SİSTEMLERİNİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE DEĞERLENDİRMESİ"** başlıklı bu çalışma, jürimiz tarafından **16/01/2024** tarihinde yapılan sınav sonucunda **oy birliği** ile başarılı bulunarak **Yüksek Lisans Tezi** olarak kabul edilmiştir.

Tez Jürisi

		İmza	**Sonuç**
DANIŞMAN	Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN İstanbul Üniversitesi-Cerrahpaşa Siber Güvenlik Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Dr. Öğr. Üyesi Özgür Can TURNA İstanbul Üniversitesi-Cerrahpaşa Bilgisayar Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret
ÜYE	Doç. Dr. Can EYÜPOĞLU Milli Savunma Üniversitesi Hava Harp Okulu Bilgisayar Mühendisliği Anabilim Dalı		☒ Kabul ☐ Ret

Ailemin her daim yanımda olduğu, desteğini esirgemediği ve çalışmalarım sırasında gösterdiği sabır için minnettarım. Bu tezi aileme ithaf ediyorum…

BÜTÇE DESTEKLERİ

ÖZNİTELİK TABANLI OLTALAMA TESPİT SİSTEMLERİNİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE DEĞERLENDİRMESİ

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.

TEŞEKKÜR

Tez çalışmam süresince değerli bilgilerini, deneyimli bakış açılarını esirgemeyen, değerli rehberlikleri ve sabırlı yönlendirmeleri ile bana her zaman ilham veren değerli tez danışman hocam, Dr. Öğretim Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN'a,

Bu tez çalışmam süresince değerli fikirlerini, değerli yönlendirmelerini esirgemeyen ve bana sonsuz destek veren değerli arkadaşım hocam, Dr. Öğretim Üyesi Ebu Yusuf GÜVEN'e,

Her an yanımda olan koşulsuz sevgi ve sabırları ile bana destek olan aileme,

Bu süreçte yanımda olduğunu hissettiğim tüm arkadaşlarıma ve dostlarıma,

En içten teşekkür ederim.

Ocak 2024 **Selahattin ALİYAZICIOĞLU**

İÇİNDEKİLER

Sayfa No

1. GİRİŞ 1
2. KAVRAMSAL ÇERÇEVE 4
2.1. Sosyal Mühendislik 5
2.2. Oltalama Saldırısı ve Türleri 5
2.2.1. Oltalama Saldırısı (Phishing Attack) Tanımı 5
2.2.2. Oltalama Saldırısı Türleri 6
2.3. Oltalama Saldırı Tespit Türleri 8
2.3.1. Liste Tabanlı Oltalama Saldırı Tespit Sistemleri 8
2.3.2. Kural Tabanlı Oltalama Saldırı Tespit Sistemleri 9
2.3.3. Görsel Benzerlik Tabanlı Oltalama Saldırı Tespit Sistemleri 10
2.4. Benzerlik Yöntemleri 11
2.4.1. Karakter Benzerliği Yöntemi 11
2.4.2. Kosinüs Benzerliği (Cosine Similarity) Yöntemi 11
2.4.3. Levenshtein Mesafe Yöntemi 12
2.5. Makine Öğrenmesi (ML - Machine Learning) 14
2.5.1. Karar Ağacı (DT - Decision Tree) Yöntemi 15
2.5.2. Rastgele Orman (RF – Random Forest) Yöntemi 15
2.5.3. Destek Vektör Makinesi (SVM - Support Vector Machine) Yöntemi 16
2.5.4. Lojistik Regresyon (LR - Logistic Regression) Yöntemi 16
2.5.5. Yapay Sinir Ağları (ANN – Artificial Neural Networks) Yöntemi 16
2.6. Literatür Taraması 17
2.6.1. Benzerlik Tabanlı Yapılan Çalışmalar 18
2.6.2. Kural Tabanlı Yapılan Çalışmalar 18
2.6.3. Makine Öğrenmesiyle Yapılan Çalışmalar 19
3. YÖNTEM 22
3.1. Veri Kaynağı 22
3.2. Veri Birleştirme 25
3.3. Veri Temizleme 26

3.3.1. Veri Setinin Oluşturulması 26
3.4. Veri Dönüştürme 27
3.4.1. Kategorik Değerden Sayısal Değere Dönüşüm İşleminin Uygulanması 31
3.4.2. Normalizasyon İşleminin Uygulanması 31
3.5. Karışıklık Matrisi (CM - Confussion Matrix) 31
3.6. Çapraz Doğrulama (CV - Cross Validation) 33
3.7. Sınıflandırma Yöntemleri Metrikleri 33
4. BULGULAR 35
5. TARTIŞMA 41
6. SONUÇ VE ÖNERİLER 44

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: Literatür Taraması için Yapılan Çalışmalara Ait Gruplama.....17

Şekil 3.1: Modelime Ait Veri Hazırlama Aşamaları.....22

Şekil 3.2: USOM Zararlı Bağlantılar Sayfası Örneği [45].....23

Şekil 3.3: USOM Örnek Bir Zararlı Bağlantı Bilgisine Ait Göster Sayfası Detayı [46].....24

Şekil 3.4: a101mobilcep.com Sahte Web Sitesinin İçeriği [47].....24

Şekil 3.5: a101.com.tr Yasal Web Sitesinin İçeriği [48].....25

Şekil 3.6: Veri Kümesi Birleştirme Adımı.....25

Şekil 3.7: Veri Setinin Oluşturulması Gösterimi.....27

Şekil 3.8: Veri Setimden 3 Farklı Veri Seti Oluşturma.....30

Şekil 3.9: Karışıklık Matrisi Modeli.....32

Şekil 3.10: Model Akış Diyagramı.....34

Şekil 4.1: Veri Setlerine Göre Yöntemlerin Yüzdelik Doğruluk Değerleri Karşılaştırma Grafiği.....38

Şekil 4.2: Veri Setlerine Göre Yöntemlerin Yüzdelik F1-skor Değerleri Karşılaştırma Grafiği.....39

TABLO LİSTESİ

Sayfa No

Tablo 3.1: Modelimde Kullanılan Makineye Ait Bilgiler 26

Tablo 3.2: Veri Kümesinden Hazırlanan Özniteliklerin Adı, Veri Tipleri ve Açıklamaları ... 28

Tablo 3.3: Veri Kümemde Bulunan "sahibindenpremium.com" Sitesine Ait Öznitelik Bilgilerinin OUT.xlsx Dosyasındaki Değeri 29

Tablo 3.4: Veri Setimden Elde Edilen 3 Veri Setlerine Ait Bilgiler 30

Tablo 4.1: VS1 Veri Setine ait Yöntemler Bazında Karışıklık Matrisi Sonuçları 35

Tablo 4.2: VS2 Veri Setine ait Yöntemler Bazında Karışıklık Matrisi Sonuçları 36

Tablo 4.3: VS3 Veri Setine ait Yöntemler Bazında Karışıklık Matrisi Sonuçları 37

Tablo 4.4: Veri Setlerine Göre Yöntemlerin Doğruluk Değerleri Sonuçları 37

Tablo 4.5: Yöntem Bazında En Yüksek Doğruluk Değerlerine ait Veri Seti ve Çalışma Süreleri 38

Tablo 4.6: Veri Seti Bazında En Yüksek Doğruluk Değerlerine ait Yöntem ve Çalışma Süreleri 39

Tablo 4.7: Veri Setlerine Göre Yöntemlerin F1-skor Değer Sonuçları 39

Tablo 4.8: Yöntem Bazında En Yüksek F1-skor Değerlerine ait Veri Seti ve Çalışma Sonuçları 40

SİMGE VE KISALTMA LİSTESİ

Simgeler	Açıklama
A	**:** A Vektörü
B	**:** B Vektörü
D[i,j]	**:** Levenshtein Mesafesi Matris Gösterimi
D[i-1,j]	**:** Levenshtein Mesafesinde Bir Karakter Ekleme Gösterimi
D[i,j-1]	**:** Levenshtein Mesafesinde Bir Karakter Çıkarma Gösterimi
D[i-1,j-1]	**:** Levenshtein Mesafesinde Karakter Dönüştürme Gösterimi
∑	**:** Toplam Sembolü
j	**:** j İndeks Bilgisi
i	**:** i İndeks Bilgisi
n	**:** Vektör Boyutu
str1[i]	**:** i İndeksindeki Karakter Bilgisi
str2[j]	**:** j İndeksindeki Karakter Bilgisi
X	**:** Orijinal Değer
X'	**:** Normalleştirilmiş Değer
$\mathbf{X_{max}}$	**:** Maksimum Değer
$\mathbf{X_{min}}$	**:** Minimum Değer

Kısaltmalar	Açıklama
AC	**:** İlişkili Sınıflandırma (Associative Classification)
AI	**:** Yapay Zeka (Artificial Intelligence)
ANN	**:** Yapay Sinir Ağları (Artificial Neural Networks)
BTK	**:** Bilgi Teknolojileri ve İletişim Kurumu
CM	**:** Karışıklık Matrisi (Confussion Matrix)
CNN	**:** Konvolüsyonel/Evrişimsel Sinir Ağları (Convolutional Neural Network)
CV	**:** Çapraz Doğrulama (Cross Validation)
DDoS	**:** Hizmet Reddi Saldırısı (Distributed Denial of Service)
DNS	**:** Alan Adı Sistemi (Domain Name System)
DT	**:** Karar Ağacı (Decision Tree)
DynaPD	**:** Dinamik Kimlik Avı Veri Kümesi (Dynamic Phishing Dataset)
FN	**:** Yanlış Negatif (False Negative)

FP **:** Yanlış Pozitif (False Positive)

HTTP **:** Bağlantılı Metin Aktarım Protokolü (Hyper Text Transfer Protokol)

HTTPS **:** Güvenli Bağlantılı Metin Aktarım Protokolü (Hypertext Transfer Protocol Secure)

IP **:** İnternet Protokolü (Internet Protocol)

LR **:** Lojistik Regresyon (Logistic Regression)

LSTM **:** Uzun Kısa Süreli Bellek (Long Short Term Memory)

MCAC **:** Çok Etiketli Sınıflandırıcı Tabanlı İlişkisel Sınıflandırma (Multi-label Classifier based Associative Classification)

ML **:** Makine Öğrenme (Machine Learning)

NN **:** Sinir Ağları (Neural Networks)

RBIDS **:** Kural Tabanlı Saldırı Tespit Sistemleri (Rule-Based Intrusion Detection Systems)

RF **:** Rastgele Orman (Random Forest)

SMTP **:** Basit Mail Aktarım Protokolü (Simple Mail Transfer Protocol)

SVM **:** Destek Vektör Makinesi (Support Vector Machine)

URL **:** Tekdüzen Kaynak Bulucu (Uniform Resource Locator)

USOM **:** Ulusal Siber Olaylara Müdahale Merkezi

TCP **:** Geçiş Kontrol Protokolü (Transmission Control Protocol)

TN **:** Gerçek Negatif (True Negative)

TP **:** Gerçek Pozitif (True Positive)

VS1 **:** Veri Seti 1

VS2 **:** Veri Seti 2

VS3 **:** Veri Seti 3

www **:** Dünya Çapında Ağ (World Wide Web)

ÖZET

YÜKSEK LİSANS TEZİ

ÖZNİTELİK TABANLI OLTALAMA TESPİT SİSTEMLERİNİN MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE DEĞERLENDİRMESİ

Selahattin ALİYAZICIOĞLU

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman : Dr. Öğr. Üyesi Gülsüm Zeynep GÜRKAŞ AYDIN
II. Danışman : Dr. Öğr. Üyesi Ebu Yusuf GÜVEN

Oltalama saldırıları, sosyal mühendislik taktiklerini kullanan ve genellikle yasal web sitelerini taklit eden sahte web siteleri oluşturarak kullanıcıları kandırmaya dayanan saldırılardır. Bu tür saldırılar, kullanıcıları yanıltmak ve hassas bilgilerini çalmak için sahte siteleri yasal siteler gibi gösterir. Özellikle finansal verilerin çalınması gibi durumlarda kullanılan oltalama saldırıları, kullanıcılar için büyük bir tehlike oluşturmaktadır. Oltalama saldırıların neden olduğu zararlar giderek artmaktadır. Ulusal Siber Olaylara Müdahale Merkezi tarafından tespit edilen sahte web siteleri, oltalama saldırılarının Türkiye'de ciddi bir siber güvenlik sorunlarından biri olduğunu göstermektedir.

Oltalama saldırılarına karşı etkili bir mücadele yürütmek için, bu tür siteleri tespit etmeden önce onların belirgin ve ortak özelliklerini tanımlamak gerekmektedir. Bu çalışma, sahte bir web sitesinin oltalama sitesi olup olmadığını tespit etmek için URL içeriğine sahip öznitelik bilgilerine, karakter benzerliği, kosinüs benzerliği, levenshtein mesafe benzerliği ile orijinal domain öznitelik bilgilerine yoğunlaşmış ve literatürde kabul görmüş yüksek doğruluk oranlarına ulaşan bir sınıflandırıcı modelin önerisini sunmaktadır.

Bu çalışmada, önerilen model kapsamında oluşan veri kümesinde 32 adet farklı öznitelik bilgisi kullanılmıştır. Önerilen modelde oltalama saldırılarını tespit etmek amacıyla hazırlanan veri kümesinden üç farklı veri seti oluşturularak, bu veri setleri üzerinde çeşitli öznitelik seçme yöntemleri ve sınıflandırma algoritmalarının performansı değerlendirilmiştir. Makine öğrenme yöntemlerinden Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları yöntemleri bu çalışmada kullanılmıştır. Veri setleri üzerinde yapılan testler sonucunda Lojistik Regresyon yönteminde %96,40 oranında doğruluk parametresi açısından başarı elde edilmiştir.

Ocak 2024 , 70 sayfa.

Anahtar kelimeler: Kimlik Avı Tespiti, Kosinüs Benzerliği, Makine Öğrenmesi, Oltalama Saldırısı, Sosyal Mühendislik

ABSTRACT

M.Sc. THESIS

EVALUATION OF ATTRIBUTE BASED PHISHING DETECTION SYSTEMS WITH MACHINE LEARNING METHODS

Selahattin ALİYAZICIOĞLU

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Computer Engineering

Computer Engineering of Programme

Supervisor : Assist. Prof. Dr. Gülsüm Zeynep GÜRKAŞ AYDIN

Co-Supervisor: Assist. Prof. Dr. Ebu Yusuf GÜVEN

Phishing attacks are attacks based on deceiving users by creating fake websites that mimic legal websites, using social engineering tactics. These attacks typically involve creating counterfeit websites that imitate legal websites and often employ social engineering tactics to deceive users. Such attacks mislead users by presenting fake sites as legitimate ones to steal sensitive information. Phishing attacks, especially those aiming to steal financial data, pose a significant threat to users and the damages caused by these attacks continue to increase. Fake websites detected by the National Cyber Incident Response Center demonstrate that phishing attacks are a serious cybersecurity issue in Turkey.

Effectively combating phishing attacks requires identifying their distinctive and common features before detecting such sites. This study focuses on attribute information related to URL content, character similarity, cosine similarity, Levenshtein distance similarity, and original domain features to determine if a website is a phishing site. It proposes a classifier model that achieves high accuracy rates, recognized in the literature.

In this study, 32 different attribute information were used in the dataset generated within the scope of the proposed model. Three different datasets were created from the dataset prepared to detect phishing attacks within the proposed model, and the performance of various feature selection methods and classification algorithms were evaluated on these datasets. Machine learning methods such as Decision Tree, Random Forest, Logistic Regression, Support Vector Machine, and Artificial Neural Networks were employed. Tests conducted on the datasets resulted in an accuracy parameter of 96,40% for the Logistic Regression method.

January 2024, 70 pages.

Keywords: Phishing Detection, Cosine Similarity, Machine Learning, Phishing Attack, Social Engineering

1. GİRİŞ

Günümüzde, siber saldırıların karmaşıklığı ve çeşitliliği arttıkça, siber güvenlik alanında etkili çözümler geliştirme ihtiyacı da katlanarak büyümektedir. Bu bağlamda, özellikle oltalama saldırıları gibi gizlilik ve güvenlik açısından ciddi tehditler oluşturan siber saldırıları tespit etme ve önleme konusu, büyük önem taşımaktadır. Oltalama saldırıları, kullanıcıları yanıltarak hassas bilgilerini ifşa etmeyi amaçlayan ve sıklıkla kimlik avı gibi taktikler kullanan zararlı saldırı türlerinden biridir [1].

Oltalama saldırıları, siber suçluların en yaygın kullandığı yöntemlerden biridir ve hedeflerinin kişisel bilgileri veya finansal verileri ele geçirmek olduğu bilinmektedir. Bu saldırı türü, kullanıcıların sahte web sitelerine veya e-posta bağlantılarına tıklamasıyla gerçekleşebilir. Oltalama saldırıları hem bireysel kullanıcıları hem de kurumları hedef alabilir ve büyük zararlara neden olabilir. Bu nedenle, bireysel ve kurumsal kullanıcıların bu tür saldırılara karşı dikkatli olmaları ve kendilerini korumaları önemlidir.

Oltalama saldırıları genellikle bir dizi adımdan oluşur. İlk adım, sahte bir e-posta veya mesaj göndermek ve kurbanın dikkatini çekmek olabilir. Saldırganlar, kurbanların dikkatini çekmek için, kurbanın güvendiği veya ilgisini çekebilecek bir kaynaktan e-postalar gönderebilirler. İkinci adım, kurbanı sahte bir web sitesine veya dosyaya yönlendirmektir. Bu, bir URL (Uniform Resource Locator) bağlantısı veya bir dosya eki şeklinde gerçekleşebilir. Üçüncü adım, kurbanın kişisel veya finansal bilgilerini girmesini sağlamak için sahte bir form göstermektir. Bu form, kurbanların girdikleri bilgileri saldırganlara gönderir.

Oltalama saldırılarına karşı korunmak için, kullanıcılar her zaman e-posta ve web sitelerinin doğruluğunu kontrol etmelidirler. E-postaların veya mesajların sahte olabileceği ve sahte web sitelerinin gerçek web sitelerine benzediği unutulmamalıdır. Kullanıcılar, sahte web sitelerine girmeden önce, web sitesinin URL bilgisinin doğruluğunu kontrol etmeli ve güvenilir olmayan web sitelerinden kaçınmalıdır. Ayrıca, kullanıcıların güçlü ve benzersiz şifreler kullanması ve iki faktörlü kimlik doğrulama sistemlerini kullanmaları önerilir. Kurumsal kullanıcılar, çalışanlarını oltalama saldırılarına karşı bilinçlendirmek ve güvenli bir şirket ağı sağlamak için ek önlemler almalıdır [2].

Bu çalışmanın kapsamı, öznitelik tabanlı oltalama tespit modelinin geliştirilmesi, geliştirilen modelin makine öğrenme yöntemlerinden Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağı yöntemleri ile sınıflandırarak analiz ve değerlendirilmesi ile ilgilidir. Bu model, oltalama saldırılarına karşı koruma sağlamak, saldırıları hızlı ve doğru bir şekilde tanımlamak amacıyla tasarlanmıştır. Öznitelik tabanlı yaklaşımlar, saldırıların özelliklerini ve davranışlarını inceleyerek bu tehditleri tanımlamak için kullanılan temel bir yöntemdir. Ulusal Siber Olaylara Müdahale Merkezi'ndeki (USOM) zararlı bağlantılar sayfasında [3] bulunan sahte siteler ile yasal sitelerden oluşan web siteleri kullanılarak, elde edilen veri seti üzerinde öznitelik çıkarımı yöntemi ile makine öğrenmesi yöntemleri uygulanarak doğruluk değerleri elde edilmiştir. Elde edilen doğruluk değerlerine göre yöntemlere ait sonuçlar karşılaştırılmış ve değerlendirilmiştir.

Bu çalışmanın amacı, oltalama saldırılarının etkili bir şekilde tespit edilmesini sağlayan bir model bilgisi geliştirmek, geliştirilen model ile makine öğrenme yöntemlerini kullanarak bir web sitesinin saldırı sitesi olup olmadığını tespit etmek ve değerlendirmektir. Bu amaç doğrultusunda, makine öğrenme yöntemlerinden Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağı yöntemlerinin oltalama tespiti üzerindeki performansını karşılaştırmak ve bu yöntemlerin oltalama saldırılarını belirleme yeteneklerini analiz etmek amaçlanmaktadır. Ayrıca, önerilen yöntemlerin gerçek dünya senaryolarında ne kadar etkili olduğunu değerlendirmek için USOM veri setinde bulunan sahte siteler ile yasal siteler kullanılmıştır.

Öznitelik çıkarımı için USOM veri setindeki sahte siteler ile yasal sitelere ait bilgiler birleştirilerek yeni bir veri kümesi oluşturulmuştur. Oluşturulan yeni veri kümesi üzerinde öznitelik çıkarımı işlemleri yapılarak analizleri ve değerlendirmeleri yapacağımız veri seti elde edilmiştir.

Oluşturulan yeni veri kümesi tamamen yasal sitelerden oluşan veri seti ile karşılaştırılarak kosinüs, karakter ve levenshtein benzerlik yöntemlerine ait benzerlik oranları, benzerlik gösteren site bilgisi ile orijinal domain ait olup olmadığının sonuçları da öznitelik bilgisi olarak veri setine eklenmiştir. Elde edilen veri kümesi üç farklı veri setine ayrılarak, her veri seti üzerinde makine öğrenme yöntemlerinden Karar Ağacı yöntemi, Rastgele Orman yöntemi, Lojistik Regresyon yöntemi, Destek Vektör Makinesi yöntemi ve Yapay Sinir Ağları yöntemleri uygulanmış ve sonuçları karşılaştırılmıştır.

Bu tez çalışması, KAVRAMSAL ÇERÇEVE başlığı altında altı ana bölümden oluşmaktadır. Sosyal Mühendislik kısmında, oltalama saldırılarının nasıl ortaya çıktığı hakkında bilgiler verilmiştir. Oltalama Saldırısı ve Türleri kısmında, oltalama saldırısının tanımı yapılarak, oltalama saldırısının ne olduğu açıklanmış ve saldırı türlerinin neler olduğu ile ilgili genel bilgiler verilmiştir. Oltalama Saldırı Tespit Türleri bölümünde, oltalama saldırı tespit türleri hakkında bilgiler verilmiştir. Benzerlik Yöntemleri kısmında, çalışmada kullanılan benzerlik yöntemleri ve yöntemler hakkında genel bilgiler açıklanmıştır. Makine Öğrenmesi kısmında, tezde kullanılan makine öğrenimi yöntemleri ve genel metodoloji açıklanmıştır. Literatür Taraması bölümünde, benzerlik tabanlı, kural tabanlı ve makine öğrenmesiyle yapılan çalışmalardan seçilen ve incelenen araştırmaların özetleri sunulmuştur. YÖNTEM kısmında, veri kaynağı, veri birleştirme, veri temizleme, tez kapsamında çıkarılan özniteliklerin neler olduğu, veri setinin oluşturulması, veri üzerine yapılan işlemler, oluşturulan model, sonuçların değerlendirilmesi için kullanılan metrikler hakkında bilgiler anlatılmış ve uygulanan yöntemler ayrıntılı açıklanmıştır. BULGULAR bölümünde, modelden elde edilen sonuçlar sunulmuştur. TARTIŞMA kısmında, yapılan çalışmanın bulguları karşılaştırılarak bu çalışmanın literatüre katkısı hakkında bilgiler verilmiştir. SONUÇ ve ÖNERİLER bölümünde, yapılan çalışmanın sonuçları ve bu çalışmanın gelecekteki çalışmalara katkısı detaylı bir şekilde ele alınmıştır.

2. KAVRAMSAL ÇERÇEVE

Bu bölümde çalışılmakta olan konu hakkında temel bilgiler, çalışmanın amacı, hedefleri ve kapsamı hakkında bilgiler açıklanmıştır. Sosyal mühendislikten, oltalama saldırısı tanımı yapılarak oltalama tespit türleri hakkında bilgiler sunulmuştur. Çalışmada kullanılan benzerlik yöntemleri olan Karakter benzerliği yöntemi, Kosinüs benzerliği yöntemi ve Levenshtein yöntemi ile makine öğrenmesi tanımı yapılarak, makine öğrenmesi yöntemleri olan Karar Ağacı yöntemi, Rastgele Orman yöntemi, Lojistik Regresyon yöntemi, Destek Vektör Makinesi ve Yapay Sinir Ağları hakkında bilgiler özetlenmiştir. Bu bölümün sonunda bu alanda yapılmış olan çalışmalar gruplara ayrılarak çalışmalar hakkında bilgiler verilmiştir.

Bu çalışmanın temel amacı, bir web sitesine ait url bilgisinin oltalama saldırısı olup olmadığını tespit eden bir model geliştirmek ve geliştirilen modelin tespit edilmesini sağlayan makine öğrenme yöntemlerini önermek ve değerlendirmektir. Bu bağlamda, modelin farklı makine öğrenme yöntemlerinin oltalama saldırısı tespiti üzerindeki performansını karşılaştırmak ve bu yöntemlerin oltalama saldırılarını belirleme yeteneklerini analiz etmek amaçlanmaktadır.

Bu çalışmanın kapsamı, oltalama saldırılarının tespiti üzerine odaklanan öznitelik tabanlı tespit modelini geliştirmek ve bu geliştirilen modelin makine öğrenme yöntemleriyle değerlendirilmesini içermektedir. Çalışma, farklı makine öğrenme yöntemlerinin kullanıldığı öznitelik tabanlı oltalama tespit modelini ele alacaktır. Değerlendirme süreci, veri kümesi ve veri kümesinden elde edilen veri setleri üzerinde makine öğrenme yöntemleri kullanılarak gerçekleştirilecektir.

Veri seti üç farklı veri setine ayrılarak, farklı veri setleri üzerinde farklı sınıflandırma yöntemlerini uygulayarak en uyumlu birlikteliği elde etmek ve bir web sitesinin oltalama saldırısı tespitinde en yüksek doğruluk değerini elde etmesini sağlamaktır. Bu çalışma ile oltalama saldırılarının temel özelliklerini ve davranışlarını tanımlamak, sahte ve yasal web siteleri kullanılarak oluşturulan veri kümesinden özniteliklerin çıkarılmasını sağlamak, öznitelik çıkarımında benzerlik yöntemlerini kullanarak oltalama tespitinde daha yüksek doğruluk değerleri elde etmek ve farklı makine öğrenme algoritmalarının oltalama tespiti üzerindeki etkisini değerlendirmek hedeflenmiştir. Ayrıca oltalama saldırılarına özgü öznitelik çıkarımı yapılarak siber saldırı uzayında önemli bir yer edinen konu, yine bu alanda

yapılacak olan bilimsel çalışmalara referans olmak ve öznitelik tabanlı oltalama saldırılarıyla birlikte yaşanabilecek güvenlik sorunlarına katkı sağlamak hedeflenmektedir.

2.1. Sosyal Mühendislik

Sosyal mühendislik, insanları manipüle etme ve yanıltma tekniklerini kullanarak hassas bilgileri ele geçirme veya belirli davranışları gerçekleştirmeye yönelik bir saldırı taktiğidir. Bu taktikler, güvenlik duvarlarını, şifreleri veya diğer teknik güvenlik önlemlerini aşmak yerine, insanların doğal davranışlarına ve güvensizliklerine odaklanır. Sosyal mühendislik saldırıları, genellikle dolandırıcılık, kimlik hırsızlığı, şifre çalma veya gizli bilgilere erişim gibi kötü amaçlarla kullanılır. Bu tür saldırılar, genellikle kurumsal veya kişisel e-posta yoluyla gelen sahte mesajlar, telefon aramaları veya sosyal medya üzerinden gerçekleştirilir. Saldırganlar, hedef kişilerin güvenini kazanarak, onları belli bir eylemi gerçekleştirmeye ikna etmeye çalışır. Sosyal mühendislik teknikleri arasında oltalama saldırısı gibi yöntemler bulunur. Bu teknikler, insanların duygusal zayıflıklarını, güvenlerini veya doğal eğilimlerini sömürerek başarılı olabilir [4].

Sosyal mühendislik saldırılarına karşı koymak için kullanıcıların dikkatli olması, şüpheli durumları rapor etmesi ve kişisel bilgilerini paylaşmaktan kaçınması önemlidir. Eğitim ve farkındalık artırma önlemleri de organizasyonların ve bireylerin sosyal mühendislik saldırılarına karşı daha dirençli olmalarına yardımcı olabilir.

2.2. Oltalama Saldırısı ve Türleri

2.2.1. Oltalama Saldırısı (Phishing Attack) Tanımı

Siber güvenlik, günümüzün dijital dünyasında giderek daha fazla önem kazanmaktadır. İnternetin yaygınlaşması ve dijitalleşmenin artmasıyla birlikte, siber saldırılar daha sofistike hale gelmiştir. Bu saldırıların içinde, oltalama saldırısı (phishing attack), özellikle kullanıcıları hedef alarak hassas bilgilere erişmeye çalışan yaygın ve tehlikeli bir siber saldırı türüdür.

Oltalama saldırısı, kötü niyetli aktörlerin (siber suçluların) sahte kimlikler altında kendilerini güvenilir bir varlık gibi göstererek kullanıcıları yanılttığı bir siber saldırı türüdür. Temel amacı, kullanıcıları kişisel, finansal veya hassas bilgilerini ifşa etmeye veya kötü amaçlı yazılım yüklemeye ikna etmektir. Bu tür saldırılar, e-posta, SMS, anlık mesajlar,

sosyal medya veya sahte web siteleri gibi çeşitli iletişim kanalları aracılığıyla gerçekleştirilir [5]. Oltalama saldırılarının işleyişi, genellikle birkaç adımdan oluşmaktadır.

Siber saldırganın hazırlığı adımında saldırgan, bir oltalama kampanyası başlatmadan önce hedef kitlesi ve amaçları hakkında bilgi toplar. Bu, hedeflenen kuruluşun veya kişinin zayıf noktalarını anlama çabasını içerir. Oltalama mesajının hazırlanması adımında saldırgan, sahte e-postalar, mesajlar veya web siteleri oluşturur. Bu iletişimler, hedefi kandırmak için güvenilir bir kaynak gibi görünür. Genellikle sahte bir kuruluşun veya hizmetin temsilcisi olduklarını iddia ederler. Kurbanların yanıltılması adımında oltalama mesajları, sahte teklifler, acil durumlar veya güncelleme gereklilikleri gibi kullanıcıların dikkatini çekecek içeriklerle hazırlanır. Kullanıcılar, bu mesajları açtıklarında veya bağlantılara tıkladıklarında yanıltılırlar. Hassas bilgilerin çalınması adımında oltalama mesajlarına yanıt veren veya sahte web sitelerine giriş yapan kullanıcılar, kişisel bilgilerini, şifrelerini, kredi kartı numaralarını veya diğer hassas verilerini ifşa edebilirler. Bu bilgiler daha sonra kötü niyetli amaçlar için kullanılabilir [6].

2.2.2. Oltalama Saldırısı Türleri

Elde ettiği başarı oranı yüksek olan oltalama saldırıları, siber saldırganların en çok tercih ettiği saldırı türleri arasında yer almaktadır. Bu saldırı türleri, saldırganların hedeflediği kuruluşlar veya bireylerin güvenliğini riske sokmakta ve ciddi maddi veya manevi zararlar doğurabilmektedir. Bu nedenle, oltalama saldırılarına karşı alınacak önlemler ve farkındalığın artırılması oldukça önemlidir.

Oltalama saldırıları, temel olarak sosyal mühendislik yöntemlerini kullanarak kurbanların güvenini kazanmak ve kişisel bilgilerini veya finansal bilgilerini ele geçirmek amacıyla gerçekleştirilir. Bu saldırı türleri, genellikle e-posta, telefon veya sosyal medya üzerinden gerçekleştirilir. Ayrıca, saldırganlar hedeflerine ulaşmak için farklı teknikler kullanmakta ve oltalama saldırıları birçok farklı şekilde sınıflandırılmaktadır [7].

2.2.2.1. Hedef Bazlı Oltalama Saldırıları

Hedef bazlı oltalama saldırıları, genellikle kurumsal veya kurumsal hedefleri olan bir birey veya grup tarafından yürütülen oltalama saldırılarıdır. Bu saldırı türünde, saldırganlar öncelikle belirli bir kuruluş veya kişiye özgü bir hedef belirlerler. Ardından, hedefin ilgi alanları, zaafları, etkinlikleri ve çalışanları hakkında ayrıntılı bilgi toplamak için sosyal mühendislik taktikleri kullanırlar.

Hedef bazlı oltalama saldırıları genellikle daha sofistike ve karmaşık olabilir. Saldırganlar, hedefe özgü bir e-posta mesajı veya web sitesi gibi özel olarak tasarlanmış araçlar kullanarak, hedefin dikkatini çekmek ve onları güvenli olmayan bir işlem yapmaya yönlendirmek için çalışırlar. Bu işlemler, örneğin, hassas bilgi talepleri veya kötü amaçlı yazılım indirme gibi hedefin güvenliği açısından önemli tehditler oluşturabilir [8].

Hedef bazlı oltalama saldırıları, birçok farklı nedenle gerçekleştirilebilir. Örneğin, kurumun ticari sırlarını çalmak, bir devlet kurumunu casusluk yapmak veya bir kişinin finansal bilgilerine erişmek için kullanılabilirler. Bu tür saldırıların gerçekleştirilmesi genellikle zaman alıcı ve maliyetlidir, ancak sonuçları oldukça zararlı olabilmektedir. Bu nedenle, hedef bazlı oltalama saldırılarına karşı korunmak için kurumlar ve bireyler, iyi bir siber güvenlik planı oluşturmalı ve düzenli olarak güvenlik eğitimleri almalıdır.

2.2.2.2. SMS Oltalama Saldırıları

SMS oltalama saldırıları, saldırganların hedef kişilerin cep telefonlarına sahte veya aldatıcı SMS mesajları göndererek kişisel bilgilerini elde etmeye veya kötü amaçlı yazılım yüklemeye çalıştığı bir tür oltalama saldırısıdır. Bu tür saldırılar genellikle bir kısa mesaj servisi (SMS) aracılığıyla gerçekleştirilmektedir. Saldırganlar genellikle hedef kişileri sahte bir işlem yapmaya, bir bağlantıya tıklamaya veya kötü amaçlı bir uygulama yüklemeye ikna etmeye çalışırlar.

SMS oltalama saldırıları genellikle bankacılık işlemleri, kredi kartı işlemleri veya diğer finansal işlemlerle ilgili olabilir. Saldırganlar, hedef kişilerin finansal bilgilerini ele geçirmek veya finansal dolandırıcılık yapmak için sahte bir banka mesajı veya kredi kartı işlemi mesajı gönderebilirler. SMS oltalama saldırıları ayrıca bir kötü amaçlı yazılım yüklemek veya hedef kişinin cep telefonu verilerini ele geçirmek için de kullanılabilir. Saldırganlar, hedef kişilere kötü amaçlı bir bağlantı isteği göndererek veya bir kötü amaçlı uygulamanın indirilmesi için yanıltıcı bir bağlantı isteği sağlayarak telefonlarına kötü amaçlı bir yazılım yüklemeyi deneyebilirler.

SMS oltalama saldırılarından korunmak için, hedef kişilerin sahte veya şüpheli mesajlara karşı dikkatli olmaları, bilinmeyen veya güvenilmeyen kaynaklardan gelen mesajlara tıklamamaları veya yanıtlamamaları önemlidir. Ayrıca, cep telefonlarının güvenlik ayarlarının düzenli olarak güncellenmesi, güncellenmeyen uygulamaların kaldırılması ve bir anti-virüs yazılımının kullanılması da önemlidir [8, 9].

2.2.2.3. Sesli Oltalama Saldırıları

Sesli oltalama saldırıları, bir saldırganın telefonda bir kişiyi arayarak, genellikle finansal bilgileri veya diğer hassas bilgileri ele geçirmeye çalıştığı bir saldırı türüdür. Saldırganlar, mağduru gerçek bir işletme, finansal kurum veya hükümet kurumu gibi tanınmış bir kuruluş gibi hissettirmek için yanıltıcı bilgiler kullanırlar. Sesli oltalama saldırıları genellikle birkaç aşamalı bir süreç olarak gerçekleştirilir. İlk olarak, saldırgan mağdurun güvenini kazanmak için genellikle bir konuda uzman olduğunu veya bir yetkilinin temsilcisi olduğunu iddia eder. İkinci aşamada, saldırgan mağdurdan özel bilgiler talep eder. Bu bilgiler, kredi kartı bilgileri, banka hesap bilgileri, sosyal güvenlik numaraları veya diğer hassas bilgiler olabilir. Son olarak, saldırgan bu bilgileri kötüye kullanabilir veya satmak için kullanabilir.

Sesli oltalama saldırılarının bir diğer yaygın şekli ise "vishing" olarak bilinir. Bu, "voice" (ses) ve "phishing" (oltalama) terimlerinin birleşiminden oluşan bir terimdir. Vishing, e-posta veya metin mesajı gibi diğer iletişim kanalları aracılığıyla gerçekleştirilen oltalama saldırılarına benzer, ancak burada saldırganlar telefon aracılığıyla kurbanlarını yanıltmaya çalışır [10].

Sesli oltalama saldırıları, kullanıcıların dikkatli olması ve kişisel bilgilerini vermeden önce her zaman doğrulama yapması gerektiği için oldukça tehlikelidir. Bu tür dolandırıcılık vakalarına karşı dikkatli olmak ve bilinmeyen numaralardan veya tanımadığımız kişilerden gelen aramalara karşı dikkatli yaklaşmak önemlidir. Özellikle işletmeler, çalışanlarına bu tür saldırıları tanımaları ve nasıl önleyeceklerini bilmeleri için eğitim vermelidir. Ayrıca, işletmeler, müşterilerini bu tür saldırılara karşı bilgilendirmeli ve korumak için gerekli önlemleri almalıdır.

2.3. Oltalama Saldırı Tespit Türleri

2.3.1. Liste Tabanlı Oltalama Saldırı Tespit Sistemleri

Liste tabanlı saldırı tespit yöntemleri, bilinen kötü amaçlı işaretlere veya tanımlanmış tehditlere dayalı olarak ağdaki anormal etkinlikleri tespit etmek için kullanılır. Bu tür tespit yöntemleri, ağ trafiğini veya sistem etkinliklerini izler ve önceden tanımlanmış kötü niyetli desenleri arar. Kara liste tabanlı saldırı tespiti, bilinen kötü amaçlı IP adresleri, URL'ler, dosya adları veya imzalar gibi önceden belirlenmiş "kara liste" öğelerini kullanır. Ağ trafiği veya sistem etkinlikleri, bu kara liste öğeleri ile eşleştiğinde, bir saldırı tespit edilir. Bu

yöntem, bilinen tehditlere karşı oldukça etkilidir, ancak yeni veya özgün saldırıları tespit etmekte zorlanabilir.

Beyaz liste tabanlı saldırı tespiti, yalnızca belirli bir liste veya "beyaz liste" içinde tanımlanmış olanlar dışında kalanları tespit eder. Örneğin, yalnızca belirli bir uygulamanın çalışmasına izin verilmesi durumunda, bu yöntem, diğer uygulamaların çalışmasını tespit edebilir. Beyaz liste tespiti, tanıdık olmayan veya izinsiz uygulamaları tespit etmek için kullanışlıdır [11].

İmza tabanlı tespit, bilinen saldırıların imzalarını içeren veritabanları kullanır. Bu imzalar, belirli bir saldırı türünü veya kötü amaçlı yazılımı tanımlar. Eğer bir sistem veya ağ üzerinde bu imzalarla eşleşen bir etkinlik tespit edilirse, saldırı olarak işaretlenir. Bu yöntem, yaygın olarak kullanılan kötü amaçlı yazılımları tespit etmek için etkilidir, ancak yeni ve bilinmeyen tehditlere karşı savunmasızdır.

Başlık tabanlı tespit, ağ trafiğinin veya sistem etkinliklerinin başlıklarını (örneğin, HTTP başlıkları) analiz eder. Belirli başlık modelleri veya yapıları, kötü amaçlı etkinliklerin belirtilerini taşıyabilir. Başlık tabanlı tespit, özellikle HTTP trafiği üzerinde kullanışlıdır ve kötü amaçlı web sayfalarını veya istekleri tespit etmek için kullanılmaktadır.

Protokol tabanlı tespit, ağ trafiğinin belirli iletişim protokollerini (örneğin, TCP/IP, DNS, SMTP) inceleyerek anormal veya kötü amaçlı davranışları tespit etmeye odaklanır. Örneğin, DNS (Domain Name System) trafiğinde anormal bir artış, bir DDoS (Distributed Denial of Service) saldırısının belirtisi olabilir. Bazı liste tabanlı tespit yöntemleri, zararlı dosyaların veya uygulamaların analizi için statik veya dinamik analiz kullanabilir. Statik analiz, dosyanın içeriğini inceleyerek potansiyel tehditleri belirler. Dinamik analiz, dosyanın gerçek dünyada nasıl davrandığını gözlemleyerek tehditleri tespit eder [12].

2.3.2. Kural Tabanlı Oltalama Saldırı Tespit Sistemleri

Kural tabanlı saldırı tespit sistemleri (Rule-Based Intrusion Detection Systems veya RBIDS), bilgisayar ağları ve sistemlerindeki anormal aktiviteleri veya belirli davranış kalıplarını tespit etmek için kullanılan bir güvenlik teknolojisi türüdür. Bu sistemler, belirli kuralları veya işlem mantığını kullanarak potansiyel tehditleri tanımak ve raporlamak için tasarlanmıştır [13]. Kural tabanlı saldırı tespit sistemleri, belirli bir tehdit türünü veya anormal davranışı tanımlamak için kullanılan kuralları içerir. Bu kurallar, ağ trafiği, sistem günlükleri

veya diğer kaynaklardan gelen verileri izler. Örnek bir kural, "Bir kullanıcı 3 (üç) başarısız oturum açma girişiminde bulunursa, bir saldırı girişimi olarak kabul edilsin" şeklinde olabilir.

Kural tabanlı sistemler, belirli bir işlem mantığını veya algoritmayı kullanarak verileri analiz eder. Bu işlem mantığı, genellikle potansiyel tehditleri tanımak için belirli davranış kalıplarını izler. Örneğin, bir kullanıcının normal oturum açma davranışından sapması veya belirli bir IP adresinden gelen aşırı trafik, işlem mantığı tarafından tespit edilebilir. Bu sistemler, genellikle ağ trafiği verileri, günlük dosyaları, güvenlik olayları ve diğer kaynaklardan veri alır. Bu veriler, saldırıları veya anormal aktiviteleri tespit etmek için incelenir.

Kural tabanlı saldırı tespit sistemleri, tanınmış tehditleri içeren bir kara liste ve güvenilir kaynakları içeren bir beyaz liste kullanabilir. Bu listeler, sistemlerin hangi trafiği izlemesi gerektiğini belirlemelerine yardımcı olur. Bu sistemler, belirli bir davranışın ne kadar anormal olduğunu değerlendirmek için eşik değerleri kullanabilir. Örneğin, ağ trafiği belirli bir hızın üzerine çıkarsa veya aynı anda çok sayıda bağlantı açılırsa, bir saldırı girişimi olarak kabul edilebilir.

2.3.3. Görsel Benzerlik Tabanlı Oltalama Saldırı Tespit Sistemleri

Görsel benzerlik tabanlı saldırı tespit sistemleri, görsel içeriği analiz ederek ve benzerlik ölçümleri kullanarak bilgisayar ağı veya dijital ortamlarda meydana gelen saldırıları tespit etmeye yönelik bir güvenlik teknolojisi türüdür. Bu sistemler, özellikle resimler, videolar veya diğer görsel medya türleri üzerinde çalışır ve bu medyanın içeriğini inceleyerek saldırıları veya tehditleri tespit etmeye çalışır.

Görsel benzerlik tabanlı saldırı tespit sistemleri, normal veya güvenli olarak kabul edilen görsel içeriğin belirli bir benzerlik eşiğini aştığı durumları tespit etmeyi amaçlar. Bu, normalden sapma gösteren veya önceden tanımlanmış güvenlik politikalarına uymayan görsel içeriği belirlemeye çalışır. Görsel benzerlik tabanlı saldırı tespit sistemleri, çeşitli kullanım alanlarında yaygın olarak kullanılır. Örneğin, video gözetimi, siber saldırı tespiti, görsel veri madenciliği ve içerik filtreleme gibi alanlarda kullanılır [14].

2.4. Benzerlik Yöntemleri

Benzerlik yöntemleri, iki veya daha fazla nesne, veri noktası veya öğe arasındaki benzerliği veya farklılığı ölçmeye ve karşılaştırmaya yönelik matematiksel veya istatistiksel bir yaklaşımı ifade eder. Bu yöntemler, makine öğrenmesi, veri madenciliği, metin işleme, görüntü işleme ve öneri sistemlerinde kullanılır.

2.4.1. Karakter Benzerliği Yöntemi

Karakter benzerliği algoritması, iki veya daha fazla karakter dizisinin benzerliğini veya farklılığını ölçmek için kullanılan hesaplama yöntemidir. Karakter benzerliği, iki karakter dizisi arasındaki benzerlik seviyesini sayısal olarak ifade eden bir değerle sonuçlanır.

Tez çalışmamda, iki url bilgisi arasındaki karakter benzerliği hesaplaması Python'ın difflib kütüphanesinin bir parçası olan SequenceMatcher sınıfını kullanılarak yapılmıştır. Denklem 2.1 de verilen kod parçasına göre SequenceMatcher sınıfı, iki sıra benzeri nesnenin karşılaştırılmasını sağlar ve benzerlik oranını döndürür. None olarak belirtilen ilk parametre, bir benzerlik oranı hesaplamasının nasıl yapılacağını belirlemek için kullanılır. Fonksiyon, similarity adında bir değişkene benzerlik oranını atar ve bu değeri geri döndürür.

```
similarity = SequenceMatcher(None, url1, url2).ratio()
```
(2.1)

Özetle, bu kod parçası iki karakter dizisinin (URL veya başka herhangi bir metin) benzerlik oranını hesaplayan bir fonksiyon içerir. Benzerlik oranı, 0 (sıfır) ile 1 (bir) arasında bir değerdir, 1 (bir)'e yaklaştıkça iki karakter dizisi birbirine daha çok benzer.

2.4.2. Kosinüs Benzerliği (Cosine Similarity) Yöntemi

Kosinüs benzerliği, iki farklı vektörün benzerliğini ölçmek için kullanılan bir metrik veya benzerlik ölçüsüdür. Kosinüs benzerliği, iki vektörün birbirine ne kadar benzediğini ölçer ve sonucu 0 ile 1 arasında bir değer olarak verir. 0, iki vektörün hiçbir benzerliğe sahip olmadığını, 1 ise iki vektörün tamamen aynı olduğunu gösterir. Kosinüs benzerliği, makine öğrenmesi, metin madenciliği, doküman benzerliği, öneri sistemleri ve çok boyutlu veri madenciliği gibi birçok alanda kullanılır. Özellikle metin belgelerinin benzerliğini hesaplarken kullanılan etkili bir ölçüdür [15]. Tezde iki url arasındaki kosinüs benzerliği hesaplaması Denklem 2.2'de verilmiştir. A ve B vektörleri, benzerliğini ölçmek istediğimiz

iki vektördür. A_i ve B_i, sırasıyla A ve B vektörlerinin i indeksli öğeleridir. n, vektörlerin boyutunu temsil eder.

$$\text{Cosine Similarity(A,B)} = \frac{\sum_{i=1}^{n} A_i x B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2 x} \quad \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{2.2}$$

Kosinüs benzerliği, iki vektörün yönleri arasındaki açıyı hesaplar ve bu açının kosinüsünü kullanarak benzerliği ölçer. Eğer açı 0 derece ise, kosinüs benzerliği 1'e eşit olur, yani iki vektör tamamen aynıdır. Eğer açı 90 derece ise, kosinüs benzerliği 0'a eşit olur, yani iki vektör arasında hiçbir benzerlik yoktur [16].

Bu durumu bir örnek ile açıklayacak olursak, iki metin belgesi vektör olarak A = [2, 3, 1, 0, 0] ve B = [1, 2, 0, 1, 1] vektörleri olsun. Bu vektörlerin kosinüs benzerliği Denklem 2.2'de yerine koyulup hesaplanır.

$$\text{Cosine Similarity(A,B)} = \frac{(2x1)+(3x2)+(1x0)+(0x1)+(0x1)}{\sqrt{2^2+3^2+1^2+0^2+0^2}.\sqrt{1^2+2^2+0^2+1^2+1^2}} = 0{,}919$$

Tez çalışmamda, iki url bilgisi arasındaki kosinüs benzerliği hesaplaması Python'ın Scikit-Learn kütüphanesinin metrics modülünden pairwise alt modülünde yer alan cosine_similarity fonksiyonunu kullanılarak yapılmıştır. Bu fonksiyon, iki url bilgisi arasındaki kosinüs benzerliğini hesaplar.

2.4.3. Levenshtein Mesafe Yöntemi

Levenshtein mesafe yöntemi, Levenshtein [17] tarafından bulunmuştur. İki metin dizisi veya kelimenin benzerliğini ölçmek için kullanılan bir metrik veya düzenleme mesafesi ölçüsüdür. Bu yöntem, iki kelimenin birbirine dönüştürülmesi gereken minimum işlem sayısını (ekleme, çıkarma veya değiştirme) hesaplar. İki kelime arasındaki harf sayıları eşitlenir. Uzun olan kelime, kısa kelimenin sonuna boşluk karakteri eklenerek eşit uzunluğa getirilir. Her iki kelimenin başından itibaren sırayla harf karşılaştırmaları yapılır. Eğer karşılaştırılan harfler aynı ise herhangi bir işlem yapılmasına gerek yoktur. Eğer farklı ise üç farklı işlem seçeneği vardır. Değiştirme (Substitution), bir harfi başka bir harf ile değiştirme işlemidir. Ekleme (Insertion), bir harf ekleme işlemidir. Çıkarma (Deletion), bir harfi çıkarma işlemidir. Bu işlemlerle her iki kelimenin birbirine dönüştürülmesi gereken minimum işlem sayısı hesaplanır.

Levenshtein mesafe yöntemi, kelime düzeltme, metin madenciliği, genetik dizilim karşılaştırması ve biyoinformatik gibi birçok alanda kullanılır. İki kelime arasındaki Levenshtein mesafesi Denklem 2.3'te verilmiştir.

$$D[i,j] = \begin{cases} 0, & \text{if } i = 0 \text{ and } j = 0 \\ i, & \text{if } i > 0 \text{ and } j = 0 \\ j, & \text{if } i = 0 \text{ and } j > 0 \\ \min \begin{cases} D[i-1,j] + 1 \\ D[i,j-1] + 1 \\ D[i-1,j-1] + (1 \text{ if } str1[i] \neq str2[j] \text{ else } 0) \end{cases} \end{cases} \tag{2.3}$$

Burada, D[i,j], Levenshtein mesafesini temsil eden bir matris elemanıdır. str1[i] ve str2[j], sırasıyla ilk ve ikinci metindeki i ve j indekslerindeki karakterleri temsil eder. D[i-1,j], bir karakter eklemeyi temsil eder. D[i,j-1], bir karakter çıkarmayı temsil eder. D[i-1,j-1], iki karakterin aynı mı yoksa farklı mı olduğuna bağlı olarak bir karakteri diğerine dönüştürmeyi temsil eder.

Örneğin, "yahoo" ve "google" kelimeleri arasındaki levenshtein mesafesini hesaplayalım. Birinci adımda iki kelimenin uzunluğunu eşit hale getirelim. "yahoo" kelimesinin uzunluğu 5 harf, "google" kelimesinin uzunluğu 6 harftır. Bu nedenle "yahoo" kelimesinin sonuna bir boşluk ekleyelim ve uzunluklarını eşit hale getirelim. "Yahoo" kelimesinin yeni hali: "yahoo " olur. İkinci adımda her iki kelimenin başından itibaren harf karşılaştırmalarını yapalım. "y" ile "g" karşılaştırıldığında fark var, bu nedenle değiştirme işlemi gerekiyor. "a" ile "o" karşılaştırıldığında fark var, bu nedenle değiştirme işlemi gerekiyor. "h" ile "o" karşılaştırıldığında fark var, bu nedenle değiştirme işlemi gerekiyor. "o" ile "g" karşılaştırıldığında fark var, bu nedenle değiştirme işlemi gerekiyor. "o" ile "o" karşılaştırıldığında fark yok. " " ile "l" karşılaştırıldığında fark var, bu nedenle değiştirme işlemi gerekiyor. Üçüncü adımda toplam işlem sayısını hesaplarız. Toplamda 5 değiştirme işlemi yapıldı. Sonuç olarak "yahoo" kelimesini "google" kelimesine dönüştürmek için 5 değiştirme işlemi gereklidir. Bu, "yahoo" ve "google" kelimeleri arasındaki Levenshtein mesafesidir.

Levenshtein mesafesi ile kelime veya metin dizileri arasındaki uzaklık sayısı arttıkça benzerlik azalır ve uzaklık sayısı azaldıkça benzerlik artar. Yani, iki kelimenin benzerliği artıkça Levenshtein mesafesi azalır.

Tez çalışmamda, iki url bilgisi arasındaki levenshtein mesafe hesaplaması Python'ın Levenshtein kütüphanesindeki distance fonksiyonunu kullanılarak yapılmıştır. Bu fonksiyon, iki url bilgisi arasındaki bir url bilgisinin diğerine dönüştürülmesi için gereken en az düzenleme (ekleme, çıkarma veya değiştirme) işlemi sayısını hesaplar.

2.5. Makine Öğrenmesi (ML - Machine Learning)

Makine öğrenmesi (Machine Learning), bilgisayar sistemlerinin verilerden öğrenmelerini ve bu öğrenilen bilgileri kullanarak belirli görevleri gerçekleştirmelerini sağlayan bir yapay zeka alt dalıdır. Makine öğrenimi, temel amacı birçok farklı görevi gerçekleştirebilen genel bir zeka yaratmak olan Yapay Zeka (AI - Artificial Intelligence) alanının önemli bir parçasıdır. Makine öğrenimi, genellikle algoritma ve istatistiksel model oluşturma süreçlerini içerir. Bu süreçler, veriyi analiz eder, modeli eğitir ve ardından yeni verilerle test ederek modelin doğruluğunu belirler. Makine öğrenimi, büyük veri miktarlarını işlemek ve karmaşık görevleri gerçekleştirmek için insan müdahalesi olmadan makinelerin öğrenmesini sağlayarak birçok alanda büyük bir potansiyele sahiptir [18]. Makine öğrenimi temelde üç kategoriye ayrılır.

Denetimli öğrenme (Supervised learning), algoritmanın belirli bir veri kümesinde giriş verileri ile hedef çıktıları (etiketleri) arasındaki ilişkiyi öğrendiği bir süreçtir. Algoritma, eğitim verilerinden örneklerle beslenir ve bu verilere dayanarak gelecekteki verilerle ilgili tahminler yapabilir. Sınıflandırma ve regresyon gibi problemlerde denetimli öğrenme kullanılır.

Denetimsiz öğrenme (Unsupervised learning), veriler arasındaki yapıyı veya kalıpları keşfetmeye çalışır. Etiketlenmiş veri yoktur, bu yüzden algoritma, veri kümesindeki benzerlikleri, grupları veya özellikleri kendiliğinden öğrenir. Kümeleme (clustering) ve boyut indirgeme (dimensionality reduction) gibi görevlerde denetimsiz öğrenme kullanılır.

Güçlendirme öğrenmesi (Reinforcement learning), bir ajanın çevresiyle etkileşimde bulunarak, bir görevi en iyi şekilde nasıl gerçekleştireceğini öğrendiği bir öğrenme türüdür. Ajan, çevresel durumları değerlendirir ve belirli bir davranışa karar verir. Sonuçlar, ajanın aldığı aksiyonlara göre ödüllerle veya cezalarla şekillendirilir [18, 19].

2.5.1. Karar Ağacı (DT - Decision Tree) Yöntemi

Karar Ağacı (Decision Tree) yöntemi, öğrenme tabanlı bir makine öğrenme yöntemidir. Bu yöntem, bir sorunu çözmek için ağaç benzeri bir yapı oluşturarak karar verir. Her iç düğüm, bir özellik veya testi temsil eder, her dal ise bu test sonucunda alınabilecek kararları gösterir [20]. Karar ağacı, kök düğümden başlar ve yaprak düğümlere (leaf nodes) kadar ilerler. Her iç düğüm, bir özellik veya koşulu temsil eder. Her yaprak düğüm, sonuç sınıfını veya tahminini içerir.

Karar ağacı, veri kümesini bölerek en fazla bilgi kazancını sağlayacak bir özellik seçmeye çalışır. Bilgi kazancı, belirli bir özellik kullanıldığında veri kümesindeki belirsizliğin azalma miktarını ifade eder. Bu, her özelliği değerlendirerek başlar ve bilgi kazancı en yüksek olanı seçer. Karar ağacının derinliği, ağacın ne kadar karmaşık olduğunu gösterir. Derin bir ağaç, daha fazla karmaşıklık ve ayrıntı sağlayabilir, ancak aynı zamanda aşırı uydurma (overfitting) riski de taşır. Bu nedenle ağacın derinliği genellikle sınırlanır. Ağaç oluşturulduktan sonra, gereksiz dalları (branch) kaldırmak için kesme işlemi yapılabilir. Bu, ağacın daha basit ve daha genelleştirilebilir hale gelmesine yardımcı olmaktadır. Örneğin, hava durumu tahminlemesi yapmak istiyoruz. Hava durumu tahminlemesi için bir Karar Ağacı oluşturabiliriz. Kök düğüm, günün saatini temsil edebilir. İlk iç düğüm, saat 12'den önce mi sonra mı olduğunu kontrol edebilir. Bu ağacın sonucunda farklı hava durumu koşulları elde edebiliriz.

2.5.2. Rastgele Orman (RF – Random Forest) Yöntemi

Rastgele Orman (Random Forest), çoklu karar ağaçlarının (decision tree) bir araya getirilerek oluşturulan bir topluluk öğrenme yöntemidir. Bu yöntem, sınıflandırma ve regresyon gibi makine öğrenme yöntemlerini gerçekleştirmek için yaygın olarak kullanılır [21]. İlk adımda, rastgele veri örnekleme (bootstrap) yapılır. Bu, veri kümesinden rastgele örnekler alınarak yeni bir veri kümesi oluşturulmasını içerir. Bu işlem, her karar ağacının farklı bir eğitim veri kümesine sahip olmasını sağlar. Oluşturulan bu yeni veri kümesi üzerinde çok sayıda karar ağacı oluşturulur. Her ağaç, veri kümesinin rastgele bir alt kümesi ile eğitilir. Bu, ağaçların birbirinden bağımsız olmasını sağlar.

Sınıflandırma görevlerinde, her karar ağacı bir sınıf tahmini yapar. Rastgele Orman, bu tahminlerin bir araya getirilmesiyle sonuç elde eder. Örneğin, birden fazla ağaç sınıf tahminleri yapar ve en çok oy alan sınıf sonucu olarak kabul edilir. Regresyon görevlerinde

ise her ağaç bir sayısal tahmin yapar ve bu tahminlerin ortalaması sonucu oluşturur. Sınıflandırma görevi için bir Rastgele Orman örneği düşünelim. Elimizde bir öğrenci veri kümesi olsun ve her öğrencinin öğrenci kimliği, notları, katılım durumu, vb. bilgileri bulunsun. Bu veri kümesini kullanarak bir öğrencinin belirli bir sınavda başarılı olup olmayacağını tahmin etmek istiyoruz. Rastgele Orman algoritması, bu veri kümesini kullanarak çok sayıda karar ağacı oluşturur. Her ağaç, farklı öğrenci örnekleriyle eğitilir. Sonuçta, her ağaç sınav başarısıyla ilgili bir tahmin yapar. Rastgele Orman, bu tahminlerin bir araya getirilmesiyle öğrencinin sınav başarısını tahmin eder.

2.5.3. Destek Vektör Makinesi (SVM - Support Vector Machine) Yöntemi

Destek Vektör Makinesi (Support Vector Machine), sınıflandırma ve regresyon problemleri için kullanılan güçlü bir makine öğrenimi yöntemidir. Özellikle sınıflandırma problemleri için etkili bir şekilde kullanılır. SVM, veri noktalarını bir çoklu boyutta (çoklu özellikler) analiz eder ve bu verileri sınıflara ayırmak için bir karar sınırı (decision boundary) oluşturur. SVM, sınıfları ayırmak için en iyi karar sınırını (hiper düzlemi) bulmaya çalışır. Bu hiper düzlem, sınıflar arasındaki marjini (mesafe) maksimize eden bir şekilde çizilir. Bu marj, iki sınıf arasındaki güvenli bir bölgeyi ifade eder [22].

2.5.4. Lojistik Regresyon (LR - Logistic Regression) Yöntemi

Lojistik Regresyon (Logistic Regression), sınıflandırma problemleri için kullanılan bir makine öğrenimi yöntemidir. İsim benzerliği olmasına rağmen, lojistik regresyon bir regresyon tekniği değil, sınıflandırma tekniğidir. Genellikle iki sınıf (ikili sınıflandırma) için kullanılır, ancak çoklu sınıflandırma problemlerine de uyarlanabilir [23]. Lojistik regresyon, sınıflandırma problemlerinde sıklıkla kullanılır. Örneğin, hastalık teşhisi tahmini (Hasta / Değil), oltalama saldırı tahmini (Saldırı / Değil), kredi başvuru sonucu (Onay / Red).

2.5.5. Yapay Sinir Ağları (ANN – Artificial Neural Networks) Yöntemi

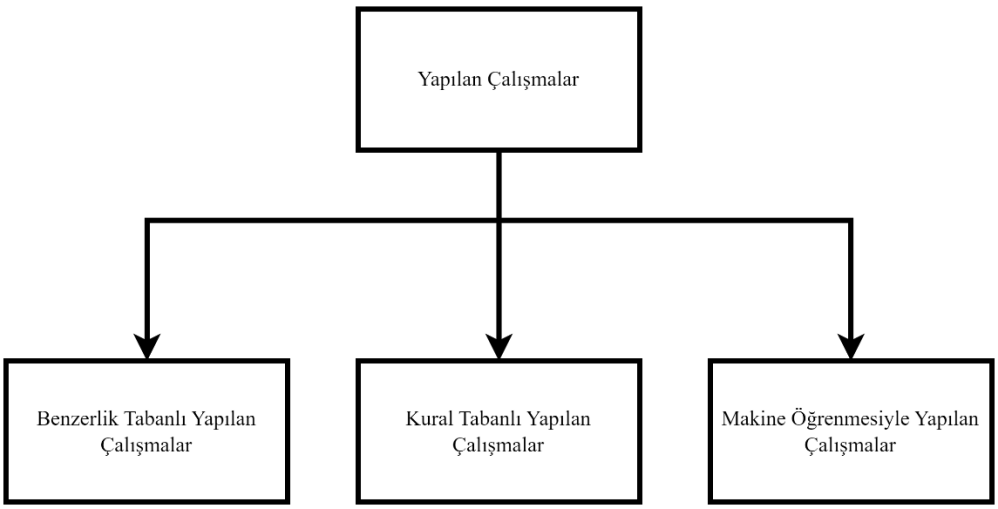
Yapay Sinir Ağları (Artificial Neural Networks), bilgisayar biliminin ve yapay zeka alanının bir alt dalı olan makine öğrenmesi içinde önemli bir rol oynayan bir yöntemdir. Biyolojik sinir ağlarından esinlenerek tasarlanmış matematiksel bir yöntemdir. İnsan beynindeki sinir hücrelerini (nöronlar) taklit eden yapay nöronlardan oluşur. Bu yapay nöronlar, veri işleme ve öğrenme görevlerini gerçekleştirmek için bir araya getirilir. Yapay sinir ağları, genellikle üç ana katmandan oluşur. Bunlar, giriş katmanı, gizli katmanlar ve çıkış

katmanıdır. Her katman, yapay nöronlardan oluşur ve katmanlar arasındaki bağlantılar ağırlıklarla temsil edilir [24].

Giriş katmanı, dış dünyadan alınan veriler bu katmana giriş yapar. Her giriş, bir yapay nörona bağlanır. Gizli katmanlar, işleme ve öğrenme işlemlerinin çoğu bu katmanlarda gerçekleşir. Bir yapay sinir ağı genellikle birden fazla gizli katmana sahip olabilir. Bu katmanlar, verileri daha karmaşık özelliklere dönüştürmek için kullanılır. Çıkış katmanı ağın sonucunu üreten katmandır. Örneğin, bir sınıflandırma probleminde, çıkış katmanı farklı sınıflara ait olasılıkları verebilir.

2.6. Literatür Taraması

Öznitelik tabanlı oltalama tespit sistemlerinin makine öğrenmesi yöntemleri ile değerlendirilmesi konusunda birçok çalışma yapılmıştır ve farklı yönleri ele alınmıştır. Bu kapsamda incelenen ve yararlanılan kaynaklardan çıkarılan çalışmalar üç gruba ayrılmıştır. Literatürde yapılan çalışmalara ait gruplama Şekil 2.1'de gösterilmiştir. Bunlar benzerlik tabanlı yapılan çalışmalar, kural tabanlı yapılan çalışmalar ve makine öğrenmesiyle yapılan çalışmalardır.



Şekil 2.1: Literatür Taraması için Yapılan Çalışmalara Ait Gruplama

2.6.1. Benzerlik Tabanlı Yapılan Çalışmalar

Liu ve diğ. [25], sahte web sitelerinin tespiti için DynaPhish adında bir yöntem geliştirmişlerdir. DynaPhish, çalışma süresi boyunca dinamik bir şekilde web sitelerini tarar, aktif olan yasal web sitelerini tespit edip bunları referans listesine ekler. Levenshtein benzerlik yöntemini kullanarak inandırıcılığı olmayan markasız web sitelerinde algılanmasını sağlayarak sahte web sitelerin tespitini sağlar. Sahte tespit ettiği web sitelerini kendi oluşturduğu DynaPD (Dynamic Phishing Dataset) veri setine kaydeder. DynaPhish, kendi oluşturduğu DynaPD veri setinde toplam 6344 adet oltalama web sitesi ile Alexa dan elde ettiği 6309 adet yasal web sitesini karşılaştırır. Karşılaştırma sonucunda DynaPhish yöntemi ile sahte web site tespitinde 9 kat daha etkili olduklarını göstermişlerdir.

Zouina ve Outtaj [26], 1000 tane yasal 1000 tane sahte web sitesi olmak üzere toplam 2000 web sitesi üzerinde SVM ve Levenshtein benzerlik algoritmasını uygulayarak sahte web sitelerini tespit etmeye çalışan bir model önermiştir. Önerdiği modelde 6 tane özellik bilgisi kullanmıştır. Bunlar; URL boyutu, tire sayısı, nokta sayısı, sayısal karakter sayısı, URL'de bir IP adresinin varlığına karşılık gelen ayrı bir değişken ve Levenshtein benzerlik indeksidir. Model sonucunda %95,80 oranında bir başarı elde etmiştir.

Wu ve diğ. [27], sahte web sitelerini benzerlik üzerinden tespit etmek için url bilgilerine odaklanan bir yöntem geliştirmişlerdir. Web sayfasının içerdiği url bilgisi ile web sayfasının kaynak kodundaki url bilgisi arasındaki benzerliği tespit etmek için Levenshtein mesafesi yöntemini uyguladılar. Uyguladıkları yöntemde %92,60 oranında bir başarı elde edilmiştir.

Liu ve diğ. [28], belirli bir alan adı içinde en benzer popüler sahte web sitelerini hızlı ve doğru bir şekilde elde etmek için geriye doğru arama yaklaşımı olan yazım hatası (TypoPegging) yöntemini geliştirmişlerdir. Geliştirilen yöntem bir url sitesinin görsel özelliklerini alan genelleştirilmiş Levenshtein mesafe yöntemine dayanmaktadır. Belirli bir web sitesi seti üzerinde bir alanın görsel benzerliğini maksimum ölçmek için üçgen eşitsizliği ve hash algoritması kullanarak arama sürecini hızlandırmışlardır. Yöntem sonucunda yazım hatası olan alan adlarını normal adlardan ayırmada oldukça etkili olduğunu göstermişlerdir.

2.6.2. Kural Tabanlı Yapılan Çalışmalar

SatheeshKumar ve diğ. [29] bir e-postada ve herhangi bir web sayfası bölümünde geçerli olan bir urlnin urlsini, etki alanını, içeriğini ve sayfa özelliklerini inceleyerek gerçek

zamanlı oltalama saldırısını tespit etmek için Chrome uzantısında bir saldırı tespit sistemi geliştirmiştir. Bilinmeyen oltalama URL'lerini tespit etmek için hafif ve proaktif kural tabanlı artımlı bir yapı yaklaşımı tanımlanmıştır. Geliştirdikleri sistem %97,13 oranında oltalama sitelerini tespit etmede başarılı olmuştur.

Fazliya ve Naleer [30], bir web sitesinin oltalama veya yasal veri seti olduğunu bulmak için web sitesine etkisi, url uzunluğu gibi özelliklere dayalı kural tabanlı bir yaklaşım önermişlerdir. Veri setine J48 Algoritması, PART Algoritması, Karar Ağacı ve JRip sınıflandırma algoritmaları uygulamışlardır. Uygulama performansları, doğrulukları ve oluşturulan kural sayıları açısından karşılaştırmalı olarak incelenmiştir. Karşılaştırma sonucunda, sınıflandırma algoritmalarından oluşturulan kurallar %96,76 oranında doğruluk değer elde etmişlerdir.

Abdelhamid ve diğ. [31], tarafından sahte web sitelerini tespit etmek için çok etiketli ilişkisel sınıflandırmaya dayalı kural tabanlı bir yöntem geliştirilmiştir. MCAC (Multi-label Classifier based Associative Classification) adını verdikleri gelişmiş bir AC (Associative Classification) yöntemini kullanarak bir web sitesinin yasal veya oltalama web sitesi olup olmadığını araştırmışlardır. MCAC ile diğer algoritmaların bulamadığı yeni kurallar üretmişler. URL bilgilerini 16 özelliğe göre sahte, yasal ve şüpheli diye üç kategoriye ayırmışlardır. MCAC yönteminin, oltalama web sitelerini diğer algoritmalara göre daha yüksek doğrulukta tespit ettiğini göstermişlerdir.

Mourtaji ve diğ. [32], oltalama web sitelerini tespit etmek için 37 farklı özniteliği içeren altı ayrı yöntemden elde edilen verileri kapsayan bir kural tabanlı yöntem geliştirmişlerdir. Yöntemleri arasında kara liste, sözlüksel ve ana bilgisayar analizi, içerik değerlendirmesi, özdeşlik incelemesi, kimlik benzerliği değerlendirmesi, görsel benzerlik analizi ve davranışsal analiz gibi teknikler bulunmaktadır. Yöntem sonucunda %97,94 oranında CNN algoritması için başarım elde etmişlerdir.

2.6.3. Makine Öğrenmesiyle Yapılan Çalışmalar

Ahmedi ve diğ. [33], 400 tane sahte web sitesi ve 400 tane de yasal web sitesi olmak üzere toplam 800 web sitesi üzerinde sahte web sayfalarını tespit etmek ve kullanıcıları oltalama saldırılarından korumak amacıyla makine öğrenimi tekniklerine dayalı PhishCatcher olarak adlandırılan bir Google Chrome uzantısı geliştirmişlerdir. Girdi olarak url parametrelerini, web sayfası içeriğini, kara liste bilgisi ve sahte web sitesine ait sitenin yasal

web sitesi ile olan görsel benzerliği olmak üzere dört farklı türdeki web özelliğini alırlar. Makine öğrenmesi yöntemlerinden Naive Bayes, SVM ve Lineer Regresyon yöntemlerini kullanarak %98,50 oranında SVM yönteminde başarı elde etmişlerdir.

Alhogail ve diğ. [34], oltalama email saldırılarını tespit etmek için oltalama saldırısının başarısına veya başarısızlığına katkıda bulunan faktörleri belirlemeyi amaçlayan url bilgisine, web trafiğine ve web içeriğine odaklana üç aşamalı bir yöntem önermişlerdir. Oltalama saldırısı tespitini üç farklı sınıflandırıcı makine öğrenme yöntemleri kullanarak değerlendirmişlerdir. Sinir Ağı (NN), Destek Vektör Makinesi (SVM) ve Rastgele Orman (RF) yöntemlerinden NN sınıflandırıcı %95,18, SVM %85,45, RF ise %78,89 doğruluk değerlerini elde etmişlerdir.

Butt ve diğ. [35], e-posta gönderirken ve alırken gerçekleşen oltalama saldırılarını tespit etmek üzere bir çalışma yapmışlardır. Mevcut yaklaşımları inceleyip değiştirilmiş virgülle ayrılmış değerler (csv) dosyası ve etiket dosyasından oluşan bir veri seti oluşturdular. E-posta oltalama saldırılarını sınıflandırmak için Destek Vektör Makinesi (SVM), Saf Bayes (NB) ve Uzun Kısa Süreli Bellek (LSTM) yöntemlerini kullanmışlar ve sırasıyla %99,62, %97 ve %98 doğruluk değerlerini elde etmişlerdir.

Rao ve Pais [36], oltalama web sitelerini tespit etmek için url bilgisinden, kaynak kodundan ve üçüncü taraf hizmetlerden çıkarılan özelliklere dayalı yeni bir sınıflandırma modeli geliştirmişlerdir. Geliştirilen model üzerinde, sekiz farklı makine öğrenimi yöntemi kullanarak değerlendirdiler ve Rastgele Orman (RF) yönteminde %99,31 doğruluk oranı ile en iyi performans değerini elde etmişlerdir. Ayrıca geliştirdikleri modelin sıfırıncı gün kimlik avı saldırılarını tespit ettiğini göstermişlerdir.

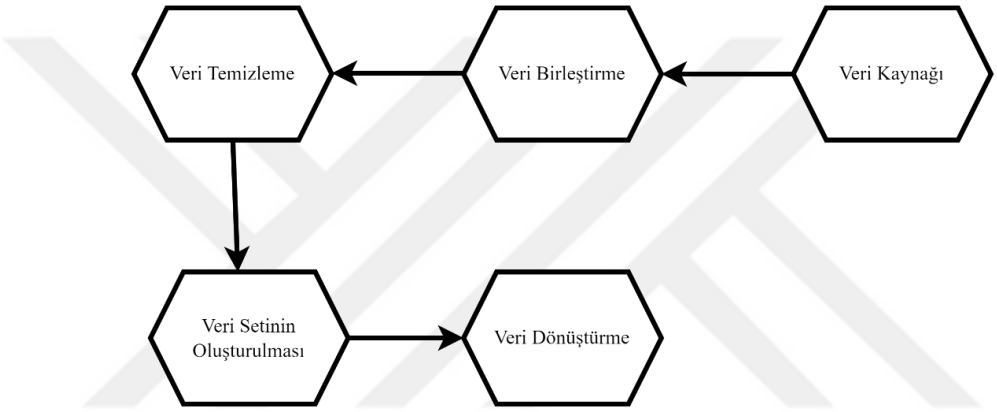
Karim ve diğ. [37], oltalama saldırılarını önlemek ve kullanıcıları korumak için Kaggle veri setini kullanarak url bilgilerinden öznitelikler çıkararak bir yöntem önermişlerdir. Makine öğrenimi yöntemlerinden Karar Ağacı, Lineer Regresyon, Rastgele Orman, Saf Bayes, Gradyan Artırma Sınıflandırıcısı, K-en yakın komşu ve Destek Vektör Makinesi sınıflandırma yöntemlerini kullanarak Rastgele Orman yönteminden %98,12 doğruluk oranı elde etmişlerdir. Ayrıca, Lojistik Regresyon, Destek Vektör Makinesi ve Karar ağacının bir kombinasyonu olan hibrit bir LSD modeli kullanılarak oltalama saldırılarına karşı %95,23 doğruluk oranı elde etmişlerdir.

Choudhary ve diğ. [38], makine öğrenme yöntemlerini kullanarak sahte url bilgileri ile yasal url bilgileri arasında ayrım yaparak oltalama web sitelerini tespit etmek üzerine bir araştırma yapmışlardır. PhishTank ve UCI den elde edilen veri setlerinden 25 tane öznitelik bilgisi çıkarmışlardır. Yasal ile oltalama url bilgilerini ayırt etmek amacıyla Extreme Gradient Boosting, Karar Ağacı, Lojistik Regresyon, Rastgele Orman ve Destek Vektör Makinesi yöntemleri ile sınıflandırıp Rastgele Orman yöntemi için PhishTank veri seti üzerinde %98,80 ve UCI veri seti üzerinde %97,87 doğruluk değerlerini elde etmişlerdir.

Islam ve Chowdhury [39], 11055 web sitesinden çıkarılan 31 öznitelik üzerinde oltalama web sitelerini tespit etmek için makine öğrenmesi yöntemlerinden Naive Bayes, Destek Vektör Makinesi, Sinir Ağı, Rastgele Orman, K-en yakın komşular ve Karar Ağacı sınıflandırma algoritmalarını kullanarak performanslarını karşılaştırma üzerine bir çalışma yapmışlardır. Yapmış oldukları çalışma kapsamında Naive Bayes, Destek Vektör Makinesi, Sinir Ağı, Rastgele Orman, K-en yakın komşular ve Karar Ağacı sınıflandırma algoritmaları için sırasıyla %93,09, %94,84, %96,65, %97,47, %97,07 ve %95,61 doğruluk değerlerini elde etmişlerdir.

3. YÖNTEM

Oltalama saldırısı yapılan alan adlarının tespit edilmesi için kullanılan veri kaynakları, veri birleştirme, veri temizleme, veri setinin oluşturulması, oluşturulan veri setinden yeni veri setlerinin elde edilmesi ve veri setlerinin dönüştürülmesine ait işlemler açıklanmıştır. Bu kapsamda Şekil 3.1'de modelime ait veri hazırlama aşamaları verilmiş ve bu aşamalar ilerleyen bölümlerde açıklanmıştır. Ayrıca tez modelimin geliştirilmesi, modelim kapsamında oltalama saldırılarını belirlemek için çıkardığım öznitelikler, karışıklık matrisinden ve çaprazlama doğrulama yöntemleri detaylıca açıklanmıştır.



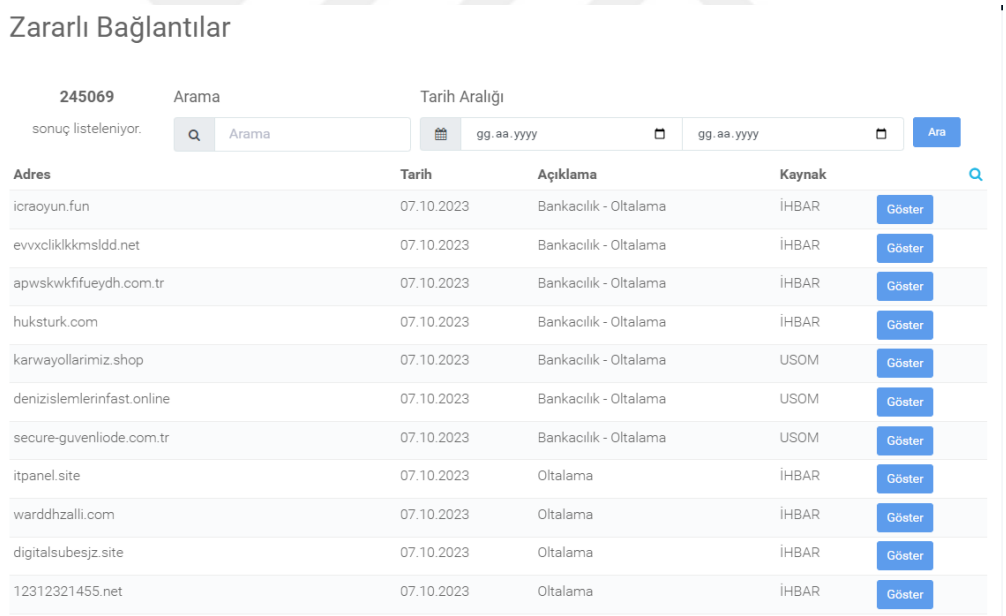
Şekil 3.1: Modelime Ait Veri Hazırlama Aşamaları

3.1. Veri Kaynağı

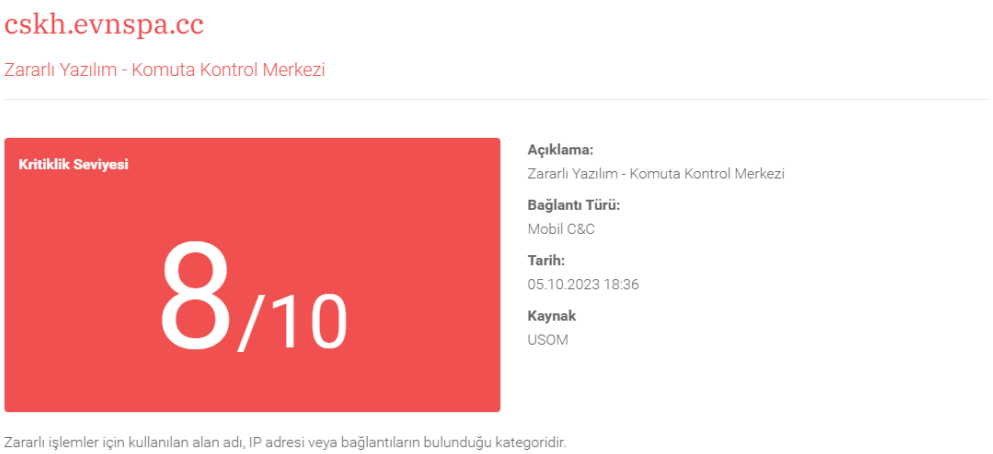
Bu bölümde sahte web siteleri ile yasal web siteleri kullanılarak oltalama web sitelerini analiz etmek için verilerin nasıl toplandığı açıklanmıştır. Sahte web siteleri Bilgi Teknolojileri ve İletişim Kurumu (BTK) bünyesinde 27/05/2013 tarihinde kurulan Ulusal Siber Olaylara Müdahale Merkezindeki (USOM, TR-CERT) zararlı bağlantılar sayfasından alınmıştır [40]. Yasal web siteleri ise Türkiye de ve dünya da faaliyet gösteren en az ile en çok ziyaret edilen kamu, üniversite, e-ticaret, banka, enerji, ulaşım vb. gibi internet siteleri Google ve Yahoo arama motorlarında manuel gezilerek elde edilmiştir. Örneğin; Türkiye de faaliyet gösteren banklara ait yasal web siteleri Türkiye Bankalar Birliği'ndeki Banka ve Şube Sayıları alanından [41], Türkiye de bulunan üniversitelere ait yasal web siteleri Yükseköğretim Kurulu'ndaki tüm üniversiteler alanından [42], Türkiye de faaliyet gösteren tüm kamu kurumları, belediyelere ait yasal web siteleri de e-devlette (türkiye.gov.tr) bulunan e-hizmetler, kurumlar, belediyeler ve firmalar alanından [43], Dünya da faaliyet gösteren

yasal web siteleri Forbes sayfasında bulunan Forbes World's Billionaires List alanındaki [44] şirketlerin web siteleri gezilerek elde edilmiştir. Böylelikle yasal web sitelerinde az bilinen ve potansiyel olarak oltalama saldırısı olabilecek web siteleri de çalışmaya dahil edilmiştir. Toplam sahte ve yasal web sitesi olmak üzere 9362 web sitesi elde edilmiştir. Bu web sitelerinden 2533 tanesi yasal site ve 6829 tanesi de oltalama sitesidir.

USOM Zararlı Bağlantılar sayfasında; zararlı web sitesine ait Adres bilgisi, ne zaman tespit edildiğine ait Tarih bilgisi, tespit edilen saldırı türüne ait Açıklama bilgisi ve sahte web sitesinin kim tarafından tespit edildiğine ait Kaynak bilgisi alanları bulunmaktadır. USOM Zararlı Bağlantılar sayfası Şekil 3.2'de verilmiştir. Tespit edilen zararlı bağlantı bilgisinin Göster alanına tıklayarak; zararlı bağlantıya ait Açıklama, Bağlantı Türü, Tarih, Kaynak ve Kritiklik Seviyesi bilgilerini görebiliriz. USOM da zararlı bağlantılar genellikle alan adları veya IP bilgilerinden oluşmaktadır. USOM Zararlı Bağlantılar sayfasında bulunan "cskh.evnspa.cc" sahte web sitesine ait Göster alanı detay bilgisi Şekil 3.3'te gösterilmiştir.

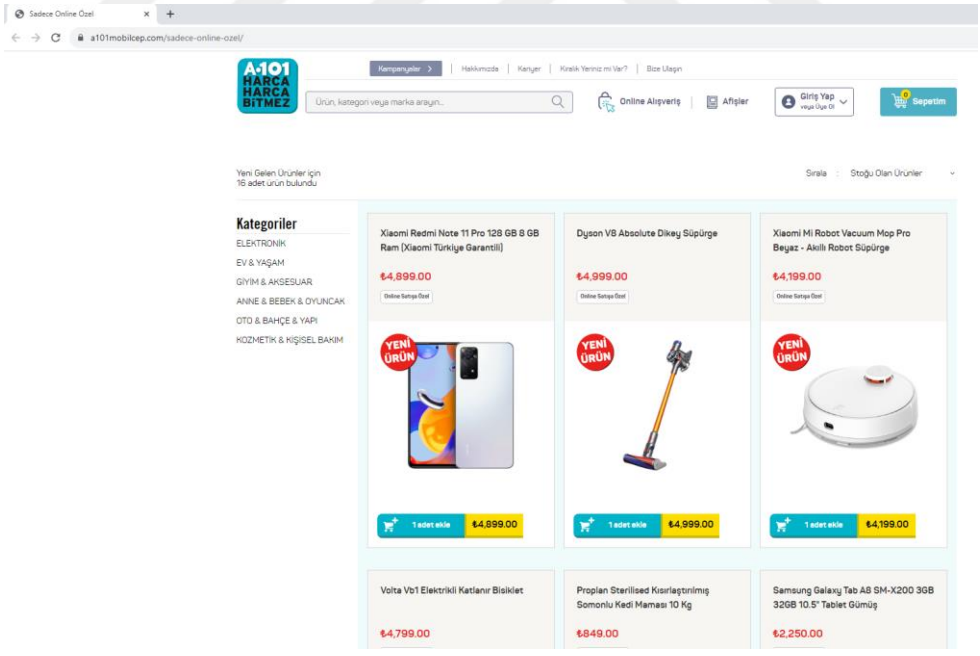


Şekil 3.2: USOM Zararlı Bağlantılar Sayfası Örneği [45]

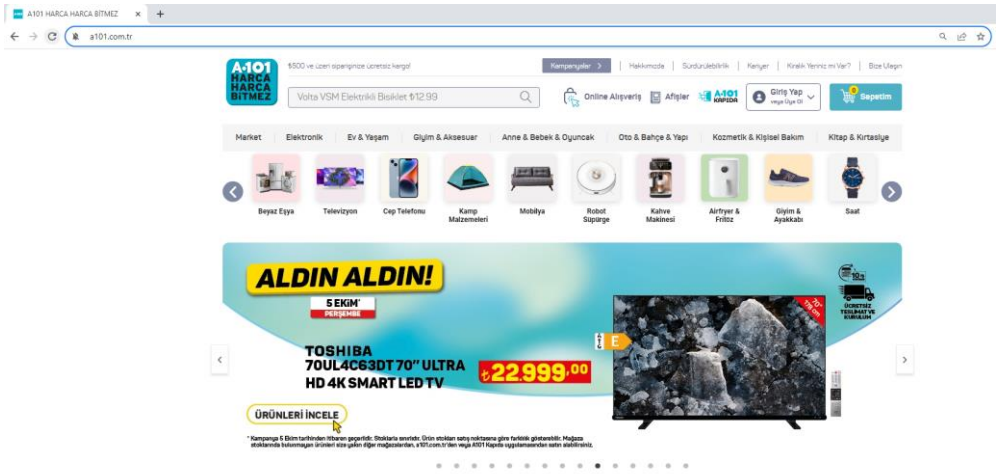


Şekil 3.3: USOM Örnek Bir Zararlı Bağlantı Bilgisine Ait Göster Sayfası Detayı [46]

USOM Zararlı Bağlantılar sayfasında bulunan örnek bir "a101mobilcep.com" zararlı bağlantısına ait web sitesinin içeriği Şekil 3.4 ile o web sitesine yakın "a101.com.tr" yasal web sitesinin içeriğini aşağıdaki Şekil 3.5'te görülmektedir.



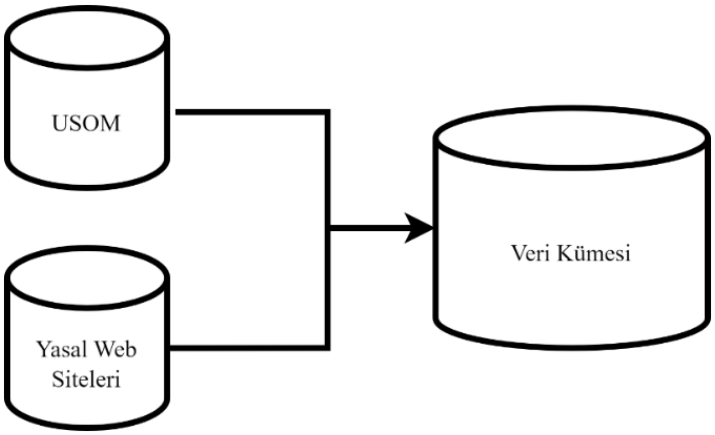
Şekil 3.4: a101mobilcep.com Sahte Web Sitesinin İçeriği [47]



Şekil 3.5: a101.com.tr Yasal Web Sitesinin İçeriği [48]

3.2. Veri Birleştirme

Bu adımda USOM dan elde edilen zararlı web siteleri ile Google ve Yahoo arama motorlarından manuel olarak aratılarak elde edilen Türkiye ve Dünya da bulunan kamu, üniversite, e-ticaret, banka vs. gibi yasal siteler bir araya getirilerek veri kümesi elde edilir. Oltalama web siteleri ile yasal web siteleri bir araya getirilir ve yeni bir veri kümesi oluşturulur. Veri kümesi oluşumu grafiği Şekil 3.6'da verilmiştir.



Şekil 3.6: Veri Kümesi Birleştirme Adımı

3.3. Veri Temizleme

Bu adımda tez çalışmasına dahil edilen web sitelerinin url bilgilerine ön ek (prefix) ve son ek (suffix) kaldırma işlemleri uygulanır. USOM da bulunan zararlı bağlantı adres bilgileri kontrol edildiği zaman “http://”, “https://”, “www.”, “/XXX” vs. gibi url bilgilerinde ön ek ve son ek bilgilerinin olmadığını görürüz. Elde edilen yasal url bilgilerinde bulunan “https://www.”, “http://www.”, “http://”, https://, “www.”, “/”, “/XXX” vs. gibi ön ek ve son ek alanları kaldırılır. Elde ettiğim örnek bir yasal url bilgisi ile yasal url bilgilerine uyguladığımız ön ek ve son ek işlemlerini açıklayalım. “https://www.a101.com.tr/aldin-aldin-gelecek-hafta-brosuru/” gibi bir url bilgisinde öncelikle urlnin başındaki “https://www” kısmını çıkarırız. Bu işlem sonucunda url şu şekilde olur: “a101.com.tr/aldin-aldin-gelecek-hafta-brosuru/”. Ardından, urlnin sonundaki “/aldin-aldin-gelecek-hafta-brosuru/” kısmını çıkarırız. Bu işlem sonucunda url bilgisi şu hale gelir: “a101.com.tr”. Böylece veri kümesinde bulunan sahte ve yasal url bilgileri aynı düzende olup sonuç bilgilerinin ve benzerliklerinin artması hedeflenmiştir.

3.3.1. Veri Setinin Oluşturulması

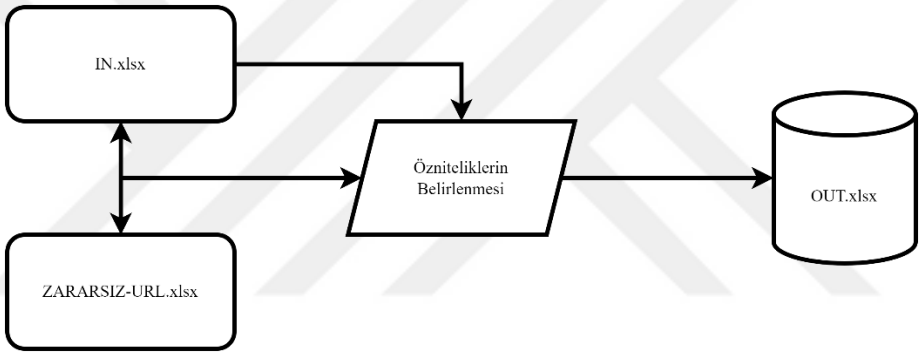
Veri kümesinden oltalama saldırılarının analizini ve tespitini yapmak için programlama dili olarak Python 3.8.0 versiyonu [49] seçilmiştir. Python programlama dili makine öğrenmesi yöntemleri ile değerlendirmede yardımcı olması için hazırlanan kütüphanelerden Numpy, Pandas ve Scikit-Learn kütüphaneleriyle birlikte bu kütüphanelere ait fonksiyonlar bu çalışmada kullanılmıştır. Tez modelimin tasarımı için Anaconda Navigator uygulamasında [50] bulunan Spyder 5.1.5 versiyonu [51] geliştirme ortamı kullanılmıştır. Tez modelimin eğitiminin ve testi için kullanılan makineye ait özellikler Tablo 3.1 de verilmiştir.

Tablo 3.1: Modelimde Kullanılan Makineye Ait Bilgiler

Makine Bilgileri	
Bellek Kapasitesi	512 GB SSD
İşlemci	11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
RAM Kapasitesi	16 GB
İşletim Sistemi	Windows 11 Pro

Veri birleştirme sonrası oluşan veri kümesi, sahte web siteleri ile yasal web siteleri, IN.xlsx excel dosyasında bulunur ve program bu dosya bilgisini girdi olarak alır. Tamamen yasal web sitelerinden oluşan 2013 adet web sitesi bilgisine sahip ZARARSIZ-URL.xlsx

excel dosyası da programa girdi olarak verilir. Böylece IN.xlsx dosyasında bulunan web siteleri ile ZARARSIZ-URL.xlsx excel dosyasında bulunan web siteleri karşılaştırılıp benzerlik algoritmalarına (karakter benzerlik algoritması, kosinüs benzerlik algoritması ve levenshtein benzerlik algoritması) göre benzerlik değeri, hangi web sitesi ile eşleştiği ve orijinal domaine sahip olup olmadığının bilgileri diğer öznitelik bilgileriyle birlikte OUT.xlsx excel dosyasına yazılır. OUT.xlsx excel dosyası programın çıktı dosyasıdır ve aynı zamanda veri setimizdir. Yani üzerinde makine öğrenmesi yöntemlerinden Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları yöntemlerinin uygulanıp analiz edilip ve değerlendirilecek veri setimizdir. OUT.xlsx dosyasının veri setinden oluşturulması adımları Şekil 3.7'de verilmiştir. Program kodlanması titiz bir şekilde yapıldığı için OUT.xlsx excel dosyasında boş değerler bulunmamaktadır.



Şekil 3.7: Veri Setinin Oluşturulması Gösterimi

3.4. Veri Dönüştürme

Elde edilen veri kümesinde bulunan url bilgilerini daha anlaşılır ve kullanılabilir hale getirmek, makine öğrenimi yöntemlerinden Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemlerinin daha doğru ve etkili olmasını sağlamak için veri kümesinden 32 adet öznitelik bilgisi oluşturulmuştur. Oluşturulan yeni veri kümesi (sahte web siteleri + yasal web siteleri) üzerinde özellik çıkarımı teknikleri kullanılarak Tablo 3.2'deki öznitelikler elde edilmiştir. Tablo 3.2'de modelimde kullanılan her bir öznitelik bilgisinin adı, öznitelik bilgisinin veri tipi ve öznitelik bilgisinin açıklamaları verilmiştir. Çıkarılan her bir öznitelik veri setinin bir sütun bilgisine ve veri setinde bulunan her web sitesi de bir satıra denk gelmektedir. Veri kümesinde bulunan örnek "sahibindenpremium.com" sahte web sitesine ait OUT.xlsx dosyasında özniteliklere göre değerleri Tablo 3.3'te verilmiştir.

Tablo 3.2: Veri Kümesinden Hazırlanan Özniteliklerin Adı, Veri Tipleri ve Açıklamaları

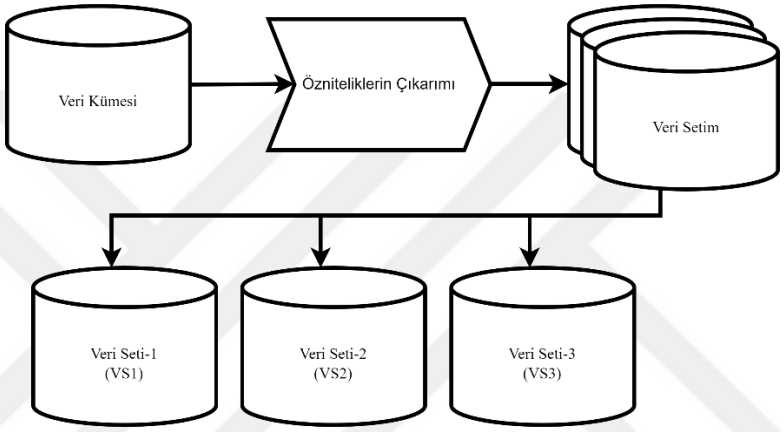
Öznitelik Bilgisi Adı	Veri Tipi	Öznitelik Açıklaması
USOM_da_mevcut_mu	Sayısal	Bir web sitesine ait url bilgisinin USOM yer alan zararlı bağlantılar sayfasında olup olmadığını belirlemek için kullanılır. Eğer varsa 1 değerini yoksa 0 değerini alır.
SSL_mevcut_mu	Sayısal	Bir web sitesine ait url bilgisinin SSL protokolü olup olmadığını belirlemek için kullanılır. Eğer varsa 1 değerini yoksa 0 değerini alır.
Kosinus_benzerligi_orani	Sayısal	İki url bilgisinin bu benzerlik ölçüsüne göre ne kadar benzer olduğu hesaplanır, 0 ile 1 arasında değer alır.
Kosinus_benzerligi_url_bilgisi	Metin	Bir web sitesine ait url bilgisinin kosinüs benzerliğine göre hesaplanıp hangi yasal site ile benzerlik gösterdiğini belirtir.
Karakter_benzerligi_orani	Sayısal	İki url bilgisinin bu benzerlik ölçüsüne göre ne kadar benzer olduğu hesaplanır, 0 ile 1 arasında değer alır.
Karakter_benzerligi_url_bilgisi	Metin	Bir web sitesine ait url bilgisinin karakter benzerliğine göre hesaplanıp hangi yasal site ile benzerlik gösterdiğini belirtir.
Levenshtein_benzerligi_orani	Sayısal	İki url bilgisinin bu benzerlik ölçüsüne göre ne kadar benzer olduğu hesaplanır.
Levenshtein_benzerligi_url_bilgisi	Metin	Bir web sitesine ait url bilgisinin levenshtein benzerliğine göre hesaplanıp hangi yasal site ile benzerlik gösterdiğini belirtir.
Orijinal_domain_bilgisi	Sayısal	Bir web sitesine ai url bilgisinin gerçek ve doğrulanmış bir kuruluşa, şirkete veya kaynağa ait olup olmadığını belirlemek için kullanılır. Eğer aitse 1 değerini ait değise 0 değerini alır.
Toplam_nokta_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam nokta sayısını belirtir.
Toplam_yatik_cizgi_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam yatık çizgi sayısını belirtir.
Toplam_cift_yatik_cizgi_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam çift yatık çizgi sayısını belirtir.
Toplam_alt_cizgi_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam alt çizgi sayısını belirtir.
Toplam_tire_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam tire işareti sayısını belirtir.
Toplam_esittir_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam eşittir işareti sayısını belirtir.
Toplam_kucuktur_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam küçüktür işareti sayısını belirtir.
Toplam_buyuktur_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam büyüktür işareti sayısını belirtir.
Toplam_tilde_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam tilde işareti sayısını belirtir.
Toplam_yildiz_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam yıldız işareti sayısını belirtir.
Toplam_arti_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam artı işareti sayısını belirtir.
Toplam_url_uzunlugu_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam url uzunluğu sayısını belirtir.
Toplam_com_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam com sayısını belirtir.
Toplam_nokta_com_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam com sayısını belirtir.
Toplam_unlem_isareti_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan toplam ünlem işareti sayısını belirtir.
Toplam_sesli_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "aeıioöuü" harflerin toplam sayısını belirtir.
Toplam_kalin_sesli_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "aıou" toplam harflerin sayısını belirtir.
Toplam_ince_sesli_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "eiöü" toplam harflerin sayısını belirtir.
Toplam_sessiz_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "bcçdfgğhjklmnprsştvyz" toplam harflerin sayısını belirtir.
Toplam_sert_unsuz_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "çfhkpsşt" toplam harflerin sayısını belirtir.
Toplam_yumusak_unsuz_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "bcdgğjlmnrvyz" toplam harflerin sayısını belirtir.
Toplam_urlde_gecen_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "AaBbCdÇçDdEeFfGgĞğHhİiIıJjKkLlMmNnOoÖöPpRrSsŞşTtUuÜüVvYyZz" toplam harflerin sayısını belirtir.
Toplam_urlde_gecen_ingilizce_harflerin_sayisi	Sayısal	Bir web sitesine ait url bilgisinde bulunan "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz" toplam harflerin sayısını belirtir.

Tablo 3.3: Veri Kümemde Bulunan "sahibindenpremium.com" Sitesine Ait Öznitelik Bilgilerinin OUT.xlsx Dosyasındaki Değeri

URL	sahibindenpremium.com
USOM_da_mevcut_mu	1
SSL_mevcut_mu	0
Kosinus_benzerligi_orani	0.5
Kosinus_benzerligi_url_bilgisi	sahibinden.com
Karakter_benzerligi_orani	0.8
Karakter_benzerligi_url_bilgisi	sahibinden.com
Levenshtein_benzerligi_orani	7
Levenshtein_benzerligi_url_bilgisi	sahibinden.com
Orijinal_domain_bilgisi	0
Toplam_nokta_sayisi	1
Toplam_yatik_cizgi_sayisi	0
Toplam_cift_yatik_cizgi_sayisi	0
Toplam_alt_cizgi_sayisi	0
Toplam_tire_isareti_sayisi	0
Toplam_esittir_isareti_sayisi	0
Toplam_kucuktur_isareti_sayisi	0
Toplam_buyuktur_isareti_sayisi	0
Toplam_tilde_isareti_sayisi	0
Toplam_yildiz_isareti_sayisi	0
Toplam_arti_isareti_sayisi	0
Toplam_url_uzunlugu_sayisi	21
Toplam_com_sayisi	1
Toplam_nokta_com_sayisi	1
Toplam_unlem_isareti_sayisi	0
Toplam_sesli_harflerin_sayisi	8
Toplam_kalin_sesli_harflerin_sayisi	3
Toplam_ince_sesli_harflerin_sayisi	5
Toplam_sessiz_harflerin_sayisi	12
Toplam_sert_unsuz_harflerin_sayisi	3
Toplam_yumusak_unsuz_harflerin_sayisi	9
Toplam_urlde_gecen_harflerin_sayisi	20
Toplam_urlde_gecen_ingilizce_harflerin_sayisi	20

Veri kümesinde bulunan her bir url bilgisi satırına karşılık çıkarılan öznitelik sütun değerlerine göre elde edilen çıktı, bizim veri setimiz, makine öğrenmesi yöntemlerinde temel alacağımız veri setimizdir. Bu veri setinden 3 farklı veri seti bilgisi elde edilmiştir. Veri

setinden üç farklı veri seti bilgisinin elde edilme gösterimi Şekil 3.8'de verilmiştir. Elde edilen bu üç veri setinden USOM_da_mevcut_mu öznitelik bilgisi çıkarılmıştır. USOM_da_mevcut_mu öznitelik bilgisinin çıkarılma nedeni bir web sitesi eğer USOM da mevcut ise 1 değeri alır ve bu bir oltalama sitesi demektir. Oltalama saldırısı tespit sonucuna doğrudan etki etmemesi için ve saldırı olup olmadığını makine öğrenme yöntemleri ile sınıflandırmak için veri setlerinden çıkarılmıştır. Tablo 3.4'te veri setinden üç farklı veri setinin oluşturulmasında yapılan işlem bilgileri, kullanılan öznitelik sayıları, veri seti adları ve tez boyunca kullanılacak kısaltma bilgileri verilmiştir.



Şekil 3.8: Veri Setimden 3 Farklı Veri Seti Oluşturma

Tablo 3.4: Veri Setimden Elde Edilen 3 Veri Setlerine Ait Bilgiler

Veri Seti Adı	Kısaltması	Yapılan İşlem	Öznitelik Sayısı
Veri Seti-1	VS1	Elde edilen veri setinden USOM_da_mevcut_mu, Kosinus_benzerligi_url_bilgisi, Karakter_benzerligi_url_bilgisi ve Levenshtein_benzerligi_url_bilgisi öznitelik bilgileri çıkarılarak elde edilmiştir.	28
Veri Seti-2	VS2	Elde edilen veri setinden USOM_da_mevcut_mu öznitelik sütun alanı çıkarılmış ve Kosinus_benzerligi_url_bilgisi, Karakter_benzerligi_url_bilgisi ve Levenshtein_benzerligi_url_bilgisi özniteliklerine kategorik değerden nümerik değere geçiş cat.codes işlemi uygulanmıştır.	31
Veri Seti-3	VS3	VS2 üzerinde 0 ile 1 arasında olacak şekilde SSL_mevcut_mu, Kosinus_benzerligi_orani, Karakter_benzerligi_orani ve Orijinal_domain_bilgisi öznitelikleri hariç diğer tüm öznitelik alanlarına normalizayon işlemi yapılmıştır.	31

Öznitelikler sonucunda oluşan VS1, VS2 ve VS3 veri setleri üzerinde makine öğrenmesi yöntemlerinden olan Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek

Vektör Makinesi ve Yapay Sinir Ağları yöntemleri Anaconda Navigator uygulamasında bulunan Jupyter Notebook 6.4.5 versiyonunda [52] uygulanır ve model eğitilir. Eğitilen model için Karışıklık Matrisi bilgileri bulunur ve sonuçlar değerlendirilir. Bu sonuçlara göre modelde bulunan url bilgisinin saldırı olup olmadığı değerlendirilir.

3.4.1. Kategorik Değerden Sayısal Değere Dönüşüm İşleminin Uygulanması

Bu çalışmada kategorik veri değerlerini sayısal olarak temsil etmek için Python da bulunan Pandas kütüphanesindeki cat.codes özelliği kullanılmıştır. VS2 de bulunan verilerimize cat.codes özelliği sayesinde her farklı kategori değeri için bir sayı değeri atanıp veri setimize ait excel dosyasına kaydedilmiştir. Örneğin; veri setimizde bulunan Kosinus_benzerligi_url_bilgisi sütunu (alanı) farklı kategorik değerlere sahiptir. Bu sütunda bulunan her kategorik değer Pandas kütüphanesindeki DataFrame yöntemi cat.codes özelliği ile kategorik bir sütunu sayısal bir değere dönüştürür. Bu sayede bu sütunu makine öğrenmesi modellerine girebilecek formata getirir.

3.4.2. Normalizasyon İşleminin Uygulanması

Bu çalışmada veri değerlerini 0 ile 1 arasında bir aralığa getirmek için min-max normalizasyonu işlemi uygulanmıştır [53]. Denklem 3.1'de bir değeri 0 ile 1 arasında bir aralığa getirmek için kullanılan normalizasyon denklemi verilmiştir.

$$\mathbf{X'} = \frac{\mathbf{X - Xmin}}{\mathbf{Xmax - Xmin}} \tag{3.1}$$

Burada X orijinal değeri, X' normalleştirilmiş değeri, Xmin minimum değeri ve Xmax maksimum değeri temsil eder. VS3 de bulunan verilerimize normalizasyon işlemi yapmak için Python da bulunan Scikit-Learn kütüphanesinden MinMaxScaler() fonksiyonunu kullanarak min-max normalizasyonu uygulanıp daha sonra Pandas kütüphanesiyle veri setimize ait excel dosyasına kaydedilmiştir.

3.5. Karışıklık Matrisi (CM - Confussion Matrix)

Karışıklık matrisi (confusion matrix), makine öğrenmesi sınıflandırma algoritmalarının performansını değerlendirmek için kullanılan bir matristir. Bu matris, tahmin edilen sınıflar ile gerçek sınıflar arasındaki ilişkiyi gösterir. Sınıflandırma problemlerinde modelin doğruluğunu değerlendirmek, hataları anlamak ve modelin performansını iyileştirmek için kullanılır. Karışıklık matris modeli Şekil 3.9'da gösterilmiştir.

Karışıklık Matrisi (Confussion Matrix)		Gerçek (Actual) Değerler	
		Pozitif (1)	Negatif (0)
Tahminlenen (Predicted) Değerler	Pozitif (1)	Gerçek Pozitif TP (1,1)	Yanlış Pozitif FP (1,0)
	Negatif (0)	Yanlış Negatif FN (0,1)	Gerçek Negatif TN (0,0)

Şekil 3.9: Karışıklık Matrisi Modeli

Bir karışıklık matrisi genellikle dört ana terim içerir:

True Positive (TP): Gerçek pozitif değerler. Modelin doğru bir şekilde pozitif olarak tahmin ettiği örnek sayısıdır.

True Negative (TN): Gerçek negatif değerler. Modelin doğru bir şekilde negatif olarak tahmin ettiği örnek sayısıdır.

False Positive (FP): Yanlış pozitif değerler. Modelin negatif olması gereken bir örneği pozitif olarak tahmin ettiği örnek sayısıdır.

False Negative (FN): Yanlış negatif değerler. Modelin pozitif olması gereken bir örneği negatif olarak tahmin ettiği örnek sayısıdır.

Hassasiyet (Precision): Hassasiyet, pozitif olarak tahmin edilen örneklerin ne kadarının gerçekten pozitif olduğunu gösterir. Yani, doğru pozitiflerin toplam pozitif tahminlere oranıdır. Denklem 3.2'de hassasiyet değerinin formülü verilmiştir. Hassasiyet, yanlış pozitifleri (gerçek negatiflerin yanlış pozitif olarak tahmin edilmesi) minimize etmeyi amaçlar. Yüksek hassasiyet, yanlış pozitiflerin düşük olduğunu gösterir.

$$\textbf{Hassasiyet} = \frac{\textbf{Gerçek Pozitifler (TP)}}{\textbf{Gerçek Pozitifler (TP)} + \textbf{Yanlış Pozitifler (FP)}} \tag{3.2}$$

Duyarlılık (Recall): Duyarlılık, gerçek pozitiflerin ne kadarının doğru bir şekilde tespit edildiğini gösterir. Yani, doğru pozitiflerin toplam gerçek pozitiflere oranıdır. Denklem 3.3'te duyarlılık değerinin formülü verilmiştir. Duyarlılık, yanlış negatifleri (gerçek pozitiflerin yanlış negatif olarak tahmin edilmesi) minimize etmeyi amaçlar.

$$\textbf{Duyarlılık} = \frac{\textbf{Gerçek Pozitifler (TP)}}{\textbf{Gerçek Pozitifler (TP)+Yanlış Negatifler (FN)}} \quad (3.3)$$

F1-skor (F1-score): F1-skor, hassasiyet ve duyarlılığın harmonik ortalamasını ifade eder. Denklem 3.4'te F1-skor değerinin formülü verilmiştir. Bu skor hem hassasiyeti hem de duyarlılığı dikkate alarak denge sağlar. F1-skor hem yanlış pozitifleri hem de yanlış negatifleri minimize etmeyi amaçlar.

$$\textbf{F1 - skor} = 2\text{x}\left(\frac{\textbf{Hassasiyet x Duyarlılık}}{\textbf{Hassasiyet+ Duyarlılık}}\right) \quad (3.4)$$

Destek (Support): Destek, her bir sınıf için gerçek değerlerin ne kadar olduğunu temsil eder. Bu değer, veri setindeki her sınıfa ait örnek sayısını ifade eder. Daha büyük bir destek değeri, o sınıfın daha fazla örnek içerdiğini gösterir.

Hassasiyet, Duyarlılık, F1-skor ve Destek sınıflandırma modellerinin performansını değerlendirmek için kullanılan önemli ölçümlerdir [54].

3.6. Çapraz Doğrulama (CV - Cross Validation)

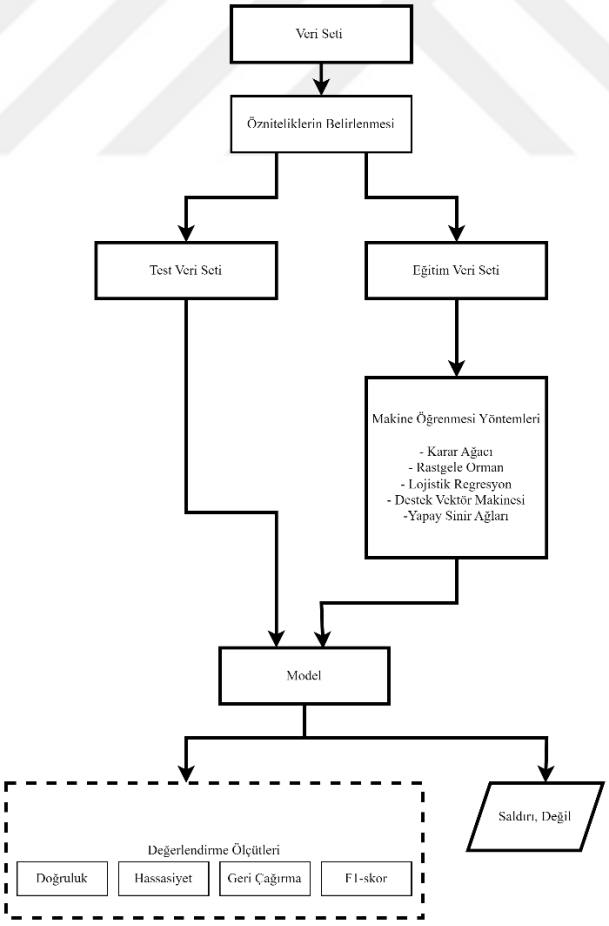
Çapraz doğrulama (cross validation), bir makine öğrenme modelinin performansını değerlendirmek için kullanılan bir yöntemdir. Temel amacı, modelin ne kadar iyi genelleme yaptığını anlamak ve aşırı uydurmayı (overfitting) veya veriye uygunluğu (data fitting) tespit etmektir. Çapraz doğrulama, veriyi eğitim ve test verisi olarak ikiye ayırmak yerine veriyi daha fazla parçaya böler ve modelin her bir parçada nasıl performans gösterdiğini değerlendirir. Bu çalışmada daha önceki çalışmalarda kullanılan çapraz doğrulama değeri 10 olarak seçilmiştir [55]. Böylece Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları yöntemlerinde çapraz doğrulama değeri 10 olarak sınıflandırma işlemleri yapılmıştır.

3.7. Sınıflandırma Yöntemleri Metrikleri

VS1, VS2 ve VS3 veri setlerine uygulanan makine öğrenme yöntemlerinden Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları yöntemlerine ait metrik bilgileri belirlenmiştir. Karar Ağacı yönteminde daha yüksek doğruluk değeri elde etmek için eğitim kümesi yüzde 75 ile test kümesi yüzde 25 olarak ayrılmış, minimum yaprak sayısı 25 ve maksimum derinlik bilgisi 20 olarak seçilmiştir. Rastgele Orman yönteminde daha yüksek doğruluk değeri elde etmek için eğitim kümesi

yüzde 75 ile test kümesi yüzde 25 olarak ayrılmış, minimum yaprak sayısı 25 ve maksimum derinlik bilgisi 20 olarak seçilmiştir. Lojistik Regresyon yönteminde daha yüksek doğruluk değeri elde etmek için eğitim kümesi yüzde 70 ile test kümesi yüzde 30 olarak seçilmiştir. Destek Vektör Matrisi yönteminde daha yüksek doğruluk değeri elde etmek için eğitim kümesi yüzde 70 ile test kümesi yüzde 30 olarak seçilmiştir. Yapay Sinir Ağları yönteminde daha yüksek doğruluk değeri elde etmek için eğitim kümesi yüzde 70 ile test kümesi yüzde 30 olarak ayrılmış ve maksimum iterasyon bilgisi 30 olarak seçilmiştir. Tüm bu seçilen bilgiler VS1, VS2 ve VS3 olmak üzere üç veri setine uygulanmıştır. Şekil 3.10'da modelime ait akış diyagramı verilmiştir.

Üç veri seti için uygulanan yöntemlerin çalışma süreleri saniye (sn) üzerinden hesaplanmıştır. Çalışma süresi, veri setine uygulanan işlemlerin başlangıç zamanından veri setinin eğitilip öğrenme ve test aşamaları sonundaki bitiş süresi arasındaki süreden bulunmaktadır.



Şekil 3.10: Model Akış Diyagramı

4. BULGULAR

Oltalama saldırılarında saldırganlar tarafından kullanılan alan adları tespiti için USOM üzerinde paylaşılan oltalama saldırısı alan adı listesi üzerinden hazırlanan, Veri Seti-1 (VS1), Veri Seti-2 (VS2) ve Veri Seti-3 (VS3) olmak üzere üç veri seti ile eğitilen Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemleri sonucunda elde edilen karışıklık matrisi değerlerine ait sonuçlar tablolar ve grafiklerle verilmiştir. Oltalama amaçlı kullanılan URL adreslerini tespit etmek için veri kümesi üç farklı veri setine ayrılmıştır. Böylece farklı sınıflandırma yöntemleri ile farklı özellik seçme yöntemlerinin birbirleriyle en uygun şekilde nasıl entegre edilebileceğini keşfederek sahte web sitelerini tespit doğruluğunu en üst düzeye çıkarmak amaçlanmıştır. VS1, VS2 ve VS3 veri setleri için seçilen öznitelik bilgisi ile sınıflandırma yöntemlerine bağlı olarak farklı sonuç değerleri elde edilmiştir. Elde edilen değerlerin tablolarda gösteriminde, ondalıklı gösterimi belirtmek için virgül (,) işareti ve virgülden sonra dört basamak bilgisi kullanılmıştır. Çalışma süresi saniye (sn) üzerinden hesaplanmıştır ve ondalıklı gösteriminde virgül işaretinden sonra iki basamak bilgisi kullanılmıştır.

VS1 veri seti üzerinde en iyi doğruluk değeri performansını %96,15 ile Lojistik Regresyon yöntemi göstermiştir. En az doğruluk değeri performansını %95,42 ile Yapay Sinir Ağları yöntemi göstermiştir. VS1 veri setinde 28 adet öznitelik bilgisi için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemlerine göre karışıklık matrisi sonuçları ve çalışma süreleri Tablo 4.1'de verilmiştir.

Tablo 4.1: VS1 Veri Setine ait Yöntemler Bazında Karışıklık Matrisi Sonuçları

	Karar Ağacı	Rastgele Orman	Lojistik Regresyon	Destek Vektör Makinesi	Yapay Sinir Ağları
Doğruluk	0,9602	0,9607	0,9615	0,9601	0,9542
Hassasiyet	0,9493	0,9495	0,9502	0,9501	0,9405
Geri Çağırma	0,9988	1,0000	1,0000	0,9980	1,0000
F1-skor	0,9734	0,9741	0,9745	0,9735	0,9693
TP	1704	1731	2064	2060	1694
FP	91	92	108	108	107
FN	2	0	0	4	0
TN	544	518	637	637	540
Süre	11,72	15,93	9,48	23,68	20,20

VS2 veri seti üzerinde en iyi doğruluk değeri performansını %96,19 ile Yapay Sinir Ağları yöntemi göstermiştir. En az doğruluk değeri performansını %94,90 ile Destek Vektör Makinesi yöntemi göstermiştir. VS2 veri setinde 31 adet öznitelik bilgisi için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemlerine göre karışıklık matrisi sonuçları ve çalışma süreleri Tablo 4.2'de verilmiştir.

Tablo 4.2: VS2 Veri Setine ait Yöntemler Bazında Karışıklık Matrisi Sonuçları

	Karar Ağacı	Rastgele Orman	Lojistik Regresyon	Destek Vektör Makinesi	Yapay Sinir Ağları
Doğruluk	0,9581	0,9611	0,9597	0,9490	0,9619
Hassasiyet	0,9452	0,9486	0,9497	0,9494	0,9537
Geri Çağırma	1,0000	1,0000	0,9980	0,9830	0,9965
F1-skor	0,9718	0,9736	0,9733	0,9659	0,9746
TP	1692	1681	2060	2029	1712
FP	98	91	109	108	83
FN	0	0	4	35	6
TN	551	569	636	637	540
Süre	18,19	17,55	15,61	522,82	26,55

VS3 veri seti üzerinde en iyi doğruluk değeri performansını %96,40 ile Lojistik Regresyon yöntemi göstermiştir. En az doğruluk değeri performansını %95,55 ile Rastgele Orman yöntemi göstermiştir. VS3 veri setinde 31 adet öznitelik bilgisi için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemlerine göre karışıklık matrisi sonuçları ve çalışma süreleri Tablo 4.3'te verilmiştir.

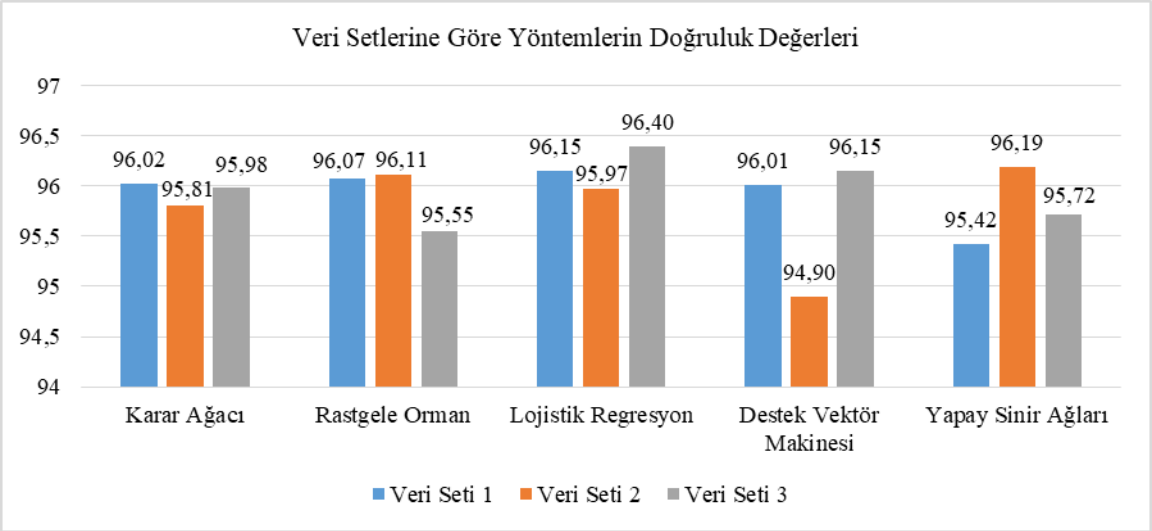
Tablo 4.3: VS3 Veri Setine ait Yöntemler Bazında Karışıklık Matrisi Sonuçları

	Karar Ağacı	Rastgele Orman	Lojistik Regresyon	Destek Vektör Makinesi	Yapay Sinir Ağları
Doğruluk	0,9598	0,9555	0,9640	0,9615	0,9572
Hassasiyet	0,9476	0,9425	0,9533	0,9502	0,9446
Geri Çağırma	1,0000	1,0000	1,0000	1,0000	1,0000
F1-skor	0,9730	0,9704	0,9761	0,9745	0,9715
TP	1700	1706	2064	2064	1706
FP	94	104	101	108	100
FN	0	0	0	0	0
TN	547	531	644	637	535
Süre	24,15	30,30	19,61	35,46	30,34

Üç veri seti üzerinde en iyi doğruluk değeri performansını %96,40 ile Lojistik Regresyon yöntemi göstermiştir. VS1, VS2 ve VS3 veri setleri için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemleri sonucunda elde edilen doğruluk değerlerine ait sonuçlar Tablo 4.4'te verilmiştir. Tablo 4.4'e gösterilen sonuçlara ait doğruluk değerlerinin yüz ile çarpılıp yüzdelik grafik gösterimi Şekil 4.1'de sunulmuştur.

Tablo 4.4: Veri Setlerine Göre Yöntemlerin Doğruluk Değerleri Sonuçları

	Karar Ağacı	Rastgele Orman	Lojistik Regresyon	Destek Vektör Makinesi	Yapay Sinir Ağları
VS1	0,9602	0,9607	0,9615	0,9601	0,9542
VS2	0,9581	0,9611	0,9597	0,9490	0,9619
VS3	0,9598	0,9555	0,9640	0,9615	0,9572



Şekil 4.1: Veri Setlerine Göre Yöntemlerin Yüzdelik Doğruluk Değerleri Karşılaştırma Grafiği

Üç veri setine uygulanan makine öğrenme yöntemlerine göre en iyi doğruluk değeri Lojistik Regresyon yöntemi %96,40 ile VS3 veri setinde 19,61 saniye çalışma süresi olarak göstermiştir. En yüksek doğruluk değerleri içerisinde çalışma süresi en kısa sürede gerçekleşen yöntem 11,72 saniye ile Karar Ağacı yöntemidir. Diğer yöntemlere göre çalışma süresi daha fazla olan yöntem 35,46 saniye ile Destek Vektör Makinesi yöntemidir. VS1, VS2 ve VS3 veri setleri için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemleri sonucunda elde edilen yöntemler bazında en yüksek doğruluk değerleri, bu doğruluk değerlerinin hangi veri setinden elde edildiği ve doğruluk değerlerine ait çalışma süreleri bilgileriyle birlikte Tablo 4.5'te verilmiştir.

Tablo 4.5: Yöntem Bazında En Yüksek Doğruluk Değerlerine ait Veri Seti ve Çalışma Süreleri

Yöntem Adı	Doğruluk Değeri	Veri Seti	Süre
Karar Ağacı	0,9602	VS1	11,72
Rastgele Orman	0,9611	VS2	17,55
Lojistik regresyon	0,9640	VS3	19,61
Destek Vektör Makinesi	0,9615	VS3	35,46
Yapay Sinir Ağları	0,9619	VS2	26,55

En kısa çalışma süresi 9,48 saniye ile VS1 veri setine uygulanan Lojistik Regresyon yöntemi vermiştir. Veri setlerine uygulanan beş farklı sınıflandırma yöntemleri sonucunda elde edilen veri setleri bazında en yüksek doğruluk değerleri, bu doğruluk değerlerinin hangi

yöntemden elde edildiği ve doğruluk değerine ait çalışma süreleriyle birlikte Tablo 4.6'da verilmiştir.

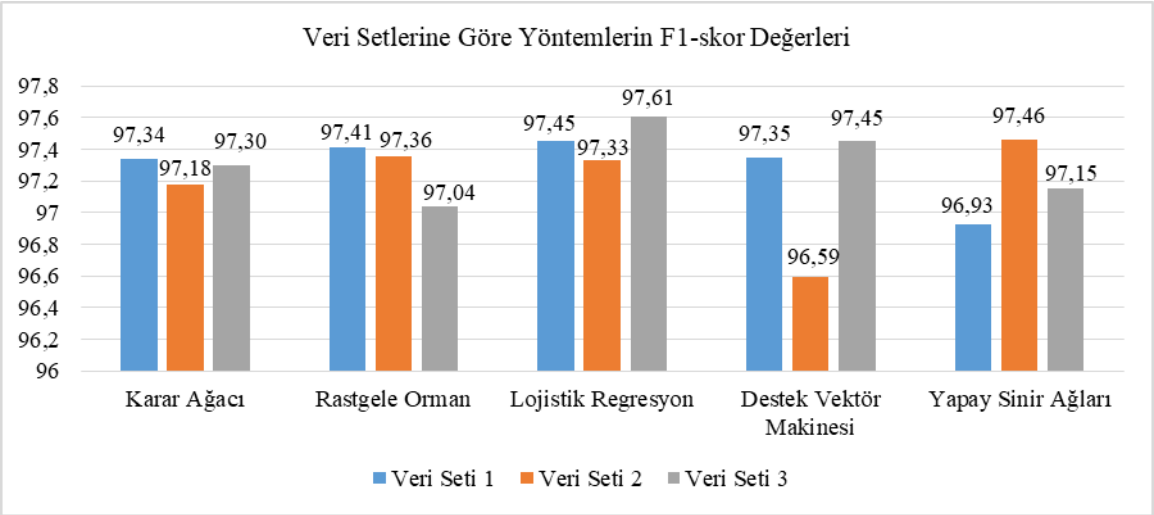
Tablo 4.6: Veri Seti Bazında En Yüksek Doğruluk Değerlerine ait Yöntem ve Çalışma Süreleri

Veri Seti	Doğruluk Değeri	Yöntem Adı	Süre
VS1	0,9615	Lojistik Regresyon	9,48
VS2	0,9619	Yapay Sinir Ağları	26,55
VS3	0,9640	Lojistik Regresyon	19,61

Üç veri setinde en iyi F1-skor değeri VS3 veri setinde %97,61 ile Lojistik Regresyon yöntemi vermiştir. VS1, VS2 ve VS3 veri setlerine göre Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemleri sonucunda elde edilen F1-skoru değerlerine ait sonuçlar Tablo 4.7'de verilmiştir. Tablo 4.7'de gösterilen sonuçlara ait F1-skor değerlerinin yüz ile çarpılıp yüzdelik grafik gösterimi Şekil 4.2'de sunulmuştur.

Tablo 4.7: Veri Setlerine Göre Yöntemlerin F1-skor Değer Sonuçları

	Karar Ağacı	Rastgele Orman	Lojistik Regresyon	Destek Vektör Makinesi	Yapay Sinir Ağları
VS1	0,9734	0,9741	0,9745	0,9735	0,9693
VS2	0,9718	0,9736	0,9733	0,9659	0,9746
VS3	0,9730	0,9704	0,9761	0,9745	0,9715



Şekil 4.2: Veri Setlerine Göre Yöntemlerin Yüzdelik F1-skor Değerleri Karşılaştırma Grafiği

Sınıflandırma yöntemleri bazında üç veri seti için en iyi F1-skor değerini VS3 veri setinde 19,61 saniye çalışma süresi ile %97,61 Lojistik Regresyon yöntemi vermiştir. En yüksek F1-skor değerleri içerisinde çalışma süresi 11,72 saniye ile en kısa sürede gerçekleşen ve diğer yöntemlere göre F1-skor değeri %97,34 ile en düşük olan yöntem Karar Ağacı yöntemidir. VS1, VS2 ve VS3 veri setleri için Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları sınıflandırma yöntemleri sonucunda elde edilen yöntemler bazında en yüksek F1-skor değerleri, bu F1-skor değerlerinin hangi veri setinden elde edildiği ve F1-skor değerlerine ait çalışma süreleri bilgileriyle birlikte Tablo 4.8'de verilmiştir.

Tablo 4.8: Yöntem Bazında En Yüksek F1-skor Değerlerine ait Veri Seti ve Çalışma Sonuçları

Yöntem Adı	F1-skor Değeri	Veri Seti	Süre
Karar Ağacı	0,9734	VS1	11,72
Rastgele Orman	0,9741	VS1	15,93
Lojistik regresyon	0,9761	VS3	19,61
Destek Vektör Makinesi	0,9745	VS3	35,46
Yapay Sinir Ağları	0,9746	VS2	26,55

5. TARTIŞMA

Tez kapsamında, oltalama saldırısı amacıyla yapılmış sahte web sitelerini yasal web sitelerinden ayırt ederek bir web sitesinin URL öznitelik bilgilerini kullanarak oltalama saldırısı olup olmadığının tespit edilmesi hedeflenmiştir. Bu hedefe ulaşmak için, literatürde kabul gören metriklere bağlı kalarak yüksek performans sunan bir sınıflandırıcı modeli amaçlanmıştır. Literatürde oltalama sitelerini tanımlamak ve tespit etmek için çeşitli yöntemler denenmiştir. Bu çalışmada makine öğrenimi yöntemlerine odaklanarak oltalama saldırılarının tespiti için bir model geliştirilmiştir.

Geliştirilen modelde oltalama saldırılarını tespit etmek için USOM da bulunan sahte web sitelerinden oluşan siteler ile yasal web sitelerinden elde edilen siteler birleştirilip veri kümesi elde edilmiştir. Elde edilen veri kümesi üzerinde URL öznitelikleri ile sadece yasal web sitelerinden oluşan web siteleri karşılaştırılıp benzerlik yöntemlerine göre öznitelik bilgileri elde edilmiştir. Elde edilen özniteliklere ait veri seti seçilen öznitelik bilgilerine ve dönüştürme işlemlerine göre üç farklı veri setine ayrılmıştır. Böylece farklı veri setleri üzerinde farklı sınıflandırma yöntemlerini kullanarak en uyumlu birlikteliği bulmak ve bir web sitesinin oltalama saldırısı olup olmadığının doğruluğunu en yüksek tespit etmektir. Bu çalışmada Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay Sinir Ağları olmak üzere beş farklı makine öğrenimi sınıflandırma yönteminde çalışılmıştır. Üç farklı veri seti üzerinde beş farklı sınıflandırma yönteminin her biri çalıştırılmış ve sonuçlar analiz edilmiştir.

En yüksek doğruluk değeri VS3 veri setinde %96,40 ile Lojistik Regresyon yöntemi olmuştur. En düşük doğruluk değeri VS2 veri setinde %94,90 ile Destek Vektör Makinesi yöntemi olmuştur. Doğruluk değerlerine göre sınıflandırma yöntemlerini en yüksek değerden en düşük değere göre sıraladığımız zaman Lojistik Regresyon yöntemi, Destek Vektör Makinesi yöntemi, Rastgele Orman yöntemi ve Karar Ağacı yöntemi olmuştur.

Çalışma süresi olarak en iyi değer Lojistik Regresyon yöntemi ile VS1 veri setinde 9,48 saniye olmuştur. Destek Vektör Makinesi yöntemi çalışma süresi olarak VS2 veri setinde 522,82 saniye ile performansı düşük yöntem olmuştur. Çalışma süresine göre sınıflandırma yöntemlerini en kısa çalışma süresinden en uzun çalışma süresine göre sıraladığımız zaman

Lojistik Regresyon yöntemi, Karar Ağacı yöntemi, Rastgele Orman yöntemi, Yapay Sinir Ağları yöntemi ve Destek Vektör Makinesi yöntemi olmuştur.

En yüksek F1-skor değeri %97,61 ile VS3 veri setinde Lojistik Regresyon yöntemi olmuştur. Destek Vektör Makinesi F1-skor değeri %96,59 ile VS2 veri setinde yöntem olmuştur. Sınıflandırma yöntemleri bazından F1-skor değerlerine göre en yüksek değerden en düşük değere göre sıraladığımız zaman Lojistik Regresyon yöntemi, Yapay Sinir Ağları yöntemi, Destek Vektör Makinesi yöntemi, Rastgele Orman yöntemi ve Karar Ağacı yöntemi olmuştur.

Bu tez çalışmasında gerçekleştirdiğim model ile ilgili sınıflandırma sonuçları, öznitelik tabanlı makine öğrenme modelinin literatürde bulunan diğer yöntemlere göre daha başarılı olduğunu göstermiştir. Ayrıca literatürdeki diğer oltalama URL saldırılarını tespit etmeye yönelik çalışmalar incelendiği zaman, bu modelin diğer makine öğrenmesi yöntemlerine göre daha etkili sonuçlar elde ettiği görülmektedir. Bu bağlamda, literatürdeki çalışmalar, oltalama URL web sitelerini tespit etmek için makine öğrenme modelinin uygun bir yaklaşım olduğunu göstermektedir. Önerilen ve geliştirilen model bir web sitesinin oltalama saldırısı olup olmadığını tespit etmede %96,40 doğruluk değeri elde etmiştir ve literatürde bulunan diğer çalışmalar içerisinde yüksek başarı göstermiştir.

Bu tez çalışması, oltalama sitelerini tespit etme amacı güderken, bu sitelerin genellikle URL yapısındaki belirgin özellikleri ve benzerlik özellikleri üzerinde durmaktadır. Oltalama siteleri ile yasal siteler arasındaki bu farkları ortaya çıkarmak için URL özniteliklerini kullanarak bir makine öğrenme yöntemleri üzerinde sınıflandırıcı model oluşturulmuştur. Literatürde yaygın olarak kabul görmüş performans metriklerine dayalı olarak yüksek doğruluk ve en iyi çalışma sürelerine ulaşılabilecek bir model geliştirilmiştir. Geliştirilen model, USOM kaynaklı sahte web siteleriyle Türkiye ve Dünyada en az ile en çok gezilen yasal web sitelerinin toplanmasından çıkarılan öznitelikler üzerinde üç farklı veri setine ayrılarak eğitilmiştir.

Bu tez çalışmasında, veri kullanımı açık bir şekilde ele alınmıştır. Veri kaynakları açıkça belirtilmiş ve izlenebilirliği sağlamak amacıyla tüm yöntemler ve sonuçlar detaylı bir şekilde açıklanmıştır. Geliştirilen modelin objektif doğruluğu için gerçek verilere dayanılmış, özgün yazılım kodları oluşturulmuştur. Model kabul görmüş yazılım araçlarıyla başarı oranları doğrulanmıştır. Veri toplama süreci özenle ve titiz bir şekilde yürütülmüş olup; sahte

web siteleri USOM'dan ve yasal web siteleri de manuel çalışmalarla bir araya getirilip en doğru veriler elde edilmiştir. Böylelikle, çalışmanın güvenilirliği ve geçerliliği en üst düzeyde sağlanmıştır. Toplanan sahte ve yasal web sitelerinin veri kümesi ile sadece yasal web sitelerinden oluşan veri kümesinin karşılaştırılarak Orijinal domain kontrolü ve benzerlik yöntemlerine göre (karakter benzerliği yöntemi, kosinüs benzerliği yöntemi ve levenshtein benzerliği yöntemi) özniteliklerin elde edilmiştir. Elde edilen öznitelikler üç farklı veri setine ayrılarak makine öğrenme yöntemleri ile sınıflandırılmıştır. Sınıflandırma işleminde çalışma sürelerinin kısa olması oltalama saldırılarının tespitinde önemli olduğu için kısa çalışma sürelerinde yüksek doğruluk ve F1-skor değerlerine ulaşmak çalışmanın özgün taraflarındandır. Ayrıca modelimde oluşturduğum veri setleri açısından çalışma literatürde bulunan diğer çalışmalardan daha öne çıkmaktadır.

Oltalama saldırılarının tespitiyle ilgili gerçek hayattaki etkilerini anlama ve kullanıcı farkındalığını artırma çalışmalarına bu tezde özel bir vurgu yapılmamıştır. Bu noktada, çalışmanın sosyal bilimler açısından daha kapsamlı bir değerlendirme yapma şansı elde edemediğimiz bir açıklık getirilemeyen nokta olarak görülmektedir. Önümüzdeki çalışmalarda, insan davranışlarına dair daha derinlemesine bir araştırma yapmak mümkün görünmektedir. Böylelikle, oltalama saldırılarının gerçek dünyada yarattığı etkileri ve bu konuda kullanıcı farkındalığını artırmaya yönelik öneriler sunmak, çalışmalarımızın kapsamını genişletmek istediğimiz bir alan olarak karşımıza çıkmaktadır.

Bu tez çalışmasında hem mevcut hem de gelişen oltalama saldırısı yöntemleri ele almaya odaklanılmıştır. Bu amaçla esnek ve özellikle özelleştirilebilir bir model sistemi tasarlanmıştır. Bu sistem, kullanıcıların ihtiyaçlarına göre öznitelik bilgilerinin kolayca değiştirilebileceği ve genişletilebileceği bir yapıya sahiptir. Bu çalışma ile bundan sonra bu alanda yapılacak olan çalışmalara referans olmak, oluşturduğum benzerlik özniteliklerini bu alanda çalışacak kişi veya kişiler tarafından yapılacak çalışmalara katkı ve fayda sağlayacağı öngörülmektedir.

6. SONUÇ VE ÖNERİLER

Oltalama saldırıları, sahte bir kimlik, kuruluş veya web sitesi gibi gösterilerek insanları hassas bilgilerini paylaşmaya, kişisel bilgilerini vermeye veya kötü amaçlı yazılımları indirmeye ikna etmeye çalışan saldırılardır. Son dönemde, özellikle kullanıcıların finansal bilgilerini çalmak amacıyla yasal web sitelerini taklit eden sahte web siteleri oluşturulmuştur. Bu tür saldırılar, kullanıcıların maddi kayıplarına yol açabilecek ciddi tehditler oluşturmaktadır. Bilgi Teknolojileri ve İletişim Kurumu Başkanlığı bünyesinde kurulan Ulusal Siber Olaylara Müdahale Merkezi tarafından sunulan verilerde ülkemizdeki oltalama saldırılarının her geçen gün arttığı görülmektedir. Bu verilere göre, tespit edilen oltalama sitelerinin sayısı sürekli olarak artış göstermektedir.

Oltalama saldırıları, kullanıcıları kandırmak ve hassas bilgilerini ele geçirmek amacıyla yapıldığı için siber güvenlik açısından oldukça ciddi bir tehdit oluşturmaktadır. Siber güvenlik kurumları, bu tür saldırılara neden olan siteleri tespit etmek için oltalama sitelerinin ortak özelliklerinin belirlenmesi önemlidir. Bu özelliklerden biri de URL yapısıdır. Oltalama saldırı siteleri genellikle URL içeriğiyle fark edilmektedir. Bu yüzden, oltalama saldırısı sitelerinin tespit edilmesi ve önlenmesi için öncelikle belirgin özellik bilgilerinin doğru bir şekilde belirlenmesi gerekmektedir.

Bu tez çalışması kapsamında, bir web sitesinin oltalama web sitesi olup olmadığını analiz ve tespit etmek için bir model sistemi amaçlanmıştır ve bu amaca yönelik yeni bir veri kümesi hazırlanmıştır. Hazırlanan veri kümesinde sahte web siteleri Ulusal Siber Olaylara Müdahale Merkezi'ndeki zararlı bağlantılar sayfasından yasal web siteleri ise Türkiye ve dünya da faaliyet gösteren en az ile en çok ziyaret edilen kamu, üniversite, e-ticaret, banka, enerji, ulaşım vb. gibi web sitelerinden arama motorlarında gezilerek elde edilmiştir. Böylece yasal web sitelerinde bilinirliği az olan ve potansiyel olarak oltalama web sitesi olabilecek sitelerde çalışmaya dahil edilmiştir. Veri kümesinden URL yapısındaki öznitelikler ile tamamen yasal web sitelerinden oluşan web siteleriyle karşılaştırılıp benzerlik yöntemlerine göre elde edilen öznitelikler bir araya getirilmiş ve 32 adet öznitelik elde edilmiştir.

Elde edilen özniteliklere ait veri seti daha iyi ve daha hızlı oltalama saldırılarını tespit etmek için üç farklı veri setine ayrılmıştır. Ayrılan üç farklı veri seti makine öğrenme yöntemi olan Karar Ağacı, Rastgele Orman, Lojistik Regresyon, Destek Vektör Makinesi ve Yapay

Sinir Ağları yöntemleri ile sınıflandırılmıştır. Böylelikle farklı veri setleri üzerinde farklı sınıflandırma yöntemlerinin sonuçları değerlendirilmiştir.

Önerilen modelde bir web sitesinin oltalama saldırı sitesi olup olmadığını tespit etmede çalışma süresi 9,48 saniye, %96,40 doğruluğa ve %97,61 F1-skor değerlerine ulaşılmıştır. Bu sonuçlar çalışmamda önerdiğim modelin diğer çalışmalar içerisinde oldukça iyi performans göstermiştir. Modelimin başarısının temel nedenlerinden biri, URL adreslerini ve benzerlik yöntemlerini dikkatlice analiz eden bir öznitelik setinin kullanılmış olmasıdır.

Bu tez çalışması ile yasal web sitelerin sahte web siteleriyle karıştırılmasına sebep olan standart dışı URL isimlendirmesi önemli bir sorun olarak görülmektedir. Bu bağlamda, bireylerin ve şirketlerin oltalama saldırılarını tanıma ve önleme konusundaki bilgi düzeylerini artırmaları önemlidir. Yasal şirketlerin web siteleri üzerinde daha güçlü güvenlik önlemlerini uygulaması kritik öneme sahiptir. Kullanıcılara sahte web siteleri konusunda uyarıcı bildirimlerinin yapılması, SSL sertifikalarının düzenli olarak kontrol edilmesi ve açıklayıcı URL yapılarının benimsenmesi oltalama saldırılarına karşı bilinçlendirilmesi konusunda tavsiye edilmektedir.

Gelecekte, modelin etkinliğini artırmak için çeşitli çalışma ve yöntemlerin birleştirilmesi planlanmaktadır. Oluşturduğum üç veri seti derin öğrenme temelli sistemlerde test edilerek başarı oranları ve çalışma süreleri gibi araştırmalar gerçekleştirilebilir. Gerçekleştirilecek sınıflandırma yöntemine için veri setine ait özniteliklere benzerlik yöntemlerine göre yeni öznitelik bilgileri eklenip sistemin verimliliği üzerine çalışmalar yapılabilir.

KAYNAKLAR

[1]. Chiew, K.L., Yong, K.S.C. and Tan, C.L., 2018, A survey of phishing attacks: Their types, vectors and technical approaches, *Expert Systems with Applications*, 106, 1-20.

[2]. James, L., 2006, *Chapter 1 - Banking on phishing*, Phishing Exposed, In: James L. (ed.), Chapter 1, Syngress, Burlington USA, ISBN: 978-1-59749-030-6, 1-35.

[3]. 2023, *Ulusal Olaylara Müdahele Merkezi*, https://www.usom.gov.tr/, [Ziyaret Tarihi: 20 Mayıs 2023].

[4]. Hameed, K. and Rahman, N., 2017, Today's Social NetworkSites: AnAnalysis of Emerging Security Risks and their Counter Measures, *2017 International Conference on Communication Technologies (ComTech)*, 19-21 April 2017 Rawalpindi, Pakistan, IEEE, ISBN: 9781509059850, 143-148.

[5]. Liu, G., Qiu, B., Wenyin, L., 2010, Automatic Detection of Phishing Target from Phishing Webpage, *2010 20th International Conference on Pattern Recognition (ICPR 2010)*, 23-26 August 2010 Istanbul, Turkey, IEEE, ISBN: 9780769541099, 4153–4156.

[6]. Abbasi, A. and Chen, H., 2007, Detecting fake escrow websites using rich fraud cues and kernel based methods, *17th Workshop on Information Technologies and Systems*, 8-9 December 2007 Montreal, Canada, arXiv, 55-60.

[7]. Jamil, A., Asif, K., Ghulam, Z., Nazir, M.K., Alam, S.M., Ashraf, R., 2018, MPMPA: A Mitigation and Prevention Model for Social Engineering Based Phishing attacks on Facebook, *2018 IEEE International Conference on Big Data (Big Data)*, 10-13 December 2018 Seattle, USA, IEEE, ISBN: 978-1-5386-5036-3, 5040–5048.

[8]. Goodrich, M. and Tamassia, R., 2013, *Introduction to Computer Security: Pearson New International Edition*, Pearson Education Limited, London, ISBN: 1292025409.

[9]. Mishra, S. and Soni, D., 2019, SMS Phishing and Mitigation Approaches, *2019 Twelfth International Conference on Contemporary Computing (IC3)*, 08-10 August 2019 Noida, India, IEEE, ISBN: 978-1-7281-2360-8, 1-5.

[10]. Brkić, M., Maksimović, A. and Miljković, A., 2023, A System Architecture for Preventing Social Engineering Attacks via E-mail, *Journal of Computer and Forensic Sciences*, 2 (1), 43–52.

[11]. Mohammad, R.M., Thabtah, F. and McCluskey, L., 2015, Tutorial and critical analysis of phishing websites methods, *Computer Science Review*, 18, 1-24.

[12]. O'Mara, A., Alsmadi, I. and AlEroud, A., 2021, Generative Adverserial Analysis of Phishing Attacks on Static and Dynamic Content of Webpages, *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 30 September 2021-03 October 2021 New York City, USA, IEEE, ISBN: 978-1-6654-1193-6, 1657-1662.

[13]. Rautray, P.K., 2018, Rule-based intrusion detection system, International Journal of Engineering, Management, *Humanities and Social Sciences Paradigms (IJEMHS)*, 30, 325-327.

[14]. Liu, W., Deng, X., Huang, G. and Fu, A.Y., 2006, An antiphishing strategy based on visual similarity assessment, *IEEE Internet Computing*, 10, 58-65.

[15]. Tata, S., and Patel, J.M., 2007, Estimating the selectivity of tf-idf based cosine similarity predicates, *ACM SIGMOD Record*, 36, 7–12.

[16]. Huang, A., 2008, Similarity Measures for Text Document Clustering, *New Zealand Computer Science Research Student Conference*, 14-18 April 2008 Christchurch, New Zealand, NZCSRSC2008, 49-56.

[17]. Levenshtein, V.I., 1965, Binary codes capable of correcting deletions, insertions, and reversals, *Doklady Akademii Nauk SSSR*, 163, 845–848.

[18]. Hakak, S., Alazab, M., Khan. S., Gadekallu, T.R., Maddikunta, P.K.R. and Khan, W.Z., 2021, An ensemble machine learning approach through effective feature extraction to classify fake news, *Future Generation Computer Systems*, 117, 47-58.

[19]. Karimi, A.M., Fada, J.S., Liu, J., Braid, J.L., Koyutürk, M. and French, R.H., 2018, Feature Extraction, Supervised and Unsupervised Machine Learning Classification of PV Cell Electroluminescence Images, *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC) (A Joint Conference of 45th IEEE PVSC, 28th PVSEC & 34th EU PVSEC), 10-15 June 2018 Waikoloa*, USA, IEEE, ISBN: 978-1-5386-8530-3, 0418-0424.

[20]. Kayode-Ajala, O., 2022, Applying Machine Learning Algorithms for Detecting Phishing Websites: Applications of SVM, KNN, Decision Trees, and Random Forests, *International Journal of Information and Cybersecurity*, 6, 43-61.

[21]. Breiman, L., 2001, Random Forests, *Kluwer Academic Publishers*, 45, 5–32.

[22]. Niu, W., Zhang, X., Yang, G., Ma, Z. and Zhuo, Z., 2017, Phishing Emails Detection Using CS-SVM, *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, 12-15 December 2017 Guangzhou, China, IEEE, ISBN: 978-1-5386-3791-3, 1054-1059.

[23]. Dedeturk, B. K. and Akay, B., 2020, Spam filtering using a logistic regression model trained by an artificial bee colony algorithm, *Applied Soft Computing*, 91 (16),106229.

[24]. Zhu, E., Ju, Y., Chen, Z., Liu, F. and Fang, X., 2020, DTOF-ANN: An Artificial Neural Network phishing detection model based on Decision Tree and Optimal Features, *Applied Soft Computing*, 95, 106505.

[25]. Liu, R., Lin, Y., Zhang, Y., Lee, P. H. and Dong, J. S., 2023, Knowledge Expansion and Counterfactual Interaction for Reference-Based Phishing Detection, *32nd USENIX Security Symposium*, 9–11 August 2023 Anaheim, USA, USENIX, 4139-4156.

[26]. Zouina, M., and Outtaj, B., 2017, A novel lightweight URL phishing detection system using SVM and similarity index, *Human-centric Computing and Information Sciences*, 7, 1–13.

[27]. Wu, C., Kuo, C. C. and Yang, C. S., 2019, A Phishing Detection System based on Machine Learning, *2019 International Conference on Intelligent Computing and Its Emerging Applications, ICEA 2019*, 30 August-1 September 2019 Tainan, Taiwan, Institute of Electrical and Electronics Engineers Inc., ISBN: 9781728131597, 28-32.

[28]. Liu, T., Zhang, Y., Shi, J., Jing, Y., Li, Q. and Guo, L., 2016, Towards quantifying visual similarity of domain names for combating typosquatting abuse, *MILCOM 2016-2016 IEEE Military Communications Conference*, 01-03 November 2016 Baltimore, USA, IEEE, ISBN: 978-1-5090-3782-7, 770-775.

[29]. SatheeshKumar, M., Srinivasagan, K.G. and UnniKrishnan, G., 2022, A lightweight and proactive rule-based incremental construction approach to detect phishing scam, *Information Technology and Management*, 23, 271–298.

[30]. Fazliya, M.H.F. and Naleer, H.M.M., 2019, A Rule Based Prediction of Phishing Websites Using Data Mining Classification Techniques, *Journal of Technology and Value Addition*, 1 (2), 31-40.

[31]. Abdelhamid, N., Ayesh, A. and Thabtah, F., 2014, Phishing detection based Associative Classification data mining, *Expert Systems with Applications*, 41, 5948-5959.

[32]. Mourtaji, Y., Bouhorma, M., Alghazzawi, D., Aldabbagh, G. and Alghamdi, A., 2021, Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network, *Wireless Communications and Mobile Computing*, 2021, 1-24.

[33]. Ahmed, M., Altamimi, A.B., Khan, W., Alsaffar, M., Ahmad, A., Khan, Z.H. and Alreshidi, A., 2023, PhishCatcher: Client-Side Defense Against Web Spoofing Attacks Using Machine Learning, *IEEE Access*, 11, 61249-61263.

[34]. Alhogail, A. and Alsabih, A., 2021, Applying machine learning and natural language processing to detect phishing email, *Computers & Security*, 110, 102414.

[35]. Butt, U.A., Amin, R., Aldabbas, H., Mohan, S., Alouffi, B. and Ahmadian, A., 2022, Cloud-based email phishing attack using machine and deep learning algorithm, *Complex & Intelligent Systems*, 9, 3043–3070.

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

[37]. Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S.B. and Joga, S.R.K., 2023, Phishing Detection System Through Hybrid Machine Learning Based on URL, *IEEE Access*, 11, 36805-36822.

[38]. Choudhary, T., Mhapankar, S., Bhddha, R., Kharuk, A. and Patil, Dr.R., 2023, A Machine Learning Approach for Phishing Attack Detection, *Journal of Artificial Intelligence and Technology*, 3 (3), 108-113.

[39]. Islam, M. and Chowdhury, N., 2016, Phishing websites detection using machine learning based classification techniques, *The 1st International Conference on Advanced Information and Communication Technology 2016 (ICAICT 2016)*, 16-17 May 2016 Chittagong, Bangladesh, CIU, 370-373.

[40]. 2023, *Ulusal Siber Olaylara Müdahale Merkezi-USOM Hakkında*, https://www.usom.gov.tr/hakkimizda, [Ziyaret Tarihi: 8 Temmuz 2023].

[41]. 2023, *Türkiye Bankalar Birliği Banka ve Şube Sayıları*, https://www.tbb.org.tr/tr/bankacilik/banka-ve-sektor-bilgileri/banka-bilgileri/banka-ve-sube-sayilari/70, [Ziyaret Tarihi: 11 Temmuz 2023].

[42]. 2023, *YÖK Universiteler*, https://www.yok.gov.tr/universiteler/universitelerimiz, [Ziyaret Tarihi: 12 Temmuz 2023].

[43]. 2023, *e-Devlet Kapısı Devletin Kısayolu*, https://www.turkiye.gov.tr/, [Ziyaret Tarihi: 14 Temmuz 2023].

[44]. 2023, *Forbes Billionaires 2023: The Richest People In The World*, https://www.forbes.com/billionaires/, [Ziyaret Tarihi: 18 Temmuz 2023].

[45]. 2023, *Ulusal Siber Olaylara Müdahale Merkezi Zararlı Bağlantılar Sayfası*, https://www.usom.gov.tr/adres, [Ziyaret tarihi: 7 Ekim 2023].

[46]. 2023, *Ulusal Siber Olaylara Müdahale Merkezi Zararlı Bağlantılar sayfasında bulunan örnek "cskh.evnspa.cc" web sitesine ait detay bilgisi*, https://www.usom.gov.tr/adres/532826, [Ziyaret tarihi: 7 Ekim 2023].

[47]. 2023, *Oltalama saldırısı tespit edilmiş sahte "a101mobilcep.com" sitesinin sayfa içeriği*, http://a101mobilcep.com/, [Ziyaret tarihi: 8 Ekim 2023].

[48]. 2023, *Yasal web sitesi "a101.com.tr" sayfa içeriği, https://www.a101.com.tr/*, [Ziyaret tarihi: 8 Ekim 2023].

[49]. 2023, *Python Release Python 3.8.0*, https://www.python.org/downloads/release/python-380/, [Ziyaret tarihi: 8 Şubat 2023].

[50]. 2022, *Free Download|Anaconda*, https://www.anaconda.com/download, [Ziyaret tarihi: 5 Nisan 2022].

[51]. 2022, *Files::Anaconda.org*, https://anaconda.org/anaconda/spyder/files?version=5.1.5, [Ziyaret tarihi: 6 Nisan 2022].

[52]. 2022, *Using Jupyter Notebook extensions-Anaconda documentation*, https://docs.anaconda.com/free/anaconda/jupyter-notebooks/use-jupyter-notebook-extensions/, [Ziyaret tarihi: 18 Nisan 2022].

[53]. Han, J., Kamber, M. and Pei, J., 2012, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, Waltham MA USA, Third Edition, ISBN: 978-0-12-381479-1.

[54]. Heydarian, M., Doyle, T.E. and Samavi, R., 2022, MLCM: Multi-Label Confusion Matrix, *IEEE Access*, 10, 19083-19095.

[55]. Aburrous, M., Hossain, M.A., Dahal, K. and Thabtah, F., 2010, Intelligent phishing detection system for e-banking using fuzzy data mining, *Expert Systems with Applications*, 37, 7913-7921.

İNTİHAL RAPORU İLK SAYFASI

Selahattin ALİYAZICIOĞLU

ORİJİNALLİK RAPORU

%11	%10	%1	%6
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	%4
2	30d4ed98-bbfd-4215-a9ab-b3bd6c00d4b3.filesusr.com İnternet Kaynağı	%2
3	acikbilim.yok.gov.tr İnternet Kaynağı	%1
4	openaccess.ogu.edu.tr:8080 İnternet Kaynağı	<%1
5	openaccess.hacettepe.edu.tr İnternet Kaynağı	<%1
6	www.tbb.org.tr İnternet Kaynağı	<%1
7	Submitted to Erciyes Ãniversitesi Öğrenci Ödevi	<%1
8	developer.aliyun.com İnternet Kaynağı	<%1
9	Submitted to Istanbul Aydin University	

KURUM İZNİ YAZILARI

Uyarı: Canlı ve cansız deneklerle yapılan tüm çalışmalar için kurum izin belgelerinin eklenmesi zorunludur. Gizlilik ve mahremiyet içeren durumlarda kurum adı kapatılmalıdır.

☐ **Kurum izni gerekmektedir.**

☒ **Kurum izni gerekmemektedir.**

Öğrenci Adı SOYADI
(İmza)